

Erweitern von ADOBE® FLASH® PROFESSIONAL CS5 & CS5.5

Rechtliche Hinweise

Rechtliche Hinweise finden Sie unter http://help.adobe.com/de_DE/legalnotices/index.html.

Inhalt

Kapitel 1: Einführung

Verwenden der JavaScript-API	1
Neu in der JavaScript-API	5
JavaScript-API-Objekte	7
Implementierungsbeispiele	13

Kapitel 2: Top-Level-Funktionen und -Methoden

activate()	15
alert()	16
configureTool()	16
confirm()	17
deactivate()	18
keyDown()	19
keyUp()	19
mouseDoubleClick()	20
mouseDown()	20
mouseMove()	21
mouseUp()	22
notifySettingsChanged()	23
prompt()	23
setCursor()	24

Kapitel 3: actionsPanel-Objekt

actionsPanel.getClassForObject()	25
actionsPanel.getScriptAssistMode()	26
actionsPanel.getSelectedText()	27
actionsPanel.getText()	27
actionsPanel.hasSelection()	28
actionsPanel.replaceSelectedText()	28
actionsPanel.setScriptAssistMode()	29
actionsPanel.setSelection()	30
actionsPanel.setText()	31

Kapitel 4: BitmapInstance-Objekt

bitmapInstance.getBits()	32
bitmapInstance.hPixels	33
bitmapInstance.setBits()	34
bitmapInstance.vPixels	35

Kapitel 5: BitmapItem-Objekt

bitmapItem.allowSmoothing	37
bitmapItem.compressionType	37
bitmapItem.exportToFile()	37
bitmapItem.fileLastModifiedDate	38

Inhalt

bitmapItem.originalCompressionType	39
bitmapItem.quality	39
bitmapItem.sourceFileExists	40
bitmapItem.sourceFilesCurrent	40
bitmapItem.sourceFilePath	41
bitmapItem.useDeblocking	41
bitmapItem.useImportedJPEGQuality	42
 Kapitel 6: CompiledClipInstance-Objekt	
compiledClipInstance.accName	43
compiledClipInstance.actionScript	44
compiledClipInstance.description	44
compiledClipInstance.forceSimple	45
compiledClipInstance.shortcut	45
compiledClipInstance.silent	46
compiledClipInstance.tabIndex	46
 Kapitel 7: compilerErrors-Objekt	
compilerErrors.clear()	47
compilerErrors.save()	48
 Kapitel 8: ComponentInstance-Objekt	
componentInstance.parameters	49
 Kapitel 9: componentsPanel-Objekt	
componentsPanel.addItemToDocument()	50
componentsPanel.reload()	51
 Kapitel 10: Contour-Objekt	
contour.fill	52
contour.getHalfEdge()	53
contour.interior	54
contour.orientation	54
 Kapitel 11: Document-Objekt	
document.accName	64
document.addDataToDocument()	64
document.addDataToSelection()	65
document.addFilter()	65
document.addItem()	66
document.addNewLine()	67
document.addNewOval()	67
document.addNewPrimitiveOval()	68
document.addNewPrimitiveRectangle()	69
document.addNewPublishProfile()	70
document.addNewRectangle()	71
document.addNewScene()	72
document.addNewText()	73

document.align()	74
document.allowScreens()	74
document.arrange()	75
document.as3AutoDeclare	76
document.as3Dialect	76
document.as3ExportFrame	76
document.as3StrictMode	77
document.as3WarningsMode	77
document.asVersion	78
document.autoLabel	79
document.backgroundColor	79
document.breakApart()	80
document.canEditSymbol()	80
document.canRevert()	81
document.canTestMovie()	81
document.canTestScene()	82
document.changeFilterOrder()	83
document.clipCopy()	83
document.clipCut()	84
document.clipPaste()	85
document.close()	85
document.convertLinesToFills()	86
document.convertToSymbol()	86
document.crop()	87
document.currentPublishProfile	88
document.currentTimeline	88
document.debugMovie()	89
document.deleteEnvelope()	89
document.deletePublishProfile()	90
document.deleteScene()	91
document.deleteSelection()	91
document.description	92
document.disableAllFilters()	92
document.disableFilter()	93
document.disableOtherFilters()	93
document.distribute()	94
document.distributeToLayers()	95
document.docClass	95
document.documentHasData()	96
document.duplicatePublishProfile()	96
document.duplicateScene()	97
document.duplicateSelection()	97
document.editScene()	98
document.enableAllFilters()	98
document.enableFilter()	99
document.enterEditMode()	100

document.exitEditMode()	100
document.exportPNG()	101
document.exportPublishProfile()	102
document.exportPublishProfileString()	102
document.exportSWF()	103
document.externalLibraryPath	104
document.forceSimple	104
document.frameRate	105
document.getAlignToDocument()	105
document.getBlendMode()	106
document.getCustomFill()	106
document.getCustomStroke()	107
document.getDataFromDocument()	108
document.getElementProperty()	109
document.getElementTextAttr()	109
document.getFilters()	110
document.getMetadata()	111
document.getMobileSettings()	111
document.getPlayerVersion()	112
document.getSelectionRect()	113
document.getTextString()	114
document.getTimeline()	114
document.getTransformationPoint()	115
document.group()	116
document.height	116
document.id	117
document.importFile()	117
document.importPublishProfile()	118
document.importPublishProfileString()	119
document.importSWF()	119
document.intersect()	120
document.library	120
document.libraryPath	121
document.livePreview	121
document.loadCuepointXML()	122
document.match()	122
document.mouseClick()	123
document.mouseDbIClk()	124
document.moveSelectedBezierPointsBy()	125
document.moveSelectionBy()	125
document.name	126
document.optimizeCurves()	126
document.path	127
document.pathURI	127
document.publish()	128
document.publishProfiles	128

document.punch()	129
document.removeAllFilters()	129
document.removeDataFromDocument()	130
document.removeDataFromSelection()	131
document.removeFilter()	131
document.renamePublishProfile()	132
document.renameScene()	132
document.reorderScene()	133
document.resetOvalObject()	133
document.resetRectangleObject()	134
document.resetTransformation()	135
document.revert()	135
document.rotate3DSelection()	136
document.rotateSelection()	136
document.save()	137
document.saveAndCompact()	138
document.scaleSelection()	138
document.screenOutline	139
document.selectAll()	140
document.selection	140
document.selectNone()	142
document.setAlignToDocument()	143
document.setBlendMode()	143
document.setCustomFill()	144
document.setCustomStroke()	145
document.setElementProperty()	145
document.setElementTextAttr()	146
document.setFillColor()	147
document.setFilterProperty()	148
document.setFilters()	148
document.setInstanceAlpha()	149
document.setInstanceBrightness()	150
document.setInstanceTint()	150
document.setMetadata()	151
document.setMobileSettings()	152
document.setOvalObjectProperty()	153
document.setPlayerVersion()	154
document.setRectangleObjectProperty()	154
document.setSelectionBounds()	155
document.setSelectionRect()	156
document.setStageVanishingPoint()	157
document.setStageViewAngle()	157
document.setStroke()	158
document.setStrokeColor()	158
document.setStrokeSize()	159
document.setStrokeStyle()	159

Inhalt

document.setTextRectangle()	160
document.setTextSelection()	161
document.setTextString()	161
document.setTransformationPoint()	162
document.silent	163
document.skewSelection()	163
document.smoothSelection()	164
document.sourcePath	165
document.space()	165
document.straightenSelection()	166
document.swapElement()	166
document.swapStrokeAndFill()	167
document.testMovie()	167
document.testScene()	168
document.timelines	169
document.traceBitmap()	169
document.translate3DCenter()	170
document.translate3DSelection()	170
document.transformSelection()	171
document.unGroup()	172
document.union()	172
document.unlockAllElements()	173
document.viewMatrix	173
document.width	174
document.xmlPanel()	174
document.zoomFactor	175
 Kapitel 12: drawingLayer-Objekt	
drawingLayer.beginDraw()	176
drawingLayer.beginFrame()	177
drawingLayer.cubicCurveTo()	178
drawingLayer.curveTo()	178
drawingLayer.drawPath()	179
drawingLayer.endDraw()	179
drawingLayer.endFrame()	180
drawingLayer.lineTo()	180
drawingLayer.moveTo()	181
drawingLayer.newPath()	181
drawingLayer.setColor()	182
drawingLayer.setFill()	183
drawingLayer.setStroke()	183
 Kapitel 13: Edge-Objekt	
edge.cubicSegmentIndex	184
edge.getControl()	185
edge.getHalfEdge()	185
edge.id	186

Inhalt

edge.isLine	186
edge.setControl()	187
edge.splitEdge()	187
edge.stroke	188

Kapitel 14: Element-Objekt

element.depth	190
element.elementType	191
element.getPersistentData()	191
element.getTransformationPoint()	192
element.hasPersistentData()	193
element.height	193
element.layer	194
element.left	194
element.locked	195
element.matrix	195
element.name	195
element.removePersistentData()	196
element.rotation	196
element.scaleX	197
element.scaleY	197
element.selected	198
element.setPersistentData()	198
element.setTransformationPoint()	199
element.skewX	200
element.skewY	200
element.top	201
element.transformX	201
element.transformY	202
element.width	202
element.x	203
element.y	203

Kapitel 15: Fill-Objekt

fill.bitmapsClipped	204
fill.bitmapPath	205
fill.color	205
fill.colorArray	206
fill.focalPoint	206
fill.linearRGB	207
fill.matrix	207
fill.overflow	208
fill.posArray	209
fill.style	209

Kapitel 16: Filter-Objekt

filter.angle	211
filter.blurX	211
filter.blurY	212
filter.brightness	212
filter.color	213
filter.contrast	214
filter.distance	214
filter.enabled	215
filter.hideObject	215
filter.highlightColor	216
filter.hue	216
filter.inner	217
filter.knockout	218
filter.name	218
filter.quality	219
filter.saturation	219
filter.shadowColor	220
filter.strength	221
filter.type	221

Kapitel 17: flash-Objekt (fl)

fl.actionsPanel	226
fl.addEventListener()	226
fl.as3PackagePaths	227
fl.browseForFileURL()	228
fl.browseForFolderURL()	229
fl.clearPublishCache()	229
fl.clipCopyString()	230
fl.closeAll()	230
fl.closeAllPlayerDocuments()	231
fl.closeDocument()	232
fl.compilerErrors	232
fl.componentsPanel	233
fl.configDirectory	233
fl.configURI	233
fl.contactSensitiveSelection	234
fl.createDocument()	234
fl.createNewDocList	235
fl.createNewDocListType	235
fl.createNewTemplateList	236
fl.documents	236
fl.drawingLayer	237
fl.exportPublishProfileString()	237
fl.externalLibraryPath	238
fl.fileExists()	238
fl.findDocumentDOM()	239

Inhalt

fl.findDocumentIndex()	239
fl.findObjectInDocByName()	240
fl.findObjectInDocByType()	241
fl.flexSDKPath	243
fl.getAppMemoryInfo()	243
fl.getDocumentDOM()	244
fl.getSwfPanel()	245
fl.installedPlayers	245
fl.isFontInstalled()	246
fl.languageCode	247
fl.libraryPath	247
fl.mapPlayerURL()	248
fl.Math	248
fl.mruRecentFileList	249
fl.mruRecentFileListType	249
fl.objectDrawingMode	250
fl.openDocument()	250
fl.openScript()	251
fl.outputPanel	252
fl.packagePaths	252
fl.presetPanel	252
fl.publishCacheDiskSizeMax	253
fl.publishCacheEnabled	253
fl.publishCacheMemoryEntrySizeLimit	254
fl.publishCacheMemorySizeMax	254
fl.publishDocument()	255
fl.quit()	255
fl.reloadEffects()	256
fl.reloadTools()	256
fl.removeEventListener()	257
fl.resetAS3PackagePaths()	258
fl.resetPackagePaths()	258
fl.revertDocument()	259
fl.runScript()	259
fl.saveAll()	260
fl.saveDocument()	261
fl.saveDocumentAs()	262
fl.scriptURI	262
fl.selectElement()	263
fl.selectTool()	264
fl.setActiveWindow()	265
fl.showIdleMessage()	266
fl.sourcePath	266
fl.swfPanels	267
fl.toggleBreakpoint()	267
fl.tools	268

Inhalt

fl.trace()	268
fl.version	269
fl.xmlui	269

Kapitel 18: FLfile-Objekt

FLfile.copy()	271
FLfile.createFolder()	272
FLfile.exists()	273
FLfile.getAttributes()	274
FLfile.getCreationDate()	275
FLfile.getCreationDateObj()	275
FLfile.getModificationDate()	276
FLfile.getModificationDateObj()	277
FLfile.getSize()	278
FLfile.listFolder()	278
FLfile.platformPathToURI()	279
FLfile.read()	280
FLfile.remove()	281
FLfile.setAttributes()	282
FLfile.uriToPlatformPath()	283
FLfile.write()	284

Kapitel 19: folderItem-Objekt**Kapitel 20: fontItem-Objekt**

fontItem.bitmap	286
fontItem.bold	287
fontItem.embeddedCharacters	287
fontItem.embedRanges	288
fontItem.embedVariantGlyphs	288
fontItem.font	290
fontItem.isDefineFont4Symbol	290
fontItem.italic	291
fontItem.size	291

Kapitel 21: Frame-Objekt

frame.convertMotionObjectTo2D()	293
frame.convertMotionObjectTo3D()	294
frame.actionScript	295
frame.duration	295
frame.elements	295
frame.getCustomEase()	296
frame.getMotionObjectXML()	297
frame.hasCustomEase	297
frame.hasMotionPath()	298
frame.is3DMotionObject()	298
frame.isMotionObject()	299
frame.labelType	299

Inhalt

frame.motionTweenOrientToPath	300
frame.motionTweenRotate	300
frame.motionTweenRotateTimes	301
frame.motionTweenScale	301
frame.motionTweenSnap	302
frame.motionTweenSync	302
frame.name	302
frame.selectMotionPath()	303
frame.setCustomEase()	303
frame.setMotionObjectDuration()	304
frame.setMotionObjectXML()	305
frame.shapeTweenBlend	305
frame.soundEffect	306
frame.soundLibraryItem	306
frame.soundLoop	307
frame.soundLoopMode	307
frame.soundName	307
frame.soundSync	308
frame.startFrame	308
frame.tweenEasing	309
frame.tweenInstanceName	309
frame.tweenType	309
frame.useSingleEaseCurve	310
 Kapitel 22: HalfEdge-Objekt	
halfEdge.getEdge()	311
halfEdge.getNext()	312
halfEdge.getOppositeHalfEdge()	312
halfEdge.getPrev()	313
halfEdge.getVertex()	314
halfEdge.id	314
halfEdge.index	315
 Kapitel 23: Instance-Objekt	
instance.instanceType	316
instance.libraryItem	317
 Kapitel 24: Item-Objekt	
item.addData()	319
item.getData()	319
item.hasData()	320
item.itemType	320
item.linkageBaseClass	321
item.linkageClassName	321
item.linkageExportForAS	322
item.linkageExportForRS	322
item.linkageExportInFirstFrame	323

Inhalt

item.linkageIdentifier	323
item.linkageImportForRS	324
item.linkageURL	324
item.name	325
item.removeData()	325

Kapitel 25: Layer-Objekt

layer.color	326
layer.frameCount	327
layer.frames	327
layer.height	328
layer.layerType	328
layer.locked	329
layer.name	329
layer.outline	329
layer.parentLayer	330
layer.visible	330

Kapitel 26: library-Objekt

library.addItemToDocument()	333
library.addNewItem()	334
library.deleteItem()	334
library.duplicateItem()	335
library.editItem()	335
library.expandFolder()	336
library.findItemIndex()	337
library.getItemProperty()	337
library.getItemType()	338
library.getSelectedItems()	338
library.importEmbeddedSWF()	339
library.itemExists()	340
library.items	340
library.moveToFolder()	340
library.newFolder()	341
library.renameItem()	342
library.selectAll()	342
library.selectItem()	343
library.selectNone()	344
library.setItemProperty()	344
library.updateItem()	345

Kapitel 27: Math-Objekt

Math.concatMatrix()	346
Math.invertMatrix()	347
Math.pointDistance()	347

Kapitel 28: Matrix-Objekt

matrix.a	349
matrix.b	350
matrix.c	350
matrix.d	351
matrix.tx	351
matrix.ty	352

Kapitel 29: outputPanel-Objekt

outputPanel.clear()	353
outputPanel.save()	354
outputPanel.trace()	354

Kapitel 30: Oval-Objekt

OvalObject.closePath	356
OvalObject.endAngle	357
OvalObject.innerRadius	357
OvalObject.startAngle	358

Kapitel 31: Parameter-Objekt

parameter.category	359
parameter.insertItem()	360
parameter.listIndex	360
parameter.name	361
parameter.removeItem()	361
parameter.value	362
parameter.valueType	363
parameter.verbose	363

Kapitel 32: Path-Objekt

path.addCubicCurve()	364
path.addCurve()	365
path.addPoint()	366
path.clear()	366
path.close()	367
path.makeShape()	367
path.newContour()	368
path.nPts	369

Kapitel 33: presetItem-Objekt

presetItem.isDefault	370
presetItem.isFolder	371
presetItem.level	371
presetItem.name	372
presetItem.open	372
presetItem.path	373

Kapitel 34: presetPanel-Objekt

presetPanel.addNewItem()	375
presetPanel.applyPreset()	375
presetPanel.deleteFolder()	376
presetPanel.deleteItem()	377
presetPanel.expandFolder()	378
presetPanel.exportItem()	378
presetPanel.findItemIndex()	379
presetPanel.getSelectedItems()	380
presetPanel.importItem()	381
presetPanel.items	382
presetPanel.moveToFolder()	382
presetPanel.newFolder()	383
presetPanel.renameItem()	384
presetPanel.selectItem()	384

Kapitel 35: Rectangle-Objekt

RectangleObject.bottomLeftRadius	386
RectangleObject.bottomRightRadius	387
RectangleObject.lockFlag	387
RectangleObject.topLeftRadius	388
RectangleObject.topRightRadius	388

Kapitel 36: Shape-Objekt

shape.beginEdit()	390
shape.contours	390
shape.deleteEdge()	391
shape.edges	391
shape.endEdit()	392
shape.getCubicSegmentPoints()	392
shape.isDrawingObject	393
shape.isGroup	393
shape.isOvalObject	394
shape.isRectangleObject	394
shape.members	395
shape.numCubicSegments	395
shape.vertices	396

Kapitel 37: SoundItem-Objekt

soundItem.bitRate	398
soundItem.bits	398
soundItem.compressionType	399
soundItem.convertStereoToMono	399
soundItem.exportToFile()	400
soundItem.fileLastModifiedDate	401
soundItem.originalCompressionType	401
soundItem.quality	402

Inhalt

soundItem.sampleRate	402
soundItem.sourceFileExists	403
soundItem.sourceFilesCurrent	403
soundItem.sourceFilePath	404
soundItem.useImportedMP3Quality	404

Kapitel 38: Stroke-Objekt

stroke.breakAtCorners	406
stroke.capType	406
stroke.color	407
stroke.curve	407
stroke.dash1	408
stroke.dash2	408
stroke.density	409
stroke.dotSize	409
stroke.dotSpace	410
stroke.hatchThickness	410
stroke.jiggle	411
stroke.joinType	411
stroke.length	411
stroke.miterLimit	412
stroke.pattern	412
stroke.rotate	413
stroke.scaleType	413
stroke.shapeFill	414
stroke.space	414
stroke.strokeHinting	415
stroke.style	415
stroke.thickness	416
stroke.variation	416
stroke.waveHeight	417
stroke.waveLength	417

Kapitel 39: swfPanel-Objekt

swfPanel.call()	419
swfPanel.name	421
swfPanel.path	422
swfPanel.setFocus()	422

Kapitel 40: SymbolInstance-Objekt

symbolInstance.accName	425
symbolInstance.actionScript	426
symbolInstance.backgroundColor	426
symbolInstance.bitmapRenderMode	427
symbolInstance.blendMode	427
symbolInstance.buttonTracking	428
symbolInstance.cacheAsBitmap	428

Inhalt

symbolInstance.colorAlphaAmount	429
symbolInstance.colorAlphaPercent	429
symbolInstance.colorBlueAmount	429
symbolInstance.colorBluePercent	430
symbolInstance.colorGreenAmount	430
symbolInstance.colorGreenPercent	430
symbolInstance.colorMode	431
symbolInstance.colorRedAmount	431
symbolInstance.colorRedPercent	432
symbolInstance.description	432
symbolInstance.filters	433
symbolInstance.firstFrame	433
symbolInstance.forceSimple	434
symbolInstance.loop	434
symbolInstance.shortcut	435
symbolInstance.silent	435
symbolInstance.symbolType	436
symbolInstance.tabIndex	436
symbolInstance.usesBackgroundColor	436
symbolInstance.visible	437

Kapitel 41: SymbolItem-Objekt

symbolItem.convertToCompiledClip()	438
symbolItem.exportSWC()	439
symbolItem.exportSWF()	439
symbolItem.scalingGrid	440
symbolItem.scalingGridRect	440
symbolItem.sourceAutoUpdate	441
symbolItem.sourceFilePath	441
symbolItem.sourceLibraryName	442
symbolItem.symbolType	442
symbolItem.timeline	442

Kapitel 42: Text-Objekt

text.accName	445
text.antiAliasSharpness	446
text.antiAliasThickness	446
text.autoExpand	447
text.border	447
text.description	448
text.embeddedCharacters	448
text.embedRanges	449
text.embedVariantGlyphs	449
text.fontRenderingMode	450
text.getTextAttr()	451
text.getTextString()	451
text.length	452

Inhalt

text.lineType	453
text.maxCharacters	453
text.orientation	453
text.renderAsHTML	454
text.scrollable	454
text.selectable	455
text.selectionEnd	455
text.selectionStart	455
text.setTextAttr()	456
text.setTextString()	457
text.shortcut	458
text.silent	458
text.tabIndex	459
text.textRuns	459
text.textType	460
text.useDeviceFonts	460
text.variableName	460

Kapitel 43: TextAttrs-Objekt

textAttrs.aliasText	463
textAttrs.alignment	463
textAttrs.autoKern	463
textAttrs.bold	464
textAttrs.characterPosition	464
textAttrs.characterSpacing	465
textAttrs.face	465
textAttrs.fillColor	465
textAttrs.indent	466
textAttrs.italic	466
textAttrs.leftMargin	467
textAttrs.letterSpacing	467
textAttrs.lineSpacing	467
textAttrs.rightMargin	468
textAttrs.rotation	468
textAttrs.size	469
textAttrs.target	469
textAttrs.url	469

Kapitel 44: TextRun-Objekt

textRun.textAttrs	471
textRun.characters	471

Kapitel 45: Timeline-Objekt

timeline.addMotionGuide()	475
timeline.addNewLayer()	476
timeline.clearFrames()	477
timeline.clearKeyframes()	477

timeline.convertToBlankKeyframes()	478
timeline.convertToKeyframes()	479
timeline.copyFrames()	479
timeline.copyLayers()	480
timeline.copyMotion()	481
timeline.copyMotionAsAS3()	481
timeline.createMotionObject()	482
timeline.createMotionTween()	483
timeline.currentFrame	484
timeline.currentLayer	484
timeline.cutFrames()	484
timeline.cutLayers()	485
timeline.deleteLayer()	486
timeline.duplicateLayers()	487
timeline.expandFolder()	487
timeline.findLayerIndex()	488
timeline.frameCount	489
timeline.getFrameProperty()	489
timeline.getGuidelines()	490
timeline.getLayerProperty()	491
timeline.getSelectedFrames()	491
timeline.getSelectedLayers()	492
timeline.insertBlankKeyframe()	492
timeline.insertFrames()	493
timeline.insertKeyframe()	494
timeline.layerCount	495
timeline.layers	496
timeline.libraryItem	496
timeline.name	496
timeline.pasteFrames()	497
timeline.pasteLayers()	498
timeline.pasteMotion()	498
timeline.removeFrames()	499
timeline.removeMotionObject()	500
timeline.reorderLayer()	500
timeline.reverseFrames()	501
timeline.selectAllFrames()	502
timeline setFrameProperty()	502
timeline.setGuidelines()	503
timeline.setLayerProperty()	504
timeline.setSelectedFrames()	504
timeline.setSelectedLayers()	505
timeline.showLayerMasking()	506
timeline.startPlayback()	507
timeline.stopPlayback()	507

Kapitel 46: ToolObj-Objekt

toolObj.depth	509
toolObj.enablePIControl()	509
toolObj.iconID	510
toolObj.position	511
toolObj.setIcon()	511
toolObj.setMenuString()	512
toolObj.setOptionsFile()	512
toolObj.setPI()	513
toolObj.setToolName()	514
toolObj.setToolTip()	514
toolObj.showPIControl()	515
toolObj.showTransformHandles()	516

Kapitel 47: Tools-Objekt

tools.activeTool	518
tools.altIsDown	518
tools.constrainPoint()	518
tools.ctrlIsDown	519
tools.getKeyDown()	519
tools.mouselsDown	520
tools.penDownLoc	520
tools.penLoc	521
tools.setCreatingBbox()	521
tools.setCursor()	522
tools.shiftIsDown	522
tools.snapPoint()	523
tools.toolObjs	523

Kapitel 48: Vertex-Objekt

vertex.getHalfEdge()	524
vertex.setLocation()	525
vertex.x	525
vertex.y	526

Kapitel 49: VideoItem-Objekt

videoItem.exportToFLV()	527
videoItem.fileLastModifiedDate	528
videoItem.sourceFileExists	528
videoItem.sourceFilesCurrent	529
videoItem.sourceFilePath	529
videoItem.videoType	530

Kapitel 50: XMLUI-Objekt

xmlui.accept()	531
xmlui.cancel()	532
xmlui.get()	532
xmlui.getControlItemElement()	533

xmlui.setEnabled()	534
xmlui.getVisible()	534
xmlui.set()	535
xmlui.setControlItemElement()	535
xmlui.setControlItemElements()	536
xmlui.setEnabled()	537
xmlui.setVisible()	538

Kapitel 51: C-Level-Erweiterbarkeit

Erweiterbarkeit	539
Integrieren von C-Funktionen	539
Datentypen	545
Die C-Level-API	545

Kapitel 1: Einführung

Als Benutzer von Adobe® Flash® Professional CS5 oder CS5.5 sind Sie unter Umständen bereits mit Adobe® ActionScript® vertraut, einem Programm zum Erstellen von Skripten, die zur Laufzeit in Adobe® Flash® Player ausgeführt werden. Bei der Flash JavaScript-API (Application Programming Interface – Anwendungsprogrammierschnittstelle), die in diesem Dokument beschrieben wird, handelt es sich um ein zusätzliches Programmierwerkzeug zur Erstellung von Skripten, die in der Authoring-Umgebung ausgeführt werden können.

In diesem Dokument finden Sie die Beschreibungen der Objekte, Methoden und Eigenschaften der JavaScript-API. Dabei wird davon ausgegangen, dass Sie bei der Arbeit mit der Authoring-Umgebung mit den dokumentierten Befehlen vertraut sind. Sollten Ihrerseits Fragen zu den Funktionen bestimmter Befehle offen sein, können Sie sich in anderen Dokumenten der Flash-Hilfe informieren, wie z. B. in *Verwenden von Flash*.

In diesem Dokument wird außerdem davon ausgegangen, dass Sie auch mit der JavaScript- bzw. der ActionScript-Syntax sowie mit grundlegenden Begriffen der Programmierung wie Funktionen, Parametern und Datentypen vertraut sind.

Verwenden der JavaScript-API

Mit der Flash JavaScript-API können Sie Skripte erstellen, die in der Flash-Authoring-Umgebung mehrere Aktionen ausführen (d. h. während das Flash-Programm geöffnet ist). Diese Funktion unterscheidet sich von der ActionScript-Sprache, mit der Sie Skripte schreiben können, die in der Flash Player-Umgebung bestimmte Aktionen ausführen (d. h. während eine SWF-Datei abgespielt wird). Diese Funktion unterscheidet sich ebenfalls von JavaScript-Befehlen, die Sie in den Seiten, die in einem Webbrowser angezeigt werden, verwenden können.

Mit der JavaScript-API können Sie Flash-Anwendungsskripte schreiben, um den Authoring-Prozess zu optimieren. So können Sie beispielsweise Skripte schreiben, die häufig auftretende Aufgaben automatisch ausführen oder benutzerdefinierte Tools zum Bedienfeld „Werkzeuge“ hinzufügen.

Die Flash JavaScript-API ähnelt der Adobe® Dreamweaver® und Adobe® Fireworks® JavaScript-API, deren Design wiederum auf der Netscape JavaScript-API beruht. Sie basiert auf einem DOM (Document Object Model – Dokumentobjektmodell), das den Zugriff auf Flash-Dokumente über JavaScript-Objekte ermöglicht, und enthält sämtliche Elemente der Netscape JavaScript-API sowie das Flash-DOM. Diese zusätzlichen Objekte und die zugehörigen Methoden und Eigenschaften werden in diesem Dokument beschrieben. Sie können in einem Flash-Skript zwar beliebige Elemente der nativen JavaScript-Sprache verwenden, aber nur Elemente, die innerhalb eines Flash-Dokuments sinnvoll sind, haben eine Wirkung.

Die JavaScript-API enthält außerdem Methoden, mit denen Sie anhand einer Kombination von JavaScript und benutzerdefiniertem C-Code Erweiterungsdateien implementieren können. Weitere Informationen finden Sie unter „[C-Level-Erweiterbarkeit](#)“ auf Seite 539.

Die Mozilla SpiderMonkey-Engine (Version 1.6) ist der JavaScript-Interpreter von Flash. Sie ist unter www.mozilla.org/js/spidermonkey/ erhältlich. SpiderMonkey ist eine der beiden von Mozilla.org entwickelten Referenzimplementierungen der JavaScript-Sprache. Diese Engine ist auch in den Mozilla-Browser eingebettet.

SpiderMonkey implementiert die JavaScript-Kernsprache in Übereinstimmung mit der ECMAScript Edition 3-Sprachenspezifikation (ECMA-262) und ist mit dieser Spezifikation konform. Nur die browserspezifischen Hostobjekte, die nicht der ECMA-262-Spezifikation angehören, werden nicht unterstützt. Viele JavaScript-Referenzhandbücher unterscheiden zwischen Core-JavaScript und clientseitigem (browserspezifischem) JavaScript. Für den Flash JavaScript-Interpreter gilt nur Core-JavaScript.

Erstellen von JSFL-Dateien

Flash JavaScript-Dateien (JSFL) können mit Adobe Flash Professional oder einem anderen Texteditor geschrieben und geändert werden. Wenn Sie mit Flash arbeiten, haben diese Dateien standardmäßig die Erweiterung .jsfl. Um ein Skript zu schreiben, wählen Sie „Datei“ > „Neu“ > „Flash JavaScript-Datei“.

Eine andere Möglichkeit zum Erstellen einer JSFL-Datei besteht darin, im Bedienfeld „Protokoll“ einen oder mehrere Befehle auszuwählen. Klicken Sie dann in dem Bedienfeld auf die Schaltfläche „Speichern“, oder wählen Sie im Bedienfeld-Menü den Befehl „Als Befehl speichern“. Die Befehlsdatei (JSFL) wird im Ordner „Commands“ gespeichert (weitere Hinweise finden Sie unter „[Speichern von JSFL-Dateien](#)“ auf Seite 2). Sie können diese Datei genau wie andere Skriptdateien öffnen und bearbeiten.

Das Bedienfeld „Protokoll“ enthält aber auch noch andere sinnvolle Funktionen. So können Sie ausgewählte Befehle in die Zwischenablage kopieren oder erstellte JavaScript-Befehle während der Arbeit in Flash anzeigen.

So kopieren Sie Befehle aus dem Bedienfeld „Protokoll“ in die Zwischenablage:

- 1 Wählen Sie im Bedienfeld „Protokoll“ mindestens einen Befehl aus.
- 2 Führen Sie einen der folgenden Schritte aus:
 - Klicken Sie auf die Schaltfläche „Kopieren“.
 - Wählen Sie im Bedienfeld-Menü die Option „Schritte kopieren“.

So zeigen Sie JavaScript-Befehle im Bedienfeld „Protokoll“ an:

- Wählen Sie im Bedienfeld-Menü den Befehl „Ansicht“ > „JavaScript in Bedienfeld“.

Speichern von JSFL-Dateien

Wenn Sie Ihre JSFL-Skripts in einem der Ordner unter „Configuration“ speichern, können Sie während der Arbeit in der Flash-Authoring-Umgebung auf diese Skripts zugreifen. In der Standardeinstellung befindet sich der Ordner „Configuration“ an folgender Stelle:

- Windows® 7™:
Startlaufwerk\Benutzer\Benutzername\Anwendungsdaten\Lokal\Adobe\Flash CS5 oder CS5.5\Sprache\Configuration
- Windows® Vista™:
Startlaufwerk\Benutzer\Benutzername\Lokale Einstellungen\Anwendungsdaten\Adobe\Flash CS5 oder CS5.5\Sprache\Configuration
- Windows XP:
Startlaufwerk\Dokumente und Einstellungen\Benutzername\Lokale Einstellungen\Anwendungsdaten\Adobe\Flash CS5 oder CS5.5\Sprache\Configuration
- Mac OS® X:

Macintosh HD/Users/*Benutzername*/Library/Application Support/Adobe/Flash CS5 *oder*
CS5.5/*Sprache*/Configuration/

Um festzustellen, wo sich der Ordner „Configuration“ befindet, verwenden Sie `fl.configDirectory` oder `fl.configURI` wie in den folgenden Beispielen:

```
// store directory to a variable
var configDir = fl.configDirectory;
// display directory in the Output panel
fl.trace(fl.configDirectory);
```

Im Ordner „Configuration“ enthalten ggf. auch die folgenden Unterverzeichnisse Skripts, zu denen Sie in der Authoring-Umgebung Zugang haben: Behaviors (zur Unterstützung der Benutzeroberfläche für Verhalten), Commands (mit Skripts für das Menü „Befehle“), JavaScript (mit Skripts für die Skripthilfe zum Ausfüllen der Steuerelemente der Benutzeroberfläche), Tools (mit erweiterbaren Werkzeugen für das Bedienfeld „Werkzeuge“) sowie WindowSWF (mit Bedienfeldern für das Menü „Fenster“). In diesem Dokument geht es hauptsächlich um die Skripts für Befehle und Werkzeuge.

Wenn Sie im Ordner „Commands“ ein Skript bearbeiten, steht Ihnen dieses Skript unmittelbar in Flash zur Verfügung. Bei den Skripts für erweiterbare Werkzeuge müssen Sie Flash schließen und erneut starten bzw. den Befehl `fl.reloadTools()` verwenden. Wenn Sie ein erweiterbares Werkzeug mithilfe eines Skripts zum Bedienfeld „Werkzeuge“ hinzugefügt haben und dieses Skript dann ändern, so müssen Sie das Werkzeug zuerst aus dem Bedienfeld „Werkzeuge“ löschen und dann wieder hinzufügen oder Flash schließen und erneut starten, damit das Werkzeug zur Verfügung steht.

Es gibt zwei Speicherorte für Befehls- und Werkzeugdateien, die einen reibungslosen Zugriff von der Authoring-Umgebung aus gewährleisten.

- Speichern Sie die JSFL-Dateien für Skripts, die dem Menü „Befehle“ in Form von Menübefehlen hinzugefügt werden, im Ordner „Commands“ an folgendem Speicherort:

Betriebssystem	Pfad
Windows 7	<i>Startlaufwerk</i> \Benutzer\ <i>Benutzername</i> \Anwendungsdaten\Lokal\Adobe\Flash CS5 <i>oder</i> CS5.5\ <i>Sprache</i> \Configuration\Commands
Windows Vista	<i>Startlaufwerk</i> \Benutzer\ <i>Benutzername</i> \Lokale Einstellungen\Anwendungsdaten\Adobe\Flash CS5\ <i>Sprache</i> \Configuration\Commands
Windows XP	<i>Startlaufwerk</i> \Dokumente und Einstellungen\ <i>Benutzername</i> \Lokale Einstellungen\Anwendungsdaten\Adobe\Flash CS5\ <i>Sprache</i> \Configuration\Commands
Mac OS X	Macintosh HD/Users/ <i>Benutzername</i> /Library/Application Support/Adobe/Flash CS5/ <i>Sprache</i> /Configuration/Commands

- Speichern Sie die JSFL-Dateien für Skripts, die dem Bedienfeld „Werkzeuge“ in Form von erweiterbaren Werkzeugen hinzugefügt werden, im Ordner „Tools“ an folgendem Speicherort:

Betriebssystem	Pfad
Windows 7	<i>Startlaufwerk</i> \Benutzer\ <i>Benutzername</i> \Anwendungsdaten\Lokal\Adobe\Flash CS5 <i>oder</i> CS5.5\ <i>Sprache</i> \Configuration\Tools

Betriebssystem	Pfad
Windows Vista	<i>Startlaufwerk\Benutzer\Benutzername\Lokale Einstellungen\Anwendungsdaten\Adobe\Flash CS5\Sprache\Configuration\Tools</i>
Windows XP	<i>Startlaufwerk\Dokumente und Einstellungen\Benutzername\Lokale Einstellungen\Anwendungsdaten\Adobe\Flash CS5\Sprache\Configuration\Tools</i>
Mac OS X	<i>Macintosh HD/Users/Benutzername/Library/Application Support/Adobe/Flash CS5/Sprache/Configuration/Tools</i>

Wenn eine JSFL-Datei durch weitere Dateien wie z. B. XML-Dateien ergänzt wird, sollten diese Dateien in demselben Verzeichnis wie die JSFL-Datei abgelegt werden.

Ausführen von Skripten

Es gibt mehrere Möglichkeiten, um Skripte auszuführen. Die gebräuchlichsten Verfahren werden in diesem Abschnitt beschrieben.

So führen Sie ein Skript aus, das Sie gerade anzeigen oder bearbeiten:

- Klicken Sie mit der rechten Maustaste bzw. bei gedrückter Befehlstaste (Macintosh) darauf und wählen Sie „Skript ausführen“.
- Klicken Sie in der Symbolleiste des Skriptfensters auf das Symbol „Skript ausführen“.

Diese Option ermöglicht es Ihnen, ein Skript auszuführen, bevor Sie es speichern. Außerdem ermöglicht sie, ein Skript auszuführen, selbst wenn keine FLA-Dateien geöffnet sind.

Wenn Sie ein Skript ausführen möchten, das sich im Ordner „Commands“ befindet, führen Sie einen der folgenden Schritte aus:

- Wählen Sie in der Authoring-Umgebung „Befehle“ > „Script Name“ aus.
- Verwenden Sie den Tastaturbefehl, den Sie dem Skript zugewiesen haben. Wählen Sie „Bearbeiten“ > „Tastaturbefehle“ und dann im Popupmenü „Befehle“ die Option „Befehle im Zeichenmenü“. Erweitern Sie im Menüverzeichnis den Punkt „Befehle“, um eine Liste der zur Verfügung stehenden Skripts anzuzeigen.

Wenn Sie dagegen ein Befehlsskript ausführen möchten, das sich nicht im Ordner „Commands“ befindet, führen Sie einen der folgenden Schritte aus:

- Wählen Sie in der Authoring-Umgebung „Befehle“ > „Befehl ausführen“ und dann das auszuführende Skript aus.
- In einem Skript können Sie ein anderes Skript mithilfe des Befehls `fl.runScript()` ausführen.
- Im Dateisystem doppelklicken Sie auf die Skriptdatei.

So fügen Sie dem Bedienfeld „Werkzeuge“ ein in einer JSFL-Datei implementiertes Werkzeug hinzu:

- 1 Kopieren Sie die JSFL-Datei für das Werkzeug sowie alle zugehörigen Dateien in den Ordner „Tools“ (siehe „Speichern von JSFL-Dateien“ auf Seite 2).
- 2 Wählen Sie „Bearbeiten“ > „Werkzeug-Bedienfeld anpassen“ (Windows) bzw. „Flash“ > „Werkzeug-Bedienfeld anpassen“ (Macintosh).
- 3 Fügen Sie das Werkzeug der Liste der verfügbaren Werkzeuge hinzu.
- 4 Klicken Sie auf „OK“.

Mithilfe der Funktion `MMEExecute()`, die im *Komponenten-Referenzhandbuch für ActionScript 3.0* ausführlich erläutert wird, können Sie einzelne JavaScript-API-Befehle zu ActionScript-Dateien hinzufügen. Allerdings hat die Funktion `MMEExecute()` nur dann eine Wirkung, wenn sie in Verbindung mit einem benutzerdefinierten Benutzeroberflächenelement eingesetzt wird, z. B. mit dem Eigenschafteninspektor einer Komponente oder mit einem SWF-Bedienfeld innerhalb der Authoring-Umgebung. JavaScript-API-Befehle haben im Flash Player oder außerhalb der Authoring-Umgebung selbst dann keine Auswirkungen, wenn sie von ActionScript aufgerufen werden.

So führen Sie einen Befehl mithilfe eines ActionScripts aus:

- Verwenden Sie die folgende Syntax (dabei können Sie mehrere Befehle auch in einem String zusammenfassen):

```
MMEExecute(Javascript command string);
```

Sie können ein Skript auch von der Befehlszeile ausführen.

So führen Sie ein Skript von der Befehlszeile von Windows aus:

- Verwenden Sie die folgende Syntax (ggf. unter Berücksichtigung von Pfadangaben):

```
"flash.exe" myTestFile.jsfl
```

So führen Sie ein Skript von der Anwendung „Terminal“ auf Macintosh-Systemen aus:

- Verwenden Sie die folgende Syntax (ggf. unter Berücksichtigung von Pfadangaben):

```
osascript -e 'tell application "flash" to open alias "Mac OS X:Users:user:myTestFile.jsfl" '
```

Der Befehl `osascript` kann auch AppleScript in einer Datei ausführen. So können Sie beispielsweise den folgenden Text in eine Datei mit dem Namen „myScript“ einfügen:

```
tell application "flash"
open alias "Mac OS X:Users:user:myTestFile.jsfl"
end tell
```

und dann das Skript mithilfe dieses Befehls ausführen:

```
osascript myScript
```

Neu in der JavaScript-API

In Flash CS5 und CS5.5 wurden einige Objekte, Methoden und Eigenschaften hinzugefügt, während andere entfernt wurden. Diese Änderungen werden unten zusammengefasst.

Wenn Sie bisher noch nicht mit der JavaScript-API gearbeitet haben, können Sie diesen Abschnitt ignorieren und über die „[JavaScript-API-Objekte](#)“ auf Seite 7 weiterlesen.

Neue Methoden und Eigenschaften

Die folgenden Methoden und Eigenschaften für vorhandene Objekte sind neu in Flash Pro CS5:

- [Document-Objekt](#)
 - `document.debugMovie()`
 - `document.loadCuepointXML()`
- [flash-Objekt \(fl\)](#)
 - `fl.languageCode`

- `fl.toggleBreakpoint`
- **Frame-Objekt**
 - `frame.convertMotionObjectTo2D()`
 - `frame.convertMotionObjectTo3D()`
 - `frame.getMotionObjectXML()`
 - `frame.hasMotionPath()`
 - `frame.isMotionObject()`
 - `frame.is3DMotionObject()`
 - `frame.selectMotionPath()`
 - `frame.setMotionObjectDuration()`
 - `frame.setMotionObjectXML()`
 - `frame.tweenInstanceName`

- **Timeline-Objekt**
 - `timeline.createMotionObject()`
 - `timeline.libraryItem`
 - `timeline.removeMotionObject()`
 - `timeline.startPlayback`
 - `timeline.stopPlayback`

Die folgenden Methoden und Eigenschaften für vorhandene Objekte sind neu in Flash Pro CS5.5:

- **SymbolInstance-Objekt**
 - `symbolInstance.bitmapRenderMode`
 - `symbolInstance.backgroundColor`
 - `symbolInstance.usesBackgroundColor`
 - `symbolInstance.visible`
- **Timeline-Objekt**
 - `timeline.copyLayers()`
 - `timeline.cutLayers()`
 - `timeline.duplicateLayers()`
 - `timeline.pasteLayers()`
- **flash-Objekt (fl)**
 - `fl.getSwfPanel()`
 - `fl.installedPlayers()`
 - `fl.publishCacheEnabled`
 - `fl.publishCacheDiskSizeMax`
 - `fl.publishCacheMemorySizeMax`
 - `fl.publishCacheMemoryEntrySizeLimit`
 - `fl.clearPublishCache()`

- [swfPanel-Objekt](#)
 - `swfPanel.setFocus()`

Weitere Änderungen

Die folgenden Methoden und Eigenschaften wurden in Flash CS5 aktualisiert:

- `fl.openScript()`
- `fl.publishDocument()`
- `fontItem.embedRanges`
- `fontItem.embeddedCharacters`
- `fontItem.embedVariantGlyphs`

Die folgenden Objekte und Methoden stehen in Flash CS5 nicht mehr zur Verfügung:

- Screen-Objekt
- ScreenOutline-Objekt
- `document.canSaveAVersion()`
- `document.revertToLastVersion()`
- `document.saveAVersion()`
- `document.synchronizeWithHeadVersion()`
- `fl.downloadLatestVersion()`
- `fl.revertDocumentToLastVersion()`
- `fl.saveAVersionOfDocument()`
- `fl.synchronizeDocumentWithHeadVersion()`

JavaScript-API-Objekte

In diesem Abschnitt finden Sie eine Übersicht über die Objekte, die in der JavaScript-API von Flash verfügbar sind, sowie Informationen zur Verwendung dieser Objekte. Alle standardmäßigen JavaScript-Befehle stehen beim Verwenden der JavaScript-API ebenfalls zur Verfügung.

In der folgenden Tabelle werden die einzelnen Objekte der JavaScript-API kurz beschrieben. Die Objekte sind in alphabetischer Reihenfolge aufgeführt.

Objekt	Beschreibung
actionsPanel-Objekt	Das actionsPanel-Objekt repräsentiert das zurzeit angezeigte Bedienfeld „Aktionen“.
BitmapInstance-Objekt	Das BitmapInstance-Objekt ist eine Unterklasse des Instance-Objekts und entspricht einer Bitmap in einem Bild.
BitmapItem-Objekt	Ein BitmapItem-Objekt bezieht sich auf eine Bitmap in der Bibliothek eines Dokuments. Das BitmapItem-Objekt ist eine Unterklasse des Item-Objekts.
CompiledClipInstance-Objekt	Das CompiledClipInstance-Objekt ist eine Unterklasse des Instance-Objekts.
compilerErrors-Objekt	Das compilerErrors-Objekt repräsentiert das Bedienfeld „Compiler-Fehler“. Es handelt sich um eine Eigenschaft des flash-Objekts (<code>fl.compilerErrors</code>).

Objekt	Beschreibung
ComponentInstance-Objekt	Das ComponentInstance-Objekt ist eine Unterklasse des SymbolInstance-Objekts und entspricht einer Komponente in einem Bild.
componentsPanel-Objekt	Das componentsPanel-Objekt, welches das Bedienfeld darstellt, ist eine Eigenschaft des flash-Objekts (<code>fl.componentsPanel</code>).
Contour-Objekt	Ein Contour-Objekt stellt einen geschlossenen Pfad von Halbkanten auf der Begrenzung einer Form dar.
Document-Objekt	Das Document-Objekt repräsentiert die Bühne.
drawingLayer-Objekt	Das drawingLayer-Objekt ist ein Child-Objekt des flash-Objekts. Der Zugriff auf dieses Objekt kann daher von JavaScript aus erfolgen.
Edge-Objekt	Das Edge-Objekt stellt eine Kante einer Form auf der Bühne dar.
Element-Objekt	Alles, was Sie auf der Bühne sehen, sind Element-Objekte.
Fill-Objekt	Das Fill-Objekt enthält alle Eigenschaften der Einstellung „Füllfarbe“ des Bedienfelds „Werkzeuge“ bzw. einer ausgewählten Form.
Filter-Objekt	Das Filter-Objekt enthält alle Eigenschaften für alle Filter.
flash-Objekt (fl)	Das flash-Objekt stellt die Flash-Anwendung dar.
FLfile-Objekt	Mit dem FLfile-Objekt können Sie Flash-Erweiterungen schreiben, die im lokalen Dateisystem auf Dateien und Ordner zugreifen und diese modifizieren und entfernen können.
folderItem-Objekt	Das folderItem-Objekt ist eine Unterklasse des Item-Objekts.
fontItem-Objekt	Das fontItem-Objekt ist eine Unterklasse des Item-Objekts.
Frame-Objekt	Das Frame-Objekt stellt Bilder in der Ebene dar.
HalfEdge-Objekt	Gerichtete Hälfte der Kante eines Shape-Objekts.
Instance-Objekt	Das Instance-Objekt ist eine Unterklasse des Element-Objekts.
Item-Objekt	Das Item-Objekt ist eine abstrakte Basisklasse.
Layer-Objekt	Das Layer-Objekt stellt eine Ebene in der Zeitleiste dar.
library-Objekt	Das library-Objekt stellt das Bedienfeld „Bibliothek“ dar
Math-Objekt	Das Math-Objekt ist als eine schreibgeschützte Eigenschaft des flash-Objekts (<code>fl.Math</code>) verfügbar.
Matrix-Objekt	Das Matrix-Objekt stellt eine Transformationsmatrix dar.
outputPanel-Objekt	Das outputPanel-Objekt stellt das Bedienfeld „Ausgabe“ dar, in dem Informationen zur Behebung von Fehlern wie z. B. Syntaxfehlern angezeigt werden. Ist eine Eigenschaft des flash-Objekts (<code>fl.outputPanel</code>).
Oval-Objekt	Das Oval-Objekt ist eine Form, die mit dem Ellipsenwerkzeug gezeichnet wird. Um festzustellen, ob ein Element ein Oval-Objekt ist, verwenden Sie <code>shape.isOvalObject</code> .
Parameter-Objekt	Der Zugriff auf das Parameter-Objekt erfolgt über das Array <code>screen.parameters</code> (entspricht dem Eigenschafteninspektor für Bildschirme im Flash-Authoring-Tool) bzw. über das Array <code>componentInstance.parameters</code> (entspricht dem Eigenschafteninspektor für Komponenten im Authoring-Tool).
Path-Objekt	Das Path-Objekt definiert eine Sequenz von Liniensegmenten (gerade, gebogen oder beides), die in der Regel beim Erstellen erweiterbarer Werkzeuge eingesetzt werden.
presetItem-Objekt	Das presetItem-Objekt repräsentiert ein Element (Voreinstellung oder Ordner) im Bedienfeld „Bewegungsvoreinstellungen“.

Objekt	Beschreibung
presetPanel-Objekt	Das presetPanel-Objekt repräsentiert das Bedienfeld „Bewegungsvoreinstellungen“ („Fenster“ > „Bewegungsvoreinstellungen“). Es ist eine Eigenschaft des flash-Objekts (<code>fl.presetPanel</code>).
Rectangle-Objekt	Das Rectangle-Objekt ist eine Form, die mit dem Rechteckwerkzeug gezeichnet wird. Um festzustellen, ob ein Element ein Rectangle-Objekt ist, verwenden Sie <code>shape.isRectangleObject</code> .
Screen-Objekt	Das Screen-Objekt stellt einen einzelnen Bildschirm in einem Folien- oder Formulardokument dar.
ScreenOutline-Objekt	Das ScreenOutline-Objekt stellt die Bildschirmgruppe dar, die sich in einem Folien- oder Formulardokument befindet.
Shape-Objekt	Das Shape-Objekt ist eine Unterklasse des Element-Objekts. Das Shape-Objekt bietet bei der Bearbeitung und beim Erstellen von Formen auf der Bühne präzisere Steuerungsmöglichkeiten als die Zeichnungs-APIs.
SoundItem-Objekt	Das SoundItem-Objekt ist eine Unterklasse des Item-Objekts. Es stellt ein Bibliothekselement dar, anhand dessen ein Sound erstellt wird.
Stroke-Objekt	Das Stroke-Objekt enthält sämtliche Einstellungen für einen Strich, einschließlich der benutzerdefinierten Einstellungen.
swfPanel-Objekt	Das swfPanel-Objekt repräsentiert ein Windows-SWF-Bedienfeld. Windows-SWF-Bedienfelder sind SWF-Dateien, die Anwendungen implementieren, die von der Flash-Authoringumgebung aus ausgeführt werden können. Das Array von swfPanel-Objekten ist eine Eigenschaft des flash-Objekts (<code>fl.swfPanels</code>).
SymbolInstance-Objekt	Das SymbolInstance-Objekt ist eine Unterklasse des Instance-Objekts und entspricht einem Symbol in einem Bild.
SymbolItem-Objekt	Das SoundItem-Objekt ist eine Unterklasse des Item-Objekts.
Text-Objekt	Das Text-Objekt stellt ein einzelnes Textelement in einem Dokument dar.
TextAttrs-Objekt	Das TextAttrs-Objekt enthält alle Texteingenschaften, die auf eine Unterauswahl angewendet werden können. Dieses Objekt ist eine Unterklasse des Text-Objekts.
TextRun-Objekt	Das TextRun-Objekt stellt eine Zeichenreihe dar, deren Attribute mit allen Eigenschaften des TextAttrs-Objekts übereinstimmen.
Timeline-Objekt	Das Timeline-Objekt entspricht der Flash-Zeitleiste. Sie können über <code>fl.getDocumentDOM().getTimeline()</code> auf die Flash-Zeitleiste des aktuellen Dokuments zugreifen.
ToolObj-Objekt	Ein ToolObj-Objekt stellt ein einzelnes Werkzeug im Bedienfeld „Werkzeuge“ dar.
Tools-Objekt	Sie können vom Flash-Objekt (<code>fl.tools</code>) aus auf das Tools-Objekt zugreifen.
Vertex-Objekt	Das Vertex-Objekt ist der Teil der Formdatenstruktur, der die Koordinatendaten enthält.
VideoItem-Objekt	Beim VideoItem-Objekt handelt es sich um eine Unterklasse des Item-Objekts.
XMLUI-Objekt	Mit dem XMLUI-Objekt können Sie die Eigenschaften eines XMLUI-Dialogfelds abrufen und festlegen, die Einstellungen eines XMLUI-Dialogfelds akzeptieren sowie ein XMLUI-Dialogfeld schließen, ohne die Einstellungen zu akzeptieren.

Das Flash-Dokumentobjektmodell

Das Flash-DOM (Document Object Model) für die Flash JavaScript-API besteht aus einem Satz Top-Level-Funktionen (siehe „[Top-Level-Funktionen und -Methoden](#)“ auf Seite 15) und den beiden Top-Level-Objekten „FLfile“ und „flash (fl)“. Beide Objekte sind immer vorhanden, wenn die Flash-Authoring-Umgebung geöffnet ist und daher grundsätzlich für Skripts verfügbar. Weitere Informationen finden Sie unter [FLfile-Objekt](#) und [flash-Objekt \(fl\)](#).

Bei dem flash-Objekt sind die beiden Schreibweisen `flash` und `fl` zulässig. Wenn Sie z. B. alle geöffneten FLA-Dateien schließen möchten, stehen Ihnen die beiden folgenden Anweisungen zur Auswahl:

```
flash.closeAll();
fl.closeAll();
```

Das flash-Objekt enthält die folgenden *Child*-Objekte:

Objekt	Zugriff
actionsPanel-Objekt	Mit <code>fl.actionsPanel</code> können Sie auf das actionsPanel-Objekt zugreifen. Dieses Objekt entspricht dem Bedienfeld „Aktionen“ in der Flash-Authoring-Umgebung.
compilerErrors-Objekt	Mit <code>fl.compilerErrors</code> können Sie auf das compilerErrors-Objekt zugreifen. Dieses Objekt entspricht dem Bedienfeld „Compiler-Fehler“ in der Flash-Authoring-Umgebung.
componentsPanel-Objekt	Mit <code>fl.componentsPanel</code> können Sie auf das componentsPanel-Objekt zugreifen. Dieses Objekt entspricht dem Bedienfeld „Komponenten“ in der Flash-Authoring-Umgebung.
Document-Objekt	Mit <code>fl.documents</code> rufen Sie ein Array aller geöffneten Dokumente ab. Mit <code>fl.documents[index]</code> können Sie auf ein bestimmtes Dokument und mit <code>fl.getDocumentDOM()</code> auf das aktuelle Dokument zugreifen (d. h. das Dokument mit dem Fokus).
drawingLayer-Objekt	Mit <code>fl.drawingLayer</code> können Sie auf das drawingLayer-Objekt zugreifen.
Math-Objekt	Mit <code>fl.Math</code> können Sie auf das Math-Objekt zugreifen.
outputPanel-Objekt	Mit <code>fl.outputPanel</code> können Sie auf das outputPanel-Objekt zugreifen. Dieses Objekt entspricht dem Bedienfeld „Ausgabe“ in der Flash-Authoring-Umgebung.
presetPanel-Objekt	Mit <code>fl.presetPanel</code> können Sie auf das outputPanel-Objekt zugreifen. Dieses Objekt entspricht dem Bedienfeld „Bewegungsvoreinstellungen“ („Fenster“ > „Bewegungsvoreinstellungen“).
swfPanel-Objekt	Mit <code>fl.swfPanels</code> greifen Sie auf ein Array von swfPanel-Objekten zu. Diese Objekte entsprechen den Window SWF-Bedienfeldern.
Tools-Objekt	Mit <code>fl.tools</code> greifen Sie auf ein Array von Tools-Objekten zu.
XMLUI-Objekt	Mit <code>fl.xmlui</code> greifen Sie auf ein XMLUI-Objekt (XML User Interface) zu. Mithilfe des XMLUI-Objekts können Sie die Eigenschaften eines XMLUI-Dialogfelds abrufen und festlegen.

Das Document-Objekt

Die Eigenschaft `fl.documents` ist eine wichtige Eigenschaft des flash-Objekts. Diese Eigenschaft enthält ein Array von Document-Objekten, die jeweils eine der gegenwärtig in der Authoring-Umgebung geöffneten FLA-Dateien darstellen. Die Eigenschaften der einzelnen Document-Objekte wiederum stellen die meisten Elemente dar, die eine FLA-Datei enthalten kann. Ein Großteil des DOM setzt sich daher aus Child-Objekten und Eigenschaften des Document-Objekts zusammen. Weitere Informationen finden Sie unter [Document-Objekt](#).

Wenn Sie z. B. auf das erste geöffnete Dokument Bezug nehmen möchten, verwenden Sie die Anweisung `flash.documents[0]` oder `fl.documents[0]`. Das erste Dokument ist das Flash-Dokument, das im Rahmen der aktuellen Sitzung in der Authoring-Umgebung zuerst geöffnet wurde. Wenn das zuerst geöffnete Dokument geschlossen wird, werden die Indizes der übrigen geöffneten Dokumente entsprechend dekrementiert.

Wenn Sie nach dem Index eines bestimmten Dokuments suchen möchten, verwenden Sie `flash.findDocumentIndex(nameOfDocument)` oder `fl.findDocumentIndex(nameOfDocument)`. Siehe [fl.findDocumentIndex\(\)](#).

Wenn Sie auf das Dokument zugreifen möchten, das gegenwärtig den Fokus hat, verwenden Sie die Anweisung `flash.getDocumentDOM()` oder `fl.getDocumentDOM()`. Siehe [fl.getDocumentDOM\(\)](#). In den in diesem Dokument angeführten Beispielen wird meist die letztere Syntax verwendet.

Wenn Sie das `fl.documents`-Array nach einem bestimmten Dokument durchsuchen möchten, müssen Sie das Array durchlaufen und die `document.name`-Eigenschaft jedes einzelnen Dokuments überprüfen. Siehe [fl.documents](#) und [document.name](#).

Über das Document-Objekt können Sie auf sämtliche Objekte in dem DOM zugreifen, die nicht in der obigen Tabelle aufgeführt sind (siehe „[Das Flash-Dokumentobjektmodell](#)“ auf Seite 9). Um z. B. auf die Bibliothek eines Dokuments zuzugreifen, verwenden Sie die `document.library`-Eigenschaft. Dadurch wird ein `library`-Objekt abgerufen:

```
fl.getDocumentDOM().library
```

Wenn Sie auf das Array der Bibliothekselemente zugreifen möchten, verwenden Sie die `library.items`-Eigenschaft. Jedes Element im Array ist ein `Item`-Objekt:

```
fl.getDocumentDOM().library.items
```

Wenn Sie auf ein bestimmtes Bibliothekselement zugreifen möchten, müssen Sie ein Mitglied des `library.items`-Arrays angeben:

```
fl.getDocumentDOM().library.items[0]
```

Das `library`-Objekt ist also ein `Child`-Objekt des `Document`-Objekts und das `Item`-Objekt ist ein `Child`-Objekt des `library`-Objekts. Weitere Informationen finden Sie unter [document.library](#), [library-Objekt](#), [library.items](#) und [Item-Objekt](#).

Angeben eines Aktionsziels

Methoden wirken sich in der Regel auf den aktuellen Fokus bzw. auf die aktuelle Auswahl aus, sofern nichts anderes angegeben ist. Mit dem folgenden Skript wird beispielsweise die Größe der aktuellen Auswahl verdoppelt, weil kein bestimmtes Objekt angegeben ist:

```
fl.getDocumentDOM().scaleSelection(2, 2);
```

Nehmen wir einmal an, Sie möchten, dass eine Aktion sich ganz spezifisch auf das in dem Flash-Dokument gegenwärtig ausgewählte Element auswirkt. Dazu müssen Sie das Array verwenden, das in der Eigenschaft `document.selection` enthalten ist (siehe [document.selection](#)). Das erste Element in dem Array stellt das gegenwärtig ausgewählte Element dar, wie im folgenden Beispiel veranschaulicht:

```
var accDescription = fl.getDocumentDOM().selection[0].description;
```

Mit dem folgenden Skript wird nicht die Größe der aktuellen Auswahl, sondern die Größe des ersten in dem `Element`-Array gespeicherten Elements auf der Bühne verdoppelt:

```
var element = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];
if (element) {
    element.width = element.width*2;
    element.height = element.height*2;
}
```

Sie können nach Wunsch auch alle Elemente auf der Bühne nacheinander durchlaufen und dabei die Breite und Höhe um einen bestimmten Betrag erhöhen, wie im folgenden Beispiel veranschaulicht:

```
var elementArray =
    fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements;
for (var i=0; i < elementArray.length; i++) {
    var offset = 10;
    elementArray[i].width += offset;
    elementArray[i].height += offset;
}
```

Übersicht über die DOM-Struktur

In der folgenden Liste finden Sie eine Übersicht über die DOM-Struktur. Die Ziffern am Zeilenanfang kennzeichnen jeweils die Objektebene. Wenn beispielsweise vor einem Objekt die Ziffern „03“ stehen, bedeutet dies, dass dieses Objekt ein Child-Objekt des nächsthöheren Objekts mit den Ziffern „02“ ist, und dieses Objekt wiederum ist dem Objekt mit den Ziffern „01“ untergeordnet.

Der Zugriff auf manche Objekte kann einfach dadurch erfolgen, dass Sie eine Eigenschaft des Parent-Objekts angeben. Beispielsweise enthält die Eigenschaft `document.timelines` ein Array von Timeline-Objekten. Diese Eigenschaften werden in der folgenden Übersicht berücksichtigt.

Einige Objekte sind keine Child-Objekte, sondern Unterklassen anderer Objekte. Ein Objekt, das eine Unterklasse eines anderen Objekts ist, hat zusätzlich zu den Methoden und Eigenschaften des übergeordneten Objekts (der übergeordneten Klasse) eigene Methoden oder Eigenschaften. Unterklassen befinden sich in der Hierarchie jeweils auf derselben Ebene wie die entsprechende übergeordnete Klasse. Beispielsweise ist das Item-Objekt eine übergeordnete Klasse des BitmapItem-Objekts. Diese Beziehungen werden in der folgenden Übersicht veranschaulicht:

```
01 Top-Level Functions and Methods
01 FLfile object
01 flash object (fl)
    02 compilerErrors object
    02 componentsPanel object
    02 Document object (fl.documents array)
    03 Filter object
    03 Matrix object
    03 Fill object
    03 Stroke object
    03 library object
        04 Item object (library.items array)
        04 BitmapItem object (subclass of Item object)
        04 folderItem object (subclass of Item object)
        04 fontItem object (subclass of Item object)
        04 SoundItem object (subclass of Item object)
        04 SymbolItem object (subclass of Item object)
        04 VideoItem object (subclass of Item object)
    03 Timeline object (document.timelines array)
    04 Layer object (timeline.layers array)
        05 Frame object (layer.frames array)
            06 Element object (frame.elements array)
                07 Matrix object (element.matrix)
            06 Instance object (abstract class, subclass of Element object)
            06 BitmapInstance object (subclass of Instance object)
            06 CompiledClipInstance object (subclass of Instance object)
            06 ComponentInstance object (subclass of SymbolInstance object)
                07 Parameter object (componentInstance.parameters array)
            06 SymbolInstance object (subclass of Instance object)
            06 Text object (subclass of Element object)
                07 TextRun object (text.textRuns array)
                08 TextAttrs object (textRun.textAttrs array)
        06 Shape object (subclass of Element object)
            07 Oval object
            07 Rectangle object
            07 Contour object (shape.contours array)
                08 HalfEdge object
                    09 Vertex object
                    09 Edge object
```

```
07 Edge object (shape.edges array)
08 HalfEdge object
09 Vertex object
09 Edge object
07 Vertex object (shape.vertices array)
08 HalfEdge object
09 Vertex object
09 Edge object
05 Parameter object (screen.parameters array)
02 drawingLayer object
03 Path object
04 Contour object
02 Math object
02 outputPanel object
02 presetPanel object
03 presetItem object (presetPanel.items array)
02 swfPanel object
02 Tools object (fl.tools array)
03 ToolObj object (tools.toolObjs array)
02 XMLUI object
```

Implementierungsbeispiele

Für Adobe Flash Professional CS5 und CS5.5 sind verschiedene JSFL-Implementierungsbeispiele verfügbar. Sie können diese Dateien ansehen und installieren, um sich mit der JavaScript-API vertraut zu machen. Die Beispiele befinden sich im Ordner „Samples/ExtendingFlash“ in der Zip-Datei unter www.adobe.com/go/learn_fl_samples_de.

Beispielbefehl „Shape“

Im Ordner „ExtendingFlash/Shape“ ist eine JavaScript-API mit dem Namen „Shape.jsfl“ gespeichert (siehe „Implementierungsbeispiele“ weiter oben). Mit diesem Skript werden im Bedienfeld „Ausgabe“ Informationen über die Konturen der Form angezeigt.

So können Sie das Shape-Skript installieren und ausführen:

- 1 Kopieren Sie die Datei „Shape.jsfl“ in den Ordner „Configuration/Commands“ (siehe „[Speichern von JSFL-Dateien](#)“ auf Seite 2).
- 2 Wählen Sie in einem Flash-Dokument (FLA-Datei) ein Formobjekt aus.
- 3 Wählen Sie „Befehle“ > „Shape“, um das Skript auszuführen.

Beispielbefehl „get and set filters“

Im Ordner „ExtendingFlash/filtersGetSet“ ist eine JavaScript-API mit dem Namen „filtersGetSet.jsfl“ gespeichert (siehe „Implementierungsbeispiele“ weiter oben). Dieses Skript fügt einem ausgewählten Objekt Filter hinzu und zeigt im Bedienfeld „Ausgabe“ Informationen über die hinzugefügten Filter an.

So können Sie das filtersGetSet-Skript installieren und ausführen:

- 1 Kopieren Sie die Datei „filtersGetSet.jsfl“ in den Ordner „Configuration/Commands“ (siehe „[Speichern von JSFL-Dateien](#)“ auf Seite 2).
- 2 Wählen Sie in einem Flash-Dokument (FLA-Datei) ein Text-, Movieclip- oder Button-Objekt aus.

- 3 Wählen Sie „Befehle“ > „filtersGetSet“, um das Skript auszuführen.

Beispiel-Polysternwerkzeug

Im Ordner „ExtendingFlash/PolyStar“ ist eine JavaScript-API mit dem Namen „PolyStar.jsfl“ gespeichert (siehe „Implementierungsbeispiele“ weiter oben).

Das Skript „PolyStar.jsfl“ repliziert das Polysternwerkzeug, das sich im Flash-Bedienfeld „Werkzeuge“ befindet. Es zeigt, wie Sie das Polysternwerkzeug mithilfe der JavaScript-API erstellen und enthält detaillierte Kommentare zu den einzelnen Codezeilen. Sie sollten diese Datei unbedingt lesen, um sich einen besseren Überblick darüber zu verschaffen, wie sich die JavaScript-API einsetzen lässt. Außerdem sollten Sie die Datei „PolyStar.xml“ im Verzeichnis „Tools“ lesen; in dieser Datei wird erläutert, wie Sie selbst Werkzeuge erstellen können.

Beispiel-Bedienfeld „Bitmap nachzeichnen“

Im Ordner „ExtendingFlash/TraceBitmapPanel“ sind zwei Dateien mit den Namen „TraceBitmap fla“ und „TraceBitmap.swf“ gespeichert (siehe „Implementierungsbeispiele“ weiter oben). Diese beiden Dateien veranschaulichen die Gestaltung und die Erstellung eines Bedienfelds zur Steuerung der Funktionen von Flash. Darüber hinaus wird dargestellt, wie JavaScript-Befehle von ActionScript-Skripts aus mithilfe der Funktion `MMExecute()` aufgerufen werden.

So führen Sie das TraceBitmap-Beispiel aus:

- 1 Sofern Flash ausgeführt wird, beenden Sie die Anwendung.
- 2 Kopieren Sie die Datei „TraceBitmap.swf“ in den Ordner „WindowSWF“, der sich im Ordner „Configuration“ befindet (siehe „[Speichern von JSFL-Dateien](#)“ auf Seite 2). Unter Windows XP befindet sich der Ordner zum Beispiel hier: *Startlaufwerk\Dokumente und Einstellungen\Benutzer\Lokale Einstellungen\Anwendungsdaten\Adobe\Flash CS5\Sprache\Configuration\WindowSWF*.
- 3 Starten Sie Flash.
- 4 Erstellen bzw. öffnen Sie ein Flash-Dokument (FLA-Datei) und importieren Sie eine Bitmap oder ein JPEG-Bild in die Datei.

Sie können hierzu ein beliebiges Bild oder aber die Datei „flower.jpg“ verwenden, die im „TraceBitmapPanel“-Ordner zur Verfügung steht.
- 5 Wählen Sie das importierte Bild aus und klicken Sie auf „Fenster“ > „Andere Bedienfelder“ > „TraceBitmap“.
- 6 Klicken Sie auf „Senden“.

Das Bild wird in mehrere Formen umgewandelt.

Beispiel-DLL

Im Ordner „ExtendingFlash/dllSampleComputeSum“ ist ein DLL-Implementierungsbeispiel gespeichert (siehe „Implementierungsbeispiele“ weiter oben). Weitere Informationen zum Erstellen von DLLs finden Sie unter „[C-Level-Erweiterbarkeit](#)“ auf Seite 539.

Kapitel 2: Top-Level-Funktionen und -Methoden

Übersicht

In diesem Abschnitt werden die Funktionen und Methoden oberster Ebene beschrieben, die bei der Arbeit mit der Adobe Flash JavaScript-API verfügbar sind. Informationen zu den Speicherorten, an denen Sie JavaScript-API-Dateien ablegen sollten, finden Sie unter „[Speichern von JSFL-Dateien](#)“ auf Seite 2.

Globale Methoden

Die folgenden Methoden können von einem beliebigen JavaScript-API-Skript aus aufgerufen werden:

```
alert()
confirm()
prompt()
```

Erweiterbare Tools

Die folgenden Funktionen sind in Skripten verfügbar, mit denen erweiterbare Werkzeuge erstellt werden:

```
activate()
configureTool()
deactivate()
keyDown()
keyUp()
mouseDoubleClick()
mouseDown()
mouseMove()
mouseUp()
notifySettingsChanged()
setCursor()
```

activate()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
function activate() {
    // statements
}
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Funktion; wird aufgerufen, wenn das erweiterbare Werkzeug aktiv wird (d. h. wenn das Werkzeug im Bedienfeld „Werkzeuge“ ausgewählt wird). Verwenden Sie diese Funktion, um alle für das Werkzeug erforderlichen Initialisierungsaufgaben auszuführen.

Beispiel

Im folgenden Beispiel wird der Wert von `tools.activeTool` festgelegt, wenn das erweiterbare Werkzeug im Bedienfeld „Werkzeuge“ ausgewählt wird:

```
function activate() {  
    var theTool = fl.tools.activeTool  
}
```

Siehe auch

`tools.activeTool`

alert()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
alert ( alertText )
```

Parameter

alertText Ein String mit der Meldung, die im Warndialogfeld angezeigt werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; zeigt in einem modalen Warndialogfeld einen String und eine OK-Schaltfläche an.

Beispiel

Im folgenden Beispiel wird in einem Warndialogfeld die Meldung „Process Complete“ angezeigt:

```
alert("Process Complete");
```

Siehe auch

`confirm()`, `prompt()`

configureTool()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
function configureTool() {  
    // statements  
}
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Funktion; wird aufgerufen, wenn Flash geöffnet und das erweiterbare Werkzeug in das Bedienfeld „Werkzeuge“ geladen wird. Stellen Sie mithilfe dieser Funktion alle Informationen bereit, die Flash zu dem Werkzeug benötigt.

Beispiel

In den folgenden Beispielen werden zwei mögliche Implementierungen dieser Funktion veranschaulicht:

```
function configureTool() {  
    theTool = fl.tools.activeTool;  
    theTool.setToolName("myTool");  
    theTool.setIcon("myTool.png");  
    theTool.setMenuString("My Tool's menu string");  
    theTool.setToolTip("my tool's tool tip");  
    theTool.setOptionsFile("mtTool.xml");  
}
```

```
function configureTool() {  
    theTool = fl.tools.activeTool;  
    theTool.setToolName("ellipse");  
    theTool.setIcon("Ellipse.png");  
    theTool.setMenuString("Ellipse");  
    theTool.setToolTip("Ellipse");  
    theTool.showTransformHandles( true );  
}
```

confirm()

Verfügbarkeit

Flash 8.

Verwendung

```
confirm ( strAlert )
```

Parameter

strAlert Ein String mit der Meldung, die im Warndialogfeld angezeigt werden soll.

Rückgabewerte

Ein boolescher Wert; `true`, wenn der Benutzer auf „OK“ klickt, bzw. `false`, wenn der Benutzer auf „Abbrechen“ klickt.

Beschreibung

Methode; zeigt in einem modalen Warndialogfeld einen String, eine OK-Schaltfläche und eine Abbrechen-Schaltfläche an.

Hinweis: Wenn keine Dokumente (FLA-Dateien) geöffnet sind, tritt ein Fehler bei der Ausführung auf.

Beispiel

Im folgenden Beispiel wird in einem Warndialogfeld die Meldung „Sort data?“ angezeigt:

```
confirm("Sort data?");
```

Siehe auch

`alert()`, `prompt()`

deactivate()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
function deactivate() {  
    // statements  
}
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Funktion; wird aufgerufen, wenn das erweiterbare Werkzeug inaktiv wird (d. h. wenn ein anderes Werkzeug als aktives Werkzeug ausgewählt wird). Verwenden Sie diese Funktion, um alle für das Werkzeug erforderlichen Bereinigungsaufgaben auszuführen.

Beispiel

Im folgenden Beispiel wird eine Nachricht im Bedienfeld „Ausgabe“ angezeigt, wenn das Werkzeug deaktiviert wird:

```
function deactivate() {  
    fl.trace( "Tool is no longer active" );  
}
```


keyDown()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
function keyDown() {  
    // statements  
}
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Funktion; wird aufgerufen, wenn das erweiterbare Werkzeug aktiv ist und der Benutzer eine Taste drückt. Das Skript sollte `tools.getKeyDown()` aufrufen, um festzustellen, welche Taste gedrückt wurde.

Beispiel

Im folgenden Beispiel werden Informationen über die gedrückte Taste angezeigt, wenn das erweiterbare Werkzeug aktiv ist und der Benutzer eine Taste drückt:

```
function keyDown() {  
    fl.trace("key " + fl.tools.getKeyDown() + " was pressed");  
}
```

Siehe auch

`keyUp()`, `tools.getKeyDown()`

keyUp()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
function keyUp() {  
    // statements  
}
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Funktion; wird aufgerufen, wenn das erweiterbare Werkzeug aktiv ist und eine Taste losgelassen wird.

Beispiel

Im folgenden Beispiel wird im Bedienfeld „Ausgabe“ eine Meldung angezeigt, wenn das erweiterbare Werkzeug aktiv ist und eine Taste losgelassen wird.

```
function keyUp() {  
    fl.trace("Key is released");  
}
```

Siehe auch

[keyDown\(\)](#)

mouseDoubleClick()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
function mouseDoubleClick() {  
    // statements  
}
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Funktion; wird aufgerufen, wenn das erweiterbare Werkzeug aktiv ist und der Benutzer auf der Bühne doppelklickt.

Beispiel

Im folgenden Beispiel wird im Bedienfeld „Ausgabe“ eine Meldung angezeigt, wenn das erweiterbare Werkzeug aktiv ist und der Benutzer doppelklickt.

```
function mouseDoubleClick() {  
    fl.trace("Mouse was double-clicked");  
}
```

mouseDown()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
function mouseDown( [ pt ] ) {  
    // statements  
}
```

Parameter

pt Ein Punkt, der die Mausposition angibt, wenn die Maustaste gedrückt wird. Er wird an die Funktion übergeben, wenn die Maustaste gedrückt wird. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Funktion; wird aufgerufen, wenn das erweiterbare Werkzeug aktiv ist und die Maustaste gedrückt wird, während der Zeiger sich über der Bühne befindet.

Beispiel

Die folgenden Beispiele veranschaulichen die Verwendung dieser Funktion, wenn das erweiterbare Werkzeug aktiv ist. Im ersten Beispiel wird im Bedienfeld „Ausgabe“ die Meldung angezeigt, dass die Maustaste gedrückt wurde. Im zweiten Beispiel werden die *x*- und *y*-Koordinaten der Position angegeben, an der die Maus sich beim Drücken der Taste befindet.

```
function mouseDown() {  
    fl.trace("Mouse button has been pressed");  
}  
function mouseDown(pt) {  
    fl.trace("x = "+ pt.x+" :: y = "+pt.y);  
}
```

mouseMove()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
function mouseMove( [ pt ] ) {  
    // statements  
}
```

Parameter

pt Ein Punkt, der die gegenwärtige Mausposition angibt. Er wird bei jeder Mausbewegung an die Funktion übergeben, d. h. die Mausposition wird kontinuierlich verfolgt. Wenn die Bühne sich im Modus „Bearbeiten“ bzw. „An Position bearbeiten“ befindet, werden die Punktkoordinaten in Relation zu dem bearbeiteten Objekt angegeben, und andernfalls in Relation zur Bühne. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Funktion; wird jedes Mal aufgerufen, wenn das erweiterbare Werkzeug aktiv ist und die Maus über einen bestimmten Punkt auf der Bühne bewegt wird. Dabei ist gleichgültig, ob die Maustaste gedrückt oder nicht gedrückt ist.

Beispiel

Die folgenden Beispiele veranschaulichen die Verwendung dieser Funktion. Im ersten Beispiel wird im Bedienfeld „Ausgabe“ die Meldung angezeigt, dass die Maus bewegt wurde. Im zweiten Beispiel werden die *x*- und *y*-Koordinaten der Mausposition während der Mausbewegung angegeben.

```
function mouseMove() {  
    fl.trace("moving");  
}  
  
function mouseMove(pt) {  
    fl.trace("x = " + pt.x + " :: y = " + pt.y);  
}
```

mouseUp()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
function mouseUp() {  
    // statements  
}
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Funktion; wird jedes Mal aufgerufen, wenn das erweiterbare Werkzeug aktiv ist und die Maustaste losgelassen wird, nachdem sie zuvor auf der Bühne gedrückt wurde.

Beispiel

Im folgenden Beispiel wird im Bedienfeld „Ausgabe“ eine Meldung angezeigt, wenn das erweiterbare Werkzeug aktiv ist und der Benutzer die Maustaste loslässt.

```
function mouseUp() {  
    fl.trace("mouse is up");  
}
```

notifySettingsChanged()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
function notifySettingsChanged() {  
    // statements  
}
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Funktion; wird aufgerufen, wenn das erweiterbare Werkzeug aktiv ist und der Benutzer die zugehörigen Optionen im Eigenschafteninspektor ändert. Mit der Eigenschaft `tools.activeTool` können Sie die aktuellen Werte der Optionen abfragen (siehe [tools.activeTool](#)).

Beispiel

Im folgenden Beispiel wird im Bedienfeld „Ausgabe“ eine Meldung angezeigt, wenn das erweiterbare Werkzeug aktiv ist und der Benutzer die Werkzeugoptionen im Eigenschafteninspektor ändert.

```
function notifySettingsChanged() {  
    var theTool = fl.tools.activeTool;  
    var newValue = theTool.myProp;  
}
```

prompt()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
prompt(promptMsg [,text])
```

Parameter

promptMsg Ein String, der im Eingabeaufforderungsdialogfeld angezeigt wird (bei Mac OS X sind maximal 256 Zeichen zulässig).

text Ein optionaler String, der in dem Textfeld als Standardwert angezeigt wird.

Rückgabewerte

Der von dem Benutzer eingegebene String, sofern der Benutzer auf „OK“ klickt, bzw. `null`, wenn er auf „Abbrechen“ klickt.

Beschreibung

Methode; zeigt in einem modalen Warndialogfeld eine Eingabeaufforderung, optionalen Text, eine OK-Schaltfläche und eine Abbrechen-Schaltfläche an.

Beispiel

Im folgenden Beispiel wird der Benutzer zur Eingabe eines Benutzernamens aufgefordert. Wenn der Benutzer einen Namen eingibt und auf „OK“ klickt, wird der Name im Bedienfeld „Ausgabe“ angezeigt.

```
var userName = prompt("Enter user name", "Type user name here");  
fl.trace(userName);
```

Siehe auch

[alert\(\)](#), [confirm\(\)](#)

setCursor()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
function setCursor() {  
    // statements  
}
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Funktion; wird aufgerufen, wenn das erweiterbare Werkzeug aktiv ist und die Maus bewegt wird, damit das Skript benutzerdefinierte Zeiger einstellen kann. Das Skript sollte `tools.setCursor()` aufrufen, um den zu verwendenden Zeiger festzulegen. Eine Übersicht über die Ganzzahlenwerte, die den verschiedenen Zeigern entsprechen, finden Sie unter [tools.setCursor\(\)](#).

Beispiel

```
function setCursor() {  
    fl.tools.setCursor( 1 );  
}
```

Kapitel 3: actionsPanel-Objekt

Verfügbarkeit

Flash CS3 Professional.

Beschreibung

Das actionsPanel-Objekt ist eine Eigenschaft des flash-Objekts (fl) und stellt das derzeit angezeigte Bedienfeld „Aktionen“ dar (siehe [fl.actionsPanel](#)).

Übersicht über Methoden

Für das actionsPanel-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>actionsPanel.getClassForObject()</code>	Gibt die Klasse einer angegebenen Variablen zurück.
<code>actionsPanel.getScriptAssistMode()</code>	Gibt an, ob die Skripthilfe aktiviert ist.
<code>actionsPanel.getSelectedText()</code>	Gibt den Text zurück, der zurzeit im Bedienfeld „Aktionen“ ausgewählt ist.
<code>actionsPanel.getText()</code>	Gibt den Text im Bedienfeld „Aktionen“ zurück.
<code>actionsPanel.hasSelection()</code>	Gibt an, ob im Bedienfeld „Aktionen“ zurzeit Text ausgewählt ist.
<code>actionsPanel.replaceSelectedText()</code>	Ersetzt den zurzeit ausgewählten Text durch den angegebenen Text.
<code>actionsPanel.setScriptAssistMode()</code>	Aktiviert oder deaktiviert die Skripthilfe.
<code>actionsPanel.setSelection()</code>	Wählt die angegebenen Zeichen im Bedienfeld „Aktionen“ aus.
<code>actionsPanel.setText()</code>	Löscht den gesamten Text im Bedienfeld „Aktionen“ und fügt dann den angegebenen Text hinzu.

actionsPanel.getClassForObject()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
actionsPanel.getClassForObject(ASvariableName)
```

Parameter

ASvariableName Ein String, der den Namen der ActionScript-Variablen darstellt.

Rückgabewerte

Ein String, der die Klasse, zu der *ASvariableName* gehört, darstellt.

Beschreibung

Methode; gibt die Klasse der angegebenen Variablen zurück, die im zurzeit angezeigten Bedienfeld „Aktionen“ definiert sein muss. Außerdem muss sich der Cursor oder der im Bedienfeld „Aktionen“ ausgewählte Text hinter der Variablendefinition befinden.

Beispiel

Im folgenden Beispiel wird die Klasse angezeigt, die der Variablen „myVar“ zugeordnet ist, wenn sich der Cursor hinter der Anweisung „varmyVar:ActivityEvent;“ im Bedienfeld „Aktionen“ befindet.

```
// Place the following code in the Actions panel,  
// and position the cursor somewhere after the end of the line  
var myVar:ActivityEvent;  
// Place the following code in the JSFL file  
var theClass = fl.actionsPanel.getClassForObject("myVar");  
fl.trace(theClass); // traces: "ActivityEvent"
```

actionsPanel.getScriptAssistMode()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
actionsPanel.getScriptAssistMode()
```

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert, der angibt, ob die Skripthilfe aktiviert ist (`true`) oder nicht (`false`).

Beschreibung

Methode; gibt an, ob die Skripthilfe aktiviert ist.

Beispiel

Im folgenden Beispiel wird eine Meldung angezeigt, falls die Skripthilfe nicht aktiviert ist.

```
mAssist = fl.actionsPanel.getScriptAssistMode();  
if (!mAssist) {  
    alert("For more guidance when writing ActionScript code, try Script Assist mode");  
}
```

Siehe auch

[`actionsPanel.setScriptAssistMode\(\)`](#)

actionsPanel.getSelectedText()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
actionsPanel.getSelectedText()
```

Parameter

Keine.

Rückgabewerte

Ein String, der den Text enthält, der zurzeit im Bedienfeld „Aktionen“ ausgewählt ist.

Beschreibung

Methode; gibt den Text zurück, der zurzeit im Bedienfeld „Aktionen“ ausgewählt ist.

Beispiel

Im folgenden Beispiel wird der Text angezeigt, der zurzeit im Bedienfeld „Aktionen“ ausgewählt ist.

```
var apText = fl.actionsPanel.getSelectedText();  
fl.trace(apText);
```

Siehe auch

[actionsPanel.getText\(\)](#), [actionsPanel.hasSelection\(\)](#), [actionsPanel.replaceSelectedText\(\)](#),
[actionsPanel.setSelection\(\)](#)

actionsPanel.getText()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
actionsPanel.getText()
```

Parameter

Keine.

Rückgabewerte

Ein String, der den gesamten Text im Bedienfeld „Aktionen“ enthält.

Beschreibung

Methode; gibt den Text im Bedienfeld „Aktionen“ zurück.

Beispiel

Im folgenden Beispiel wird der Text angezeigt, der sich im Bedienfeld „Aktionen“ befindet.

```
var apText = fl.actionsPanel.getText();  
fl.trace(apText);
```

Siehe auch

[actionsPanel.getSelectedText\(\)](#), [actionsPanel.setText\(\)](#)

actionsPanel.hasSelection()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
actionsPanel.hasSelection()
```

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert, der angibt, ob im Bedienfeld „Aktionen“ Text ausgewählt ist (`true`) oder nicht (`false`).

Beschreibung

Methode; gibt an, ob im Bedienfeld „Aktionen“ zurzeit Text ausgewählt ist.

Beispiel

Im folgenden Beispiel wird Text angezeigt, der zurzeit im Bedienfeld „Aktionen“ ausgewählt ist. Wenn kein Text ausgewählt ist, wird der gesamte Text aus dem Bedienfeld „Aktionen“ angezeigt.

```
if (fl.actionsPanel.hasSelection()) {  
    var apText = fl.actionsPanel.getSelectedText();  
}  
else {  
    var apText = fl.actionsPanel.getText();  
}  
fl.trace(apText);
```

Siehe auch

[actionsPanel.getSelectedText\(\)](#), [actionsPanel.getText\(\)](#), [actionsPanel.replaceSelectedText\(\)](#),
[actionsPanel.setSelection\(\)](#)

actionsPanel.replaceSelectedText()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
actionsPanel.replaceSelectedText(replacementText)
```

Parameter

replacementText Ein String, der den Text darstellt, der den im Bedienfeld „Aktionen“ ausgewählten Text ersetzen soll.

Rückgabewerte

Ein boolescher Wert; `true`, wenn das Bedienfeld „Aktionen“ gefunden wurde, ansonsten `false`.

Beschreibung

Methode; ersetzt den zurzeit ausgewählten Text durch den Text, der mit *replacementText* angegeben wird. Wenn *replacementText* mehr Zeichen als der ausgewählte Text enthält, stehen alle Zeichen hinter dem ausgewählten Text jetzt hinter *replacementText*; d. h., sie werden nicht überschrieben.

Beispiel

Im folgenden Beispiel wird der Text ersetzt, der zurzeit im Bedienfeld „Aktionen“ ausgewählt ist.

```
if (fl.actionsPanel.hasSelection()) {  
    fl.actionsPanel.replaceSelectedText("// © 2006 Adobe Inc.");  
}
```

Siehe auch

`actionsPanel.getSelectedText()`, `actionsPanel.hasSelection()`, `actionsPanel.setSelection()`,
`actionsPanel.setText()`

actionsPanel.setScriptAssistMode()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
actionsPanel.setScriptAssistMode(bScriptAssist)
```

Parameter

bScriptAssist Ein boolescher Wert, der angibt, ob die Skripthilfe aktiviert oder deaktiviert werden soll.

Rückgabewerte

Ein boolescher Wert, der angibt, ob die Skripthilfe erfolgreich aktiviert oder deaktiviert wurde.

Beschreibung

Methode; aktiviert oder deaktiviert die Skripthilfe.

Beispiel

Im folgenden Beispiel wird der Status der Skripthilfe umgeschaltet.

```
fl.trace(fl.actionsPanel.getScriptAssistMode());  
if (fl.actionsPanel.getScriptAssistMode()) {  
    fl.actionsPanel.setScriptAssistMode(false);  
}  
else {  
    fl.actionsPanel.setScriptAssistMode(true);  
}  
fl.trace(fl.actionsPanel.getScriptAssistMode());
```

Siehe auch

[actionsPanel.getScriptAssistMode\(\)](#)

actionsPanel.setSelection()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
actionsPanel.setSelection(startIndex, numberOfChars)
```

Parameter

startIndex Eine auf Null basierende Ganzzahl, die das erste auszuwählende Zeichen angibt.

numberOfChars Eine Ganzzahl, die die Anzahl der auszuwählenden Zeichen angibt.

Rückgabewerte

Ein boolescher Wert, der angibt, ob die angeforderten Zeichen ausgewählt werden können (`true`) oder nicht (`false`).

Beschreibung

Methode; wählt die angegebenen Zeichen im Bedienfeld „Aktionen“ aus.

Beispiel

Im folgenden Beispiel werden die Zeichen „2006“ im Bedienfeld „Aktionen“ durch den angegebenen Text ersetzt.

```
// Type the following as the first line in the Actions panel  
// 2006 - Addresses user request 40196  
// Type the following in the JSFL file  
fl.actionsPanel.setSelection(3,4);  
fl.actionsPanel.replaceSelectedText("// Last updated: 2007");
```

Siehe auch

[actionsPanel.getSelectedText\(\)](#), [actionsPanel.hasSelection\(\)](#),
[actionsPanel.replaceSelectedText\(\)](#)

actionsPanel.setText()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
actionsPanel.setText(replacementText)
```

Parameter

replacementText Ein String, der den Text darstellt, der in das Bedienfeld „Aktionen“ eingefügt werden soll.

Rückgabewerte

Ein boolescher Wert, der `true` lautet, wenn der angegebene Text in das Bedienfeld „Aktionen“ geschrieben wurde, andernfalls `false`.

Beschreibung

Methode; löscht den gesamten Text im Bedienfeld „Aktionen“ und fügt dann den mit *replacementText* festgelegten Text hinzu.

Beispiel

Im folgenden Beispiel wird der gesamte Text im Bedienfeld „Aktionen“ durch den angegebenen Text ersetzt.

```
fl.actionsPanel.setText("// Deleted this code - no longer needed");
```

Siehe auch

[actionsPanel.getText\(\)](#), [actionsPanel.replaceSelectedText\(\)](#)

Kapitel 4: BitmapInstance-Objekt

Vererbung [Element-Objekt](#) > [Instance-Objekt](#) > BitmapInstance-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das BitmapInstance-Objekt ist eine Unterklasse des Instance-Objekts und entspricht einer Bitmap in einem Bild (siehe [Instance-Objekt](#)).

Übersicht über Methoden

Abgesehen von den Methoden für das [Instance-Objekt](#) stehen für das BitmapInstance-Objekt die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>bitmapInstance.getBits()</code>	Dient zum Erstellen von Bitmap-Effekten. Dabei werden die Bits aus der Bitmap extrahiert und manipuliert und schließlich an Flash zurückgegeben.
<code>bitmapInstance.setBits()</code>	Stellt die Bits eines vorhandenen Bitmap-Elements ein.

Übersicht über Eigenschaften

Abgesehen von den Eigenschaften für das [Instance-Objekt](#) stehen für das BitmapInstance-Objekt die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>bitmapInstance.hPixels</code>	Schreibgeschützt; eine Ganzzahl, die die Breite der Bitmap in Pixel angibt.
<code>bitmapInstance.vPixels</code>	Schreibgeschützt; eine Ganzzahl, die die Höhe der Bitmap in Pixel angibt.

bitmapInstance.getBits()

Verfügbarkeit

Flash MX 2004.

Verwendung

`bitmapInstance.getBits()`

Parameter

Keine.

Rückgabewerte

Ein Objekt mit den Eigenschaften `width`, `height`, `depth`, `bits` und, sofern die Bitmap eine Farbtabelle hat, `cTab`. Das Element `bits` ist ein Byte-Array. Das Element `cTab` ist ein Array von Farbwerten im Format "#RRGGBB". Die Länge des Arrays entspricht der Länge der Farbtabelle.

Das Byte-Array hat nur dann eine Bedeutung, wenn eine DLL oder eine gemeinsame Bibliothek (Shared Library) darauf verweist. Es wird in der Regel nur beim Erstellen eines erweiterbaren Werkzeugs oder eines Effekts eingesetzt. Weitere Hinweise zur Erstellung von DLLs, die mit Flash JavaScript genutzt werden können, finden Sie in „[C-Level-Erweiterbarkeit](#)“ auf Seite 539.

Beschreibung

Methode; dient zum Erstellen von Bitmap-Effekten. Dabei werden die Bits aus der Bitmap extrahiert und manipuliert und schließlich an Flash zurückgegeben.

Beispiel

Im folgenden Beispiel wird zuerst ein Verweis auf das gegenwärtig ausgewählte Objekt erstellt. Dann wird festgestellt, ob es sich bei dem Objekt um eine Bitmap handelt, und zum Schluss werden die Höhe, Breite und Farbtiefe der Bitmap ermittelt:

```
var isBitmap = fl.getDocumentDOM().selection[0].instanceType;
if (isBitmap == "bitmap") {
    var bits = fl.getDocumentDOM().selection[0].getBits();
    fl.trace("height = " + bits.height);
    fl.trace("width = " + bits.width);
    fl.trace("depth = " + bits.depth);
}
```

Siehe auch

[bitmapInstance.setBits\(\)](#)

bitmapInstance.hPixels

Verfügbarkeit

Flash MX 2004.

Verwendung

`bitmapInstance.hPixels`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die die Breite der Bitmap darstellt, also die Anzahl der Pixel in horizontaler Richtung.

Beispiel

Im folgenden Beispiel wird die Breite der Bitmap in Pixel abgerufen:

```
// Get the number of pixels in the horizontal dimension.  
var bmObj = fl.getDocumentDOM().selection[0];  
var isBitmap = bmObj.instanceType;  
if(isBitmap == "bitmap"){  
    var numHorizontalPixels = bmObj.hPixels;  
}
```

Siehe auch

[bitmapInstance.vPixels](#)

bitmapInstance.setBits()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
bitmapInstance.setBits(bitmap)
```

Parameter

bitmap Ein Objekt mit den Eigenschaften `height`, `width`, `depth`, `bits` und `cTab`. Die Eigenschaften `height`, `width` und `depth` sind Ganzzahlen. Die Eigenschaft `bits` ist ein Byte-Array. Die Eigenschaft `cTab` ist nur bei Bitmaps erforderlich, deren Farbtiefe 8 Bit oder weniger beträgt. Sie ist ein String, der einen Farbwert im Format "#RRGGBB" angibt.

***Hinweis:** Das Byte-Array hat nur eine Bedeutung, wenn eine externe Bibliothek darauf verweist. Es wird in der Regel nur beim Erstellen eines erweiterbaren Werkzeugs oder eines Effekts eingesetzt.*

Rückgabewerte

Keine.

Beschreibung

Methode; stellt die Bits eines vorhandenen Bitmap-Elements ein. Mit dieser Methode können Sie Bitmap-Effekte erstellen. Dabei werden die Bits aus der Bitmap extrahiert und manipuliert und zum Schluss wird die Bitmap an Flash zurückgegeben.

Beispiel

Im folgenden Beispiel wird zunächst festgestellt, ob es sich bei der aktuellen Auswahl um eine Bitmap handelt. Dann wird die Höhe der Bitmap auf 150 Pixel festgelegt:

```
var isBitmap = fl.getDocumentDOM().selection[0].instanceType;  
if(isBitmap == "bitmap"){  
    var bits = fl.getDocumentDOM().selection[0].getBits();  
    bits.height = 150;  
    fl.getDocumentDOM().selection[0].setBits(bits);  
}
```

Siehe auch

[bitmapInstance.getBits\(\)](#)

bitmapInstance.vPixels

Verfügbarkeit

Flash MX 2004.

Verwendung

`bitmapInstance.vPixels`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die die Höhe der Bitmap darstellt, also die Anzahl der Pixel in vertikaler Richtung.

Beispiel

Im folgenden Beispiel wird die Höhe der Bitmap in Pixel abgerufen:

```
// Get the number of pixels in the vertical dimension.  
var bmObj = fl.getDocumentDOM().selection[0];  
var isBitmap = bmObj.instanceType;  
if(isBitmap == "bitmap"){  
    var numVerticalPixels = bmObj.vPixels;  
}
```

Siehe auch

[bitmapInstance.hPixels](#)

Kapitel 5: BitmapItem-Objekt

Vererbung [Item-Objekt](#) > BitmapItem-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Ein BitmapItem-Objekt bezieht sich auf eine Bitmap in der Bibliothek eines Dokuments. Das BitmapItem-Objekt ist eine Unterklasse des Item-Objekts (siehe [Item-Objekt](#)).

Übersicht über Eigenschaften

Neben den Eigenschaften für das [Item-Objekt](#) verfügt das BitmapItem-Objekt über die folgenden Eigenschaften:

Eigenschaft	Beschreibung
<code>bitmapItem.allowSmoothing</code>	Ein boolescher Wert, der angibt, ob das Glätten einer Bitmap zulässig ist.
<code>bitmapItem.compressionType</code>	Ein String, der die auf die Bitmap angewendete Art der Bildkomprimierung festlegt.
<code>bitmapItem.fileLastModifiedDate</code>	Die Anzahl der Sekunden, die seit dem 1. Januar 1970 bis zum Änderungsdatum der Originaldatei vergangen sind.
<code>bitmapItem.originalCompressionType</code>	Gibt an, ob das Element als eine JPEG-Datei importiert wurde.
<code>bitmapItem.sourceFileExists</code>	Gibt an, ob die Datei, die in die Bibliothek importiert wurde, an ihrem ursprünglichen Speicherort, von dem sie importiert wurde, immer noch vorhanden ist.
<code>bitmapItem.sourceFileIsCurrent</code>	Gibt an, ob das Dateiänderungsdatum des Bibliothekselements mit dem Dateiänderungsdatum der importierten Datei auf der Festplatte übereinstimmt.
<code>bitmapItem.sourceFilePath</code>	Der Pfad und der Name der Datei, die in die Bibliothek importiert wurde.
<code>bitmapItem.useDeblocking</code>	Gibt an, ob Deblocking aktiviert ist.
<code>bitmapItem.useImportedJPEGQuality</code>	Ein boolescher Wert, der angibt, ob die standardmäßige importierte JPEG-Qualität verwendet werden soll.

Übersicht über Methoden

Neben den Eigenschaften für das [Item-Objekt](#) verfügt das BitmapItem-Objekt über die folgenden Methoden:

Methode	Beschreibung
<code>bitmapItem.exportToFile()</code>	Exportiert das angegebene Element in eine PNG- oder JPG-Datei.

bitmapItem.allowSmoothing

Verfügbarkeit

Flash MX 2004.

Verwendung

`bitmapItem.allowSmoothing`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob eine Bitmap geglättet (`true`) bzw. nicht geglättet (`false`) werden darf.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `allowSmoothing` des ersten Elements in der Bibliothek des aktuellen Dokuments auf `true` eingestellt:

```
fl.getDocumentDOM().library.items[0].allowSmoothing = true;  
alert(fl.getDocumentDOM().library.items[0].allowSmoothing);
```

bitmapItem.compressionType

Verfügbarkeit

Flash MX 2004.

Verwendung

`bitmapItem.compressionType`

Beschreibung

Eigenschaft; ein String, der die auf die Bitmap angewendete Art der Bildkomprimierung festlegt. Die zulässigen Werte lauten "photo" und "lossless". Wenn der Wert für `bitmapItem.useImportedJPEGQuality` `false` lautet, entspricht "photo" einer JPEG-Qualität von 0 bis 100. Lautet der Wert für `bitmapItem.useImportedJPEGQuality` hingegen `true`, entspricht "photo" einer JPEG-Qualität, die auf dem Standardwert für die Dokumentqualität basiert. Der Wert "lossless" entspricht dem GIF- bzw. dem PNG-Format (siehe [bitmapItem.useImportedJPEGQuality](#)).

Beispiel

Im folgenden Beispiel wird die Eigenschaft `compressionType` des ersten Elements in der Bibliothek des aktuellen Dokuments auf "photo" eingestellt:

```
fl.getDocumentDOM().library.items[0].compressionType = "photo";  
alert(fl.getDocumentDOM().library.items[0].compressionType);
```

bitmapItem.exportToFile()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
bitmapItem.exportToFile(fileURI)
```

Parameter

fileURI Ein String im Format file:/// URI, der den Namen der exportierten Datei angibt.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die Datei erfolgreich exportiert wurde; andernfalls `false`.

Beschreibung

Methode; exportiert das angegebene Element in eine PNG- oder JPG-Datei.

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Bitmap-Element handelt, wird es im folgenden Beispiel als JPG-Datei exportiert:

```
var imageFileURL = "file:///C:/exportTest/out.jpg";  
var libItem = fl.getDocumentDOM().library.items[0];  
libItem.exportToFile(imageFileURL);
```

bitmapItem.fileLastModifiedDate

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
bitmapItem.fileLastModifiedDate
```

Beschreibung

Schreibgeschützte Eigenschaft; ein String mit einer hexadezimalen Zahl, die die Anzahl der Sekunden angibt, die seit dem 1. Januar 1970 bis zum Änderungsdatum der ursprünglichen Datei zu dem Zeitpunkt vergangen sind, an dem die Datei in die Bibliothek importiert wurde. Sollte die Datei nicht mehr existieren, lautet dieser Wert "00000000".

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Bitmap-Element handelt, wird im folgenden Beispiel wie oben beschrieben eine hexadezimale Zahl angezeigt:

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("Mod date when imported = " + libItem.fileLastModifiedDate);
```

Siehe auch

[bitmapItem.sourceFileExists](#), [bitmapItem.sourceFileIsCurrent](#), [bitmapItem.sourceFilePath](#),
[FLfile.getModificationDate\(\)](#)

bitmapItem.originalCompressionType

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`bitmapItem.originalCompressionType`

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der angibt, ob das angegebene Element als jpeg-Datei importiert wurde. Mögliche Werte für diese Eigenschaft sind „photo“ (für jpeg-Dateien) und „lossless“ (für nicht komprimierte Dateitypen, wie etwa GIF und PNG).

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Bitmap-Element handelt, wird im folgenden Beispiel "photo" angezeigt, wenn die Datei als jpeg-Datei in die Bibliothek importiert wurde, andernfalls "lossless":

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("Imported compression type = "+ libItem.originalCompressionType);
```

Siehe auch

[bitmapItem.compressionType](#)

bitmapItem.quality

Verfügbarkeit

Flash MX 2004.

Verwendung

`bitmapItem.quality`

Beschreibung

Eigenschaft; eine Ganzzahl, die die Qualität der Bitmap angibt. Geben Sie -1 ein, wenn Sie die standardmäßige Dokumentqualität verwenden möchten, und andernfalls eine Ganzzahl von 0 bis 100. Nur bei JPEG-Komprimierung verfügbar.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `quality` des ersten Elements in der Bibliothek des aktuellen Dokuments auf 65 eingestellt:

```
fl.getDocumentDOM().library.items[0].quality = 65;  
alert(fl.getDocumentDOM().library.items[0].quality);
```

bitmapItem.sourceFileExists

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
bitmapItem.sourceFileExists
```

Beschreibung

Schreibgeschützte Eigenschaft; der boolesche Wert `true`, wenn die in die Bibliothek importierte Datei noch an dem Speicherort existiert, von dem aus sie importiert wurde; andernfalls `false`.

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Bitmap-Element handelt, wird im folgenden Beispiel "true" angezeigt, wenn die in die Bibliothek importierte Datei noch existiert.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("sourceFileExists = "+ libItem.sourceFileExists);
```

Siehe auch

[bitmapItem.sourceFileIsCurrent](#),

[bitmapItem.sourceFilePath](#)

bitmapItem.sourceFileIsCurrent

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
bitmapItem.sourceFileIsCurrent
```

Beschreibung

Schreibgeschützte Eigenschaft; der boolesche Wert `true`, wenn das Dateiänderungsdatum des Bibliothekselements mit dem Änderungsdatum der importierten Datei auf der Festplatte identisch ist; andernfalls `false`.

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Bitmap-Element handelt, wird im folgenden Beispiel "true" angezeigt, wenn die importierte Datei auf der Festplatte seit dem Import nicht geändert wurde.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("fileIsCurrent = "+ libItem.sourceFileIsCurrent);
```

Siehe auch

[bitmapItem.fileLastModifiedDate](#), [bitmapItem.sourceFilePath](#)

bitmapItem.sourceFilePath

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`bitmapItem.sourceFilePath`

Beschreibung

Schreibgeschützte Eigenschaft; ein String im Format `file:///` URI, der den Pfad und den Namen der in die Bibliothek importierten Datei angibt.

Beispiel

Im folgenden Beispiel werden die Namen und die Quelldateipfade aller Bibliothekselemente angezeigt, die dem Typ "bitmap" angehören:

```
for (idx in fl.getDocumentDOM().library.items) {  
    if (fl.getDocumentDOM().library.items[idx].itemType == "bitmap") {  
        var myItem = fl.getDocumentDOM().library.items[idx];  
        fl.trace(myItem.name + " source is " + myItem.sourceFilePath);  
    }  
}
```

Siehe auch

[bitmapItem.sourceFileExists](#)

bitmapItem.useDeblocking

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`bitmapItem.useDeblocking`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob Deblocking aktiviert (`true`) oder nicht aktiviert (`false`) ist.

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Bitmap-Element handelt, wird im folgenden Beispiel das Deblocking für das Element aktiviert:

```
var libItem = fl.getDocumentDOM().library.items[0];  
libItem.useDeblocking = true;
```

bitmapItem.useImportedJPEGQuality

Verfügbarkeit

Flash MX 2004.

Verwendung

`bitmapItem.useImportedJPEGQuality`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob die standardmäßige importierte JPEG-Qualität verwendet (`true`) bzw. nicht verwendet (`false`) wird. Nur bei JPEG-Komprimierung verfügbar.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `useImportedJPEGQuality` des ersten Elements in der Bibliothek des aktuellen Dokuments auf `true` eingestellt:

```
fl.getDocumentDOM().library.items[0].useImportedJPEGQuality = true;  
alert(fl.getDocumentDOM().library.items[0].useImportedJPEGQuality);
```


Kapitel 6: CompiledClipInstance-Objekt

Vererbung [Element-Objekt](#) > [Instance-Objekt](#) > CompiledClipInstance-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das CompiledClipInstance-Objekt ist eine Unterklasse des Instance-Objekts. Es ist im Wesentlichen eine Instanz eines Movieclips, die in ein kompiliertes Clipbibliothekselement konvertiert wurde (siehe [Instance-Objekt](#)).

Übersicht über Eigenschaften

Abgesehen von den Eigenschaften für das [Instance-Objekt](#) stehen für das CompiledClipInstance-Objekt die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>compiledClipInstance.accName</code>	Ein String, der dem Feld „Name“ im Bedienfeld „Eingabehilfen“ entspricht.
<code>compiledClipInstance.actionScript</code>	Ein String, der das ActionScript für diese Instanz darstellt. Entspricht <code>symbolInstance.actionScript</code> .
<code>compiledClipInstance.description</code>	Ein String, der dem Feld „Beschreibung“ im Bedienfeld „Eingabehilfen“ entspricht.
<code>compiledClipInstance.forceSimple</code>	Ein boolescher Wert, der die Eingabehilfe für die Child-Objekte des Objekts aktiviert und deaktiviert.
<code>compiledClipInstance.shortcut</code>	Ein String, der dem Feld „Kurzbehl“ im Bedienfeld „Eingabehilfen“ entspricht.
<code>compiledClipInstance.silent</code>	Ein boolescher Wert, der die Eingabehilfe für das Objekt aktiviert oder deaktiviert; entspricht der umgekehrten Logik der Einstellung „Objekt mit Eingabehilfen versehen“ im Bedienfeld „Eingabehilfen“.
<code>compiledClipInstance.tabIndex</code>	Eine Ganzzahl, die dem Feld „Reihenfolgenposition“ im Bedienfeld „Eingabehilfen“ entspricht.

compiledClipInstance.accName

Verfügbarkeit

Flash MX 2004.

Verwendung

`compiledClipInstance.accName`

Beschreibung

Eigenschaft; ein String, der dem Feld „Name“ im Bedienfeld „Eingabehilfen“ entspricht. Bildschirmleseprogramme geben die Namen von Objekten akustisch wieder, um diese zu identifizieren.

Beispiel

Im folgenden Beispiel wird der Eingabehilfenname des ersten ausgewählten Objekts abgerufen und festgelegt:

```
// Get the name of the object.  
var theName = fl.getDocumentDOM().selection[0].accName;  
// Set the name of the object.  
fl.getDocumentDOM().selection[0].accName = 'Home Button';
```

compiledClipInstance.actionScript

Verfügbarkeit

Flash MX 2004.

Verwendung

`compiledClipInstance.actionScript`

Beschreibung

Eigenschaft; ein String, der das ActionScript für diese Instanz angibt. Entspricht `symbolInstance.actionScript`.

Beispiel

Mit dem folgenden Code wird das ActionScript bestimmten Elementen zugewiesen:

```
// Assign some ActionScript to a specified Button compiled clip instance.  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0]  
    .actionScript = "on(click) {trace('button is clicked');}";  
// Assign some ActionScript to the currently selected Button compiled clip instance.  
fl.getDocumentDOM().selection[0].actionScript =  
    "on(click) {trace('button is clicked');}";
```

compiledClipInstance.description

Verfügbarkeit

Flash MX 2004.

Verwendung

`compiledClipInstance.description`

Beschreibung

Eigenschaft; ein String, der dem Feld „Beschreibung“ im Bedienfeld „Eingabehilfen“ entspricht. Die Beschreibung wird von Bildschirmleseprogrammen akustisch wiedergegeben.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `description` abgerufen und festgelegt:

```
// Get the description of the current selection.  
var theDescription = fl.getDocumentDOM().selection[0].description;  
// Set the description of the current selection.  
fl.getDocumentDOM().selection[0].description =  
    "This is compiled clip number 1";
```

compiledClipInstance.forceSimple

Verfügbarkeit

Flash MX 2004.

Verwendung

`compiledClipInstance.forceSimple`

Beschreibung

Eigenschaft; ein boolescher Wert, der die Eingabehilfe für die Child-Objekte des Objekts aktiviert und deaktiviert. Dies entspricht der umgekehrten Logik der Einstellung „Child-Objekte mit Eingabehilfen versehen“ im Bedienfeld „Eingabehilfen“. Wenn der Wert für `forceSimpletrue` lautet, entspricht dies einer Deaktivierung der Option „Child-Objekte mit Eingabehilfen versehen“. Wenn der Wert für `forceSimplefalse` lautet, entspricht dies einer Aktivierung der Option „Child-Objekte mit Eingabehilfen versehen“.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `forceSimple` abgerufen und festgelegt:

```
// Query if the children of the object are accessible.  
var areChildrenAccessible = fl.getDocumentDOM().selection[0].forceSimple;  
// Allow the children of the object to be accessible.  
fl.getDocumentDOM().selection[0].forceSimple = false;
```

compiledClipInstance.shortcut

Verfügbarkeit

Flash MX 2004.

Verwendung

`compiledClipInstance.shortcut`

Beschreibung

Eigenschaft; ein String, der dem Feld „Kurzbehl“ im Bedienfeld „Eingabehilfen“ entspricht. Der Kurzbehl wird vom Bildschirmleseprogramm akustisch wiedergegeben. Diese Eigenschaft ist nicht für dynamische Textfelder verfügbar.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `shortcut` abgerufen und festgelegt:

```
// Get the shortcut key of the object.  
var theShortcut = fl.getDocumentDOM().selection[0].shortcut;  
// Set the shortcut key of the object.  
fl.getDocumentDOM().selection[0].shortcut = "Ctrl+I";
```

compiledClipInstance.silent

Verfügbarkeit

Flash MX 2004.

Verwendung

`compiledClipInstance.silent`

Beschreibung

Eigenschaft; ein boolescher Wert, der die Eingabehilfe für das Objekt aktiviert oder deaktiviert. Entspricht der umgekehrten Logik der Einstellung „Objekt mit Eingabehilfen versehen“ im Bedienfeld „Eingabehilfen“. Wenn der Wert für `silenttrue` lautet, ist „Objekt mit Eingabehilfen versehen“ deaktiviert. Wenn der Wert für `silentfalse` lautet, ist „Objekt mit Eingabehilfen versehen“ aktiviert.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `silent` abgerufen und festgelegt:

```
// Query if the object is accessible.  
var isSilent = fl.getDocumentDOM().selection[0].silent;  
// Set the object to be accessible.  
fl.getDocumentDOM().selection[0].silent = false;
```

compiledClipInstance.tabIndex

Verfügbarkeit

Flash MX 2004.

Verwendung

`compiledClipInstance.tabIndex`

Beschreibung

Eigenschaft; eine Ganzzahl, die dem Feld „Reihenfolgenposition“ im Bedienfeld „Eingabehilfen“ entspricht. Erstellt die Reihenfolge, in der ein Benutzer Objekte durch Drücken der Tabulatortaste ansteuern kann.

Beispiel

Das folgende Beispiel verdeutlicht, wie man die Eigenschaft `tabIndex` abrufen und einstellt:

```
// Get the tabIndex of the object.  
var theTabIndex = fl.getDocumentDOM().selection[0].tabIndex;  
// Set the tabIndex of the object.  
fl.getDocumentDOM().selection[0].tabIndex = 1;
```

Kapitel 7: compilerErrors-Objekt

Verfügbarkeit

Flash CS3 Professional.

Beschreibung

Beim compilerErrors-Objekt, das das Bedienfeld „Compiler-Fehler“ darstellt, handelt es sich um eine Eigenschaft des flash-Objekts (fl), auf das mithilfe von `fl.compilerErrors` zugegriffen wird (siehe [flash-Objekt \(fl\)](#)).

Übersicht über Methoden

Die folgenden Methoden können in Verbindung mit dem compilerErrors-Objekt verwendet werden:

Methode	Beschreibung
<code>compilerErrors.clear()</code>	Löscht den Inhalt des Bedienfelds „Compiler-Fehler“.
<code>compilerErrors.save()</code>	Speichert den Inhalt des Bedienfelds „Compiler-Fehler“ in einer lokalen Textdatei.

compilerErrors.clear()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
compilerErrors.clear()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; löscht den Inhalt des Bedienfelds „Compiler-Fehler“.

Beispiel

Im folgenden Beispiel wird der Inhalt des Bedienfelds „Compiler-Fehler“ gelöscht.

```
fl.compilerErrors.clear();
```

Siehe auch

[compilerErrors.save\(\)](#)

compilerErrors.save()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
compilerErrors.save(fileURI [, bAppendToFile [, bUseSystemEncoding]])
```

Parameter

fileURI Ein String im Format file:/// URI, der den Dateinamen der gespeicherten Datei angibt. Wenn *fileURI* bereits vorhanden ist und Sie nicht den Wert `true` für *bAppendToFile* angegeben haben, wird *fileURI* ohne Vorwarnung überschrieben.

bAppendToFile Ein optionaler boolescher Wert, der festlegt, ob der Inhalt des Bedienfelds „Compiler-Fehler“ an *fileURI* angehängt werden soll (`true`) oder nicht (`false`). Der Standardwert lautet `false`.

bUseSystemEncoding Ein optionaler boolescher Wert, der festlegt, ob der Text im Bedienfeld „Compiler-Fehler“ mithilfe der Systemkodierung gespeichert werden soll. Falls dieser Wert auf `false` gesetzt ist (der Standardwert), wird der Text im Bedienfeld „Compiler-Fehler“ mithilfe der UTF-8-Kodierung gespeichert, wobei der Text mit Bytereihenfolgenkennungen beginnt. Der Standardwert lautet `false`.

Rückgabewerte

Keine.

Beschreibung

Methode; speichert den Inhalt des Bedienfelds „Compiler-Fehler“ in einer lokalen Textdatei.

Beispiel

Im folgenden Beispiel wird der Inhalt des Bedienfelds „Compiler-Fehler“ im Ordner „C:\tests“ in der Datei „errors.log“ gespeichert:

```
fl.compilerErrors.save("file:///c:/tests/errors.log");
```

Siehe auch

[compilerErrors.clear\(\)](#)

Kapitel 8: ComponentInstance-Objekt

Vererbung [Element-Objekt](#) > [Instance-Objekt](#) > [SymbolInstance-Objekt](#) > ComponentInstance-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das ComponentInstance-Objekt ist eine Unterklasse des SymbolInstance-Objekts und entspricht einer Komponente in einem Bild (siehe [SymbolInstance-Objekt](#)).

Übersicht über Eigenschaften

Abgesehen von den Eigenschaften für das [SymbolInstance-Objekt](#) steht für das ComponentInstance-Objekt die folgende Eigenschaft zur Verfügung:

Eigenschaft	Beschreibung
<code>componentInstance.parameters</code>	Schreibgeschützt; ein Array der ActionScript 2.0-Eigenschaften, auf die vom Eigenschafteninspektor für Komponenten aus zugegriffen werden kann.

componentInstance.parameters

Verfügbarkeit

Flash MX 2004.

Verwendung

`componentInstance.parameters`

Beschreibung

Schreibgeschützt; ein Array der ActionScript 2.0-Eigenschaften, auf die vom Eigenschafteninspektor für Komponenten aus zugegriffen werden kann. Siehe [Parameter-Objekt](#).

Beispiel

Im folgenden Beispiel wird die Eigenschaft `parameters` abgerufen und eingestellt:

```
var parms = fl.getDocumentDOM().selection[0].parameters;
parms[0].value = "some value";
```

Siehe auch

[Parameter-Objekt](#)

Kapitel 9: componentsPanel-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das componentsPanel-Objekt ist eine Eigenschaft des flash-Objekts (fl) und stellt das Bedienfeld „Komponenten“ dar. Der Zugriff auf dieses Objekt kann über `fl.componentsPanel` erfolgen (siehe [flash-Objekt \(fl\)](#)).

Übersicht über Methoden

Für das componentsPanel-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>componentsPanel.addItemToDocument()</code>	Fügt die angegebene Komponente an der angegebenen Stelle dem Dokument hinzu.
<code>componentsPanel.reload()</code>	Aktualisiert die im Bedienfeld „Komponenten“ aufgeführte Komponentenliste.

componentsPanel.addItemToDocument()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
componentsPanel.addItemToDocument(position, categoryName, componentName)
```

Parameter

position Ein Punkt (z. B. {x:0, y:100}), der die Stelle angibt, an der die Komponente eingefügt werden soll. *position* muss in Relation zum Mittelpunkt der Komponente und nicht in Relation zum Registrierungspunkt (auch *Bezugspunkt* oder *Nullpunkt* genannt) der Komponente angegeben werden.

categoryName Ein String, der den Namen der Komponentenkategorie angibt (z. B. "Data"). Eine Liste der gültigen Kategorienamen finden Sie im Bedienfeld „Komponenten“.

componentName Ein String, der den Namen der Komponente in der festgelegten Kategorie angibt (z. B. "WebServiceConnector"). Eine Liste der gültigen Komponentennamen finden Sie im Bedienfeld „Komponenten“.

Rückgabewerte

Keine.

Beschreibung

Fügt die angegebene Komponente an der angegebenen Stelle dem Dokument hinzu.

Beispiel

In den folgenden Beispielen werden einige Verwendungsmöglichkeiten dieser Methode demonstriert:


```
fl.componentsPanel.addItemToDocument({x:0, y:0}, "User Interface", "CheckBox");  
fl.componentsPanel.addItemToDocument({x:0, y:100}, "Data", "WebServiceConnector");  
fl.componentsPanel.addItemToDocument({x:0, y:200}, "User Interface", "Button");
```

componentsPanel.reload()

Verfügbarkeit

Flash 8.

Verwendung

```
componentsPanel.reload()
```

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die Liste im Bedienfeld „Komponenten“ aktualisiert wurde, ansonsten `false`.

Beschreibung

Methode; aktualisiert die im Bedienfeld „Komponenten“ aufgeführte Komponentenliste.

Beispiel

Im folgenden Beispiel wird das Bedienfeld „Komponenten“ aktualisiert:

```
fl.componentsPanel.reload();
```

Kapitel 10: Contour-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Ein Contour-Objekt stellt einen geschlossenen Pfad von Halbkanten auf der Begrenzung einer Form dar.

Übersicht über Methoden

Für das Contour-Objekt steht die folgende Methode zur Verfügung:

Methode	Beschreibung
<code>contour.getHalfEdge()</code>	Gibt ein HalfEdge-Objekt auf der Kontur der Auswahl zurück.

Übersicht über Eigenschaften

Für das Contour-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>contour.fill</code>	Ein Fill-Objekt .
<code>contour.interior</code>	Schreibgeschützt; der Wert lautet <code>true</code> , wenn die Kontur einen Bereich umschließt, ansonsten <code>false</code> .
<code>contour.orientation</code>	Schreibgeschützt; eine Ganzzahl, die die Richtung der Kontur angibt.

contour.fill

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`contour.fill`

Beschreibung

Eigenschaft; ein [Fill-Objekt](#).

Beispiel

Sofern Sie über eine Kontur verfügen, für die eine Füllung ausgewählt wurde, wird im folgenden Beispiel im Bedienfeld „Ausgabe“ die Füllfarbe der Kontur angezeigt:

```
var insideContour = fl.getDocumentDOM().selection[0].contours[1];
var insideFill = insideContour.fill;
fl.trace(insideFill.color);
```

contour.getHalfEdge()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
contour.getHalfEdge()
```

Parameter

Keine.

Rückgabewerte

Ein [HalfEdge-Objekt](#).

Beschreibung

Methode; gibt ein [HalfEdge-Objekt](#) auf der Kontur der Auswahl zurück.

Beispiel

In diesem Beispiel werden alle Konturen der ausgewählten Form nacheinander geprüft und dann die Koordinaten der Scheitelpunkte im Bedienfeld „Ausgabe“ angezeigt:

```
// with a shape selected

var elt = fl.getDocumentDOM().selection[0];
elt.beginEdit();

var contourArray = elt.contours;
var contourCount = 0;
for (i=0;i<contourArray.length;i++)
{
    var contour = contourArray[i];
    contourCount++;
    var he = contour.getHalfEdge();

    var iStart = he.id;
    var id = 0;
    while (id != iStart)
    {
        // Get the next vertex.
        var vrt = he.getVertex();

        var x = vrt.x;
        var y = vrt.y;
        fl.trace("vrt: " + x + ", " + y);

        he = he.getNext();
        id = he.id;
    }
}
elt.endEdit();
```

contour.interior

Verfügbarkeit

Flash MX 2004.

Verwendung

`contour.interior`

Beschreibung

Schreibgeschützte Eigenschaft; der Wert lautet `true`, wenn die Kontur einen Bereich umschließt, ansonsten `false`.

Beispiel

In diesem Beispiel werden alle Konturen der ausgewählten Form nacheinander geprüft und dann wird für jede Kontur der Wert der Eigenschaft `interior` im Bedienfeld „Ausgabe“ angezeigt:

```
var elt = fl.getDocumentDOM().selection[0];
elt.beginEdit();

var contourArray = elt.contours;

var contourCount = 0;
for (i=0;i<contourArray.length;i++) {
    var contour = contourArray[i];
    fl.trace("Next Contour, interior:" + contour.interior );
    contourCount++;
}
elt.endEdit();
```

contour.orientation

Verfügbarkeit

Flash MX 2004.

Verwendung

`contour.orientation`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die die Richtung der Kontur angibt. Der Wert der Ganzzahl beträgt -1, wenn die Kontur im Uhrzeigersinn verläuft, und 1, wenn sie gegen den Uhrzeigersinn verläuft. Bei einer Kontur, die keinen Bereich umschließt, beträgt er 0.

Beispiel

Im folgenden Beispiel werden alle Konturen der ausgewählten Form nacheinander geprüft und dann wird für jede Kontur der Wert der Eigenschaft `orientation` im Bedienfeld „Ausgabe“ angezeigt:

```
var elt = fl.getDocumentDOM().selection[0];
elt.beginEdit();

var contourArray = elt.contours;

var contourCount = 0;
for (i=0;i<contourArray.length;i++) {
    var contour = contourArray[i];
    fl.trace("Next Contour, orientation:" + contour.orientation);
    contourCount++;
}
elt.endEdit();
```

Kapitel 11: Document-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Document-Objekt repräsentiert die Bühne. Es werden also nur FLA-Dateien als Dokumente betrachtet. Um das Document-Objekt für das aktuelle Dokument zurückzugeben, verwenden Sie `fl.getDocumentDOM()`.

Übersicht über Methoden

Mit dem Document-Objekt können Sie die folgenden Methoden verwenden:

Methode	Beschreibung
<code>document.addDataToDocument()</code>	Speichert die angegebenen Daten mit einem Dokument.
<code>document.addDataToSelection()</code>	Speichert die angegebenen Daten mit dem bzw. den ausgewählten Objekt(en).
<code>document.addFilter()</code>	Wendet einen Filter auf die ausgewählten Objekte an.
<code>document.addItem()</code>	Fügt dem angegebenen Document-Objekt ein Element aus einem beliebigen geöffneten Dokument bzw. aus einer Bibliothek hinzu.
<code>document.addNewLine()</code>	Fügt zwischen zwei Punkten einen neuen Pfad hinzu.
<code>document.addNewOval()</code>	Fügt im angegebenen Begrenzungsrechteck ein neues Oval-Objekt hinzu.
<code>document.addNewPrimitiveOval()</code>	Fügt eine neue Ellipsengrundform hinzu, die in den angegebenen Bereich passt.
<code>document.addNewPrimitiveRectangle()</code>	Fügt eine neue Rechteckgrundform hinzu, die in den angegebenen Bereich passt.
<code>document.addNewPublishProfile()</code>	Fügt ein neues Veröffentlichungsprofil hinzu und wählt es als aktuelles Profil aus.
<code>document.addNewRectangle()</code>	Fügt ein neues Rechteck bzw. ein Rechteck mit abgerundeten Ecken hinzu und passt es in den angegebenen Bereich ein.
<code>document.addNewScene()</code>	Fügt eine neue Szene (Timeline-Objekt) nach der gegenwärtig ausgewählten Szene als nächste Szene hinzu und wählt diese neue Szene als aktuelle Szene aus.
<code>document.addNewText()</code>	Fügt ein neues leeres Textfeld hinzu.
<code>document.align()</code>	Richtet die Auswahl aus.
<code>document.allowScreens()</code>	Verwenden Sie diese Methode vor der Eigenschaft „ <code>document.screenOutline</code> “ auf Seite 139.
<code>document.arrange()</code>	Ordnet die Auswahl auf der Bühne an.
<code>document.breakApart()</code>	Teilt die aktuelle Auswahl.
<code>document.canEditSymbol()</code>	Gibt an, ob das Menü und die Funktion „Symbole bearbeiten“ aktiviert sind.

Methode	Beschreibung
<code>document.canRevert()</code>	Stellt fest, ob Sie die <code>document.revert()</code> -Methode oder die <code>fl.revertDocument()</code> -Methode erfolgreich verwenden können.
<code>document.canTestMovie()</code>	Stellt fest, ob Sie die <code>document.testMovie()</code> -Methode erfolgreich verwenden können.
<code>document.canTestScene()</code>	Stellt fest, ob Sie die <code>document.testScene()</code> -Methode erfolgreich verwenden können.
<code>document.changeFilterOrder()</code>	Ändert den Index des Filters in der Filterliste.
<code>document.clipCopy()</code>	Kopiert die aktuelle Auswahl aus dem Dokument in die Zwischenablage.
<code>document.clipCut()</code>	Schneidet die aktuelle Auswahl aus dem Dokument aus und speichert sie in der Zwischenablage.
<code>document.clipPaste()</code>	Fügt den Inhalt der Zwischenablage in das Dokument ein.
<code>document.close()</code>	Schließt das angegebene Dokument.
<code>document.convertLinesToFills()</code>	Konvertiert die Linien der ausgewählten Objekte in Füllungen.
<code>document.convertToSymbol()</code>	Konvertiert die ausgewählten Bühnenelemente in ein neues Symbol.
<code>document.crop()</code>	Verwendet das zuoberst liegende ausgewählte Zeichnungsobjekt, um alle darunter liegenden ausgewählten Zeichnungsobjekte zuzuschneiden.
<code>document.debugMovie()</code>	Leitet eine Debugging-Sitzung mit dem Dokument ein.
<code>document.deleteEnvelope()</code>	Löscht die Hülle (eine Begrenzungsbox, die ein oder mehrere Objekte enthält) aus dem ausgewählten Objekt.
<code>document.deletePublishProfile()</code>	Löscht das gegenwärtig aktive Profil, sofern es mehrere Profile gibt.
<code>document.deleteScene()</code>	Löscht die aktuelle Szene (Timeline-Objekt) und, falls die gelöschte Szene nicht die letzte Szene war, wählt die nächste Szene als das aktuelle Timeline-Objekt aus.
<code>document.deleteSelection()</code>	Löscht die aktuelle Auswahl auf der Bühne.
<code>document.disableAllFilters()</code>	Deaktiviert alle Filter, die für die ausgewählten Objekte gelten.
<code>document.disableFilter()</code>	Deaktiviert den in der Filterliste angegebenen Filter.
<code>document.disableOtherFilters()</code>	Deaktiviert alle Filter mit Ausnahme des Filters, der sich an der angegebenen Stelle in der Filterliste befindet.
<code>document.distribute()</code>	Verteilt die Auswahl.
<code>document.distributeToLayers()</code>	Verteilt die aktuelle Auswahl auf Ebenen; entspricht dem Befehl „Auf Ebenen verteilen“.
<code>document.documentHasData()</code>	Stellt fest, ob das Dokument permanente Daten mit dem angegebenen Namen enthält.
<code>document.duplicatePublishProfile()</code>	Dupliziert das gegenwärtig aktive Profil und weist dem Duplikat den Fokus zu.
<code>document.duplicateScene()</code>	Erstellt von der gegenwärtig ausgewählten Szene eine Kopie, weist der neuen Szene einen eindeutigen Namen zu und wählt die neue Szene als aktuelle Szene aus.

Methode	Beschreibung
<code>document.duplicateSelection()</code>	Dupliziert die Auswahl auf der Bühne.
<code>document.editScene()</code>	Wählt die angegebene Szene als aktuelle Szene aus, damit sie bearbeitet werden kann.
<code>document.enableAllFilters()</code>	Aktiviert alle Filter in der Filterliste für die ausgewählten Objekte.
<code>document.enableFilter()</code>	Aktiviert den angegebenen Filter für die ausgewählten Objekte.
<code>document.enterEditMode()</code>	Versetzt das Authoring-Tool in den von dem Parameter angegebenen Bearbeitungsmodus.
<code>document.exitEditMode()</code>	Beendet den Symbolbearbeitungsmodus und verlagert den Fokus vom Bearbeitungsmodus aus auf die nächsthöhere Ebene.
<code>document.exportPNG()</code>	Exportiert das Dokument als eine oder mehrere PNG-Dateien.
<code>document.exportPublishProfile()</code>	Exportiert das gegenwärtig aktive Profil in eine XML-Datei.
<code>document.exportPublishProfileString()</code>	Gibt einen String zurück, der im XML-Format das angegebene Profil angibt.
<code>document.exportSWF()</code>	Exportiert das Dokument in das Flash SWC-Format.
<code>document.getAlignToDocument()</code>	Entspricht dem Abrufen des Werts der Schaltfläche „An Bühne“ im Bedienfeld „Ausrichten“.
<code>document.getBlendMode()</code>	Gibt einen String zurück, der den Mischmodus für das ausgewählte Objekt bzw. die ausgewählten Objekte angibt.
<code>document.getCustomFill()</code>	Gibt das Füllobjekt der ausgewählten Form zurück oder das Bedienfeld „Werkzeuge“ und den Eigenschafteninspektor.
<code>document.getCustomStroke()</code>	Gibt das Strichobjekt der ausgewählten Form zurück oder das Bedienfeld „Werkzeuge“ und den Eigenschafteninspektor.
<code>document.getDataFromDocument()</code>	Ruft den Wert der angegebenen Daten ab.
<code>document.getElementProperty()</code>	Ruft die angegebene <code>Element</code> -Eigenschaft der aktuellen Auswahl ab.
<code>document.getElementTextAttr()</code>	Ruft eine angegebene <code>TextAttrs</code> -Eigenschaft des ausgewählten Text-Objekts ab.
<code>document.getFilters()</code>	Gibt ein Array zurück, das die Liste der auf die gegenwärtig ausgewählten Objekte angewendeten Filter enthält.
<code>document.getMetadata()</code>	Gibt einen String zurück, der die mit dem Dokument verknüpften XML-Metadaten enthält.
<code>document.getMobileSettings()</code>	Gibt den String zurück, der an <code>document.setMobileSettings()</code> übergeben wurde.
<code>document.getPlayerVersion()</code>	Gibt einen String zurück, der die Player-Zielversion für das angegebene Dokument darstellt.
<code>document.getSelectionRect()</code>	Ruft das Begrenzungsrechteck der aktuellen Auswahl ab.
<code>document.getTextString()</code>	Ruft den gegenwärtig ausgewählten Text ab.
<code>document.getTimeline()</code>	Ruft das aktuelle Timeline-Objekt im Dokument ab.

Methode	Beschreibung
<code>document.getTransformationPoint()</code>	Ruft die Position des Transformierungspunkts der aktuellen Auswahl ab.
<code>document.group()</code>	Konvertiert die aktuelle Auswahl in eine Gruppe.
<code>document.importFile()</code>	Importiert eine Datei in das Dokument.
<code>document.importPublishProfile()</code>	Importiert ein Profil aus einer Datei.
<code>document.importPublishProfileString()</code>	Importiert einen XML-String, der ein Veröffentlichungsprofil darstellt, und legt es als aktuelles Profil fest.
<code>document.importSWF()</code>	Importiert eine SWF-Datei in das Dokument.
<code>document.intersect()</code>	Erstellt aus allen ausgewählten Zeichnungsobjekten ein Schnittobjekt.
<code>document.loadCuepointXML()</code>	Lädt eine Cue-Point-XML-Datei.
<code>document.match()</code>	Gibt allen ausgewählten Objekten die gleiche Größe.
<code>document.mouseClick()</code>	Führt mit dem Auswahlwerkzeug einen Mausklick aus.
<code>document.mouseDoubleClick()</code>	Führt mit dem Auswahlwerkzeug einen doppelten Mausklick aus.
<code>document.moveSelectedBezierPointsBy()</code>	Wenn die Auswahl mindestens einen Pfad mit mindestens einem ausgewählten Bézierpunkt enthält, werden mit dieser Methode alle ausgewählten Bézierpunkte auf den ausgewählten Pfaden um den angegebenen Betrag verschoben.
<code>document.moveSelectionBy()</code>	Verschiebt ausgewählte Objekte um einen bestimmten Betrag.
<code>document.optimizeCurves()</code>	Optimiert die Glättung für die aktuelle Auswahl, wobei mehrere Durchgänge zulässig sind, sofern angegeben; entspricht dem Befehl „Modifizieren“ > „Form“ > „Optimieren“.
<code>document.publish()</code>	Veröffentlicht das Dokument gemäß den aktiven Veröffentlichungseinstellungen („Datei“ > „Einstellungen für Veröffentlichungen“; entspricht dem Befehl „Datei“ > Veröffentlichen).
<code>document.punch()</code>	Verwendet das zuoberst liegende ausgewählte Zeichnungsobjekt, um alle darunter liegenden ausgewählten Zeichnungsobjekte auszustanzen.
<code>document.removeAllFilters()</code>	Entfernt alle Filter aus den ausgewählten Objekten.
<code>document.removeDataFromDocument()</code>	Entfernt permanente Daten mit dem angegebenen Namen, die mit dem Dokument verknüpft wurden.
<code>document.removeDataFromSelection()</code>	Entfernt permanente Daten mit dem angegebenen Namen, die mit der Auswahl verknüpft wurden.
<code>document.removeFilter()</code>	Entfernt den angegebenen Filter aus der Filterliste der ausgewählten Objekte.
<code>document.renamePublishProfile()</code>	Benennt das aktuelle Profil um.
<code>document.renameScene()</code>	Benennt die im Bedienfeld „Szene“ gegenwärtig ausgewählte Szene um.
<code>document.reorderScene()</code>	Verschiebt die angegebene Szene vor eine andere angegebene Szene.

Methode	Beschreibung
<code>document.resetOvalObject()</code>	Setzt alle Werte im Eigenschafteninspektor auf die Standardeinstellungen für das Oval-Objekt zurück.
<code>document.resetRectangleObject()</code>	Setzt alle Werte im Eigenschafteninspektor auf die Standardeinstellungen für das Rectangle-Objekt zurück.
<code>document.resetTransformation()</code>	Setzt die Transformationsmatrix zurück; entspricht der Auswahl von „Modifizieren“ > „Transformieren“ > „Transformation entfernen“.
<code>document.revert()</code>	Stellt die zuvor gespeicherte Version des angegebenen Dokuments wieder her; entspricht dem Befehl „Datei“ > „Wiederherstellen“.
<code>document.rotate3DSelection()</code>	Wendet eine 3D-Drehung auf die Auswahl an.
<code>document.rotateSelection()</code>	Dreht die Auswahl um einen in Grad angegebenen Betrag.
<code>document.save()</code>	Speichert das Dokument an seinem Standardspeicherort; entspricht der Auswahl von „Datei“ > „Speichern“.
<code>document.saveAndCompact()</code>	Speichert und komprimiert die Datei; entspricht der Auswahl von „Datei“ > „Speichern und komprimieren“.
<code>document.scaleSelection()</code>	Skaliert die Auswahl um einen angegebenen Betrag; entspricht der Skalierung des Objekts mit dem Werkzeug „Frei transformieren“.
<code>document.selectAll()</code>	Wählt alle Elemente auf der Bühne aus; entspricht dem Tastaturbefehl <Strg>+<A> (Windows) bzw. <Befehl>+<A> (Macintosh) sowie der Auswahl von „Bearbeiten“ > „Alles auswählen“.
<code>document.selectNone()</code>	Hebt die vorhandene Elementauswahl auf.
<code>document.setAlignToDocument()</code>	Legt fest, dass die Voreinstellungen für <code>document.align()</code> , <code>document.distribute()</code> , <code>document.match()</code> und <code>document.space()</code> für das Dokument gelten; entspricht der Aktivierung der Schaltfläche „An Bühne“ im Bedienfeld „Ausrichten“.
<code>document.setBlendMode()</code>	Legt den Mischmodus für die ausgewählten Objekte fest.
<code>document.setCustomFill()</code>	Legt die Füllungseinstellungen für das Bedienfeld „Werkzeuge“, den Eigenschafteninspektor und alle ausgewählten Formen fest.
<code>document.setCustomStroke()</code>	Legt die Stricheinstellungen für das Bedienfeld „Werkzeuge“, den Eigenschafteninspektor und alle ausgewählten Formen fest.
<code>document.setElementProperty()</code>	Legt die angegebene Element-Eigenschaft für ausgewählte Objekte im Dokument fest.
<code>document.setElementTextAttr()</code>	Weist der angegebenen TextAttrs-Eigenschaft der ausgewählten Textelemente den angegebenen Wert zu.
<code>document.setFillColor()</code>	Ersetzt die Füllfarbe der Auswahl durch die angegebene Farbe.
<code>document.setFilterProperty()</code>	Stellt eine angegebene Filtereigenschaft für die gegenwärtig ausgewählten Objekte ein.
<code>document.setFilters()</code>	Wendet Filter auf die ausgewählten Objekte an.
<code>document.setInstanceAlpha()</code>	Legt die Transparenz der Instanz fest.

Methode	Beschreibung
<code>document.setInstanceBrightness()</code>	Legt die Helligkeit der Instanz fest.
<code>document.setInstanceTint()</code>	Legt den Farbton der Instanz fest.
<code>document.setMetadata()</code>	Legt die XML-Metadaten für das angegebene Dokument fest und überschreibt dabei alle vorhandenen Metadaten.
<code>document.setMobileSettings()</code>	Legt den Wert eines XML-Einstellungsstrings in einer mobilen FLA-Datei fest.
<code>document.setOvalObjectProperty()</code>	Legt einen Wert für eine angegebene Eigenschaft von Ellipsengrundobjekten fest.
<code>document.setPlayerVersion()</code>	Legt die Flash Player-Version fest, die für das angegebene Dokument vorgesehen ist.
<code>document.setRectangleObjectProperty()</code>	Legt einen Wert für eine angegebene Eigenschaft von rechteckigen Grundobjekten fest.
<code>document.setSelectionBounds()</code>	Verschiebt die Auswahl und ändert gleichzeitig ihre Größe.
<code>document.setSelectionRect()</code>	Zeichnet anhand der angegebenen Koordinaten einen rechteckigen Auswahlrahmen in Relation zur Bühne.
<code>document.setStageVanishingPoint()</code>	Gibt den Fluchtpunkt bei der Anzeige von 3D-Objekten an.
<code>document.setStageViewAngle()</code>	Gibt den perspektivischen Winkel bei der Anzeige von 3D-Objekten an.
<code>document.setStroke()</code>	Stellt die Farbe, die Breite und den Stil der ausgewählten Striche ein.
<code>document.setStrokeColor()</code>	Ersetzt die Strichfarbe der Auswahl durch die angegebene Farbe.
<code>document.setStrokeSize()</code>	Ändert die Strichgröße der Auswahl auf die angegebene Größe.
<code>document.setStrokeStyle()</code>	Ersetzt den Strichstil der Auswahl durch den angegebenen Stil.
<code>document.setTextRectangle()</code>	Ändert das Begrenzungsrechteck des ausgewählten Textelements auf die angegebene Größe.
<code>document.setTextSelection()</code>	Stellt die Textauswahl des gegenwärtig ausgewählten Textfelds auf die von den Werten <i>startIndex</i> und <i>endIndex</i> angegebenen Werte ein.
<code>document.setTextString()</code>	Fügt einen Textstring ein.
<code>document.setTransformationPoint()</code>	Verschiebt den Transformationspunkt der aktuellen Auswahl.
<code>document.skewSelection()</code>	Neigt die Auswahl um einen angegebenen Betrag.
<code>document.smoothSelection()</code>	Glättet die Kurve der ausgewählten Füllungskonturen oder gekrümmten Linien.
<code>document.space()</code>	Ordnet die ausgewählten Objekte in gleichmäßigen Abständen an.
<code>document.straightenSelection()</code>	Begradigt die gegenwärtig ausgewählten Striche; entspricht der Schaltfläche „Begradigen“ im Bedienfeld „Werkzeuge“.
<code>document.swapElement()</code>	Tauscht die aktuelle Auswahl gegen die angegebene Auswahl aus.
<code>document.swapStrokeAndFill()</code>	Vertauscht die Strich- und Füllfarben.

Methode	Beschreibung
<code>document.testMovie()</code>	Führt im Dokument den Vorgang „Film testen“ aus.
<code>document.testScene()</code>	Führt für die aktuelle Szene des Dokuments den Vorgang „Szene testen“ aus.
<code>document.traceBitmap()</code>	Führt für die aktuelle Auswahl den Vorgang „Bitmap nachzeichnen“ aus; entspricht der Auswahl von „Modifizieren“ > „Bitmap“ > „Bitmap nachzeichnen“.
<code>document.transformSelection()</code>	Führt für die aktuelle Auswahl durch Anwenden der in den Argumenten angegebenen Matrix eine allgemeine Transformation durch.
<code>document.translate3DCenter()</code>	Legt die XYZ-Position fest, um die herum die Auswahl versetzt oder gedreht wird.
<code>document.translate3DSelection()</code>	Wendet eine 3D-Versetzung auf die Auswahl an.
<code>document.unGroup()</code>	Hebt die Gruppierung der aktuellen Auswahl auf.
<code>document.union()</code>	Verbindet alle ausgewählten Formen zu einem Zeichnungsobjekt.
<code>document.unlockAllElements()</code>	Hebt die Elementsperrung im gegenwärtig ausgewählten Bild auf.
<code>document.xmlPanel()</code>	Stellt ein XMLUI-Dialogfeld bereit.

Übersicht über Eigenschaften

Für das Document-Objekt stehen die folgenden Eigenschaften zur Auswahl.

Eigenschaft	Beschreibung
<code>document.accName</code>	Ein String, der dem Feld „Name“ im Bedienfeld „Eingabehilfen“ entspricht.
<code>document.as3AutoDeclare</code>	Ein boolescher Wert, der angibt, ob die auf der Bühne platzierten Instanzen automatisch den benutzerdefinierten timeline-Klassen hinzugefügt werden.
<code>document.as3Dialect</code>	Ein String, der angibt, welcher ActionScript 3.0-Dialekt im angegebenen Dokument verwendet wird.
<code>document.as3ExportFrame</code>	Eine Ganzzahl, die angibt, in welches Bild ActionScript 3.0-Klassen exportiert werden sollen.
<code>document.as3StrictMode</code>	Ein boolescher Wert, der angibt, ob der ActionScript 3.0-Compiler mit aktiviertem oder mit deaktiviertem striktem Modus kompilieren soll.
<code>document.as3WarningsMode</code>	Ein boolescher Wert, der angibt, ob der ActionScript 3.0-Compiler mit aktiviertem oder mit deaktiviertem Warnmodus kompilieren soll.
<code>document.asVersion</code>	Eine Ganzzahl, die angibt, welche ActionScript-Version im der angegebenen Datei verwendet wird.
<code>document.autoLabel</code>	Ein boolescher Wert, der dem Kontrollkästchen „Automatische Bezeichnung“ im Bedienfeld „Eingabehilfen“ entspricht.
<code>document.backgroundColor</code>	Ein String, ein hexadezimaler Wert oder eine Ganzzahl zur Darstellung der Hintergrundfarbe.
<code>document.currentPublishProfile</code>	Ein String, der den Namen des aktiven Veröffentlichungsprofils für das angegebene Dokument angibt.
<code>document.currentTimeline</code>	Eine Ganzzahl, die den Index der aktiven Zeitleiste angibt.

Eigenschaft	Beschreibung
<code>document.description</code>	Ein String, der dem Feld „Beschreibung“ im Bedienfeld „Eingabehilfen“ entspricht.
<code>document.docClass</code>	Gibt an, welche ActionScript 3.0-Klasse der höchsten Ebene dem Dokument zugeordnet ist.
<code>document.externalLibraryPath</code>	Ein String, der eine Liste von Elementen enthält, die sich im Pfad der externen ActionScript 3.0-Bibliothek des Dokuments befinden, und dieser Pfad wiederum gibt den Speicherort von SWC-Dateien an, die zur Laufzeit als gemeinsame Bibliotheken verwendet werden.
<code>document.forceSimple</code>	Ein boolescher Wert, der angibt, ob die Child-Objekte des angegebenen Objekts mit Eingabehilfen versehen sind.
<code>document.frameRate</code>	Ein Gleitkommawert, der angibt, wie viele Bilder beim Abspielen der SWF-Datei pro Sekunde angezeigt werden; der Standardwert lautet 12.
<code>document.height</code>	Eine Ganzzahl, die die Höhe des Dokuments (Bühne) in Pixel angibt.
<code>document.id</code>	Eine eindeutige Ganzzahl (automatisch zugewiesen), die ein Dokument für die Dauer einer Flash-Sitzung identifiziert.
<code>document.library</code>	Schreibgeschützt; das library-Objekt eines Dokuments.
<code>document.libraryPath</code>	Ein String, der eine Liste von Elementen enthält, die sich im Pfad der ActionScript 3.0-Bibliothek des Dokuments befinden, und dieser Pfad wiederum gibt den Speicherort von SWC-Dateien oder SWC-Dateien enthaltenden Ordnern an.
<code>document.livePreview</code>	Ein boolescher Wert, der angibt, ob die Live-Vorschau aktiviert ist.
<code>document.name</code>	Schreibgeschützt; ein String, der den Namen eines Dokuments angibt (FLA-Datei).
<code>document.path</code>	Schreibgeschützt; ein String, der den Pfad des Dokuments in einem plattformspezifischen Format angibt.
<code>document.pathURI</code>	Schreibgeschützt; ein String im Format file:/// URI, der den Pfad des Dokuments angibt.
<code>document.publishProfiles</code>	Schreibgeschützt; ein Array der Veröffentlichungsprofilnamen für das Dokument.
<code>document.screenOutline</code>	Schreibgeschützt; das aktuelle ScreenOutline-Objekt des Dokuments. Das ScreenOutline-Objekt ist ab Flash Professional CS5 veraltet.
<code>document.selection</code>	Ein Array der im Dokument ausgewählten Objekte.
<code>document.silent</code>	Ein boolescher Wert, der angibt, ob das Objekt mit Eingabehilfen versehen ist.
<code>document.sourcePath</code>	Ein String, der eine Liste von Elementen enthält, die sich im ActionScript 3.0-Quellpfad des Dokuments befinden, und dieser Pfad wiederum gibt den Speicherort von Dateien der ActionScript-Klasse an.
<code>document.timelines</code>	Schreibgeschützt; ein Array von Timeline-Objekten (siehe Timeline-Objekt).
<code>document.viewMatrix</code>	Schreibgeschützt, ein Matrix-Objekt .
<code>document.width</code>	Eine Ganzzahl, die die Breite des Dokuments (Bühne) in Pixel angibt.
<code>document.zoomFactor</code>	Gibt den Zoomfaktor auf der Bühne während des Authorings an.

document.accName

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.accName
```

Beschreibung

Eigenschaft; ein String, der dem Feld „Name“ im Bedienfeld „Eingabehilfen“ entspricht. Bildschirmleseprogramme geben die Namen von Objekten akustisch wieder, um diese zu identifizieren.

Beispiel

Im folgenden Beispiel wird dem Dokument der Eingabehilfenname "Main Movie" zugewiesen:

```
fl.getDocumentDOM().accName = "Main Movie";
```

Im folgenden Beispiel wird der Eingabehilfenname des Dokuments abgerufen:

```
fl.trace(fl.getDocumentDOM().accName);
```

document.addDataToDocument()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.addDataToDocument(name, type, data)
```

Parameter

name Ein String, der den Namen der hinzuzufügenden Daten angibt.

type Ein String, der den Typ der hinzuzufügenden Daten angibt. Die zulässigen Werte lauten "integer", "integerArray", "double", "doubleArray", "string" und "byteArray".

data Der hinzuzufügende Wert. Zulässige Typen hängen vom Parameter *type* ab.

Rückgabewerte

Keine.

Beschreibung

Methode; speichert die angegebenen Daten mit einem Dokument. Die Daten werden der FLA-Datei hinzugefügt und sind für JavaScript verfügbar, wenn die Datei erneut geöffnet wird.

Beispiel

Im folgenden Beispiel wird dem aktuellen Dokument der Ganzzahlenwert 12 hinzugefügt:

```
fl.getDocumentDOM().addDataToDocument("myData", "integer", 12);
```

Im folgenden Beispiel wird der Wert der Daten mit dem Namen "myData" zurückgegeben und im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace(fl.getDocumentDOM().getDataFromDocument("myData"));
```

Siehe auch

[document.getDataFromDocument\(\)](#), [document.removeDataFromDocument\(\)](#)

document.addDataToSelection()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.addDataToSelection(name, type, data)
```

Parameter

name Ein String, der den Namen der permanenten Daten angibt.

type Definiert den Typ der Daten. Die zulässigen Werte lauten "integer", "integerArray", "double", "doubleArray", "string" und "byteArray".

data Der hinzuzufügende Wert. Zulässige Typen hängen vom Parameter *type* ab.

Rückgabewerte

Keine.

Beschreibung

Methode; speichert die angegebenen Daten mit dem bzw. den ausgewählten Objekt(en). Die Daten werden der FLA-Datei hinzugefügt und sind für JavaScript verfügbar, wenn die Datei erneut geöffnet wird. Nur Symbole und Bitmaps unterstützen permanente Daten.

Beispiel

Im folgenden Beispiel wird dem ausgewählten Objekt der Ganzzahlenwert 12 hinzugefügt:

```
fl.getDocumentDOM().addDataToSelection("myData", "integer", 12);
```

Siehe auch

[document.removeDataFromSelection\(\)](#)

document.addFilter()

Verfügbarkeit

Flash 8.

Verwendung

```
document.addFilter(filterName)
```

Parameter

filterName Ein String, der den Filter angibt, der der Filterliste hinzugefügt und für die ausgewählten Objekte aktiviert werden soll. Die zulässigen Werte lauten "adjustColorFilter", "bevelFilter", "blurFilter", "dropShadowFilter", "glowFilter", "gradientBevelFilter" und "gradientGlowFilter".

Rückgabewerte

Keine.

Beschreibung

Methode; wendet einen Filter auf die ausgewählten Objekte an und fügt den Filter am Ende der Filterliste ein.

Beispiel

Im folgenden Beispiel wird auf die ausgewählten Objekte ein Glühen-Filter angewendet:

```
fl.getDocumentDOM().addFilter("glowFilter");
```

Siehe auch

[document.changeFilterOrder\(\)](#), [document.disableFilter\(\)](#), [document.enableFilter\(\)](#),
[document.getFilters\(\)](#), [document.removeFilter\(\)](#), [document.setBlendMode\(\)](#),
[document.setFilterProperty\(\)](#)

document.addItem()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.addItem(position, item)
```

Parameter

position Ein Punkt, der anhand von *x*- und *y*-Koordinaten die Stelle angibt, an der das Element eingefügt werden soll. Er orientiert sich am Mittelpunkt eines Symbols bzw. an der oberen linken Ecke einer Bitmap oder eines Videos.

item Ein Item-Objekt, das angibt, welches Element hinzugefügt werden soll und aus welcher Bibliothek dieses Element stammt (siehe [Item-Objekt](#)).

Rückgabewerte

Ein boolescher Wert; *true*, wenn erfolgreich, ansonsten *false*.

Beschreibung

Methode; fügt dem angegebenen Document-Objekt ein Element aus einem beliebigen geöffneten Dokument bzw. aus einer Bibliothek hinzu.

Beispiel

Im folgenden Beispiel wird das erste Bibliothekselement an der für das ausgewählte Symbol, die ausgewählte Bitmap oder das ausgewählte Video angegebenen Stelle im ersten Dokument hinzugefügt:


```
var item = fl.documents[0].library.items[0];  
fl.documents[0].addItem({x:0,y:0}, item);
```

Im folgenden Beispiel wird das aus der Bibliothek des aktuellen Dokuments stammende Symbol `myMovieClip` dem aktuellen Dokument hinzugefügt:

```
var itemIndex = fl.getDocumentDOM().library.findItemIndex("myMovieClip");  
var theItem = fl.getDocumentDOM().library.items[itemIndex];  
fl.getDocumentDOM().addItem({x:0,y:0}, theItem);
```

Im folgenden Beispiel wird das Symbol `myMovieClip`, das aus dem zweiten Dokument des Dokumentarrays stammt, dem dritten Dokument des Dokumentarrays hinzugefügt:

```
var itemIndex = fl.documents[1].library.findItemIndex("myMovieClip");  
var theItem = fl.documents[1].library.items[itemIndex];  
fl.documents[2].addItem({x:0,y:0}, theItem);
```

document.addNewLine()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.addNewLine(startPoint, endpoint)
```

Parameter

startpoint Zwei Gleitkommazahlen, die den Anfangspunkt der Linie anhand von *x*- und *y*-Koordinaten angeben.

endpoint Zwei Gleitkommazahlen, die den Endpunkt der Linie anhand von *x*- und *y*-Koordinaten angeben.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt zwischen zwei Punkten einen neuen Pfad hinzu. Die Methode verwendet die aktuellen Strichattribute des Dokuments und fügt den Pfad im aktuellen Bild und auf der aktuellen Ebene ein. Das gleiche Ergebnis erzielen Sie, wenn Sie auf das Linienwerkzeug klicken und eine Linie zeichnen.

Beispiel

Im folgenden Beispiel wird eine Linie hinzugefügt, die von dem angegebenen Anfangspunkt bis zu dem angegebenen Endpunkt reicht:

```
fl.getDocumentDOM().addNewLine({x:216.7, y:122.3}, {x:366.8, y:165.8});
```

document.addNewOval()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.addNewOval(boundingBox [, bSuppressFill [, bSuppressStroke ]])
```

Parameter

boundingRectangle Ein Rechteck, das die Grenzen der hinzuzufügenden Ellipse angibt. Informationen zum Format von *boundingRectangle* finden Sie unter [document.addNewRectangle\(\)](#).

bSuppressFill Ein boolescher Wert; bei dem Wert `true` wird die Form ohne Füllung erstellt. Der Standardwert lautet `false`. Dieser Parameter ist optional.

bSuppressStroke Ein boolescher Wert; bei dem Wert `true` wird die Form ohne Strich erstellt. Der Standardwert lautet `false`. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt in dem angegebenen Begrenzungsrechteck ein neues Oval-Objekt hinzu. Diese Methode führt denselben Vorgang wie das Ellipsenwerkzeug aus. Sie verwendet die aktuellen standardmäßigen Strich- und Füllungsattribute des Dokuments und fügt die Ellipse im aktuellen Bild und auf der aktuellen Ebene ein. Wenn sowohl für *bSuppressFill* als auch für *bSuppressStroke* der Wert `true` angegeben wurde, ist die Methode wirkungslos.

Beispiel

Im folgenden Beispiel wird eine neue Ellipse, die 164 Pixel breit und 178 Pixel hoch ist, innerhalb der angegebenen Koordinaten hinzugefügt:

```
fl.getDocumentDOM().addNewOval({left:72,top:50,right:236,bottom:228});
```

Im folgenden Beispiel wird die Ellipse ohne Füllung gezeichnet:

```
fl.getDocumentDOM().addNewOval({left:72,top:50,right:236,bottom:228}, true);
```

Im folgenden Beispiel wird die Ellipse ohne Strich gezeichnet:

```
fl.getDocumentDOM().addNewOval({left:72,top:50,right:236,bottom:228}, false, true);
```

Siehe auch

[document.addNewPrimitiveOval\(\)](#)

document.addNewPrimitiveOval()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
document.addNewPrimitiveOval( boundingRectangle [, bSpupressFill [, bSuppressStroke ] ] )
```

Parameter

boundingRectangle Ein Rechteck, das die Grenzen angibt, innerhalb derer die neue Ellipsengrundform hinzugefügt wird. Informationen zum Format von *boundingRectangle* finden Sie unter [document.addNewRectangle\(\)](#).

bSuppressFill Ein boolescher Wert; bei dem Wert `true` wird die Ellipse ohne Füllung erstellt. Der Standardwert lautet `false`. Dieser Parameter ist optional.

bSuppressStroke Ein boolescher Wert; bei dem Wert `true` wird die Ellipse ohne Strich erstellt. Der Standardwert lautet `false`. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt eine neue Ellipsengrundform hinzu, die in den angegebenen Bereich passt. Diese Methode führt denselben Vorgang wie das Werkzeug für Ellipsengrundform aus. Die Ellipsengrundform verwendet die aktuellen standardmäßigen Strich- und Füllungsattribute des Dokuments und wird im aktuellen Bild und auf der aktuellen Ebene hinzugefügt. Wenn sowohl für *bSuppressFill* als auch für *bSuppressStroke* der Wert `true` angegeben wurde, ist die Methode wirkungslos.

Beispiel

Im folgenden Beispiel werden Ellipsengrundformen mit und ohne Füllung und Strich innerhalb der angegebenen Koordinaten hinzugefügt:

```
// Add an oval primitive with fill and stroke
fl.getDocumentDOM().addNewPrimitiveOval({left:0,top:0,right:100,bottom:100});
// Add an oval primitive without a fill
fl.getDocumentDOM().addNewPrimitiveOval({left:100,top:100,right:200,bottom:200}, true);
// Add an oval primitive without a stroke
fl.getDocumentDOM().addNewPrimitiveOval({left:200,top:200,right:300,bottom:300},false,true);
```

Siehe auch

[document.addNewOval\(\)](#)

document.addNewPrimitiveRectangle()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
document.addNewPrimitiveRectangle( boundingRectangle, roundness, [, bSuppressFill [,
bSuppressStroke ]] ) )
```

Parameter

rect Ein Rechteck, das die Grenzen angibt, innerhalb derer die neue Rechteckgrundform hinzugefügt wird. Informationen zum Format von *boundingRectangle* finden Sie unter [document.addNewRectangle\(\)](#).

roundness Eine Ganzzahl zwischen 0 und 999, die die Anzahl der Punkte zur Abrundung der Ecken angibt.

bSuppressFill Ein boolescher Wert; bei dem Wert `true` wird das Rechteck ohne Füllung erstellt. Der Standardwert lautet `false`. Dieser Parameter ist optional.

bSuppressStroke Ein boolescher Wert; bei dem Wert `true` wird das Rechteck ohne Strich erstellt. Der Standardwert lautet `false`. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt eine neue Rechteckgrundform hinzu, die in den angegebenen Bereich passt. Diese Methode führt denselben Vorgang wie das Werkzeug für Rechteckgrundform aus. Die Rechteckgrundform verwendet die aktuellen standardmäßigen Strich- und Füllungsattribute des Dokuments und wird im aktuellen Bild und auf der aktuellen Ebene hinzugefügt. Wenn sowohl für *bSuppressFill* als auch für *bSuppressStroke* der Wert `true` angegeben wurde, ist die Methode wirkungslos.

Beispiel

Im folgenden Beispiel werden Rechteckgrundformen mit und ohne Füllung und Strich sowie mit unterschiedlicher Abrundung innerhalb der angegebenen Koordinaten hinzugefügt:

```
// Add a rectangle primitive with fill and stroke
fl.getDocumentDOM().addNewPrimitiveRectangle({left:0,top:0,right:100,bottom:100}, 0);
// Add a rectangle primitive without a fill
fl.getDocumentDOM().addNewPrimitiveRectangle({left:100,top:100,right:200,bottom:200}, 20,
true);
// Add a rectangle primitive without a stroke
fl.getDocumentDOM().addNewPrimitiveRectangle({left:200,top:200,right:300,bottom:300},
50,false,true);
```

Siehe auch

[document.addNewRectangle\(\)](#)

document.addNewPublishProfile()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.addNewPublishProfile([profileName])
```

Parameter

profileName Der eindeutige Name des neuen Profils. Wenn Sie keinen Namen angeben, wird ein Standardname verwendet. Dieser Parameter ist optional.

Rückgabewerte

Eine Ganzzahl, die den Index des neuen Profils in der Profilliste angibt. Gibt -1 zurück, wenn kein neues Profil erstellt werden kann.

Beschreibung

Methode; fügt ein neues Veröffentlichungsprofil hinzu und wählt es als aktuelles Profil aus.

Beispiel

Im folgenden Beispiel wird ein neues Veröffentlichungsprofil mit einem Standardnamen hinzugefügt und anschließend wird der Name des Profils im Bedienfeld „Ausgabe“ angezeigt:

```
fl.getDocumentDOM().addNewPublishProfile();  
fl.outputPanel.trace(fl.getDocumentDOM().currentPublishProfile);
```

Im folgenden Beispiel wird ein neues Veröffentlichungsprofil mit dem Namen "my profile" hinzugefügt:

```
fl.getDocumentDOM().addNewPublishProfile("my profile");
```

Siehe auch

[document.deletePublishProfile\(\)](#)

document.addNewRectangle()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.addNewRectangle(boundingBox, roundness  
    [, bSuppressFill [, bSuppressStroke]])
```

Parameter

boundingRectangle Ein Rechteck, das die Begrenzungen angibt, innerhalb derer das neue Rechteck hinzugefügt wird. Dabei wird folgendes Format verwendet: {left:Wert1,top:Wert2,right:Wert3,bottom:Wert4}. Die Werte **left** und **top** geben die Lage der oberen linken Ecke an (z. B. kennzeichnet **left:0,top:0** die obere linke Ecke der Bühne) und die Werte **right** und **bottom** die Lage der unteren rechten Ecke. Die Breite des Rechtecks entspricht somit der Differenz der Werte **left** und **right** und die Differenz der Werte **top** und **bottom** ergibt die Höhe.

Das bedeutet, dass die Begrenzung des Rechtecks nicht mit den im Eigenschafteninspektor angegebenen Werten übereinstimmt. Die Werte für **left** und **top** entsprechen zwar den X- und Y-Werten im Eigenschafteninspektor, aber **right** und **bottom** stimmen mit den Angaben von „W“ und „H“ im Eigenschafteninspektor nicht überein. Stellen Sie sich beispielsweise ein Rechteck mit folgender Begrenzung vor:

```
{left:10,top:10,right:50,bottom:100}
```

Für dieses Rechteck würden im Eigenschafteninspektor folgende Werte angezeigt:

```
X = 10, Y = 10, W = 40, H = 90
```

roundness Eine Ganzzahl zwischen 0 und 999, die die Abrundung der Ecken angibt. Der Wert wird als Punktzahl angegeben. Je größer der Wert, desto stärker die Abrundung.

bSuppressFill Ein boolescher Wert; bei dem Wert **true** wird die Form ohne Füllung erstellt. Der Standardwert lautet **false**. Dieser Parameter ist optional.

bSuppressStroke Ein boolescher Wert; bei dem Wert **true** wird das Rechteck ohne Strich erstellt. Der Standardwert lautet **false**. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt ein neues Rechteck bzw. ein Rechteck mit abgerundeten Ecken hinzu und passt es in den angegebenen Bereich ein. Diese Methode führt denselben Vorgang wie das Rechteckwerkzeug aus. Sie verwendet die aktuellen standardmäßigen Strich- und Füllungsattribute des Dokuments und fügt das Rechteck im aktuellen Bild und auf der aktuellen Ebene ein. Wenn sowohl für *bSuppressFill* als auch für *bSuppressStroke* der Wert `true` angegeben wurde, ist die Methode wirkungslos.

Beispiel

Im folgenden Beispiel wird ein neues Rechteck ohne Abrundung innerhalb der angegebenen Koordinaten eingefügt. Es ist 100 Pixel breit und 100 Pixel hoch:

```
fl.getDocumentDOM().addNewRectangle({left:0,top:0,right:100,bottom:100},0);
```

Im folgenden Beispiel wird ein neues Rechteck hinzugefügt, das keine abgerundeten Ecken und keine Füllung aufweist. Es ist 100 Pixel breit und 200 Pixel hoch:

```
fl.getDocumentDOM().addNewRectangle({left:10,top:10,right:110,bottom:210},0, true);
```

Im folgenden Beispiel wird ein neues Rechteck hinzugefügt, das keine abgerundeten Ecken und keinen Strich aufweist. Es ist 200 Pixel breit und 200 Pixel hoch:

```
fl.getDocumentDOM().addNewRectangle({left:20,top:20,right:220,bottom:120},0, false, true);
```

Siehe auch

`document.addNewPrimitiveRectangle()`

document.addNewScene()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.addNewScene ( [name] )
```

Parameter

name Gibt den Namen der Szene an. Wenn Sie keinen Namen angeben, wird automatisch ein neuer Szenenname erstellt.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die Szene erfolgreich hinzugefügt wurde, ansonsten `false`.

Beschreibung

Methode; fügt eine neue Szene ([Timeline-Objekt](#)) nach der gegenwärtig ausgewählten Szene als nächste Szene hinzu und wählt diese neue Szene als aktuelle Szene aus. Wenn der angegebene Szenenname bereits vorhanden ist, wird die Szene nicht hinzugefügt und die Methode gibt einen Fehler zurück.

Beispiel

Im folgenden Beispiel wird in dem aktuellen Dokument eine neue Szene mit dem Namen `myScene` nach der aktuellen Szene hinzugefügt. Der Wert der Variablen `success` lautet `true`, wenn die neue Szene erstellt wurde, ansonsten `false`.

```
var success = fl.getDocumentDOM().addNewScene("myScene");
```

Im folgenden Beispiel wird eine neue Szene hinzugefügt und der Szene ein Standardname zugewiesen. Wenn nur eine Szene vorhanden ist, erhält die neu erstellte Szene den Namen "Szene 2".

```
fl.getDocumentDOM().addNewScene();
```

document.addNewText()

Verfügbarkeit

Flash MX 2004; optionaler *text*-Parameter, der in Flash CS3 Professional hinzugefügt wurde.

Verwendung

```
document.addNewText(boundingBox [, text ])
```

Parameter

boundingRectangle Legt die Größe und die Lage des Textfelds fest. Informationen zum Format von *boundingRectangle* finden Sie unter [document.addNewRectangle\(\)](#).

text Ein optionaler String, der den Text, der im Feld platziert werden soll, angibt. Wenn Sie diesen Parameter auslassen, wechselt die Auswahl im Bedienfeld „Werkzeuge“ zum Textwerkzeug. Wenn Sie nicht möchten, dass die Werkzeugauswahl geändert wird, übergeben Sie einen Wert für *text*.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt ein neues Textfeld ein und gibt optional Text in das Feld ein. Wenn Sie den *text*-Parameter auslassen, können Sie [document.setTextString\(\)](#) aufrufen, um das Textfeld auszufüllen.

Beispiel

Im folgenden Beispiel wird in der oberen linken Ecke der Bühne ein neues Textfeld erstellt und der Textstring auf "Hello World" eingestellt:

```
fl.getDocumentDOM().addNewText({left:0, top:0, right:100, bottom:100} , "Hello World!" );  
fl.getDocumentDOM().setTextString('Hello World!');
```

Siehe auch

[document.setTextString\(\)](#)

document.align()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.align(alignmode [, bUseDocumentBounds])
```

Parameter

alignmode Ein String, der angibt, wie die Auswahl ausgerichtet werden soll. Die zulässigen Werte lauten "left", "right", "top", "bottom", "vertical center" und "horizontal center".

bUseDocumentBounds Ein boolescher Wert; bei dem Wert `true` erfolgt die Ausrichtung an den Dokumentgrenzen. Andernfalls orientiert die Methode sich an den Grenzen der ausgewählten Objekte. Der Standardwert lautet `false`. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; richtet die Auswahl aus.

Beispiel

Im folgenden Beispiel werden Objekte links und an der Bühne ausgerichtet. Die gleiche Wirkung erzielen Sie, wenn Sie die Einstellung „An Bühne“ im Bedienfeld „Ausrichten“ aktivieren und auf „Ausrichten: Linke Kante“ klicken:

```
fl.getDocumentDOM().align("left", true);
```

Siehe auch

[document.distribute\(\)](#), [document.getAlignToDocument\(\)](#), [document.setAlignToDocument\(\)](#)

document.allowScreens()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.allowScreens()
```

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; `true`, wenn `document.screenOutline` problemlos verwendet werden kann, ansonsten `false`.

Beschreibung

Methode; wird vor der Eigenschaft `document.screenOutline` verwendet. Wenn diese Methode den Wert `true` zurückgibt, können Sie problemlos auf `document.screenOutline` zugreifen. Falls Sie versuchen, in einem Dokument ohne Bildschirme `document.screenOutline` zu verwenden, zeigt Flash einen Fehler an.

Beispiel

Im folgenden Beispiel wird festgestellt, ob die `screens`-Methoden in dem aktuellen Dokument eingesetzt werden können:

```
if(fl.getDocumentDOM().allowScreens()) {  
    fl.trace("screen outline is available.");  
}  
else {  
    fl.trace("whoops, no screens.");  
}
```

Siehe auch

[document.screenOutline](#)

document.arrange()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.arrange(arrangeMode)
```

Parameter

arrangeMode Gibt an, in welche Richtung die Auswahl verschoben werden soll. Die zulässigen Werte lauten "back", "backward", "forward" und "front". Dieser Parameter bietet die gleiche Funktionalität wie die Optionen im Menü „Modifizieren“ > „Anordnen“.

Rückgabewerte

Keine.

Beschreibung

Methode; ordnet die Auswahl auf der Bühne an. Diese Methode gilt nur für Objekte, bei denen es sich nicht um Formen handelt.

Beispiel

Im folgenden Beispiel wird die aktuelle Auswahl in den Vordergrund verschoben:

```
fl.getDocumentDOM().arrange("front");
```

document.as3AutoDeclare

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`document.as3AutoDeclare`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob die auf der Bühne platzierten Instanzen automatisch den benutzerdefinierten timeline-Klassen hinzugefügt werden. Der Standardwert ist `true`.

Beispiel

Im folgenden Beispiel wird festgelegt, dass Instanzen, die im aktuellen Dokument auf der Bühne platziert werden, manuell zu benutzerdefinierten timeline-Klassen hinzugefügt werden müssen.

```
fl.getDocumentDOM().as3AutoDeclare=false;
```

document.as3Dialect

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`document.as3Dialect`

Beschreibung

Eigenschaft; ein String, der angibt, welcher ActionScript 3.0-Dialekt im angegebenen Dokument verwendet wird. Der Standardwert ist `"AS3"`. Wenn Sie Prototypklassen zulassen möchten, wie es in früheren ECMAScript-Spezifikationen zulässig war, setzen Sie diesen Wert auf `"ES"`.

Beispiel

Im folgenden Beispiel wird festgelegt, dass im aktuellen Dokument der Dialekt ECMAScript verwendet wird:

```
fl.getDocumentDOM().as3Dialect="ES";
```

Siehe auch

[document.asVersion](#)

document.as3ExportFrame

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`document.as3ExportFrame`

Beschreibung

Eigenschaft; eine Ganzzahl, die angibt, in welches Bild ActionScript 3.0-Klassen exportiert werden sollen. Standardmäßig werden Klassen in Bild 1 exportiert.

Beispiel

Im folgenden Beispiel wird das Bild, in das Klassen exportiert werden, von der Standardeinstellung 1 in 5 geändert.

```
var myDocument = fl.getDocumentDOM();
fl.outputPanel.trace("Export classes in frame:" value before modification is " +
myDocument.as3ExportFrame);
myDocument.as3ExportFrame = 5;
fl.outputPanel.trace("Export classes in frame:" value after modification is " +
myDocument.as3ExportFrame);
```

document.as3StrictMode

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`document.as3StrictMode`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob der ActionScript 3.0-Compiler mit aktiviertem (`true`) oder mit deaktiviertem (`false`) striktem Modus kompilieren soll. Im strikten Modus werden Warnungen als Fehler gemeldet, d. h. die Kompilierung wird erst dann gelingen, wenn diese Fehler beseitigt sind. Der Standardwert lautet `true`.

Beispiel

Im folgenden Beispiel wird die Compiler-Option für den strikten Modus deaktiviert.

```
var myDocument = fl.getDocumentDOM();
fl.outputPanel.trace("Strict Mode value before modification is " + myDocument.as3StrictMode);
myDocument.as3StrictMode = false;
fl.outputPanel.trace("Strict Mode value after modification is " + myDocument.as3StrictMode);
```

Siehe auch

[document.as3WarningsMode](#)

document.as3WarningsMode

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`document.as3WarningsMode`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob der ActionScript 3.0-Compiler mit aktiviertem (`true`) oder mit deaktiviertem (`false`) Warnmodus kompilieren soll. Im Warnmodus werden zusätzliche Warnungen gemeldet, die nützlich sind, um beim Aktualisieren von ActionScript 2.0-Code auf ActionScript 3.0 Inkompatibilitäten zu entdecken. Der Standardwert lautet `true`.

Beispiel

Im folgenden Beispiel wird die Compiler-Option für den Warnmodus deaktiviert.

```
var myDocument = fl.getDocumentDOM();
fl.outputPanel.trace("Warnings Mode value before modification is " +
myDocument.as3WarningsMode);
myDocument.as3WarningsMode = false;
fl.outputPanel.trace("Warnings Mode value after modification is " +
myDocument.as3WarningsMode);
```

Siehe auch

[document.as3StrictMode](#)

document.asVersion

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`document.asVersion`

Beschreibung

Eigenschaft; eine Ganzzahl, die angibt, welche ActionScript-Version im angegebenen Dokument verwendet wird. Die zulässigen Werte sind 1, 2 und 3.

Um die vorgesehene Player-Version des angegebenen Dokuments festzustellen, verwenden Sie [document.getPlayerVersion\(\)](#); diese Methode gibt einen String zurück, sodass die Verwendung mit Flash® Lite™-Playern möglich ist.

Beispiel

Im folgenden Beispiel wird die ActionScript-Version des aktuellen Dokuments auf ActionScript 2.0 gesetzt, falls sie zurzeit als ActionScript 1.0 festgelegt ist.

```
if (fl.getDocumentDOM().asVersion == 1) {
    fl.getDocumentDOM().asVersion = 2;
}
```

Siehe auch

[document.as3Dialect](#), [document.getPlayerVersion\(\)](#)

document.autoLabel

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.autoLabel`

Beschreibung

Eigenschaft; ein boolescher Wert, der dem Kontrollkästchen „Automatische Bezeichnung“ im Bedienfeld „Eingabehilfen“ entspricht. Mit dieser Eigenschaft können Sie Flash anweisen, dass Objekten auf der Bühne automatisch der mit ihnen verknüpfte Text als Bezeichnung zugewiesen werden soll.

Beispiel

Im folgenden Beispiel wird der Wert der Eigenschaft `autoLabel` abgerufen und im Bedienfeld „Ausgabe“ angezeigt:

```
var isAutoLabel = fl.getDocumentDOM().autoLabel;  
fl.trace(isAutoLabel);
```

Im folgenden Beispiel wird der Eigenschaft `autoLabel` der Wert `true` zugewiesen, d. h. Flash wird angewiesen, Objekten auf der Bühne automatisch eine Bezeichnung zuzuweisen:

```
fl.getDocumentDOM().autoLabel = true;
```

document.backgroundColor

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.backgroundColor`

Beschreibung

Eigenschaft; die Farbe des Hintergrunds. Sie wird in einem der folgenden Formate angegeben:

- als String im Format `"#RRGGBB"` oder `"#RRGGBBAA"`
- als Hexadezimalzahl im Format `"0xRRGGBB"`
- als Ganzzahl, die die dezimale Entsprechung einer hexadezimalen Zahl darstellt

Beispiel

Im folgenden Beispiel wird dem Hintergrund eine schwarze Farbe zugewiesen:

```
fl.getDocumentDOM().backgroundColor = '#000000';
```

document.breakApart()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.breakApart()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; teilt die aktuelle Auswahl.

Beispiel

Im folgenden Beispiel wird die aktuelle Auswahl geteilt:

```
fl.getDocumentDOM().breakApart();
```

document.canEditSymbol()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.canEditSymbol()
```

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; `true`, wenn das Menü und die Funktion „Symbole bearbeiten“ verwendet werden können, ansonsten `false`.

Beschreibung

Methode; gibt an, ob das Menü und die Funktion „Symbole bearbeiten“ aktiviert sind. Dies hat nichts damit zu tun, ob die Auswahl bearbeitet werden kann. Diese Methode sollte nicht verwendet werden, um festzustellen, ob `fl.getDocumentDOM().enterEditMode()` zulässig ist.

Beispiel

Im folgenden Beispiel wird der Status des Menüs und der Funktion „Symbole bearbeiten“ im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace("fl.getDocumentDOM().canEditSymbol() returns: " +  
fl.getDocumentDOM().canEditSymbol());
```

document.canRevert()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.canRevert()
```

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; `true`, wenn Sie die Methoden `document.revert()` oder `fl.revertDocument()` erfolgreich einsetzen können, ansonsten `false`.

Beschreibung

Methode; stellt fest, ob die Methode `document.revert()` oder `fl.revertDocument()` erfolgreich eingesetzt werden kann.

Beispiel

Im folgenden Beispiel wird festgestellt, ob die zuvor gespeicherte Version des aktuellen Dokuments wiederhergestellt werden kann. Falls ja, wird die zuvor gespeicherte Version von `fl.getDocumentDOM().revert()` wiederhergestellt:

```
if (fl.getDocumentDOM().canRevert()) {  
    fl.getDocumentDOM().revert();  
}
```

document.canTestMovie()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.canTestMovie()
```

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; `true`, wenn Sie die Methode `document.testMovie()` erfolgreich einsetzen können, ansonsten `false`.

Beschreibung

Methode; stellt fest, ob die Methode `document.testMovie()` erfolgreich eingesetzt werden kann.

Beispiel

Im folgenden Beispiel wird geprüft, ob `fl.getDocumentDOM().testMovie()` eingesetzt werden kann. Ist dies der Fall, wird die Methode aufgerufen.

```
if (fl.getDocumentDOM().canTestMovie()) {  
    fl.getDocumentDOM().testMovie();  
}
```

Siehe auch

`document.canTestScene()`, `document.testScene()`

document.canTestScene()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.canTestScene()`

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; `true`, wenn Sie die Methode `document.testScene()` erfolgreich einsetzen können, ansonsten `false`.

Beschreibung

Methode; stellt fest, ob die Methode `document.testScene()` erfolgreich eingesetzt werden kann.

Beispiel

Im folgenden Beispiel wird zuerst geprüft, ob „`fl.getDocumentDOM().testScene()`“ erfolgreich eingesetzt werden kann. Ist dies der Fall, wird die Methode aufgerufen.

```
if (fl.getDocumentDOM().canTestScene()) {  
    fl.getDocumentDOM().testScene();  
}
```

Siehe auch

`document.canTestMovie()`, `document.testMovie()`

document.changeFilterOrder()

Verfügbarkeit

Flash 8.

Verwendung

```
document.changeFilterOrder(oldIndex, newIndex)
```

Parameter

oldIndex Eine Ganzzahl, die die aktuelle, auf Null basierende Indexposition des Filters darstellt, den Sie in der Filterliste an eine andere Stelle verschieben möchten.

newIndex Eine Ganzzahl, die die neue Indexposition des Filters in der Liste darstellt.

Rückgabewerte

Keine.

Beschreibung

Methode; ändert den Index des Filters in der Filterliste. Alle Filter über oder unter *newIndex* werden entsprechend nach oben oder nach unten verschoben. Wenn Sie beispielsweise auf die unten genannten Filter den Befehl „`fl.getDocumentDOM().changeFilterOrder(3, 0)`“ anwenden, werden die Filter folgendermaßen neu angeordnet:

Davor	Danach
blurFilterdropShadowFilterglowFiltergradientBevelFilter	gradientBevelFilterblurFilterdropShadowFilterglowFilter

Wenn Sie dann den Befehl „`fl.getDocumentDOM().changeFilterOrder(0, 2)`“ ausgeben, werden die Filter folgendermaßen neu angeordnet:

Davor	Danach
gradientBevelFilterblurFilterdropShadowFilterglowFilter	blurFilterdropShadowFiltergradientBevelFilterglowFilter

Beispiel

Im folgenden Beispiel wird der Filter, der in der Filterliste zurzeit an der zweiten Stelle steht, an die erste Stelle verschoben:

```
fl.getDocumentDOM().changeFilterOrder(1, 0);
```

Siehe auch

[document.addFilter\(\)](#), [document.disableFilter\(\)](#), [document.enableFilter\(\)](#), [document.getFilters\(\)](#), [document.removeFilter\(\)](#), [Filter-Objekt](#)

document.clipCopy()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.clipCopy()`

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; kopiert die aktuelle Auswahl aus dem Dokument in die Zwischenablage.

Um einen String in die Zwischenablage zu kopieren, verwenden Sie `fl.clipCopyString()`.

Beispiel

Im folgenden Beispiel wird die aktuelle Auswahl aus dem Dokument in die Zwischenablage kopiert:

```
fl.getDocumentDOM().clipCopy();
```

Siehe auch

`document.clipCut()`, `document.clipPaste()`

document.clipCut()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.clipCut()`

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; schneidet die aktuelle Auswahl aus dem Dokument aus und speichert sie in der Zwischenablage.

Beispiel

Im folgenden Beispiel wird die aktuelle Auswahl aus dem Dokument ausgeschnitten und in der Zwischenablage gespeichert:

```
fl.getDocumentDOM().clipCut();
```

Siehe auch

`document.clipCopy()`, `document.clipPaste()`, `fl.clipCopyString()`

document.clipPaste()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.clipPaste([bInPlace])
```

Parameter

bInPlace Ein boolescher Wert; bei dem Wert `true` wird der Inhalt der Zwischenablage an der ausgewählten Position eingefügt. Der Standardwert lautet `false`; dabei wird der Inhalt der Zwischenablage in der Mitte des Dokuments eingefügt. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt den Inhalt der Zwischenablage in das Dokument ein.

Beispiel

In den folgenden Beispielen wird der Inhalt der Zwischenablage in der Dokumentmitte eingefügt:

```
fl.getDocumentDOM().clipPaste();
```

Im folgenden Beispiel wird der Inhalt der Zwischenablage an der ausgewählten Position eingefügt:

```
fl.getDocumentDOM().clipPaste(true);
```

Siehe auch

[document.clipCopy\(\)](#), [document.clipCut\(\)](#), [fl.clipCopyString\(\)](#)

document.close()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.close([bPromptToSaveChanges])
```

Parameter

bPromptToSaveChanges Ein boolescher Wert; bei dem Wert `true`, wird der Benutzer über ein Dialogfeld zum Speichern aufgefordert, sofern das Dokument ungespeicherte Änderungen enthält. Wenn *bPromptToSaveChanges* den Wert `false` hat, wird der Benutzer nicht zum Speichern geänderter Dokumente aufgefordert. Der Standardwert lautet `true`. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; schließt das angegebene Dokument.

Beispiel

Im folgenden Beispiel wird das aktuelle Dokument geschlossen und der Benutzer über ein Dialogfeld zum Speichern der Änderungen aufgefordert:

```
fl.getDocumentDOM().close();
```

Im folgenden Beispiel wird das aktuelle Dokument geschlossen, ohne vorher die Änderungen zu speichern:

```
fl.getDocumentDOM().close(false);
```

document.convertLinesToFills()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.convertLinesToFills()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; konvertiert die Linien der ausgewählten Objekte in Füllungen.

Beispiel

Im folgenden Beispiel werden die gegenwärtig ausgewählten Linien in Füllungen konvertiert:

```
fl.getDocumentDOM().convertLinesToFills();
```

document.convertToSymbol()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.convertToSymbol(type, name, registrationPoint)
```

Parameter

type Ein String, der den zu erstellenden Symboltyp angibt. Die zulässigen Werte lauten "movie clip", "button" und "graphic".

name Ein String, der den Namen des neuen Symbols angibt. Dieser Name muss eindeutig sein. Sie können einen leeren String übergeben, wenn Sie wünschen, dass ein eindeutiger Symbolname von der Methode automatisch erstellt wird.

registration point Gibt den Punkt an, an dem sich die Position „0.0“ des Symbols befindet. Zulässige Werte sind: "top left", "top center", "top right", "center left", "center", "center right", "bottom left", "bottom center" und "bottom right".

Rückgabewerte

Ein Objekt für das neu erstellte Symbol bzw. `null`, wenn das Symbol nicht erstellt werden kann.

Beschreibung

Methode; konvertiert die ausgewählten Bühnenelemente in ein neues Symbol. Wie Sie für ein Symbol Verknüpfungen und gemeinsame Bestandseigenschaften definieren, erfahren Sie unter [Item-Objekt](#).

Beispiel

In den folgenden Beispielen werden die folgenden Symbole erstellt: ein Movieclipsymbol mit einem angegebenen Namen, ein Schaltflächensymbol mit einem angegebenen Namen und ein Movieclipsymbol mit einem Standardnamen:

```
newMc = fl.getDocumentDOM().convertToSymbol("movie clip", "mcSymbolName", "top left");
newButton = fl.getDocumentDOM().convertToSymbol("button", "btnSymbolName", "bottom right");
newClipWithDefaultName = fl.getDocumentDOM().convertToSymbol("movie clip", "", "top left");
```

document.crop()

Verfügbarkeit

Flash 8.

Verwendung

```
document.crop()
```

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; `true`, wenn erfolgreich, ansonsten `false`.

Beschreibung

Methode; verwendet das zuoberst liegende ausgewählte Zeichnungsobjekt, um alle darunter liegenden ausgewählten Zeichnungsobjekte zuzuschneiden. Die Methode gibt `false` zurück, wenn keine Zeichnungsobjekte ausgewählt sind oder wenn nicht alle ausgewählten Elemente Zeichnungsobjekte sind.

Beispiel

Im folgenden Beispiel werden die gegenwärtig ausgewählten Objekte zugeschnitten:

```
fl.getDocumentDOM().crop();
```

Siehe auch

`document.deleteEnvelope()`, `document.intersect()`, `document.punch()`, `document.union()`,
`shape.isDrawingObject`

document.currentPublishProfile

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.currentPublishProfile`

Beschreibung

Eigenschaft; ein String, der den Namen des aktiven Veröffentlichungsprofils für das angegebene Dokument angibt.

Beispiel

Im folgenden Beispiel wird ein neues Veröffentlichungsprofil mit dem Standardnamen hinzugefügt und dann der Name des Profils im Bedienfeld „Ausgabe“ angezeigt:

```
fl.getDocumentDOM().addNewPublishProfile();  
fl.outputPanel.trace(fl.getDocumentDOM().currentPublishProfile);
```

Im folgenden Beispiel wird anstelle des ausgewählten Veröffentlichungsprofils das Profil "Default" ausgewählt:

```
fl.getDocumentDOM().currentPublishProfile = "Default";
```

document.currentTimeline

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.currentTimeline`

Beschreibung

Eigenschaft; eine Ganzzahl, die den Index der aktiven Zeitleiste angibt. Sie können die aktive Zeitleiste einstellen, indem Sie den Wert dieser Eigenschaft ändern; die Wirkung ist mit dem Aufruf von `document.editScene()` vergleichbar. Der einzige Unterschied besteht darin, dass keine Fehlermeldung angezeigt wird, wenn der Index der Zeitleiste ungültig ist (d. h. die Eigenschaft wird nicht eingestellt und dies führt zu einem nicht gemeldeten Fehler).

Beispiel

Im folgenden Beispiel wird der Index der aktuellen Zeitleiste angezeigt:

```
var myCurrentTL = fl.getDocumentDOM().currentTimeline;  
fl.trace("The index of the current timeline is: " + myCurrentTL);
```

Im folgenden Beispiel löst eine Szene mit dem Namen "myScene" die Hauptzeitleiste als aktive Zeitleiste ab:

```
var i = 0;
var curTimelines = fl.getDocumentDOM().timelines;
while(i < fl.getDocumentDOM().timelines.length){
    if(curTimelines[i].name == "myScene"){
        fl.getDocumentDOM().currentTimeline = i;
    }
    ++i;
}
```

Siehe auch

[document.getTimeline\(\)](#)

document.debugMovie()

Verfügbarkeit

Flash Professional CS5.

Verwendung

```
document.DebugMovie([Boolean abortIfErrorsExist])
```

Beschreibung

Methode; ruft den Befehl „Debuggen“ für einen Film im Dokument auf.

Parameter

abortIfErrorsExist Boolescher Wert; der Standardwert lautet „false“. Wenn dieser Wert auf „true“ eingestellt ist, wird die Debug-Sitzung nicht gestartet und das Fenster der .swf-Datei wird nicht geöffnet, wenn Kompilierfehler vorliegen. Durch Compiler-Warnmeldungen wird der Befehl nicht abgebrochen.

Beispiel

Im folgenden Beispiel wird das aktuelle Dokument im Debug-Modus geöffnet, doch der Vorgang wird abgebrochen, wenn Kompilierfehler vorliegen:

```
fl.getDocumentDOM().debugMovie(1);
```

document.deleteEnvelope()

Verfügbarkeit

Flash 8.

Verwendung

```
document.deleteEnvelope()
```

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; `true`, wenn erfolgreich, ansonsten `false`.

Beschreibung

Methode; löscht die Hülle (eine Begrenzungsbox, die ein oder mehrere Objekte enthält) aus den ausgewählten Objekten.

Beispiel

Im folgenden Beispiel wird die Hülle aus den ausgewählten Objekten gelöscht:

```
fl.getDocumentDOM().deleteEnvelope();
```

Siehe auch

[document.crop\(\)](#), [document.intersect\(\)](#), [document.punch\(\)](#), [document.union\(\)](#), [shape.isDrawingObject](#)

document.deletePublishProfile()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.deletePublishProfile()
```

Parameter

Keine.

Rückgabewerte

Eine Ganzzahl, die den Index des neuen aktuellen Profils angibt. Wenn kein neues Profil verfügbar ist, bleibt das aktuelle Profil unverändert und die Methode gibt den Index des aktuellen Profils zurück.

Beschreibung

Methode; löscht das gegenwärtig aktive Profil, sofern es mehrere Profile gibt. Es muss mindestens ein Profil übrig bleiben.

Beispiel

Im folgenden Beispiel wird das gegenwärtig aktive Profil gelöscht, sofern es mehrere Profile gibt. Anschließend wird der Index des neuen aktiven Profils angezeigt:

```
alert(fl.getDocumentDOM().deletePublishProfile());
```

Siehe auch

[document.addNewPublishProfile\(\)](#)

document.deleteScene()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.deleteScene()
```

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die Szene erfolgreich gelöscht wurde, ansonsten `false`.

Beschreibung

Methode; löscht die aktuelle Szene ([Timeline-Objekt](#)). Falls die gelöschte Szene nicht die letzte Szene war, wird die nächste Szene als das aktuelle Timeline-Objekt festgelegt. Wenn die gelöschte Szene die letzte Szene war, wird das erste Objekt als aktuelles Timeline-Objekt festgelegt. Ist nur ein Timeline-Objekt (Szene) vorhanden, lautet der Rückgabewert `false`.

Beispiel

Im folgenden Beispiel wird im aktuellen Dokument, das drei Szenen (`Scene0`, `Scene1` und `Scene2`) enthält, `Scene2` als aktuelle Szene ausgewählt und dann gelöscht:

```
fl.getDocumentDOM().editScene(2);  
var success = fl.getDocumentDOM().deleteScene();
```

document.deleteSelection()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.deleteSelection()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; löscht die aktuelle Auswahl auf der Bühne. Zeigt eine Fehlermeldung an, wenn es keine Auswahl gibt.

Beispiel

Im folgenden Beispiel wird die aktuelle Auswahl aus dem Dokument gelöscht:

```
fl.getDocumentDOM().deleteSelection();
```

document.description

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.description
```

Beschreibung

Eigenschaft; ein String, der dem Feld „Beschreibung“ im Bedienfeld „Eingabehilfen“ entspricht. Die Beschreibung wird von Bildschirmleseprogrammen akustisch wiedergegeben.

Beispiel

Im folgenden Beispiel wird die Beschreibung des Dokuments festgelegt:

```
fl.getDocumentDOM().description= "This is the main movie";
```

Im folgenden Beispiel wird die Beschreibung des Dokuments abgerufen und im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace(fl.getDocumentDOM().description);
```

document.disableAllFilters()

Verfügbarkeit

Flash 8.

Verwendung

```
document.disableAllFilters()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; deaktiviert alle Filter, die für die ausgewählten Objekte gelten.

Beispiel

Im folgenden Beispiel werden alle Filter deaktiviert, die für die ausgewählten Objekte gelten:

```
fl.getDocumentDOM().disableAllFilters();
```

Siehe auch

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableFilter\(\)](#),
[document.disableOtherFilters\(\)](#), [document.enableAllFilters\(\)](#), [document.getFilters\(\)](#),
[document.removeAllFilters\(\)](#), [Filter-Objekt](#)

document.disableFilter()

Verfügbarkeit

Flash 8.

Verwendung

```
document.disableFilter(filterIndex)
```

Parameter

filterIndex Eine Ganzzahl, die den auf Null basierenden Index des Filters in der Filterliste darstellt.

Rückgabewerte

Keine.

Beschreibung

Methode; deaktiviert den in der Filterliste angegebenen Filter.

Beispiel

Im folgenden Beispiel werden der erste und der dritte Filter (mit den Indexwerten 0 und 2) in der Filterliste für die ausgewählten Objekte deaktiviert:

```
fl.getDocumentDOM().disableFilter(0);  
fl.getDocumentDOM().disableFilter(2);
```

Siehe auch

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableAllFilters\(\)](#),
[document.disableOtherFilters\(\)](#), [document.enableFilter\(\)](#), [document.getFilters\(\)](#),
[document.removeFilter\(\)](#), [Filter-Objekt](#)

document.disableOtherFilters()

Verfügbarkeit

Flash 8.

Verwendung

```
document.disableOtherFilters(enabledFilterIndex)
```

Parameter

enabledFilterIndex Eine Ganzzahl, die den auf Null basierenden Index des Filters angibt, der nach der Aktivierung anderer Filter aktiviert bleiben soll.

Rückgabewerte

Keine.

Beschreibung

Methode; deaktiviert alle Filter mit Ausnahme des Filters, der sich an der angegebenen Stelle in der Filterliste befindet.

Beispiel

Im folgenden Beispiel werden alle in der Liste aufgeführten Filter mit Ausnahme des zweiten Filters (Indexwert 1) deaktiviert:

```
fl.getDocumentDom().disableOtherFilters(1);
```

Siehe auch

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableAllFilters\(\)](#),
[document.disableFilter\(\)](#), [document.enableFilter\(\)](#), [document.getFilters\(\)](#),
[document.removeFilter\(\)](#), [Filter-Objekt](#)

document.distribute()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.distribute(distributemode [, bUseDocumentBounds])
```

Parameter

distributemode Ein String, der angibt, wie die ausgewählten Objekte verteilt werden sollen. Die zulässigen Werte lauten "left edge", "horizontal center", "right edge", "top edge", "vertical center" und "bottom edge".

bUseDocumentBounds Ein boolescher Wert; bei dem Wert `true` werden die ausgewählten Objekte anhand der Dokumentgrenzen verteilt. Andernfalls orientiert die Methode sich an den Grenzen der ausgewählten Objekte. Der Standardwert lautet `false`.

Rückgabewerte

Keine.

Beschreibung

Methode; verteilt die Auswahl.

Beispiel

Im folgenden Beispiel werden die ausgewählten Objekte anhand ihrer Oberkante verteilt:

```
fl.getDocumentDOM().distribute("top edge");
```

Im folgenden Beispiel werden die ausgewählten Objekte anhand ihrer Oberkante verteilt und der Parameter `bUseDocumentBounds` wird ausdrücklich festgelegt:

```
fl.getDocumentDOM().distribute("top edge", false);
```

Im folgenden Beispiel werden die ausgewählten Objekte an ihren Oberkanten und anhand der Dokumentgrenzen verteilt:

```
fl.getDocumentDOM().distribute("top edge", true);
```

Siehe auch

`document.getAlignToDocument()`, `document.setAlignToDocument()`

document.distributeToLayers()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.distributeToLayers()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; verteilt die aktuelle Auswahl auf Ebenen; entspricht dem Befehl „Auf Ebenen verteilen“. Wenn es keine Auswahl gibt, zeigt diese Methode einen Fehler an.

Beispiel

Im folgenden Beispiel wird die aktuelle Auswahl auf Ebenen verteilt:

```
fl.getDocumentDOM().distributeToLayers();
```

document.docClass

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
document.docClass
```

Beschreibung

Eigenschaft; ein String, der angibt, welche ActionScript 3.0-Klasse der höchsten Ebene dem Dokument zugeordnet ist. Wenn das Dokument nicht für die Verwendung von ActionScript 3.0 konfiguriert ist, wird diese Eigenschaft ignoriert.

Beispiel

Im folgenden Beispiel wird festgelegt, dass es sich bei der ActionScript 3.0-Klasse, die dem Dokument zugeordnet ist, um „com.mycompany.ManagerClass“ handelt. Dies ist in „com/mycompany/ManagerClass.as“ definiert:

```
var myDocument = fl.getDocumentDOM();  
// set the property  
myDocument.docClass = "com.mycompany.ManagerClass";  
// get the property  
fl.outputPanel.trace("document.docClass has been set to " + myDocument.docClass);
```

Siehe auch

[item.linkageBaseClass](#)

document.documentHasData()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.documentHasData(name)
```

Parameter

name Ein String, der den Namen der zu überprüfenden Daten angibt.

Rückgabewerte

Ein boolescher Wert; `true`, wenn das Dokument permanente Daten enthält, ansonsten `false`.

Beschreibung

Methode; stellt fest, ob das Dokument permanente Daten mit dem angegebenen Namen enthält.

Beispiel

Im folgenden Beispiel wird geprüft, ob das Dokument permanente Daten mit dem Namen "myData" enthält:

```
var hasData = fl.getDocumentDOM().documentHasData("myData");
```

Siehe auch

[document.addDataToDocument\(\)](#), [document.getDataFromDocument\(\)](#),
[document.removeDataFromDocument\(\)](#)

document.duplicatePublishProfile()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.duplicatePublishProfile([profileName])
```

Parameter

profileName Ein String, der den eindeutigen Namen des duplizierten Profils angibt. Wenn Sie keinen Namen angeben, wird der Standardname verwendet. Dieser Parameter ist optional.

Rückgabewerte

Eine Ganzzahl, die den Index des neuen Profils in der Profilliste angibt. Gibt -1 zurück, wenn das Profil nicht dupliziert werden kann.

Beschreibung

Methode; dupliziert das gegenwärtig aktive Profil und weist dem Duplikat den Fokus zu.

Beispiel

Im folgenden Beispiel wird das gegenwärtig aktive Profil dupliziert und der Index des neuen Profils im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace(fl.getDocumentDOM().duplicatePublishProfile("dup profile"));
```

document.duplicateScene()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.duplicateScene()
```

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; *true*, wenn die Szene erfolgreich dupliziert wurde, ansonsten *false*.

Beschreibung

Methode; erstellt von der gegenwärtig ausgewählten Szene eine Kopie, weist der neuen Szene einen eindeutigen Namen zu und wählt die neue Szene als aktuelle Szene aus.

Beispiel

Im folgenden Beispiel wird die zweite Szene in dem aktuellen Dokument dupliziert:

```
fl.getDocumentDOM().editScene(1); //Set the middle scene to current scene.  
var success = fl.getDocumentDOM().duplicateScene();
```

document.duplicateSelection()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.duplicateSelection()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; dupliziert die Auswahl auf der Bühne.

Beispiel

Im folgenden Beispiel wird die aktuelle Auswahl dupliziert. Eine ähnliche Wirkung erzielen Sie, wenn Sie bei gedrückter Alt-Taste klicken und dann ein Element ziehen:

```
fl.getDocumentDOM().duplicateSelection();
```

document.editScene()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.editScene(index)
```

Parameter

index Eine auf Null basierende Ganzzahl, die angibt, welche Szene bearbeitet werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; wählt die angegebene Szene als aktuelle Szene aus, damit sie bearbeitet werden kann.

Beispiel

Im folgenden Beispiel wird im aktuellen Dokument, das drei Szenen (Scene0, Scene1 und Scene2) enthält, Scene2 als aktuelle Szene festgelegt und dann gelöscht:

```
fl.getDocumentDOM().editScene(2);  
fl.getDocumentDOM().deleteScene();
```

document.enableAllFilters()

Verfügbarkeit

Flash 8.

Verwendung

```
document.enableAllFilters()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; aktiviert alle Filter in der Filterliste für die ausgewählten Objekte.

Beispiel

Im folgenden Beispiel werden alle Filter in der Filterliste für die ausgewählten Objekte aktiviert:

```
fl.getDocumentDOM().enableAllFilters();
```

Siehe auch

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableAllFilters\(\)](#),
[document.enableFilter\(\)](#), [document.getFilters\(\)](#), [document.removeAllFilters\(\)](#), [Filter-Objekt](#)

document.enableFilter()

Verfügbarkeit

Flash 8.

Verwendung

```
document.enableFilter(filterIndex)
```

Parameter

filterIndex Eine Ganzzahl, die den auf Null basierenden Index des Filters in der Filterliste angibt, der aktiviert werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; aktiviert den angegebenen Filter für die ausgewählten Objekte.

Beispiel

Im folgenden Beispiel wird der zweite Filter für die ausgewählten Objekte aktiviert:

```
fl.getDocumentDOM().enableFilter(1);
```

Siehe auch

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableFilter\(\)](#),
[document.enableAllFilters\(\)](#), [document.getFilters\(\)](#), [document.removeFilter\(\)](#), [Filter-Objekt](#)

document.enterEditMode()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.enterEditMode([editMode])
```

Parameter

editMode Ein String, der den Bearbeitungsmodus angibt. Die zulässigen Werte lauten "inPlace" und "newWindow". Wenn kein Parameter angegeben wurde, wird automatisch der Symbolbearbeitungsmodus aktiviert. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; versetzt das Authoring-Tool in den von dem Parameter angegebenen Bearbeitungsmodus. Wenn kein Parameter angegeben wurde, wird automatisch der Symbolbearbeitungsmodus aktiviert. Das gleiche Ergebnis erzielen Sie, wenn Sie mit der rechten Maustaste auf das Symbol klicken und dann im Kontextmenü „Bearbeiten“ auswählen.

Beispiel

Im folgenden Beispiel wird für das ausgewählte Symbol der Modus „An Position bearbeiten“ aktiviert:

```
fl.getDocumentDOM().enterEditMode('inPlace');
```

Im folgenden Beispiel wird für das ausgewählte Symbol der Modus „In neuem Fenster bearbeiten“ aktiviert:

```
fl.getDocumentDOM().enterEditMode('newWindow');
```

Siehe auch

[document.exitEditMode\(\)](#)

document.exitEditMode()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.exitEditMode()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; beendet den Symbolbearbeitungsmodus und verlagert den Fokus vom Bearbeitungsmodus aus auf die nächsthöhere Ebene. Wenn Sie beispielsweise ein Symbol innerhalb eines anderen Symbols bearbeiten, führt diese Methode Sie von dem Symbol, das Sie gerade bearbeiten, auf die nächsthöhere Ebene, d. h. zum übergeordneten Symbol.

Beispiel

Die folgende Methode beendet den Symbolbearbeitungsmodus:

```
fl.getDocumentDOM().exitEditMode();
```

Siehe auch

[document.enterEditMode\(\)](#)

document.exportPNG()

Verfügbarkeit

Flash 8.

Verwendung

```
document.exportPNG([fileURI [, bCurrentPNGSettings [, bCurrentFrame]])
```

Parameter

fileURI Ein String im Format file:/// URI, der den Dateinamen der exportierten Datei angibt. Wurde *fileURI* leer gelassen oder nicht angegeben, wird in Flash das Dialogfeld „Film exportieren“ angezeigt.

bCurrentPNGSettings Ein boolescher Wert, der angibt, ob die aktuellen PNG-Veröffentlichungseinstellungen verwendet werden (*true*) oder das Dialogfeld „PNG exportieren“ (*false*) angezeigt werden soll. Dieser Parameter ist optional. Der Standardwert lautet *false*.

bCurrentFrame Ein boolescher Wert, der angibt, ob nur das aktuelle Bild exportiert wird (*true*) oder ob alle Bilder exportiert werden, wobei für jedes Bild eine eigene PNG-Datei erstellt wird (*false*). Dieser Parameter ist optional. Der Standardwert lautet *false*.

Rückgabewerte

Der boolesche Wert *true*, wenn der Filter erfolgreich verschoben wurde, ansonsten *false*.

Beschreibung

Methode; exportiert das Dokument als eine oder mehrere PNG-Dateien. Wenn *fileURI* angegeben wurde und die betreffende Datei bereits vorhanden ist, so wird sie ohne Warnung überschrieben.

Beispiel

Mit dem folgenden Beispiel wird das aktuelle Bild des aktuellen Dokuments in die Datei „myFile.png“ mit den aktuellen PNG-Veröffentlichungseinstellungen exportiert:

```
fl.getDocumentDOM().exportPNG("file:///C:/myProject/myFile.png", true, true);
```

document.exportPublishProfile()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.exportPublishProfile(fileURI)
```

Parameter

fileURI Ein String im Format file:/// URI, der den Pfad der XML-Datei angibt, in die das Profil exportiert wird.

Rückgabewerte

Keine.

Beschreibung

Methode; exportiert das gegenwärtig aktive Profil in eine XML-Datei.

Beispiel

Im folgenden Beispiel wird das gegenwärtig aktive Profil in die Datei „profile.xml“ exportiert, die sich auf Laufwerk C im Ordner /Dokumente und Einstellungen/Benutzername/Desktop befindet:

```
fl.getDocumentDOM().exportPublishProfile('file:///C:/Documents and  
Settings/username/Desktop/profile.xml');
```

Siehe auch

[document.exportPublishProfileString\(\)](#), [document.importPublishProfile\(\)](#)

document.exportPublishProfileString()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
document.exportPublishProfileString( [profileName] )
```

Parameter

profileName Ein String, der den Namen des Profils angibt, das in einen XML-String exportiert werden soll. Dieser Parameter ist optional.

Rückgabewerte

Ein XML-String.

Beschreibung

Methode; gibt einen String zurück, der das genannte Profil im XML-Format angibt. Wenn Sie für *profileName* keinen Wert übergeben, wird das aktuelle Profil exportiert.

Beispiel

Im folgenden Beispiel wird ein XML-String, der das aktuelle Profil darstellt, in einer Variablen namens `profileXML` gespeichert und anschließend wird es im Bedienfeld „Ausgabe“ angezeigt:

```
var profileXML=fl.getDocumentDOM().exportPublishProfileString();  
fl.trace(profileXML);
```

Siehe auch

`document.exportPublishProfile()`, `document.importPublishProfileString()`

document.exportSWF()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.exportSWF([fileURI [, bCurrentSettings]])
```

Parameter

fileURI Ein String im Format `file:/// URI`, der den Namen der exportierten Datei angibt. Wenn *fileURI* leer gelassen oder nicht angegeben wurde, wird in Flash das Dialogfeld „Film exportieren“ angezeigt. Dieser Parameter ist optional.

bCurrentSettings Ein boolescher Wert; bei dem Wert `true` verwendet Flash die aktuellen SWF-Veröffentlichungseinstellungen. Andernfalls wird das Dialogfeld „Flash Player exportieren“ angezeigt. Der Standardwert lautet `false`. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; exportiert das Dokument in das Flash SWC-Format.

Beispiel

Im folgenden Beispiel wird das Dokument anhand der aktuellen Veröffentlichungseinstellungen exportiert und am angegebenen Speicherort abgelegt:

```
fl.getDocumentDOM().exportSWF("file:///C:/Documents and  
Settings/joe_user/Desktop/qwerty.swf");
```

Im folgenden Beispiel werden die Dialogfelder „Film exportieren“ und „Flash Player exportieren“ angezeigt und dann wird das Dokument anhand der festgelegten Einstellungen exportiert:

```
fl.getDocumentDOM().exportSWF("", true);
```

Im folgenden Beispiel wird das Dialogfeld „Film exportieren“ angezeigt und dann das Dokument anhand der festgelegten Einstellungen exportiert:

```
fl.getDocumentDOM().exportSWF();
```

document.externalLibraryPath

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`document.externalLibraryPath`

Beschreibung

Eigenschaft; ein String, der eine Liste von Elementen enthält, die sich im Pfad der externen ActionScript 3.0-Bibliothek des Dokuments befinden, und dieser Pfad wiederum gibt den Speicherort von SWC-Dateien an, die zur Laufzeit als gemeinsame Bibliotheken verwendet werden. Elemente innerhalb des Strings sind durch Semikolons getrennt. Zur Angabe der Elemente wählen Sie im Authoring-Tool „Datei“ > „Einstellungen für Veröffentlichungen“ und wählen dann auf der Registerkarte „Flash“ die Option „ActionScript 3.0-Einstellungen“.

Beispiel

Im folgenden Beispiel wird der Pfad der externen Bibliothek als „.“ und „../mySWCLibrary“ festgelegt:

```
var myDocument = fl.getDocumentDOM();  
myDocument.externalLibraryPath = ".;../mySWCLibrary";  
fl.trace(myDocument.externalLibraryPath);
```

Siehe auch

[document.libraryPath](#), [document.sourcePath](#), [fl.externalLibraryPath](#)

document.forceSimple

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.forceSimple`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob die Child-Objekte des angegebenen Objekts mit Eingabehilfen versehen sind. Dies entspricht der umgekehrten Logik der Einstellung „Child-Objekte mit Eingabehilfen versehen“ im Bedienfeld „Eingabehilfen“. Wenn der Wert für `forceSimple` `true` lautet, entspricht dies einer Deaktivierung der Option „Child-Objekte mit Eingabehilfen versehen“. Wenn der Wert für `forceSimple` `false` lautet, entspricht dies einer Aktivierung der Option „Child-Objekte mit Eingabehilfen versehen“.

Beispiel

Im folgenden Beispiel wird der Variablen `areChildrenAccessible` der Wert der Eigenschaft `forceSimple` zugewiesen: Der Wert `false` bedeutet, dass auf die untergeordneten Objekte zugegriffen werden kann.

```
var areChildrenAccessible = fl.getDocumentDOM().forceSimple;
```

Im folgenden Beispiel wird die Eigenschaft `forceSimple` so eingestellt, dass Child-Dokumente mit Eingabehilfen versehen sind:

```
fl.getDocumentDOM().forceSimple = false;
```

document.frameRate

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.frameRate
```

Beschreibung

Eigenschaft; ein Gleitkommawert, der angibt, wie viele Bilder beim Abspielen der SWF-Datei pro Sekunde angezeigt werden; der Standardwert lautet 12. Das Einstellen dieser Eigenschaft entspricht dem Einstellen der Bildrate im Dialogfeld „Dokumenteigenschaften“ („Modifizieren“ > „Dokument“) der FLA-Datei.

Beispiel

Im folgenden Beispiel wird die Bildrate auf 25,5 Bilder pro Sekunde eingestellt:

```
fl.getDocumentDOM().frameRate = 25.5;
```

document.getAlignToDocument()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.getAlignToDocument()
```

Parameter

Keine.

Rückgabewerte

Eine boolescher Wert; `true`, wenn die Objekte gemäß Voreinstellung an der Bühne ausgerichtet werden, ansonsten `false`.

Beschreibung

Methode; entspricht dem Abrufen des Werts der Schaltfläche „An Bühne“ im Bedienfeld „Ausrichten“. Ruft die Voreinstellung ab, die für die Methoden `document.align()`, `document.distribute()`, `document.match()` und `document.space()` im Dokument verwendet werden kann.

Beispiel

Im folgenden Beispiel wird der Wert der Schaltfläche „An Bühne“ im Bedienfeld „Ausrichten“ abgerufen. Lautet der zurückgegebene Wert `true`, ist die Schaltfläche „An Bühne“ aktiv, ansonsten ist sie inaktiv.

```
var isAlignToDoc = fl.getDocumentDOM().getAlignToDocument();  
fl.getDocumentDOM().align("left", isAlignToDoc);
```

Siehe auch

`document.setAlignToDocument()`

document.getBlendMode()

Verfügbarkeit

Flash 8.

Verwendung

`document.getBlendMode()`

Parameter

Keine.

Rückgabewerte

Ein String, der den Mischmodus für die ausgewählten Objekte angibt. Wenn mehrere Objekte mit unterschiedlichen Mischmodi ausgewählt sind, gibt der String den Mischmodus des zuoberst liegenden Objekts an.

***Hinweis:** Der Rückgabewert ist unvorhersehbar, wenn die Auswahl Objekte enthält, die keine Mischmodi unterstützen bzw. den Mischmoduswert "normal" aufweisen.*

Beschreibung

Methode; gibt einen String zurück, der den Mischmodus für die ausgewählten Objekte angibt.

Beispiel

Im folgenden Beispiel wird der Name des Mischmodus im Ausgabefenster angezeigt:

```
fl.trace(fl.getDocumentDom().getBlendMode());
```

document.getCustomFill()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.getCustomFill([objectToFill])`

Parameter

objectToFill Ein String, der angibt, wo sich das Fill-Objekt befindet. Folgende Werte sind gültig:

- "toolbar" gibt das Fill-Objekt des Bedienfelds „Werkzeuge“ und des Eigenschafteninspektors zurück.
- "selection" gibt das Fill-Objekt der Auswahl zurück.

Wenn dieser Parameter fehlt, wird der Standardwert "selection" verwendet. Wenn keine Auswahl vorhanden ist, gibt die Methode den Wert undefined zurück. Dieser Parameter ist optional.

Rückgabewerte

Falls erfolgreich, das [Fill-Objekt](#), das vom Parameter *objectToFill* angegeben wurde; andernfalls *undefined*.

Beschreibung

Methode; ruft das Fill-Objekt der ausgewählten Form oder, falls angegeben, des Bedienfelds „Werkzeuge“ und des Eigenschafteninspektors ab.

Beispiel

Im folgenden Beispiel wird das Fill-Objekt der Auswahl abgerufen und dann die Farbe der Auswahl in Weiß geändert:

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.color = 'FFFFFF';  
fill.style = "solid";  
fl.getDocumentDOM().setCustomFill(fill);
```

Im folgenden Beispiel wird das Fill-Objekt des Bedienfelds „Werkzeuge“ und des Eigenschafteninspektors zurückgegeben und dann das Farbfeld in einen linearen Farbverlauf geändert:

```
var fill = fl.getDocumentDOM().getCustomFill("toolbar");  
fill.style = "linearGradient";  
fill.colorArray = [ 0x00ff00, 0xff0000, 0x0000ff ];  
fill.posArray = [0, 100, 200];  
fl.getDocumentDOM().setCustomFill( fill );
```

Siehe auch

[document.setCustomFill\(\)](#)

document.getCustomStroke()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.getCustomStroke([locationOfStroke])
```

Parameter

locationOfStroke Ein String, der angibt, wo sich das Stroke-Objekt befindet. Folgende Werte sind gültig:

- "toolbar" gibt das Stroke-Objekt des Bedienfelds „Werkzeuge“ und des Eigenschafteninspektors zurück.
- "selection" gibt das Stroke-Objekt der Auswahl zurück.

Wenn dieser Parameter fehlt, wird der Standardwert "selection" verwendet. Wenn es keine Auswahl gibt, wird der Wert *undefined* zurückgegeben. Dieser Parameter ist optional.

Rückgabewerte

Falls erfolgreich, das [Stroke-Objekt](#), das vom Parameter *locationOfStroke* angegeben wurde; andernfalls *undefined*.

Beschreibung

Gibt das Stroke-Objekt der ausgewählten Form oder, falls angegeben, des Bedienfelds „Werkzeuge“ und des Eigenschafteninspektors zurück.

Beispiel

Im folgenden Beispiel werden die aktuellen Stricheinstellungen der Auswahl zurückgegeben und anschließend wird die Strichstärke in 2 geändert:

```
var stroke = fl.getDocumentDOM().getCustomStroke("selection");
stroke.thickness = 2;
fl.getDocumentDOM().setCustomStroke(stroke);
```

Im folgenden Beispiel werden die aktuellen Stricheinstellungen des Bedienfelds „Werkzeuge“ und des Eigenschafteninspektors zurückgegeben und anschließend wird die Strichfarbe in Rot geändert:

```
var stroke = fl.getDocumentDOM().getCustomStroke("toolbar");
stroke.color = "#FF0000";
fl.getDocumentDOM().setCustomStroke(stroke);
```

Siehe auch

[document.setCustomStroke\(\)](#)

document.getDataFromDocument()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.getDataFromDocument (name)
```

Parameter

name Ein String, der den Namen der zurückzugebenden Daten angibt.

Rückgabewerte

Die angegebenen Daten.

Beschreibung

Methode; ruft den Wert der angegebenen Daten ab. Der zurückgegebene Typ hängt von dem Typ der gespeicherten Daten ab.

Beispiel

Im folgenden Beispiel wird dem aktuellen Dokument der Ganzzahlenwert 12 hinzugefügt, der dann mit dieser Methode im Bedienfeld „Ausgabe“ angezeigt wird:

```
fl.getDocumentDOM().addDataToDocument("myData", "integer", 12);
fl.trace(fl.getDocumentDOM().getDataFromDocument("myData"));
```

Siehe auch

[document.addDataToDocument\(\)](#), [document.documentHasData\(\)](#), [document.removeDataFromDocument\(\)](#)

document.getElementProperty()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.getElementProperty(propertyName)
```

Parameter

propertyName Ein String, der den Namen der Element-Eigenschaft angibt, für die der Wert abgerufen werden soll.

Rückgabewerte

Der Wert der angegebenen Eigenschaft. Gibt `null` zurück, wenn die Eigenschaft nicht ermittelt werden kann. Dies ist beispielsweise der Fall, wenn mehrere Elemente mit unterschiedlichen Eigenschaftswerten ausgewählt sind. Gibt `undefined` zurück, wenn die Eigenschaft keine gültige Eigenschaft des ausgewählten Elements ist.

Beschreibung

Methode; ruft die angegebene Element-Eigenschaft der aktuellen Auswahl ab. Eine Liste der zulässigen Werte finden Sie in der Übersicht über die Eigenschaften für das [Element-Objekt](#).

Beispiel

Im folgenden Beispiel wird der Wert `name` der Element-Eigenschaft für die aktuelle Auswahl abgerufen:

```
// elementName = the instance name of the selected object.  
var elementName = fl.getDocumentDOM().getElementProperty("name");
```

Siehe auch

[document.setElementProperty\(\)](#)

document.getElementTextAttr()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.getElementTextAttr(attrName [, startIndex [, endIndex]])
```

Parameter

attrName Ein String mit dem Namen der zurückzugebenden `TextAttrs`-Eigenschaft. Eine Liste der Eigenschaftsnamen und erwarteten Werte finden Sie in der Übersicht über die Eigenschaften für das [TextAttrs-Objekt](#).

startIndex Eine Ganzzahl, die die Indexposition des ersten Zeichens angibt. Dabei bezeichnet 0 (null) die erste Position. Dieser Parameter ist optional.

endIndex Eine Ganzzahl, die die Indexposition des letzten Zeichens angibt. Dieser Parameter ist optional.

Rückgabewerte

Wenn ein Textfeld ausgewählt ist und in dem Text nur ein Wert verwendet wird, wird die gewünschte Eigenschaft zurückgegeben. Gibt `undefined` zurück, wenn in dem Textfeld mehrere Werte verwendet werden. Wenn mehrere Textfelder ausgewählt sind und diese Textfelder denselben Textausrichtungswert verwenden, gibt die Methode diesen Wert zurück. Wenn mehrere Textfelder ausgewählt sind und diese Textfelder unterschiedliche Textausrichtungswerte verwenden, gibt die Methode den Wert `undefined` zurück. Wenn die optionalen Argumente nicht übergeben werden, gelten diese Regeln für den gegenwärtig ausgewählten Textbereich bzw. für das gesamte Textfeld, falls der Text zurzeit nicht bearbeitet wird. Wenn nur *startIndex* übergeben wird und alle ausgewählten Textobjekte denselben Wert aufweisen, wird die Eigenschaft des Zeichens rechts neben der Indexposition zurückgegeben. Wenn *startIndex* und *endIndex* übergeben werden, bezieht sich der zurückgegebene Wert auf den gesamten Zeichenbereich von *startIndex* bis (ausschließlich) *endIndex*.

Beschreibung

Methode; ruft eine angegebene `TextAttrs`-Eigenschaft der ausgewählten Textobjekte ab. Ausgewählte Objekte, bei denen es sich nicht um Textfelder handelt, werden ignoriert. Eine Liste der Eigenschaftsnamen und erwarteten Werte finden Sie in der Übersicht über die Eigenschaften für das [TextAttrs-Objekt](#). Siehe auch `document.setTextAttrs()`.

Beispiel

Im folgenden Beispiel wird die Größe der ausgewählten Textfelder abgerufen:

```
fl.getDocumentDOM().getElementTextAttr("size");
```

Im folgenden Beispiel wird die Farbe des Zeichens abgerufen, das sich in den ausgewählten Textfeldern an Indexposition 3 befindet:

```
fl.getDocumentDOM().getElementTextAttr("fillColor", 3);
```

Im folgenden Beispiel wird in den ausgewählten Textfeldern der Schriftname des Textes abgerufen, der von Indexposition 2 bis (ausschließlich) Indexposition 10 reicht:

```
fl.getDocumentDOM().getElementTextAttr("face", 2, 10);
```

document.getFilters()

Verfügbarkeit

Flash 8.

Verwendung

```
document.getFilters()
```

Parameter

Keine.

Rückgabewerte

Ein Array, das die Liste der auf die gegenwärtig ausgewählten Objekte angewendeten Filter enthält.

Beschreibung

Methode; gibt ein Array zurück, das die Liste der auf die gegenwärtig ausgewählten Objekte angewendeten Filter enthält. Wenn mehrere Objekte mit unterschiedlichen Filtern ausgewählt sind, gibt diese Methode die Liste der auf das erste ausgewählte Objekt angewendeten Filter zurück.

Beispiel

Siehe `document.setFilters()`.

Siehe auch

`document.addFilter()`, `document.changeFilterOrder()`, `document.setFilters()`, [Filter-Objekt](#)

document.getMetadata()

Verfügbarkeit

Flash 8.

Verwendung

```
document.getMetadata()
```

Parameter

Keine.

Rückgabewerte

Ein String, der die dem Dokument zugeordneten XML-Metadaten enthält, bzw. ein leerer String, wenn keine Metadaten vorhanden sind.

Beschreibung

Methode; gibt einen String zurück, der die dem Dokument zugeordneten XML-Metadaten enthält, bzw. einen leeren String, wenn keine Metadaten vorhanden sind.

Beispiel

Im folgenden Beispiel werden die XML-Metadaten des aktuellen Dokuments im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace("XML Metadata is :" + fl.getDocumentDOM().getMetadata());
```

Siehe auch

`document.setMetadata()`

document.getMobileSettings()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
document.getMobileSettings()
```

Parameter

Keine.

Rückgabewerte

Ein String, der die XML-Einstellungen für das Dokument darstellt. Wenn kein Wert festgelegt wurde, wird ein leerer String zurückgegeben.

Beschreibung

Methode; gibt die mobilen XML-Einstellungen für das Dokument zurück.

Beispiel

Im folgenden Beispiel wird der String mit den XML-Einstellungen für das aktuelle Dokument angezeigt:

```
fl.trace(fl.getDocumentDOM().getMobileSettings());  
//traces a string like the following"<? xml version="1.0" encoding="UTF-16" standalone="no"  
?><mobileSettings> <contentType id="standalonePlayer" name="Standalone Player"/>  
<testDevices> <testDevice id="1170" name="Generic Phone" selected="yes"/> </testDevices>  
<outputMsgFiltering info="no" trace="yes" warning="yes"/> <testWindowState height="496"  
splitterClosed="No" splitterXPos="400" width="907"/> </mobileSettings>"
```

Siehe auch

[document.setMobileSettings\(\)](#)

document.getPlayerVersion()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
document.getPlayerVersion()
```

Parameter

Keine.

Rückgabewerte

Ein String, der die Flash Player-Version, die mit `document.setPlayerVersion()` festgelegt wurde, darstellt. Wenn kein Wert festgelegt wurde, wird der Wert zurückgegeben, der im Dialogfeld „Einstellungen für Veröffentlichungen“ angegeben wurde.

Beschreibung

Methode, gibt einen String zurück, der die vorgesehene Player-Version für das angegebene Dokument darstellt. Eine Liste der Werte, die diese Methode zurückgeben kann, finden Sie unter [document.setPlayerVersion\(\)](#).

Um festzustellen, welche ActionScript-Version in der angegebenen Datei vorgesehen ist, verwenden Sie [document.asVersion](#).

Beispiel

Im folgenden Beispiel wird das Ansprechen bestimmter Player-Versionen für das aktuelle Dokument veranschaulicht. Außerdem werden die entsprechenden Werte abgerufen:

```
fl.getDocumentDOM().setPlayerVersion("6");  
var version = fl.getDocumentDOM().getPlayerVersion();  
fl.trace(version) // displays "6"  
fl.getDocumentDOM().setPlayerVersion("FlashPlayer10");  
var version = fl.getDocumentDOM().getPlayerVersion();  
fl.trace(version) // displays "FlashPlayer10"
```

Siehe auch

[document.setPlayerVersion\(\)](#)

document.getSelectionRect()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.getSelectionRect()
```

Parameter

Keine.

Rückgabewerte

Das Begrenzungsrechteck der aktuellen Auswahl bzw. 0, wenn nichts ausgewählt ist. Informationen zum Format des Rückgabewerts finden Sie unter [document.addNewRectangle\(\)](#).

Beschreibung

Methode; ruft das Begrenzungsrechteck der aktuellen Auswahl ab. Bei einer nicht rechteckigen Auswahl wird das kleinstmögliche Rechteck zurückgegeben, in das die gesamte Auswahl eingepasst werden kann. Das Rechteck basiert auf dem Dokumentraum oder, falls der Bearbeitungsmodus aktiviert ist, auf dem Registrierungspunkt (auch *Bezugspunkt* oder *Nullpunkt* genannt) des bearbeiteten Symbols.

Beispiel

Im folgenden Beispiel wird das Begrenzungsrechteck der aktuellen Auswahl abgerufen und dann werden die Eigenschaften angezeigt:

```
var newRect = fl.getDocumentDOM().getSelectionRect();  
var outputStr = "left: " + newRect.left + " top: " + newRect.top + " right: " + newRect.right  
+ " bottom: " + newRect.bottom;  
alert(outputStr);
```

Siehe auch

[document.selection](#), [document.setSelectionRect\(\)](#)

document.getTextString()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.getTextString([startIndex [, endIndex]])
```

Parameter

startIndex Eine Ganzzahl, die die Indexposition des ersten abzurufenden Zeichens angibt. Dieser Parameter ist optional.

endIndex Eine Ganzzahl, die die Indexposition des letzten abzurufenden Zeichens angibt. Dieser Parameter ist optional.

Rückgabewerte

Ein String mit dem ausgewählten Text.

Beschreibung

Methode; ruft den gegenwärtig ausgewählten Text ab. Wenn die optionalen Parameter nicht übergeben werden, wird die aktuelle Textauswahl verwendet. Wenn gegenwärtig kein Text zur Bearbeitung geöffnet ist, wird der gesamte Textstring zurückgegeben. Wenn nur *startIndex* übergeben wird, wird der String zurückgegeben, der an dieser Indexposition beginnt und am Ende des Felds endet. Wenn *startIndex* und *endIndex* übergeben werden, wird ein String zurückgegeben, der von *startIndex* bis (ausschließlich) *endIndex* reicht.

Wenn mehrere Textfelder ausgewählt sind, werden sämtliche Strings verkettet und dann zurückgegeben.

Beispiel

Im folgenden Beispiel wird der String in den ausgewählten Textfeldern abgerufen:

```
fl.getDocumentDOM().getTextString();
```

Im folgenden Beispiel wird in den ausgewählten Textfeldern der String abgerufen, der sich an der Indexposition 5 befindet:

```
fl.getDocumentDOM().getTextString(5);
```

Im folgenden Beispiel wird der String abgerufen, der von der Indexposition 2 bis (ausschließlich) zur Indexposition 10 reicht:

```
fl.getDocumentDOM().getTextString(2, 10);
```

Siehe auch

[document.setTextString\(\)](#)

document.getTimeline()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.getTimeline()
```

Parameter

Keine.

Rückgabewerte

Das aktuelle Timeline-Objekt.

Beschreibung

Methode; ruft das aktuelle [Timeline-Objekt](#) in dem Dokument ab. Bei der aktuellen Zeitleiste kann es sich um die aktuelle Szene, das gegenwärtig bearbeitete Symbol oder den aktuellen Bildschirm handeln.

Beispiel

Im folgenden Beispiel wird das Timeline-Objekt abgerufen und die Anzahl der Bilder in der längsten Ebene zurückgegeben:

```
var longestLayer = fl.getDocumentDOM().getTimeline().frameCount;
fl.trace("The longest layer has" + longestLayer + "frames");
```

Im folgenden Beispiel wird der Modus „An Position bearbeiten“ für das auf der Bühne ausgewählte Symbol aktiviert und ein Bild in der Zeitleiste des Symbols eingefügt:

```
fl.getDocumentDOM().enterEditMode("inPlace");
fl.getDocumentDOM().getTimeline().insertFrames();
```

Im folgenden Beispiel wird das Timeline-Objekt abgerufen und sein Name angezeigt:

```
var timeline = fl.getDocumentDOM().getTimeline();
alert(timeline.name);
```

Siehe auch

[document.currentTimeline](#), [document.timelines](#), [symbolItem.timeline](#)

document.getTransformationPoint()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.getTransformationPoint()
```

Parameter

Keine.

Rückgabewerte

Ein Punkt (zum Beispiel {x:10, y:20}, wobei x und y Gleitkommazahlen sind), der die Position des Transformationspunkts (auch *Bezugspunkt* oder *Nullpunkt* genannt) innerhalb des Koordinatensystems des ausgewählten Elements angibt.

Beschreibung

Methode; ruft die Koordinaten des Transformationspunkts der aktuellen Auswahl ab. Mit dem Transformationspunkt können Sie Vorgänge wie Drehen und Neigen durchführen.

***Hinweis:** Transformationspunkte stehen je nach Typ des ausgewählten Elements in Relation zu verschiedenen Positionen. Weitere Informationen finden Sie unter [document.setTransformationPoint\(\)](#).*

Beispiel

Im folgenden Beispiel wird der Transformationspunkt der aktuellen Auswahl abgerufen. Die Eigenschaft `transPoint.x` gibt die *x*-Koordinate des Transformationspunkts an und die Eigenschaft `transPoint.y` die *y*-Koordinate.

```
var transPoint = fl.getDocumentDOM().getTransformationPoint();
```

Siehe auch

[document.setTransformationPoint\(\)](#), [element.getTransformationPoint\(\)](#)

document.group()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.group()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; konvertiert die aktuelle Auswahl in eine Gruppe.

Beispiel

Im folgenden Beispiel werden die Objekte der aktuellen Auswahl in eine Gruppe konvertiert:

```
fl.getDocumentDOM().group();
```

Siehe auch

[document.unGroup\(\)](#)

document.height

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.height`

Beschreibung

Eigenschaft; eine Ganzzahl, die die Höhe des Dokuments (Bühne) in Pixel angibt.

Beispiel

Im folgenden Beispiel wird die Höhe der Bühne auf 400 Pixel festgesetzt:

```
fl.getDocumentDOM().height = 400;
```

Siehe auch

[document.width](#)

document.id

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`document.id`

Beschreibung

Schreibgeschützte Eigenschaft; eine eindeutige Ganzzahl (automatisch zugewiesen), die ein Dokument für die Dauer einer Flash-Sitzung identifiziert. Verwenden Sie diese Eigenschaft zusammen mit [fl.findDocumentDOM\(\)](#), um ein bestimmtes Dokument für eine Aktion festzulegen.

Beispiel

Im folgenden Beispiel wird die Dokument-ID für das aktuelle Dokument getestet:

```
fl.trace("Current doc's internal ID is: " + fl.getDocumentDOM().id);
```

Siehe auch

[fl.findDocumentDOM\(\)](#)

document.importFile()

Verfügbarkeit

Flash 8.

Verwendung

```
document.importFile(fileURI [, importToLibrary])
```

Parameter

fileURI Ein String im Format file:/// URI, der den Pfad der zu importierenden Datei angibt.

importToLibrary Ein boolescher Wert, der angibt, ob die Datei nur in die Bibliothek des Dokuments (`true`) oder auch unmittelbar auf die Bühne (`false`) importiert werden soll. Der Standardwert lautet `false`.

Rückgabewerte

Keine.

Beschreibung

Methode; importiert eine Datei in das Dokument. Diese Methode führt die gleiche Operation durch wie die Menübefehle „In Bibliothek importieren“ bzw. „In Bühne importieren“. Ein Veröffentlichungsprofil können Sie mit `document.importPublishProfile()` importieren.

Beispiel

Mit dem folgenden Beispiel kann ein Benutzer eine Datei für den Import auf die Bühne suchen und auswählen:

```
var dom = fl.getDocumentDOM();  
var URI = fl.browseForFileURL("select", "Import File");  
dom.importFile(URI);
```

Siehe auch

`document.importSWF()`, `fl.browseForFileURL()`

document.importPublishProfile()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.importPublishProfile( fileURI )
```

Parameter

fileURI Ein String im Format `file:/// URI`, der den Pfad der XML-Datei angibt, die das zu importierende Profil definiert.

Rückgabewerte

Eine Ganzzahl, bei der es sich um den Index des importierten Profils in der Profilliste handelt. Gibt -1 zurück, wenn das Profil nicht importiert werden kann.

Beschreibung

Methode; importiert ein Profil aus einer Datei.

Beispiel

Im folgenden Beispiel wird das in der Datei „profile.xml“ enthaltene Profil importiert und der Index dieses Profils in der Profilliste angezeigt:

```
alert(fl.getDocumentDOM().importPublishProfile('file:///C:/Documents and  
Settings/janeUser/Desktop/profile.xml'));
```

document.importPublishProfileString()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
document.importPublishProfileString(xmlString)
```

Parameter

xmlString Ein String, der die als aktuelles Profil zu importierenden XML-Daten enthält.

Rückgabewerte

Ein boolescher Wert; `true`, wenn der String erfolgreich importiert wurde; andernfalls `false`.

Beschreibung

Methode; importiert einen XML-String, der ein Veröffentlichungsprofil darstellt, und legt es als aktuelles Profil fest. Bevor Sie diese Methode verwenden, müssen Sie den zu importierenden XML-String mit `document.exportPublishProfileString()` erzeugen.

Beispiel

Im folgenden Beispiel wird das Standardprofil als XML-String exportiert. Der XML-String wird mit dem standardmäßigen JavaScript-Befehl `replace` geändert. Anschließend wird der String importiert und die standardmäßige Ausgabeeinstellung „ActionScript 3“ wird auf „ActionScript 1“ gesetzt.

```
var profileXML=fl.getDocumentDOM().exportPublishProfileString('Default');
fl.trace(profileXML);
var newProfileXML = profileXML.replace("<ActionScriptVersion>3</ActionScriptVersion>",
"<ActionScriptVersion>1</ActionScriptVersion>");
fl.getDocumentDOM().importPublishProfileString(newProfileXML);
```

document.importSWF()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.importSWF(fileURI)
```

Parameter

fileURI Ein String im Format `file:///` URI, der den Pfad der zu importierenden SWF-Datei angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; importiert eine SWF-Datei in das Dokument. Diese Methode hat das gleiche Ergebnis wie die Auswahl einer SWF-Datei über den Menübefehl „Importieren“. In Flash ab Version 8 können Sie SWF-Dateien (und auch andere Dateitypen) mithilfe von `document.importFile()` importieren.

Beispiel

Im folgenden Beispiel wird die Datei "mySwf.swf" aus dem Flash-Konfigurationsordner importiert:

```
fl.getDocumentDOM().importSWF(fl.configURI+"mySwf.swf");
```

Siehe auch

[document.importFile\(\)](#)

document.intersect()

Verfügbarkeit

Flash 8.

Verwendung

```
document.intersect()
```

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; `true`, wenn erfolgreich, ansonsten `false`.

Beschreibung

Methode; erstellt aus allen ausgewählten Zeichnungsobjekten ein Schnittobjekt. Die Methode gibt `false` zurück, wenn keine Zeichnungsobjekte ausgewählt sind oder wenn nicht alle ausgewählten Elemente Zeichnungsobjekte sind.

Beispiel

Im folgenden Beispiel wird aus allen ausgewählten Zeichnungsobjekten ein Schnittobjekt erstellt:

```
fl.getDocumentDOM().intersect();
```

Siehe auch

[document.crop\(\)](#), [document.deleteEnvelope\(\)](#), [document.punch\(\)](#), [document.union\(\)](#),
[shape.isDrawingObject](#)

document.library

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.library`

Beschreibung

Schreibgeschützte Eigenschaft; das [library-Objekt](#) eines Dokuments.

Beispiel

Im folgenden Beispiel wird die Bibliothek des Dokuments abgerufen, das gegenwärtig den Fokus hat:

```
var myCurrentLib = fl.getDocumentDOM().library;
```

Sofern es sich bei dem Dokument mit dem aktuellen Fokus nicht um `fl.documents[1]` handelt, wird im folgenden Beispiel die Bibliothek einer Bibliothek ohne Fokus bzw. einer Bibliothek abgerufen, die Sie mit „Datei“ > „Öffnen“ als externe Bibliothek geöffnet haben:

```
var externalLib = fl.documents[1].library;
```

document.libraryPath

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`document.libraryPath`

Beschreibung

Eigenschaft; ein String, der eine Liste von Elementen enthält, die sich im Pfad der ActionScript 3.0-Bibliothek des Dokuments befinden, und dieser Pfad wiederum gibt den Speicherort von SWC-Dateien oder SWC-Dateien enthaltenden Ordnern an. Elemente innerhalb des Strings sind durch Semikolons getrennt. Zur Angabe der Elemente wählen Sie im Authoring-Tool „Datei“ > „Einstellungen für Veröffentlichungen“ und wählen dann auf der Registerkarte „Flash“ die Option „ActionScript 3.0-Einstellungen“.

Beispiel

Im folgenden Beispiel wird der Ordner „../Files“ zum Bibliothekspfad des Dokuments hinzugefügt und anschließend wird der Bibliothekspfad im Bedienfeld „Ausgabe“ angezeigt:

```
var myDoc = fl.getDocumentDOM();
fl.trace(myDoc.libraryPath);
myDoc.libraryPath = "../Files;" + myDoc.libraryPath;
fl.trace(myDoc.libraryPath);
```

Siehe auch

[document.externalLibraryPath](#), [document.sourcePath](#), [fl.libraryPath](#)

document.livePreview

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.livePreview
```

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob die Live-Vorschau aktiviert ist. Bei dem Wert `true` werden Komponenten so auf der Bühne angezeigt, wie sie im veröffentlichten Flash-Inhalt aussehen. Dabei wird auch die Größe der Komponenten weitgehend reproduziert. Lautet der Wert `false`, werden Komponenten nur in Form von Konturen angezeigt. Der Standardwert lautet `true`.

Beispiel

Im folgenden Beispiel wird der Live-Vorschau der Wert `false` zugewiesen:

```
fl.getDocumentDOM().livePreview = false;
```

document.loadCuepointXML()

Verfügbarkeit

Flash Professional CS5

Verwendung

```
document.loadCuepointXML(String URI)
```

Parameter

URI String; der absolute Pfad der Cue-Point-XML-Datei.

Beschreibung

Methode; lädt eine Cue-Point-XML-Datei. Format und Standarddokumenttyp (DTD) der XML-Datei sind mit der Datei identisch, die vom Eigenschafteninspektor für Cue-Points importiert und exportiert wird. Der Rückgabewert ist mit dem String identisch, der in der Cue-Point-Eigenschaft des Objekts serialisiert ist, das die Instanz einer FLVPlayback-Komponente enthält.

Beispiel

Im folgenden Beispiel befindet sich die Cue-Points-XML-Datei unter `C:\testCuePoints.xml`:

```
var cuePoints = fl.getDocumentDOM().LoadCuepointXML("c:\\testCuePoints.xml");
```

document.match()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.match(bWidth, bHeight [, bUseDocumentBounds])
```


Parameter

width Ein boolescher Wert; bei dem Wert `true` wird allen ausgewählten Elementen eine identische Breite zugewiesen.

height Ein boolescher Wert; bei dem Wert `true` wird allen ausgewählten Elementen eine identische Höhe zugewiesen.

useDocumentBounds Ein boolescher Wert; bei dem Wert `true` wird die Größe der Objekte an die Dokumentgrenzen angepasst. Andernfalls orientiert die Methode sich an den Grenzen des größten Objekts. Der Standardwert lautet `false`. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; weist den ausgewählten Objekten eine identische Größe zu.

Beispiel

Im folgenden Beispiel wird den ausgewählten Objekten eine identische Breite zugewiesen:

```
fl.getDocumentDOM().match(true, false);
```

Im folgenden Beispiel wird nur die Höhe vereinheitlicht:

```
fl.getDocumentDOM().match(false, true);
```

Im folgenden Beispiel wird nur die Breite an die Dokumentgrenzen angepasst:

```
fl.getDocumentDOM().match(true, false, true);
```

Siehe auch

[document.getAlignToDocument\(\)](#), [document.setAlignToDocument\(\)](#)

document.mouseClick()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.mouseClick(position, bToggleSel, bShiftSel)
```

Parameter

position Zwei Gleitkommawerte, die die *x*- und *y*-Koordinaten des Klicks in Pixel angeben.

bToggleSel Ein boolescher Wert, der den Zustand der Umschalttaste angibt: Bei `true` ist die Taste gedrückt, bei `false` ist sie nicht gedrückt.

bShiftSel Ein boolescher Wert, der den Zustand der Anwendungsvoreinstellung „Mit Umschalttaste auswählen“ angibt: Bei `true` ist die Voreinstellung aktiviert, bei `false` ist sie deaktiviert.

Rückgabewerte

Keine.

Beschreibung

Methode; führt mit dem Auswahlwerkzeug einen Mausklick aus.

Beispiel

Im folgenden Beispiel wird an der angegebenen Stelle ein Mausklick ausgeführt:

```
fl.getDocumentDOM().mouseClick({x:300, y:200}, false, false);
```

Siehe auch

[document.mouseDblClick\(\)](#)

document.mouseDblClick()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.mouseDblClick(position, bAltDown, bShiftDown, bShiftSelect)
```

Parameter

position Zwei Gleitkommawerte, die die *x*- und *y*-Koordinaten des Klicks in Pixel angeben.

bAltDown Ein boolescher Wert, der angibt, ob die Alt-Taste zum Zeitpunkt des Ereignisses gedrückt ist: Bei `true` ist die Taste gedrückt, bei `false` ist sie nicht gedrückt.

bShiftDown Ein boolescher Wert, der angibt, ob die Umschalttaste zum Zeitpunkt des Ereignisses gedrückt ist: Bei `true` ist die Taste gedrückt, bei `false` ist sie nicht gedrückt.

bShiftSelect Ein boolescher Wert, der den Zustand der Anwendungsvoreinstellung „Mit Umschalttaste auswählen“ angibt: Bei `true` ist die Voreinstellung aktiviert, bei `false` ist sie deaktiviert.

Rückgabewerte

Keine.

Beschreibung

Methode; führt mit dem Auswahlwerkzeug einen doppelten Mausklick aus.

Beispiel

Im folgenden Beispiel wird an der angegebenen Stelle ein doppelter Mausklick ausgeführt:

```
fl.getDocumentDOM().mouseDblClick({x:392.9, y:73}, false, false, true);
```

Siehe auch

[document.mouseClick\(\)](#)

document.moveSelectedBezierPointsBy()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.moveSelectedBezierPointsBy(delta)
```

Parameter

delta Zwei Gleitkommawerte, die anhand von *x*- und *y*-Koordinaten in Pixel angeben, um welchen Betrag die ausgewählten Bézierpunkte verschoben werden. Beispielsweise bezeichnet $(\{x:1, y:2\})$ eine Stelle, die gegenüber der aktuellen Stelle um einen Pixel nach rechts und um zwei Pixel nach unten versetzt ist.

Rückgabewerte

Keine.

Beschreibung

Methode; wenn die Auswahl mindestens einen Pfad mit mindestens einem ausgewählten Bézierpunkt enthält, werden mit dieser Methode alle ausgewählten Bézierpunkte auf den ausgewählten Pfaden um den angegebenen Betrag verschoben.

Beispiel

Im folgenden Beispiel werden die ausgewählten Bézierpunkte um 10 Pixel nach rechts und um 5 Pixel nach unten verschoben:

```
fl.getDocumentDOM().moveSelectedBezierPointsBy({x:10, y:5});
```

document.moveSelectionBy()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.moveSelectionBy(distanceToMove)
```

Parameter

distanceToMove Zwei Gleitkommawerte, die anhand von *x*- und *y*-Koordinaten angeben, wie weit die Auswahl von der Methode verschoben werden soll. Beispielsweise bezeichnet $(\{x:1, y:2\})$ eine Stelle, die gegenüber der aktuellen Stelle um einen Pixel nach rechts und um zwei Pixel nach unten versetzt ist.

Rückgabewerte

Keine.

Beschreibung

Methode; verschiebt ausgewählte Objekte um einen bestimmten Betrag.

Hinweis: Wenn der Benutzer das Element mit den Pfeiltasten verschiebt, werden sämtliche Tastenaktionen im Bedienfeld „Protokoll“ zu einem Verschiebungsschritt zusammengefasst. Wenn der Benutzer die Pfeiltasten wiederholt drückt, führt die Methode dann anstelle mehrerer Schritte nur einen einzelnen Schritt aus. Die Argumente werden entsprechend aktualisiert, damit sie den wiederholten Gebrauch der Pfeiltasten widerspiegeln.

Wie Sie eine Auswahl erstellen, erfahren Sie unter `document.setSelectionRect()`, `document.mouseClick()`, `document.mouseDbClick()` und [Element-Objekt](#).

Beispiel

Im folgenden Beispiel wird das ausgewählte Element um 62 Pixel nach rechts und um 84 Pixel nach unten verschoben:

```
fl.getDocumentDOM().moveSelectionBy({x:62, y:84});
```

document.name

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.name
```

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der den Namen eines Dokuments angibt (FLA-Datei).

Beispiel

Im folgenden Beispiel wird die Variable `fileName` auf den Dateinamen des ersten Dokuments in dem Dokumentarray gesetzt:

```
var fileName = flash.documents[0].name;
```

Im folgenden Beispiel werden die Namen aller geöffneten Dokumente im Bedienfeld „Ausgabe“ angezeigt:

```
var openDocs = fl.documents;  
for(var i=0;i < openDocs.length; i++){  
    fl.trace(i + " " + openDocs[i].name + "\n");  
}
```

document.optimizeCurves()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.optimizeCurves(smoothing, bUseMultiplePasses)
```

Parameter

smoothing Eine Ganzzahl von 0 (keine Glättung) bis 100 (maximale Glättung).

bUseMultiplePasses Ein boolescher Wert; bei dem Wert `true` soll die Methode mehrere Durchgänge durchführen. Dies dauert zwar länger, liefert aber bessere Ergebnisse. Dieser Parameter hat die gleiche Wirkung wie die Schaltfläche „Mehrere Durchgänge“ im Dialogfeld „Kurven optimieren“.

Rückgabewerte

Keine.

Beschreibung

Methode; optimiert die Glättung der aktuellen Auswahl, wobei mehrere Durchgänge (sofern angegeben) zulässig sind. Diese Methode entspricht dem Befehl „Modifizieren“ > „Form“ > „Optimieren“.

Beispiel

Im folgenden Beispiel wird die Kurvenglättung der aktuellen Auswahl auf 50° optimiert, wobei mehrere Durchgänge zulässig sind:

```
fl.getDocumentDOM().optimizeCurves(50, true);
```

document.path

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.path`

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der den Pfad des Dokuments in einem plattformspezifischen Format angibt. Wenn das Dokument bisher noch nicht gespeichert wurde, lautet diese Eigenschaft `undefined`.

Beispiel

Im folgenden Beispiel wird der Pfad des ersten Dokuments in dem Dokumentarray im Bedienfeld „Ausgabe“ angezeigt: Vor Ausführung des Skripts muss das Dokument gespeichert werden. Im folgenden Beispiel wird die Datei „test fla“ auf einem Windows-Computer im Ordner „Eigene Dateien“ gespeichert.

```
var filePath = flash.documents[0].path;  
fl.trace(filePath);  
// displays C:\Documents and Settings\<user name>\My Documents\test fla
```

Siehe auch

[document.pathURI](#)

document.pathURI

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`document.pathURI`

Beschreibung

Schreibgeschützte Eigenschaft; ein String im Format `file:///` URI, der den Pfad des Dokuments angibt. Wenn das Dokument bisher noch nicht gespeichert wurde, lautet diese Eigenschaft `undefined`.

Beispiel

Im folgenden Beispiel wird der Pfad des ersten Dokuments in dem Dokumentarray im Bedienfeld „Ausgabe“ als URI-String im Format `file:///` angezeigt: Vor Ausführung des Skripts muss das Dokument gespeichert werden. Im folgenden Beispiel wird die Datei „test fla“ auf einem Windows-Computer im Ordner „Eigene Dateien“ gespeichert.

```
var filePathURI = flash.documents[0].pathURI;
fl.trace(filePathURI);
// displays file:///C:/Documents%20and%20Settings/<userName>/My%20Documents/test fla
```

Siehe auch

[document.path](#)

document.publish()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.publish()`

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; veröffentlicht das Dokument anhand der aktiven Veröffentlichungseinstellungen („Datei“ > „Einstellungen für Veröffentlichungen“). Diese Methode entspricht dem Befehl „Datei“ > „Veröffentlichen“.

Beispiel

Im folgenden Beispiel wird das aktuelle Dokument veröffentlicht:

```
fl.getDocumentDOM().publish();
```

document.publishProfiles

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.publishProfiles`

Beschreibung

Schreibgeschützte Eigenschaft; ein Array der Veröffentlichungsprofilnamen für das Dokument.

Beispiel

Im folgenden Beispiel werden die Namen der Veröffentlichungsprofile für das Dokument angezeigt:

```
var myPubProfiles = fl.getDocumentDOM().publishProfiles;
for (var i=0; i < myPubProfiles.length; i++){
    fl.trace(myPubProfiles[i]);
}
```

document.punch()

Verfügbarkeit

Flash 8.

Verwendung

`document.punch()`

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; `true`, wenn erfolgreich, ansonsten `false`.

Beschreibung

Methode; verwendet das zuoberst liegende ausgewählte Zeichnungsobjekt, um alle darunter liegenden ausgewählten Zeichnungsobjekte auszustanzen. Die Methode gibt `false` zurück, wenn keine Zeichnungsobjekte ausgewählt sind oder wenn nicht alle ausgewählten Elemente Zeichnungsobjekte sind.

Beispiel

Im folgenden Beispiel werden die Zeichnungsobjekte unter dem ausgewählten Zeichnungsobjekt ausgestanzt:

```
fl.getDocumentDOM().punch();
```

Siehe auch

`document.crop()`, `document.deleteEnvelope()`, `document.intersect()`, `document.union()`,
`shape.isDrawingObject`

document.removeAllFilters()

Verfügbarkeit

Flash 8.

Verwendung

```
document.removeAllFilters()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; entfernt alle Filter aus den ausgewählten Objekten.

Beispiel

Im folgenden Beispiel werden alle Filter aus den ausgewählten Objekten entfernt:

```
fl.getDocumentDOM().removeAllFilters();
```

Siehe auch

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableAllFilters\(\)](#),
[document.getFilters\(\)](#), [document.removeFilter\(\)](#), [Filter-Objekt](#)

document.removeDataFromDocument()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.removeDataFromDocument(name)
```

Parameter

name Ein String, der den Namen der zu entfernenden Daten angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; entfernt permanente Daten mit dem angegebenen Namen, die mit dem Dokument verknüpft wurden.

Beispiel

Im folgenden Beispiel werden die permanenten Daten mit dem Namen "myData" aus dem Dokument entfernt:

```
fl.getDocumentDOM().removeDataFromDocument("myData");
```

Siehe auch

[document.addDataToDocument\(\)](#), [document.documentHasData\(\)](#), [document.getDataFromDocument\(\)](#)

document.removeDataFromSelection()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.removeDataFromSelection(name)
```

Parameter

name Ein String, der den Namen der zu entfernenden permanenten Daten angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; entfernt permanente Daten mit dem angegebenen Namen, die mit der Auswahl verknüpft wurden.

Beispiel

Im folgenden Beispiel werden die permanenten Daten mit dem Namen "myData" aus der Auswahl entfernt:

```
fl.getDocumentDOM().removeDataFromSelection("myData");
```

Siehe auch

[document.addDataToSelection\(\)](#)

document.removeFilter()

Verfügbarkeit

Flash 8.

Verwendung

```
document.removeFilter(filterIndex)
```

Parameter

filterIndex Eine Ganzzahl, die den auf Null basierenden Index des Filters angibt, der aus den ausgewählten Objekten entfernt werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; entfernt den angegebenen Filter aus der Filterliste der ausgewählten Objekte.

Beispiel

Im folgenden Beispiel wird der erste Filter (Indexwert 0) aus der Filterliste der ausgewählten Objekte entfernt:

```
fl.getDocumentDOM().removeFilter(0);
```

Siehe auch

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableFilter\(\)](#),
[document.getFilters\(\)](#), [document.removeAllFilters\(\)](#), [Filter-Objekt](#)

document.renamePublishProfile()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.renamePublishProfile([profileNewName])
```

Parameter

profileNewName Ein optionaler Parameter, der den neuen Namen des Profils angibt. Der neue Name muss eindeutig sein. Wenn Sie keinen Namen angeben, wird ein Standardname verwendet.

Rückgabewerte

Ein boolescher Wert; `true`, wenn der Name erfolgreich geändert wurde, ansonsten `false`.

Beschreibung

Methode; benennt das aktuelle Profil um.

Beispiel

Im folgenden Beispiel wird dem aktuellen Profil ein neuer Name (Standardname) zugewiesen und dieser Name dann angezeigt:

```
alert(fl.getDocumentDOM().renamePublishProfile());
```

document.renameScene()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.renameScene(name)
```

Parameter

name Ein String, der den neuen Namen der Szene angibt.

Rückgabewerte

Ein boolescher Wert; `true`, wenn der Name erfolgreich geändert wurde, ansonsten `false`. Die Methode gibt beispielsweise `false` zurück, wenn der neue Name nicht eindeutig ist.

Beschreibung

Methode; benennt die im Bedienfeld „Szene“ gegenwärtig ausgewählte Szene um. Der neue Name der ausgewählten Szene muss eindeutig sein.

Beispiel

Im folgenden Beispiel wird die aktuelle Szene in "new name" umbenannt:

```
var success = fl.getDocumentDOM().renameScene("new name");
```

document.reorderScene()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.reorderScene(sceneToMove, sceneToPutItBefore)
```

Parameter

sceneToMove Eine Ganzzahl, die die zu verschiebende Szene angibt. Der Wert 0 (null) bezeichnet die erste Szene.

sceneToPutItBefore Eine Ganzzahl, die die Szene angibt, vor die die mit *sceneToMove* ausgewählte Szene verschoben werden soll. Die erste Szene hat den Wert 0 (null). Wenn Sie beispielsweise für *sceneToMove* den Wert 1 angeben und für *sceneToPutItBefore* den Wert 0, wird die zweite Szene vor die erste verschoben. Mit -1 wird die Szene an das Ende verschoben.

Rückgabewerte

Keine.

Beschreibung

Methode; verschiebt die angegebene Szene vor eine andere angegebene Szene.

Beispiel

Im folgenden Beispiel wird die zweite Szene vor die erste verschoben:

```
fl.getDocumentDOM().reorderScene(1, 0);
```

document.resetOvalObject()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
document.resetOvalObject()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; setzt alle Werte im Eigenschafteninspektor auf die Standardeinstellungen für das Oval-Objekt zurück. Wenn Oval-Objekte ausgewählt sind, werden ihre Eigenschaften ebenfalls auf die Standardwerte zurückgesetzt.

Beispiel

Im folgenden Beispiel werden die Eigenschaften von Oval-Objekten im aktuellen Dokument auf die Standardwerte zurückgesetzt:

```
fl.getDocumentDOM().resetOvalObject();
```

Siehe auch

[document.resetRectangleObject\(\)](#)

document.resetRectangleObject()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
document.resetRectangleObject();
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; setzt alle Werte im Eigenschafteninspektor auf die Standardeinstellungen für das Rectangle-Objekt zurück. Wenn Rectangle-Objekte ausgewählt sind, werden ihre Eigenschaften ebenfalls auf die Standardwerte zurückgesetzt.

Beispiel

Im folgenden Beispiel werden die Eigenschaften von Rectangle-Objekten im aktuellen Dokument auf die Standardwerte zurückgesetzt:

```
fl.getDocumentDOM().resetRectangleObject();
```

Siehe auch

[document.resetOvalObject\(\)](#)

document.resetTransformation()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.resetTransformation()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; setzt die Transformationsmatrix zurück. Diese Methode entspricht dem Befehl „Modifizieren“ > „Transformieren“ > „Transformation entfernen“.

Beispiel

Im folgenden Beispiel wird die Transformationsmatrix der aktuellen Auswahl zurückgesetzt:

```
fl.getDocumentDOM().resetTransformation();
```

document.revert()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.revert()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; stellt die zuvor gespeicherte Version des angegebenen Dokuments wieder her. Diese Methode entspricht dem Befehl „Datei“ > „Wiederherstellen“.

Beispiel

Im folgenden Beispiel wird die zuvor gespeicherte Version des aktuellen Dokuments wiederhergestellt:

```
fl.getDocumentDOM().revert();
```

Siehe auch

`document.canRevert()`, `fl.revertDocument()`

document.rotate3DSelection()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`document.rotate3DSelection(xyzCoordinate, bGlobalTransform)`

Parameter

xyzCoordinate Ein XYZ-Koordinatenpunkt, der die 3D-Drehungsachsen angibt.

bGlobalTransform Ein boolescher Wert, der angibt, ob es sich um den Transformationsmodus (`true`) oder nicht (`false`) handelt.

Rückgabewerte

Keine.

Beschreibung

Methode; wendet eine 3D-Drehung auf die Auswahl an. Diese Methode ist nur für MovieClips verfügbar.

Beispiel

Im folgenden Beispiel wird die Auswahl zuerst relativ zur Bühne gedreht (global) und dann relativ zu sich selbst (lokal).

```
var myDocument = fl.getDocumentDOM();
myDocument.rotate3DSelection({x:52.0, y:0, z:0}, true);
myDocument.rotate3DSelection({x:52.0, y:0, z:-55.2}, false);
```

document.rotateSelection()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.rotateSelection(angle [, rotationPoint])`

Parameter

angle Ein Gleitkommawert, der den Drehwinkel angibt.

rotationPoint Ein String, der angibt, welche Seite des Begrenzungsrahmens gedreht werden soll. Die zulässigen Werte lauten "top right", "top left", "bottom right", "bottom left", "top center", "right center", "bottom center" und "left center". Wenn kein Wert angegeben wurde, verwendet die Methode den Transformationspunkt. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; dreht die Auswahl um einen in Grad angegebenen Betrag. Die gleiche Wirkung erzielen Sie, wenn Sie das Objekt mit dem Werkzeug „Frei transformieren“ drehen.

Beispiel

Im folgenden Beispiel wird die Auswahl um 45° um den Transformationspunkt gedreht:

```
fl.getDocumentDOM().rotateSelection(45);
```

Im folgenden Beispiel wird die Auswahl um 45° um die untere linke Ecke gedreht:

```
fl.getDocumentDOM().rotateSelection(45, "bottom left");
```

document.save()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.save([bOkToSaveAs])
```

Parameter

bOkToSaveAs Ein optionaler Parameter; wenn der Wert `true` lautet oder der Parameter fehlt und die Datei noch nicht gespeichert wurde, wird das Dialogfeld „Speichern unter“ geöffnet. Wenn der Wert `false` lautet und die Datei bisher noch nicht gespeichert wurde, wird die Datei nicht gespeichert.

Rückgabewerte

Ein boolescher Wert; `true`, wenn der Speichervorgang erfolgreich war, ansonsten `false`.

Beschreibung

Methode; speichert das aktuelle Dokument an seinem Standardspeicherort. Diese Methode entspricht dem Befehl „Datei“ > „Speichern“.

Um einen Namen für die Datei anzugeben (anstatt sie mit demselben Namen zu speichern), verwenden Sie

```
fl.saveDocument().
```

Hinweis: Wenn die Datei neu ist und nicht geändert oder gespeichert wurde oder wenn die Datei seit dem letzten Speichern nicht geändert wurde, hat diese Methode keinerlei Auswirkungen und es wird `false` zurückgegeben. Wenn Sie zulassen möchten, dass eine nicht gespeicherte bzw. nicht geänderte Datei gespeichert werden soll, verwenden Sie `document.saveAndCompact()` oder `fl.saveDocumentAs()`.

Beispiel

Im folgenden Beispiel wird das aktuelle Dokument an seinem Standardspeicherort gespeichert:

```
fl.getDocumentDOM().save();
```

Siehe auch

`document.saveAndCompact()`, `fl.saveAll()`, `fl.saveDocument()`, `fl.saveDocumentAs()`

document.saveAndCompact()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.saveAndCompact ( [bOkToSaveAs] )
```

Parameter

bOkToSaveAs Ein optionaler Parameter; wenn der Wert `true` lautet oder der Parameter fehlt und die Datei noch nicht gespeichert wurde, wird das Dialogfeld „Speichern unter“ geöffnet. Wenn der Wert `false` lautet und die Datei bisher noch nicht gespeichert wurde, wird die Datei nicht gespeichert. Der Standardwert lautet `true`.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die Datei erfolgreich gespeichert und komprimiert wurde, ansonsten `false`.

Beschreibung

Methode; speichert und komprimiert die Datei. Diese Methode entspricht der Auswahl von „Datei“ > „Speichern und komprimieren“.

Hinweis: Wurde die Datei noch nicht gespeichert, gibt diese Methode auch dann den Wert `true` zurück, wenn der Benutzer das Dialogfeld „Speichern unter“ abbricht. Um genau festzustellen, ob die Datei gespeichert wurde, verwenden Sie `fl.saveDocumentAs()`.

Beispiel

Im folgenden Beispiel wird das aktuelle Dokument gespeichert und komprimiert:

```
fl.getDocumentDOM().saveAndCompact();
```

Siehe auch

`document.save()`, `fl.saveDocumentAs()`, `fl.saveDocument()`, `fl.saveAll()`

document.scaleSelection()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.scaleSelection(xScale, yScale [, whichCorner])
```

Parameter

xscale Ein Gleitkommawert, der den x-Betrag für die Skalierung angibt.

yscale Ein Gleitkommawert, der den *y*-Betrag für die Skalierung angibt.

whichCorner Ein String, der die Ecke angibt, an der die Transformation durchgeführt werden soll. Fehlt er, orientiert die Skalierung sich am Transformationspunkt. Die zulässigen Werte lauten "bottom left", "bottom right", "top right", "top left", "top center", "right center", "bottom center" und "left center". Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; skaliert die Auswahl um einen angegebenen Betrag. Das gleiche Ergebnis erzielen Sie, wenn Sie das Objekt mit dem Werkzeug „Frei transformieren“ skalieren.

Beispiel

Im folgenden Beispiel wird die ursprüngliche Breite der aktuellen Auswahl verdoppelt und die Höhe um die Hälfte reduziert:

```
fl.getDocumentDOM().scaleSelection(2.0, 0.5);
```

Im folgenden Beispiel wird die Auswahl vertikal gespiegelt:

```
fl.getDocumentDOM().scaleSelection(1, -1);
```

Im folgenden Beispiel wird die Auswahl horizontal gespiegelt:

```
fl.getDocumentDOM().scaleSelection(-1, 1);
```

Im folgenden Beispiel wird die Auswahl von der oberen Mitte aus um den Faktor 1,9 skaliert:

```
fl.getDocumentDOM().scaleSelection(1, 1.90, 'top center');
```

document.screenOutline

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.screenOutline`

Beschreibung

Schreibgeschützte Eigenschaft; das aktuelle ScreenOutline-Objekt des Dokuments. Bevor Sie zum ersten Mal auf das Objekt zugreifen, sollten Sie mit `document.allowScreens()` feststellen, ob die Eigenschaft vorhanden ist.

Beispiel

Im folgenden Beispiel wird das Array der Werte der Eigenschaft `screenOutline` angezeigt:

```
var myArray = new Array();
for(var i in fl.getDocumentDOM().screenOutline) {
    myArray.push(" "+i+" : "+fl.getDocumentDOM().screenOutline[i]) ;
}
fl.trace("Here is the property dump for screenOutline: "+myArray);
```

Siehe auch

`document.allowScreens()`

document.selectAll()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.selectAll()`

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; wählt alle Elemente auf der Bühne aus. Dies entspricht dem Tastaturbefehl <Strg>+<A> (Windows) bzw. <Befehl>+<A> (Macintosh) sowie dem Befehl „Bearbeiten“ > „Alles auswählen“.

Beispiel

Im folgenden Beispiel werden alle Elemente ausgewählt, die gegenwärtig sichtbar sind:

```
fl.getDocumentDOM().selectAll();
```

Siehe auch

`document.selection`, `document.selectNone()`

document.selection

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.selection`

Beschreibung

Eigenschaft; ein Array der im Dokument ausgewählten Objekte. Wenn nichts ausgewählt ist, wird ein Array mit der Länge null zurückgegeben. Ist kein Dokument geöffnet, wird `null` zurückgegeben.

Wenn Sie dem Array Objekte hinzufügen möchten, müssen Sie diese zuerst auswählen. Dazu stehen verschiedene Möglichkeiten zur Auswahl:

- Wählen Sie Objekte auf der Bühne manuell aus.

- Verwenden Sie eine der Auswahlmethoden, z. B. `document.setSelectionRect()`, `document.setSelectionBounds()`, `document.mouseClick()`, `document.mouseDb1C1k()` oder `document.selectAll()`.
- Wählen Sie ein oder mehrere Bilder manuell aus.
- Verwenden Sie zur Auswahl eines oder mehrerer Bilder eine Methode des **Timeline-Objekt**, zum Beispiel `timeline.getSelectedFrames()`, `timeline.setSelectedFrames()` oder `timeline.selectAllFrames()`.
- Geben Sie alle Elemente in einem bestimmten Bild an (siehe **Element-Objekt**). Betrachten Sie dazu das erste nachstehende Beispiel.
- Erstellen Sie ein Array mit einem oder mehreren Elementen und weisen Sie dieses Array dem Array `document.selection` zu. Betrachten Sie dazu das dritte nachstehende Beispiel.

Beispiel

Im folgenden Beispiel werden alle Elemente in Bild 11 der aktuellen Auswahl zugewiesen (beachten Sie hierbei, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().currentFrame = 10;  
fl.getDocumentDOM().selection =  
fl.getDocumentDOM().getTimeline().layers[0].frames[10].elements;
```

Im folgenden Beispiel werden in der linken oberen Ecke der Bühne ein Rechteck und darunter ein Textstring erstellt. Dann werden beide Objekte mithilfe von `document.setSelectionRect()` ausgewählt und dem Array `document.selection` hinzugefügt. Zum Schluss wird der Inhalt von `document.selection` im Bedienfeld „Ausgabe“ angezeigt.

```
fl.getDocumentDOM().addNewRectangle({left:0, top:0, right:99, bottom:99}, 0);  
fl.getDocumentDOM().addNewText({left:-1, top:117.3, right:9.2, bottom:134.6});  
fl.getDocumentDOM().setTextString('Hello World');  
fl.getDocumentDOM().setSelectionRect({left:-28, top:-22, right:156.0, bottom:163});  
  
var theSelectionArray = fl.getDocumentDOM().selection;  
  
for(var i=0;i<theSelectionArray.length;i++){  
  fl.trace("fl.getDocumentDOM().selection["+i+"] = " + theSelectionArray[i]);  
}
```

Das folgende Beispiel ist etwas anspruchsvoller. Es demonstriert, wie man das Ebenen- und Elementarray durchläuft, um Instanzen eines bestimmten Symbols zu identifizieren und auszuwählen. Falls Sie zusätzlich mehrere Bilder oder Szenen berücksichtigen möchten, ließe dieses Beispiel sich auch erweitern. In diesem Beispiel werden alle Instanzen des Movieclips `myMovieClip` im ersten Bild der aktuellen Auswahl zugewiesen:

```
// Assigns the layers array to the variable "theLayers".
var theLayers = fl.getDocumentDOM().getTimeline().layers;
// Creates an array to hold all the elements
// that are instances of "myMovieClip".
var myArray = new Array();
// Counter variable
var x = 0;
// Begin loop through all the layers.
for (var i = 0; i < theLayers.length; i++) {
    // Gets the array of elements in Frame 1
    // and assigns it to the array "theElems".
    var theElems = theLayers[i].frames[0].elements;
    // Begin loop through the elements on a layer.
    for (var c = 0; c < theElems.length; c++) {
        // Checks to see if the element is of type "instance".
        if (theElems[c].elementType == "instance") {
            // If the element is an instance, it checks
            // if it is an instance of "myMovieClip".
            if (theElems[c].libraryItem.name == "myMovieClip") {
                // Assigns elements that are instances of "myMovieClip" to "myArray".
                myArray[x] = theElems[c];
                // Increments counter variable.
                x++;
            }
        }
    }
}
// Now that you have assigned all the instances of "myMovieClip"
// to "myArray", you then set the document.selection array
// equal to myArray. This selects the objects on the Stage.
fl.getDocumentDOM().selection = myArray;
```

document.selectNone()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.selectNone()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; hebt die vorhandene Elementauswahl auf.

Beispiel

Im folgenden Beispiel wird die vorhandene Elementauswahl aufgehoben:

```
fl.getDocumentDOM().selectNone();
```

Siehe auch

[document.selectAll\(\)](#), [document.selection](#)

document.setAlignToDocument()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setAlignToDocument(bToStage)
```

Parameter

bToStage Ein boolescher Wert; bei dem Wert `true` werden Objekte an der Bühne ausgerichtet. Bei dem Wert `false` ist dies nicht der Fall.

Rückgabewerte

Keine.

Beschreibung

Methode; legt die Voreinstellungen für die Methoden [document.align\(\)](#), [document.distribute\(\)](#), [document.match\(\)](#) und [document.space\(\)](#) fest, die für das Dokument gelten sollen. Diese Methode entspricht der Aktivierung der Schaltfläche „An Bühne“ im Bedienfeld „Ausrichten“.

Beispiel

Im folgenden Beispiel wird die Schaltfläche „An Bühne“ im Bedienfeld „Ausrichten“ aktiviert, damit Objekte an der Bühne ausgerichtet werden:

```
fl.getDocumentDOM().setAlignToDocument(true);
```

Siehe auch

[document.getAlignToDocument\(\)](#)

document.setBlendMode()

Verfügbarkeit

Flash 8.

Verwendung

```
document.setBlendMode(mode)
```

Parameter

mode Ein String, der den gewünschten Mischmodus für die ausgewählten Objekte angibt. Die zulässigen Werte lauten "normal", "layer", "multiply", "screen", "overlay", "hardlight", "lighten", "darken", "difference", "add", "subtract", "invert", "alpha" und "erase".

Rückgabewerte

Keine.

Beschreibung

Methode; legt den Mischmodus für die ausgewählten Objekte fest.

Beispiel

Im folgenden Beispiel wird dem ausgewählten Objekt der Mischmodus "add" zugewiesen.

```
fl.getDocumentDOM().setBlendMode("add");
```

Siehe auch

[document.addFilter\(\)](#), [document.setFilterProperty\(\)](#), [symbolInstance.blendMode](#)

document.setCustomFill()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setCustomFill(fill)
```

Parameter

fill Ein Fill-Objekt, das die zu verwendenden Füllungseinstellungen angibt. Siehe [Fill-Objekt](#).

Rückgabewerte

Keine.

Beschreibung

Methode; legt die Füllungseinstellungen für das Bedienfeld „Werkzeuge“, den Eigenschafteninspektor und alle ausgewählten Formen fest. Um die Füllungseinstellungen zu ändern, muss ein Skript normalerweise zuerst ein Objekt erstellen und dieses Objekt dann auswählen; dank dieser Methode kann ein Skript jedoch die Füllungseinstellungen bereits vor dem Zeichnen des Objekts festlegen. Darüber hinaus kann es die Füllungseinstellungen des Bedienfelds „Werkzeuge“ und des Eigenschafteninspektors ändern.

Beispiel

Im folgenden Beispiel wird das Farbfeld für die Füllung im Bedienfeld „Werkzeuge“, im Eigenschafteninspektor und in beliebigen ausgewählten Formen in Weiß geändert:

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.color = '#FFFFFF';  
fill.style = "solid";  
fl.getDocumentDOM().setCustomFill(fill);
```

Siehe auch

`document.getCustomFill()`

document.setCustomStroke()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.setCustomStroke(stroke)`

Parameter

stroke Ein [Stroke-Objekt](#).

Rückgabewerte

Keine.

Beschreibung

Methode; legt die Stricheinstellungen für das Bedienfeld „Werkzeuge“, den Eigenschafteninspektor und alle ausgewählten Formen fest. Um die Stricheinstellungen zu ändern, muss ein Skript normalerweise zuerst ein Objekt erstellen und dieses Objekt dann auswählen; dank dieser Methode kann ein Skript jedoch die Stricheinstellungen bereits vor dem Zeichnen des Objekts festlegen. Darüber hinaus kann es die Stricheinstellungen des Bedienfelds „Werkzeuge“ und des Eigenschafteninspektors ändern.

Beispiel

Im folgenden Beispiel wird die Strichstärkeneinstellung im Bedienfeld „Werkzeuge“, im Eigenschafteninspektor und in allen ausgewählten Formen geändert:

```
var stroke = fl.getDocumentDOM().getCustomStroke();  
stroke.thickness += 2;  
fl.getDocumentDOM().setCustomStroke(stroke);
```

Siehe auch

`document.getCustomStroke()`

document.setElementProperty()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.setElementProperty(property, value)`

Parameter

property Ein String mit dem Namen der festzulegenden `Element`-Eigenschaft. Eine vollständige Liste der Eigenschaften und Werte finden Sie in der Übersicht über die Eigenschaften des [Element-Objekt](#).

Mit dieser Methode können Sie keine Werte für schreibgeschützte Eigenschaften festlegen, wie z. B. `element.elementType`, `element.top` oder `element.left`.

value Eine Ganzzahl, die den in der angegebenen `Element`-Eigenschaft festzulegenden Wert angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; legt die angegebene `Element`-Eigenschaft für ausgewählte Objekte in dem Dokument fest. Wenn nichts ausgewählt ist, ist diese Methode wirkungslos.

Beispiel

Im folgenden Beispiel wird allen ausgewählten Objekten ein Breitenwert von 100 und ein Höhenwert von 50 zugewiesen:

```
fl.getDocumentDOM().setElementProperty("width", 100);  
fl.getDocumentDOM().setElementProperty("height", 50);
```

document.setElementTextAttr()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setElementTextAttr(attrName, attrValue [, startIndex [, endIndex]])
```

Parameter

attrName Ein String mit dem Namen der zu ändernden `TextAttrs`-Eigenschaft.

attrValue Der Wert, auf den die Eigenschaft `TextAttrs` eingestellt werden soll. Eine Liste der Eigenschaftsnamen und erwarteten Werte finden Sie in der Übersicht über die Eigenschaften des [TextAttrs-Objekt](#).

startIndex Eine Ganzzahl, die die Indexposition des ersten betroffenen Zeichens angibt. Dieser Parameter ist optional.

endIndex Eine Ganzzahl, die die Indexposition des letzten betroffenen Zeichens angibt. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn mindestens eine Textattributeigenschaft geändert wurde, ansonsten `false`.

Beschreibung

Methode; weist der angegebenen `textAttrs`-Eigenschaft der ausgewählten Textelemente den angegebenen Wert zu. Eine Liste der Eigenschaftsnamen und zulässigen Werte finden Sie in der Übersicht über die Eigenschaften für das [TextAttrs-Objekt](#). Wenn die optionalen Argumente nicht übergeben werden, legt die Methode den Stil des gegenwärtig ausgewählten Textbereichs fest bzw. des gesamten Textfelds, falls kein Text ausgewählt ist. Wenn Sie nur `startIndex` übergeben, werden die Attribute des Zeichens an dieser Indexposition eingestellt. Wenn Sie `startIndex` und `endIndex` übergeben, werden die Attribute des Zeichenbereichs eingestellt, der von `startIndex` bis (ausschließlich) `endIndex` reicht. Wenn Absatzstile festgelegt wurden, sind alle Absätze betroffen, die in diesen Bereich fallen.

Beispiel

In den folgenden Beispielen werden die Textattribute `fillColor`, „italic“ und „bold“ für die ausgewählten Textelemente eingestellt:

```
var success = fl.getDocumentDOM().setElementTextAttr("fillColor", "#00ff00");  
var pass = fl.getDocumentDOM().setElementTextAttr("italic", true, 10);  
var ok = fl.getDocumentDOM().setElementTextAttr("bold", true, 5, 15);
```

document.setFillColor()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setFillColor(color)
```

Parameter

color Die Farbe der Füllung. Sie wird in einem der folgenden Formate angegeben:

- als String im Format „#RRGGBB“ oder „#RRGGBBAA“
- als Hexadezimalzahl im Format „0xRRGGBB“
- als Ganzzahl, die die dezimale Entsprechung einer hexadezimalen Zahl darstellt

Lautet der Wert `null`, wird keine Füllfarbe zugewiesen. Das gleiche Ergebnis erzielen Sie, wenn Sie in der Benutzeroberfläche für das Farbfeld „Füllung“ keine Füllung auswählen.

Rückgabewerte

Keine.

Beschreibung

Methode; ersetzt die Füllfarbe der Auswahl durch die angegebene Farbe. Weitere Informationen zum Ändern der Füllfarbe im Bedienfeld „Werkzeuge“ oder im Eigenschafteninspektor finden Sie unter [document.setCustomFill\(\)](#).

Beispiel

In den ersten drei Anweisungen des folgenden Beispiels wird die Füllfarbe jeweils anhand eines der drei verfügbaren Farbformate festgelegt. In der vierten Anweisung wird keine Füllung (`null`) zugewiesen.

```
fl.getDocumentDOM().setFillColor("#cc00cc");  
fl.getDocumentDOM().setFillColor(0xcc00cc);  
fl.getDocumentDOM().setFillColor(120000);  
fl.getDocumentDOM().setFillColor(null);
```

document.setFilterProperty()

Verfügbarkeit

Flash 8.

Verwendung

```
document.setFilterProperty(property, filterIndex, value)
```

Parameter

property Ein String, der die festzulegende Eigenschaft angibt. Die zulässigen Werte lauten "blurX", "blurY", "quality", "angle", "distance", "strength", "knockout", "inner", "bevelType", "color", "shadowColor" und "highlightColor".

filterIndex Eine Ganzzahl, die den auf Null basierenden Index des Filters in der Filterliste angibt.

value Eine Zahl oder ein String, die bzw. der den für die angegebene Filtereigenschaft einzustellenden Wert angibt. Welche Werte zulässig sind, hängt jeweils von der Eigenschaft und dem Filter ab.

Rückgabewerte

Keine.

Beschreibung

Methode; stellt eine angegebene Filtereigenschaft für die gegenwärtig ausgewählten Objekte ein (vorausgesetzt, dass die Objekte den angegebenen Filter unterstützen).

Beispiel

Im folgenden Beispiel wird der `quality`-Eigenschaft des zweiten Filters in der Filterliste (Indexwert 1) der ausgewählten Objekte der Wert 2 zugewiesen und dann wird die `shadowColor`-Eigenschaft des ersten Filters in der Filterliste der ausgewählten Objekte festgelegt:

```
fl.getDocumentDOM().setFilterProperty("quality", 1, 2);  
fl.getDocumentDOM().setFilterProperty("shadowColor", 0, "#FF00FF");
```

Siehe auch

[document.addFilter\(\)](#), [document.getFilters\(\)](#), [document.setBlendMode\(\)](#), [document.setFilters\(\)](#), [Filter-Objekt](#)

document.setFilters()

Verfügbarkeit

Flash 8.

Verwendung

```
document.setFilters(filterArray)
```

Parameter

filterArray Das Array der gegenwärtig festgelegten Filter.

Rückgabewerte

Keine.

Beschreibung

Methode; wendet auf die ausgewählten Objekte Filter an. Verwenden Sie diese Methode, nachdem Sie `document.getFilters()` aufgerufen und an den Filtern alle gewünschten Änderungen vorgenommen haben.

Beispiel

Im folgenden Beispiel werden die Filter des ausgewählten Objekts abgerufen und dann wird der Eigenschaft `blurX` aller Weichzeichnungsfilter der Wert 50 zugewiesen:

```
var myFilters = fl.getDocumentDOM().getFilters();
for (i=0; i < myFilters.length; i++) {
    if (myFilters[i].name == "blurFilter") {
        myFilters[i].blurX = 50;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

Siehe auch

[document.addFilter\(\)](#), [document.getFilters\(\)](#), [document.setFilterProperty\(\)](#), [Filter-Objekt](#)

document.setInstanceAlpha()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setInstanceAlpha(opacity)
```

Parameter

opacity Eine Ganzzahl von 0 (transparent) bis 100 (vollständig gesättigt), die die Transparenz der Instanz ändert.

Rückgabewerte

Keine.

Beschreibung

Methode; legt die Transparenz der Instanz fest.

Beispiel

Im folgenden Beispiel wird die Transparenz des Farbtons auf den Wert 50 festgesetzt:

```
fl.getDocumentDOM().setInstanceAlpha(50);
```

document.setInstanceBrightness()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setInstanceBrightness(brightness)
```

Parameter

Helligkeit Eine Ganzzahl von -100 (schwarz) bis 100 (weiß), die die Helligkeit angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; legt die Helligkeit der Instanz fest.

Beispiel

Im folgenden Beispiel wird die Helligkeit der Instanz auf den Wert 50 festgesetzt:

```
fl.getDocumentDOM().setInstanceBrightness(50);
```

document.setInstanceTint()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setInstanceTint( color, strength )
```

Parameter

color Die Farbe des Farbtons. Sie wird in einem der folgenden Formate angegeben:

- als String im Format "#RRGGBB" oder "#RRGGBBAA"
- als Hexadezimalzahl im Format "0xRRGGBB"
- als Ganzzahl, die die dezimale Entsprechung einer hexadezimalen Zahl darstellt

strength Eine Ganzzahl von 0 bis 100, die die Transparenz des Farbtons angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; legt den Farbton der Instanz fest.

Beispiel

Im folgenden Beispiel werden der ausgewählten Instanz der Farbton Rot und der Transparenzwert 50 zugewiesen:

```
fl.getDocumentDOM().setInstanceTint(0xff0000, 50);
```

document.setMetadata()

Verfügbarkeit

Flash 8.

Verwendung

```
document.setMetadata(strMetadata)
```

Parameter

strMetadata Ein String, der die mit dem Dokument zu verknüpfenden XML-Metadaten enthält. Weitere Informationen finden Sie in der nachfolgenden Beschreibung.

Rückgabewerte

Ein boolescher Wert; `true`, wenn erfolgreich, ansonsten `false`.

Beschreibung

Methode; legt die XML-Metadaten für das angegebene Dokument fest und überschreibt dabei alle vorhandenen Metadaten. Die XML-Daten, die als *strMetadata* übergeben werden, werden geprüft und vor der Speicherung möglicherweise neu geschrieben. Wenn es sich bei den Daten nicht um gültige XML-Daten handelt oder wenn die Daten gegen bestimmte Regeln verstoßen, werden die XML-Metadaten nicht festgelegt und es wird `false` zurückgegeben. (Bei der Rückgabe von `false` ist es nicht möglich, detailliertere Fehlerinformationen abzurufen.)

Hinweis: Selbst wenn `true` zurückgegeben wird, stimmen die eingestellten XML-Daten möglicherweise nicht vollständig mit dem String überein, der ursprünglich übergeben wurde. Wenn Sie den exakten Wert abrufen möchten, auf den die XML-Daten eingestellt wurden, verwenden Sie `document.getMetadata()`.

Die Metadaten liegen im RDF-Format vor, das mit der XMP-Spezifikation kompatibel ist. Weitere Informationen zu RDF und XMP finden Sie in den folgenden Quellen:

- RDF Primer unter www.w3.org/TR/rdf-primer/
- RDF-Spezifikation unter www.w3.org/TR/1999/REC-rdf-syntax-19990222/
- XMP-Homepage unter www.adobe.com/products/xmp/

Beispiel

In den folgenden Beispielen finden Sie verschiedene legitime Möglichkeiten zur Darstellung derselben Daten. Abgesehen vom zweiten Fall gilt in allen Fällen, dass die Daten beim Senden an `Document.setMetadata()` nicht neu geschrieben werden, sondern dass lediglich die Zeilenumbrüche entfernt werden.

Im ersten Beispiel befinden sich die Metadaten in Tags und für jedes Schema gibt es ein eigenes `rdf:Description`-Tag:

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
<rdf:Description rdf:about='' xmlns:dc='http://purl.org/dc/1.1/'>
<dc:title>Simple title</dc:title>
<dc:description>Simple description</dc:description>
</rdf:Description>
<rdf:Description rdf:about='' xmlns:xmp='http://ns.adobe.com/xap/1.0/'>
<xmp:CreateDate>2004-10-12T10:29-07:00</xmp:CreateDate>
<xmp:CreatorTool>Flash Authoring WIN 8,0,0,215</xmp:CreatorTool>
</rdf:Description>
</rdf:RDF>
```

Im zweiten Beispiel befinden sich die Metadaten in Tags und für jedes Schema gibt es ein gemeinsames `rdf:Description`-Tag. Dieses Beispiel enthält außerdem Kommentare, die von `Document.setMetadata()` ignoriert und verworfen werden:

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
  <!-- This is before the first rdf:Description tag -->
<rdf:Description rdf:about='' xmlns:dc='http://purl.org/dc/1.1/'>
<dc:title>Simple title</dc:title>
<dc:description>Simple description</dc:description>
</rdf:Description>
  <!-- This is between the two rdf:Description tags -->
<rdf:Description rdf:about='' xmlns:xmp='http://ns.adobe.com/xap/1.0/'>
<xmp:CreateDate>2004-10-12T10:29-07:00</xmp:CreateDate>
<xmp:CreatorTool>Flash Authoring WIN 8,0,0,215</xmp:CreatorTool>
</rdf:Description>
  <!-- This is after the second rdf:Description tag -->
</rdf:RDF>
```

Im dritten Beispiel befinden sich die Metadaten in Attributen und für alle Schemas gibt es ein gemeinsames `rdf:Description`-Tag:

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
<rdf:Description rdf:about='' xmlns:dc='http://purl.org/dc/1.1/' dc:title='Simple title'
dc:description='Simple description' />
<rdf:Description rdf:about='' xmlns:xmp='http://ns.adobe.com/xap/1.0/'
xmp:CreateDate='2004-10-12T10:29-07:00' xmp:CreatorTool='Flash Authoring WIN 8,0,0,215' />
</rdf:RDF>
```

Siehe auch

[document.getMetadata\(\)](#)

document.setMobileSettings()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`document.setMobileSettings(xmlString)`

Parameter

xmlString Ein String, der die XML-Einstellungen in einer mobilen FLA-Datei beschreibt.

Rückgabewerte

Der Wert `true`, wenn die Einstellungen erfolgreich festgelegt wurden; andernfalls `false`.

Beschreibung

Methode; legt den Wert eines XML-Einstellungsstrings in einer mobilen FLA-Datei fest. (Die meisten mobilen FLA-Dateien verfügen über einen XML-String, der die Einstellungen innerhalb des Dokuments beschreibt.)

Beispiel

Im folgenden Beispiel wird der String mit den XML-Einstellungen für eine mobile FLA-Datei festgelegt. Beachten Sie, dass das nachstehende Beispiel eine einzelne Codezeile darstellt.

```
fl.getDocumentDOM().setMobileSettings("<? xml version='1.0' encoding='UTF-16' standalone='no' ?> <mobileSettings> <contentType id='standalonePlayer' name='Standalone Player'/> <testDevices> <testDevice id='1170' name='Generic Phone' selected='yes'/> </testDevices> <outputMsgFiltering info='no' trace='yes' warning='yes'/> <testWindowState height='496' splitterClosed='No' splitterXPos='400' width='907'/> </mobileSettings>");
```

Siehe auch

[document.getMobileSettings\(\)](#)

document.setOvalObjectProperty()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
document.setOvalObjectProperty(propertyName, value)
```

Parameter

propertyName Ein String, der die festzulegende Eigenschaft angibt. Eine Liste der zulässigen Werte finden Sie in der Übersicht über die Eigenschaften für das [Oval-Objekt](#).

value Der Wert, der der angegebenen Eigenschaft zugewiesen werden soll. Die zulässigen Werte richten sich nach der Eigenschaft, die Sie in *propertyName* festgelegt haben.

Rückgabewerte

Keine.

Beschreibung

Methode; legt einen Wert für eine angegebene Eigenschaft von Ellipsengrundobjekten fest.

Beispiel

Beispiele finden Sie zu den einzelnen Eigenschaften unter [Oval-Objekt](#).

Siehe auch

[Oval-Objekt](#), [shape.isOvalObject](#)

document.setPlayerVersion()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
document.setPlayerVersion(version)
```

Parameter

version Ein String, der die von dem angegebenen Dokument vorgesehene Flash Player-Version darstellt. Die zulässigen Werte lauten "FlashLite", "FlashLite1", "FlashLite2", "FlashLite3", "1", "2", "3", "4", "5", "6", "7", "8", "9", "FlashPlayer10" und "AdobeAIR1_1".

Rückgabewerte

Der Wert `true`, wenn die Player-Version erfolgreich festgelegt wurde; andernfalls `false`.

Beschreibung

Methode; legt die Flash Player-Version fest, die für das angegebene Dokument vorgesehen ist. Dies ist derselbe Wert, der im Dialogfeld „Einstellungen für Veröffentlichungen“ festgelegt wurde.

Beispiel

Im folgenden Beispiel ist Flash Player 6 als Player-Version für das aktuelle Dokument vorgesehen:

```
fl.getDocumentDOM().setPlayerVersion("6");
```

Siehe auch

[document.getPlayerVersion\(\)](#)

document.setRectangleObjectProperty()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
document.setRectangleObjectProperty(propertyName, value)
```

Parameter

propertyName Ein String, der die festzulegende Eigenschaft angibt. Eine Liste der zulässigen Werte finden Sie in der Übersicht über die Eigenschaften für das [Rectangle-Objekt](#).

value Der Wert, der der angegebenen Eigenschaft zugewiesen werden soll. Die zulässigen Werte richten sich nach der Eigenschaft, die Sie in *propertyName* festgelegt haben.

Rückgabewerte

Keine.

Beschreibung

Methode; legt einen Wert für eine angegebene Eigenschaft von rechteckigen Grundobjekten fest.

Beispiel

Beispiele finden Sie zu den einzelnen Eigenschaften unter [Rectangle-Objekt](#).

Siehe auch

[Rectangle-Objekt](#), [shape.isRectangleObject](#)

document.setSelectionBounds()

Verfügbarkeit

Flash MX 2004; Parameter *bContactSensitiveSelection* ist neu in Flash 8.

Verwendung

```
document.setSelectionBounds(boundingBox [, bContactSensitiveSelection])
```

Parameter

boundingRectangle Ein Rechteck, das die neue Lage und Größe der Auswahl angibt. Informationen zum Format von *boundingRectangle* finden Sie unter [document.addNewRectangle\(\)](#).

bContactSensitiveSelection Ein boolescher Wert, der angibt, ob der Auswahlmodus „Kontaktempfindlich“ während der Objektauswahl aktiviert (*true*) oder deaktiviert (*false*) wird. Der Standardwert lautet *false*.

Rückgabewerte

Keine.

Beschreibung

Methode; verschiebt die Auswahl und ändert gleichzeitig ihre Größe.

Wenn Sie für *bContactSensitiveSelection* einen Wert angeben, so gilt dieser nur für diese Methode und hat keinen Einfluss auf den Auswahlmodus „Kontaktempfindlich“ des Dokuments (siehe [fl.contactSensitiveSelection](#)).

Beispiel

Im folgenden Beispiel wird die aktuelle Auswahl an die Position 10, 20 verschoben und ihre Größe auf 100, 200 geändert:

```
var l = 10;  
var t = 20;  
fl.getDocumentDOM().setSelectionBounds({left:l, top:t, right:(100+l), bottom:(200+t)});
```

Siehe auch

[document.selection](#), [document.setSelectionRect\(\)](#)

document.setSelectionRect()

Verfügbarkeit

Flash MX 2004; Parameter *bContactSensitiveSelection* ist neu in Flash 8.

Verwendung

```
document.setSelectionRect(rect [, bReplaceCurrentSelection [, bContactSensitiveSelection]])
```

Parameter

rect Ein Rechteckobjekt, das ausgewählt sein soll. Informationen zum Format von *rect* finden Sie unter [document.addNewRectangle\(\)](#).

bReplaceCurrentSelection Ein boolescher Wert, der angibt, ob die Methode die aktuelle Auswahl ersetzt (*true*) oder erweitert (*false*). Der Standardwert lautet *true*.

bContactSensitiveSelection Ein boolescher Wert, der angibt, ob der Auswahlmodus „Kontaktempfindlich“ während der Objektauswahl aktiviert (*true*) oder deaktiviert (*false*) wird. Der Standardwert lautet *false*.

Rückgabewerte

Keine.

Beschreibung

Methode; zeichnet anhand der angegebenen Koordinaten einen rechteckigen Auswahlrahmen in Relation zur Bühne. Im Gegensatz dazu wird das Rechteck bei `document.getSelectionRect()` in Relation zu dem bearbeiteten Objekt erstellt.

Das gleiche Ergebnis erzielen Sie, wenn Sie ein Rechteck mit dem Auswahlwerkzeug ziehen. Eine Instanz kann nur dann ausgewählt werden, wenn sie sich vollständig innerhalb des Rechtecks befindet.

Wenn Sie für *bContactSensitiveSelection* einen Wert angeben, so gilt dieser nur für diese Methode und hat keinen Einfluss auf den Auswahlmodus „Kontaktempfindlich“ des Dokuments (siehe [fl.contactSensitiveSelection](#)).

Hinweis: Wenn Sie `setSelectionRect()` mit dem Bedienfeld oder Menüelement „Protokoll“ wiederholen, wird der Schritt vor der `setSelectionRect()`-Operation wiederholt.

Beispiel

Im folgenden Beispiel ersetzt die zweite Auswahl die erste:

```
fl.getDocumentDOM().setSelectionRect({left:1, top:1, right:200, bottom:200});  
fl.getDocumentDOM().setSelectionRect({left:364.0, top:203.0, right:508.0, bottom:434.0},  
true);
```

Im folgenden Beispiel wird die zweite Auswahl der ersten Auswahl hinzugefügt. Das gleiche Ergebnis erzielen Sie, wenn Sie die Umschalttaste gedrückt halten und ein zweites Objekt auswählen.

```
fl.getDocumentDOM().setSelectionRect({left:1, top:1, right:200, bottom:200});  
fl.getDocumentDOM().setSelectionRect({left:364.0, top:203.0, right:508.0, bottom:434.0},  
false);
```

Siehe auch

[document.getSelectionRect\(\)](#), [document.selection](#), [document.setSelectionBounds\(\)](#)

document.setStageVanishingPoint()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
document.setStageVanishingPoint(point)
```

Parameter

point Ein Punkt, der anhand von *x*- und *y*-Koordinaten angibt, an welcher Stelle sich bei der Anzeige von 3D-Objekten der Fluchtpunkt befindet.

Rückgabewerte

Keine.

Beschreibung

Gibt den Fluchtpunkt bei der Anzeige von 3D-Objekten an.

Beispiel

Im folgenden Beispiel wird der Fluchtpunkt auf der Bühne festgelegt:

```
fl.getDocumentDOM().setStageVanishingPoint({x:45, y:45});
```

document.setStageViewAngle()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
document.setStageViewAngle(angle)
```

Parameter

angle Ein Gleitkommawert zwischen 0,0 und 179,0.

Rückgabewerte

Keine.

Beschreibung

Gibt den perspektivischen Winkel bei der Anzeige von 3D-Objekten an.

Beispiel

Im folgenden Beispiel wird auf der Bühne ein perspektivischer Winkel von 70 Grad festgelegt:

```
fl.getDocumentDOM().setStageViewAngle(70);
```

document.setStroke()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setStroke(color, size, strokeType)
```

Parameter

color Die Farbe des Strichs. Sie wird in einem der folgenden Formate angegeben:

- als String im Format "#RRGGBB" oder "#RRGGBBAA"
- als Hexadezimalzahl im Format "0xRRGGBB"
- als Ganzzahl, die die dezimale Entsprechung einer hexadezimalen Zahl darstellt

size Ein Gleitkommawert, der die neue Strichgröße für die Auswahl angibt.

strokeType Ein String, der den neuen Strichtyp für die Auswahl angibt. Die zulässigen Werte lauten "hairline", "solid", "dashed", "dotted", "ragged", "stipple" und "hatched".

Rückgabewerte

Keine.

Beschreibung

Methode; stellt die Farbe, die Breite und den Stil des ausgewählten Strichs ein. Weitere Informationen zum Ändern des Strichs im Bedienfeld „Werkzeuge“ und im Eigenschafteninspektor finden Sie unter [document.setCustomStroke\(\)](#).

Beispiel

Im folgenden Beispiel erhält der Strich die Farbe Rot, die Größe 3.25 und den Typ „dashed“:

```
fl.getDocumentDOM().setStroke("#ff0000", 3.25, "dashed");
```

document.setStrokeColor()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setStrokeColor(color)
```

Parameter

color Die Farbe des Strichs. Sie wird in einem der folgenden Formate angegeben:

- als String im Format "#RRGGBB" oder "#RRGGBBAA"
- als Hexadezimalzahl im Format "0xRRGGBB"
- als Ganzzahl, die die dezimale Entsprechung einer hexadezimalen Zahl darstellt

Rückgabewerte

Keine.

Beschreibung

Methode; ersetzt die Strichfarbe der Auswahl durch die angegebene Farbe. Weitere Informationen zum Ändern des Strichs im Bedienfeld „Werkzeuge“ und im Eigenschafteninspektor finden Sie unter [document.setStrokeColor\(\)](#).

Beispiel

In den drei Anweisungen des folgenden Beispiels wird die Strichfarbe jeweils anhand eines der drei verfügbaren Farbformate festgelegt:

```
fl.getDocumentDOM().setStrokeColor("#cc00cc");  
fl.getDocumentDOM().setStrokeColor(0xcc00cc);  
fl.getDocumentDOM().setStrokeColor(120000);
```

document.setStrokeSize()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setStrokeSize(size)
```

Parameter

size Ein Gleitkommawert von 0,25 bis 10, der die Strichgröße angibt. Es werden maximal zwei Dezimalstellen berücksichtigt.

Rückgabewerte

Keine.

Beschreibung

Methode; ändert die Strichgröße der Auswahl auf die angegebene Größe. Weitere Informationen zum Ändern des Strichs im Bedienfeld „Werkzeuge“ und im Eigenschafteninspektor finden Sie unter [document.setStrokeStyle\(\)](#).

Beispiel

Im folgenden Beispiel wird die Strichgröße der Auswahl auf 5 geändert:

```
fl.getDocumentDOM().setStrokeSize(5);
```

document.setStrokeStyle()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setStrokeStyle(strokeType)
```

Parameter

strokeType Ein String, der den Strichstil der aktuellen Auswahl angibt. Die zulässigen Werte lauten "hairline", "solid", "dashed", "dotted", "ragged", "stipple" und "hatched".

Rückgabewerte

Keine.

Beschreibung

Methode; ersetzt den Strichstil der Auswahl durch den angegebenen Stil. Weitere Informationen zum Ändern des Strichs im Bedienfeld „Werkzeuge“ und im Eigenschafteninspektor finden Sie unter [document.setCustomStroke\(\)](#).

Beispiel

Im folgenden Beispiel wird der Strichstil der Auswahl in "dashed" geändert:

```
fl.getDocumentDOM().setStrokeStyle("dashed");
```

document.setTextRectangle()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setTextRectangle(boundingRectangle)
```

Parameter

boundingRectangle Ein Rechteckobjekt für Text, das die neue Größe des Rechtecks angibt, in dem das Textelement sich befinden soll. Informationen zum Format von *boundingRectangle* finden Sie unter [document.addNewRectangle\(\)](#).

Rückgabewerte

Ein boolescher Wert; *true*, wenn die Größe von mindestens einem Textfeld geändert wird, ansonsten *false*.

Beschreibung

Methode; ändert das Begrenzungsrechteck des ausgewählten Textelements auf die angegebene Größe. Dadurch fließt der Text neu in das neue Rechteck ein. Das Textelement wird weder skaliert noch transformiert. Die von *boundingRectangle* übergebenen Werte werden wie folgt eingesetzt:

- Wenn der Text horizontal und statisch ist, berücksichtigt die Methode nur den an *boundingRectangle* übergebenen Breitenwert; die Höhe wird automatisch berechnet und an die Textmenge angepasst.
- Wenn der Text vertikal (und daher statisch) ist, berücksichtigt die Methode nur den an *boundingRectangle* übergebenen Höhenwert; die Breite wird automatisch berechnet und an die Textmenge angepasst.
- Bei dynamischem oder Eingabetext berücksichtigt die Methode sowohl den an *boundingRectangle* übergebenen Breitenwert als auch den Höhenwert. Das resultierende Rechteck ist möglicherweise größer als für die Textmenge erforderlich. Wenn durch die Parameter aber eine Rechteckgröße festgelegt wird, die für die Textmenge zu klein ist, berücksichtigt die Methode nur den an *boundingRectangle* übergebenen Breitenwert (der Höhenwert wird automatisch berechnet und an die Textmenge angepasst).

Beispiel

Im folgenden Beispiel wird die Größe des Begrenzungsrechtecks für Text auf die angegebenen Maße geändert:

```
fl.getDocumentDOM().setTextRectangle({left:0, top:0, right:50, bottom:200})
```

document.setTextSelection()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setTextSelection(startIndex, endIndex)
```

Parameter

startIndex Eine Ganzzahl, die die Position des ersten auszuwählenden Zeichens angibt. Das erste Zeichen befindet sich an der Position 0 (null).

endIndex Eine Ganzzahl, die die Position des letzten Zeichens der Auswahl angibt, wobei das Zeichen an der Position *endIndex* nicht mehr zur Auswahl gehört. Das erste Zeichen befindet sich an der Position 0 (null).

Rückgabewerte

Ein boolescher Wert; `true`, wenn der Text ausgewählt werden kann, ansonsten `false`.

Beschreibung

Methode; stellt die Textauswahl des gegenwärtig ausgewählten Textfelds auf die von den Werten *startIndex* und *endIndex* angegebenen Werte ein. Die Textbearbeitung wird aktiviert, sofern sie noch nicht aktiviert ist.

Beispiel

Im folgenden Beispiel wird der Textbereich ausgewählt, der vom sechsten bis zum fünfundzwanzigsten Zeichen reicht:

```
fl.document.setTextSelection(5, 25);
```

document.setTextString()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setTextString(text [, startIndex [, endIndex]])
```

Parameter

text Ein String mit den in das Textfeld einzufügenden Zeichen.

startIndex Eine Ganzzahl, die das letzte zu ersetzende Zeichen angibt. Das erste Zeichen befindet sich an der Position 0 (null). Dieser Parameter ist optional.

endIndex Eine Ganzzahl, die das letzte zu ersetzende Zeichen angibt. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn der Text von mindestens einem Textstring eingestellt wird, ansonsten `false`.

Beschreibung

Methode; fügt einen Textstring ein. Wenn die optionalen Parameter nicht übergeben werden, wird die vorhandene Textauswahl ersetzt; wenn das Textobjekt zurzeit nicht bearbeitet wird, wird der ganze Textstring ersetzt. Wenn nur *startIndex* übergeben wird, wird der übergebene String an dieser Position eingefügt. Wenn *startIndex* und *endIndex* übergeben werden, ersetzt der übergebene String den Textbereich, der von *startIndex* bis *endIndex* reicht, das Zeichen an der letzten Position aber nicht einschließt.

Beispiel

Im folgenden Beispiel wird die aktuelle Textauswahl durch „Hello World“ ersetzt:

```
var success = fl.getDocumentDOM().setTextString("Hello World!");
```

Im folgenden Beispiel wird an Position 6 der aktuellen Textauswahl das Wort „hello“ eingefügt:

```
var pass = fl.getDocumentDOM().setTextString("hello", 6);
```

Im folgenden Beispiel wird das Wort „Howdy“ in dem von Position 2 bis Position 7 reichenden Bereich eingefügt, wobei das Zeichen an der letzten Position unverändert bleibt:

```
var ok = fl.getDocumentDOM().setTextString("Howdy", 2, 7);
```

Siehe auch

[document.getTextString\(\)](#)

document.setTransformationPoint()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.setTransformationPoint( transformationPoint )
```

Parameter

transformationPoint Ein Punkt (zum Beispiel {*x*:10, *y*:20}, wobei *x* und *y* Gleitkommazahlen sind), der Werte für die Transformationspunkte der folgenden Elemente angibt:

- Formen: *transformationPoint* wird relativ zum Dokument festgelegt (0,0 ist die obere linke Ecke der Bühne).
- Symbole: *transformationPoint* wird relativ zum Registrierungspunkt des Symbols festgelegt (0,0 befindet sich am Registrierungspunkt).
- Text: *transformationPoint* wird relativ zum Textfeld festgelegt (0,0 ist die obere linke Ecke des Textfelds).
- Bitmaps/Videos: *transformationPoint* wird relativ zur Bitmap/zum Video festgelegt (0,0 ist die obere linke Ecke der Bitmap oder des Videos).
- Zeichenobjekte, Ellipsen- und Rechteckgrundformen sowie Gruppen: *transformationPoint* wird relativ zum Dokument festgelegt (0,0 ist die obere linke Ecke der Bühne). Um *transformationPoint* relativ zur Mitte des Objekts, der Grundform oder der Gruppe festzulegen, verwenden Sie () [element.setTransformationPoint\(\)](#).

Rückgabewerte

Keine.

Beschreibung

Methode; legt die Position des Transformationspunkts der aktuellen Auswahl fest.

Beispiel

Im folgenden Beispiel wird der Transformationspunkt der aktuellen Auswahl auf 100, 200 festgesetzt:

```
fl.getDocumentDOM().setTransformationPoint({x:100, y:200});
```

Siehe auch

`document.getTransformationPoint()`, `element.setTransformationPoint()`

document.silent

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.silent`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob das Objekt mit Eingabehilfen versehen ist. Dies entspricht der umgekehrten Logik der Einstellung „Film mit Eingabehilfen versehen“ im Bedienfeld „Eingabehilfen“. Wenn der Wert für `document.silent` `true` lautet, entspricht dies einer Deaktivierung der Option „Film mit Eingabehilfen versehen“. Lautet der Wert dagegen `false`, entspricht dies einer Aktivierung der Option „Film mit Eingabehilfen versehen“.

Beispiel

Im folgenden Beispiel wird der Variablen `isSilent` der Wert der Eigenschaft `silent` zugewiesen:

```
var isSilent = fl.getDocumentDOM().silent;
```

Im folgenden Beispiel wird der Eigenschaft `silent` der Wert `false` zugewiesen, was bedeutet, dass das Dokument mit Eingabehilfen versehen ist:

```
fl.getDocumentDOM().silent = false;
```

document.skewSelection()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.skewSelection(xSkew, ySkew [, whichEdge])`

Parameter

xSkew Eine Gleitkommazahl, die angibt, um welchen *x*-Betrag die Auswahl geneigt werden soll (gemessen in Grad).

ySkew Eine Gleitkommazahl, die angibt, um welchen *y*-Betrag die Auswahl geneigt werden soll (gemessen in Grad).

whichEdge Ein String, der angibt, an welcher Kante die Transformation stattfindet; fehlt diese Angabe, wird der Transformationspunkt verwendet. Die zulässigen Werte lauten "top center", "right center", "bottom center" und "left center". Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; neigt die Auswahl um einen angegebenen Betrag. Die gleiche Wirkung erzielen Sie, wenn Sie das Objekt mit dem Werkzeug „Frei transformieren“ neigen.

Beispiel

Im ersten Beispiel wird das ausgewählte Objekt vertikal um 2.0 und horizontal um 1.5 geneigt. Im zweiten Beispiel wird das Objekt an der Mitte der Oberkante transformiert.

```
fl.getDocumentDOM().skewSelection(2.0, 1.5);  
fl.getDocumentDOM().skewSelection(2.0, 1.5, "top center");
```

document.smoothSelection()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.smoothSelection()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; glättet die Kurve der ausgewählten Füllungskonturen oder gekrümmten Linien. Sie führt den gleichen Vorgang wie die Schaltfläche „Glätten“ im Bedienfeld „Werkzeuge“ aus.

Beispiel

Im folgenden Beispiel wird die Kurve der aktuellen Auswahl geglättet:

```
fl.getDocumentDOM().smoothSelection();
```

document.sourcePath

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`document.sourcePath`

Beschreibung

Eigenschaft; ein String, der eine Liste von Elementen enthält, die sich im ActionScript 3.0-Quellpfad des Dokuments befinden, und dieser Pfad wiederum gibt den Speicherort von Dateien der ActionScript-Klasse an. Elemente innerhalb des Strings sind durch Semikolons getrennt. Zur Angabe der Elemente wählen Sie im Authoring-Tool „Datei“ > „Einstellungen für Veröffentlichungen“ und wählen dann auf der Registerkarte „Flash“ die Option „ActionScript 3.0-Einstellungen“.

Beispiel

Das folgende Beispiel fügt den Ordner „./Class files“ dem Quellpfad des Dokuments hinzu:

```
var myDoc = fl.getDocumentDOM();  
fl.trace(myDoc.sourcePath);  
myDoc.sourcePath = "./Class files;" + myDoc.sourcePath;  
fl.trace(myDoc.sourcePath);
```

Siehe auch

[document.externalLibraryPath](#), [document.libraryPath](#), [fl.sourcePath](#)

document.space()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.space(direction [, bUseDocumentBounds])`

Parameter

direction Ein String, der die Richtung angibt, in der die ausgewählten Objekte in gleichmäßigen Abständen angeordnet werden sollen. Die zulässigen Werte lauten "horizontal" und "vertical".

bUseDocumentBounds Ein boolescher Wert; bei dem Wert `true` werden die Objekte anhand der Dokumentgrenzen in gleichmäßigen Abständen angeordnet. Andernfalls orientiert die Methode sich an den Grenzen der ausgewählten Objekte. Der Standardwert lautet `false`. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; ordnet die ausgewählten Objekte in gleichmäßigen Abständen an.

Beispiel

Im folgenden Beispiel werden die Objekte horizontal und in Relation zur Bühne in gleichmäßigen Abständen angeordnet:

```
fl.getDocumentDOM().space("horizontal",true);
```

Im folgenden Beispiel werden die Objekte horizontal und in Relation zueinander in gleichmäßigen Abständen angeordnet:

```
fl.getDocumentDOM().space("horizontal");
```

Im folgenden Beispiel werden die Objekte horizontal und in Relation zueinander in gleichmäßigen Abständen angeordnet, wobei *bUseDocumentBounds* ausdrücklich auf *false* eingestellt ist:

```
fl.getDocumentDOM().space("horizontal",false);
```

Siehe auch

`document.getAlignToDocument()`, `document.setAlignToDocument()`

document.straightenSelection()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.straightenSelection()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; begradigt die gegenwärtig ausgewählten Striche. Diese Methode entspricht der Schaltfläche „Begradigen“ im Bedienfeld „Werkzeuge“.

Beispiel

Im folgenden Beispiel wird die Kurve der aktuellen Auswahl begradigt:

```
fl.getDocumentDOM().straightenSelection();
```

document.swapElement()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.swapElement(name)`

Parameter

name Ein String, der den Namen des zu verwendenden Bibliothekselements angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; tauscht die aktuelle Auswahl gegen die angegebene Auswahl aus. Die Auswahl muss eine Grafik, eine Schaltfläche, einen Movieclip, ein Video oder eine Bitmap enthalten. Diese Methode zeigt eine Fehlermeldung an, wenn kein Objekt ausgewählt ist oder wenn das angegebene Objekt nicht gefunden wurde.

Beispiel

Im folgenden Beispiel wird die aktuelle Auswahl gegen `Symbol 1` aus der Bibliothek ausgetauscht:

```
fl.getDocumentDOM().swapElement('Symbol 1');
```

document.swapStrokeAndFill()

Verfügbarkeit

Flash 8.

Verwendung

`document.swapStrokeAndFill()`

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; vertauscht die Strich- und Füllfarben.

Beispiel

Im folgenden Beispiel werden die Strich- und Füllfarben im aktuellen Dokument vertauscht:

```
fl.getDocumentDOM().swapStrokeAndFill();
```

document.testMovie()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.testMovie([Boolean abortIfErrorsExist])
```

Parameter

abortIfErrorsExist Boolescher Wert; der Standardwert lautet „false“. Wenn dieser Wert auf „true“ eingestellt ist, wird die Sitzung zum Testen des Films nicht gestartet und das Fenster der .swf-Datei wird nicht geöffnet, wenn Kompilierfehler vorliegen. Durch Compiler-Warnmeldungen wird der Befehl nicht abgebrochen. Dieser Parameter wurde in Flash Professional CS5 eingeführt.

Rückgabewerte

Keine.

Beschreibung

Methode; führt im Dokument den Vorgang „Film testen“ aus.

Beispiel

Im folgenden Beispiel wird der Film für das aktuelle Dokument getestet, doch der Test wird abgebrochen, wenn Kompilierfehler vorliegen:

```
fl.getDocumentDOM().testMovie(1);
```

Siehe auch

[document.canTestMovie\(\)](#), [document.testScene\(\)](#)

document.testScene()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.testScene()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; führt für die aktuelle Szene des Dokuments den Vorgang „Szene testen“ aus.

Beispiel

Im folgenden Beispiel wird die aktuelle Szene im Dokument getestet:

```
fl.getDocumentDOM().testScene();
```

Siehe auch

`document.canTestScene()`, `document.testMovie()`

document.timelines

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.timelines`

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von Timeline-Objekten (siehe [Timeline-Objekt](#)).

Beispiel

Im folgenden Beispiel wird das Array der aktuellen Zeitleisten in dem aktiven Dokument abgerufen und die Namen werden im Bedienfeld „Ausgabe“ angezeigt:

```
var i = 0;
var curTimelines = fl.getDocumentDOM().timelines;
while(i < fl.getDocumentDOM().timelines.length) {
    alert(curTimelines[i].name);
    ++i;
}
```

Siehe auch

`document.currentTimeline`, `document.getTimeline()`

document.traceBitmap()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.traceBitmap(threshold, minimumArea, curveFit, cornerThreshold)`

Parameter

threshold Eine Ganzzahl, die die Anzahl der Farben in der nachgezeichneten Bitmap steuert. Es sind Ganzzahlen von 0 bis 500 zulässig.

minimumArea Eine Ganzzahl, die den Radius in Pixel angibt. Es sind Ganzzahlen von 1 bis 1000 zulässig.

curveFit Ein String, der angibt, wie glatt die gezeichneten Konturen sind. Die zulässigen Werte lauten "pixels", "very tight", "tight", "normal", "smooth" und "very smooth".

cornerThreshold Ein String, der *curveFit* ähnelt, sich aber auf die Ecken der Bitmap bezieht. Die zulässigen Werte lauten "many corners", "normal" und "few corners".

Rückgabewerte

Keine.

Beschreibung

Methode; führt für die aktuelle Auswahl den Vorgang „Bitmap nachzeichnen“ aus. Sie entspricht der Auswahl von „Modifizieren“ > „Bitmap“ > „Bitmap nachzeichnen“.

Beispiel

Im folgenden Beispiel wird die ausgewählte Bitmap anhand der angegebenen Parameter nachgezeichnet:

```
fl.getDocumentDOM().traceBitmap(0, 500, 'normal', 'normal');
```

document.translate3DCenter()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
document.translate3DCenter(xyzCoordinate)
```

Parameter

xyzCoordinate Ein XYZ-Koordinatenpunkt, der den Mittelpunkt für eine 3D-Drehung oder -Umsetzung angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; legt die XYZ-Position fest, um die herum die Auswahl umgesetzt oder gedreht wird. Diese Methode ist nur für MovieClips verfügbar.

Beispiel

Im folgenden Beispiel werden die XYZ-Achsen für die 3D-Umsetzung angegeben:

```
fl.getDocumentDOM().translate3DCenter({x:180, y:18,z:-30});
```

document.translate3DSelection()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
document.translate3DSelection(xyzCoordinate, bGlobalTransform)
```

Parameter

xyzCoordinate Eine XYZ-Koordinate, die die 3D-Umsetzungsachsen angibt.

bGlobalTransform Ein boolescher Wert, der angibt, ob es sich um den Transformationsmodus (`true`) oder nicht (`false`) handelt.

Rückgabewerte

Keine.

Beschreibung

Methode; wendet eine 3D-Umsetzung auf die Auswahl an. Diese Methode ist nur für MovieClips verfügbar.

Beispiel

Im folgenden Beispiel wird die Auswahl zuerst relativ zur Bühne umgesetzt (`global`) und dann relativ zu sich selbst (`lokal`).

```
var myDocument = fl.getDocumentDOM();  
myDocument.translate3DSelection({x:52.0, y:0, z:0}, true);  
myDocument.translate3DSelection({x:52.0, y:0, z:-55.2}, false);
```

Siehe auch

[document.translate3DCenter\(\)](#)

document.transformSelection()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.transformSelection(a, b, c, d)
```

Parameter

- a** Eine Gleitkommazahl, die das Element (0,0) der Transformationsmatrix angibt.
- b** Eine Gleitkommazahl, die das Element (0,1) der Transformationsmatrix angibt.
- c** Eine Gleitkommazahl, die das Element (1,0) der Transformationsmatrix angibt.
- d** Eine Gleitkommazahl, die das Element (1,1) der Transformationsmatrix angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; führt für die aktuelle Auswahl durch Anwenden der in den Argumenten angegebenen Matrix eine allgemeine Transformation durch. Weitere Informationen finden Sie im Abschnitt zur Eigenschaft [element.matrix](#).

Beispiel

Im folgenden Beispiel wird die Auswahl in der x-Achse um den Faktor 2 gedehnt:

```
fl.getDocumentDOM().transformSelection(2.0, 0.0, 0.0, 1.0);
```

document.unGroup()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.unGroup()`

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; hebt die Gruppierung der aktuellen Auswahl auf.

Beispiel

Im folgenden Beispiel wird die Gruppierung der Elemente in der aktuellen Auswahl aufgehoben:

```
fl.getDocumentDOM().unGroup();
```

Siehe auch

[document.group\(\)](#)

document.union()

Verfügbarkeit

Flash 8.

Verwendung

`document.union()`

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; `true`, wenn erfolgreich, ansonsten `false`.

Beschreibung

Methode; verbindet alle ausgewählten Formen zu einem Zeichnungsobjekt.

Beispiel

Im folgenden Beispiel werden alle ausgewählten Formen zu einem Zeichnungsobjekt verbunden:

```
fl.getDocumentDOM().union();
```

Siehe auch

`document.crop()`, `document.deleteEnvelope()`, `document.intersect()`, `document.punch()`,
`shape.isDrawingObject`

document.unlockAllElements()

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.unlockAllElements()`

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; hebt die Elementsperrung in dem gegenwärtig ausgewählten Bild auf.

Beispiel

Im folgenden Beispiel wird die Elementsperrung im gegenwärtig ausgewählten Bild aufgehoben:

```
fl.getDocumentDOM().unlockAllElements();
```

Siehe auch

`element.locked`

document.viewMatrix

Verfügbarkeit

Flash MX 2004.

Verwendung

`document.viewMatrix`

Beschreibung

Schreibgeschützte Eigenschaft; ein Matrix-Objekt. Mit `viewMatrix` wird der Objektraum in den Dokumentraum übertragen, wenn das Dokument sich im Bearbeitungsmodus befindet. Die Mausposition, die ein Werkzeug empfängt, wird in Relation zu dem gegenwärtig bearbeiteten Objekt angegeben. Siehe [Matrix-Objekt](#).

Wenn Sie beispielsweise ein Symbol erstellen, auf das Symbol doppelklicken, um es zu bearbeiten, und dann mit dem Polysternwerkzeug zeichnen, befindet sich der Punkt (0,0) am Registrierungspunkt des Symbols. Das `drawingLayer`-Objekt erwartet allerdings Werte im Dokumentraum; wenn Sie also eine Linie von (0,0) aus mithilfe von `drawingLayer` zeichnen, beginnt diese Linie in der oberen linken Ecke der Bühne. Mit der `viewMatrix`-Eigenschaft ist es möglich, den Raum des bearbeiteten Objekts in den Dokumentraum zu übertragen.

Beispiel

Im folgenden Beispiel wird der Wert der Eigenschaft `viewMatrix` abgerufen:

```
var mat = fl.getDocumentDOM().viewMatrix;
```

document.width

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.width
```

Beschreibung

Eigenschaft; eine Ganzzahl, die die Breite des Dokuments (Bühne) in Pixel angibt.

Beispiel

Im folgenden Beispiel wird die Breite der Bühne auf 400 Pixel festgesetzt:

```
fl.getDocumentDOM().width= 400;
```

Siehe auch

[document.height](#)

document.xmlPanel()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
document.xmlPanel (fileURI)
```

Parameter

fileURI Ein String im Format `file:/// URI`, der den Pfad zu der XML-Datei angibt, die die Steuerelemente in dem Bedienfeld definiert. Der Pfad muss vollständig angegeben werden.

Rückgabewerte

Ein Objekt, in dem die Eigenschaften aller in der XML-Datei aufgeführten Steuerelemente definiert sind. Alle Eigenschaften werden als Strings zurückgegeben. Das zurückgegebene Objekt hat eine vordefinierte Eigenschaft mit dem Namen "dismiss" mit dem Stringwert "accept" oder "cancel".

Beschreibung

Methode; stellt ein XMLUI-Dialogfeld bereit. Siehe [fl.xmlui](#).

Beispiel

Im folgenden Beispiel wird die Datei „Test.xml“ geladen und alle darin enthaltenen Eigenschaften werden angezeigt:

```
var obj = fl.getDocumentDOM().xmlPanel(fl.configURI + "Commands/Test.xml");
for (var prop in obj) {
    fl.trace("property " + prop + " = " + obj[prop]);
}
```

document.zoomFactor

Verfügbarkeit

Flash 8.

Verwendung

`document.zoomFactor`

Beschreibung

Eigenschaft; gibt den Zoomfaktor auf der Bühne während des Authorings an. Der Wert 1 entspricht dem Zoomfaktor 100 %, 8 entspricht 800 %, 0.5 entspricht 50 % usw.

Beispiel

Im folgenden Beispiel wird der Zoomfaktor auf der Bühne auf 200 % festgelegt:

```
fl.getDocumentDOM().zoomFactor = 2;
```

Kapitel 12: drawingLayer-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das drawingLayer-Objekt ist ein Child-Objekt des flash-Objekts. Der Zugriff auf dieses Objekt kann daher von JavaScript aus erfolgen. Das drawingLayer-Objekt wird für erweiterbare Werkzeuge verwendet, wenn der Benutzer während des Ziehens vorübergehend zeichnen möchte (zum Beispiel beim Erstellen eines Auswahlrahmens). Sie sollten `drawingLayer.beginFrame()` aufrufen, bevor Sie eine andere drawingLayer-Methode aufrufen.

Übersicht über Methoden

Für das drawingLayer-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>drawingLayer.beginDraw()</code>	Versetzt Flash in den Zeichenmodus.
<code>drawingLayer.beginFrame()</code>	Löscht die mit dem drawingLayer-Objekt zuvor erstellten Zeichnungen und wartet auf weitere Zeichenbefehle.
<code>drawingLayer.cubicCurveTo()</code>	Zeichnet von der aktuellen Stiftposition aus eine kubische Kurve, wobei die Parameter als Koordinaten des kubischen Segments dienen.
<code>drawingLayer.curveTo()</code>	Zeichnet ein quadratisches Kurvensegment, das an der aktuellen Zeichnungsposition beginnt und an dem angegebenen Punkt endet.
<code>drawingLayer.drawPath()</code>	Zeichnet den angegebenen Pfad.
<code>drawingLayer.endDraw()</code>	Beendet den Zeichenmodus.
<code>drawingLayer.endFrame()</code>	Signalisiert das Ende einer Zeichenbefehlsgruppe.
<code>drawingLayer.lineTo()</code>	Zeichnet eine Linie, die von der aktuellen Zeichnungsposition bis zum Punkt mit den Koordinaten (x,y) reicht.
<code>drawingLayer.moveTo()</code>	Legt die aktuelle Zeichnungsposition fest.
<code>drawingLayer.newPath()</code>	Gibt ein neues Path-Objekt zurück.
<code>drawingLayer.setColor()</code>	Legt die Farbe der nachfolgend gezeichneten Daten fest.
<code>drawingLayer.setFill()</code>	Diese Methode steht nicht zur Verfügung.
<code>drawingLayer.setStroke()</code>	Diese Methode steht nicht zur Verfügung.

drawingLayer.beginDraw()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
drawingLayer.beginDraw([persistentDraw])
```

Parameter

persistentDraw Ein boolescher Wert (optional). Bei dem Wert `true` bleibt die Zeichnung in dem letzten Bild auf der Bühne stehen, bis ein neuer Aufruf des Typs `beginDraw()` oder `beginFrame()` erfolgt. (In diesem Zusammenhang bezieht sich *frame* nicht auf Zeitleistenbilder, sondern auf den Zeitpunkt, an dem der Zeichnungsvorgang beginnt bzw. endet.) Wenn ein Benutzer beispielsweise ein Rechteck zeichnet, kann er während des Ziehens der Maus eine Vorschau der Formkontur anzeigen. Soll diese Vorschauform sichtbar bleiben, nachdem der Benutzer die Maustaste losgelassen hat, weisen Sie *persistentDraw* den Wert `true` zu.

Rückgabewerte

Keine.

Beschreibung

Methode; versetzt Flash in den Zeichenmodus. Der Zeichenmodus dient zum vorübergehenden Zeichnen, während die Maustaste gedrückt ist. Diese Methode wird in der Regel nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Im folgenden Beispiel wird Flash in den Zeichenmodus versetzt:

```
fl.drawingLayer.beginDraw();
```

drawingLayer.beginFrame()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
drawingLayer.beginFrame()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; löscht die mit dem drawingLayer-Objekt zuvor erstellten Zeichnungen und wartet auf weitere Zeichenbefehle. Sollte nach `drawingLayer.beginDraw()` aufgerufen werden. Alle zwischen `drawingLayer.beginFrame()` und `drawingLayer.endFrame()` erstellten Zeichnungen bleiben bis zum nächsten Aufruf von `beginFrame()` und `endFrame()` auf der Bühne stehen. (In diesem Zusammenhang bezieht sich *frame* nicht auf Zeitleistenbilder, sondern auf den Zeitpunkt, an dem der Zeichnungsvorgang beginnt bzw. endet.) Diese Methode wird in der Regel nur beim Erstellen erweiterbarer Werkzeuge eingesetzt. Siehe [drawingLayer.beginDraw\(\)](#).

drawingLayer.cubicCurveTo()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
drawingLayer.cubicCurveTo(x1Ctrl, y1Ctrl, x2Ctrl, y2Ctrl, xEnd, yEnd)
```

Parameter

x1Ctrl Ein Gleitkommawert, der die *x*-Koordinate des ersten Steuerpunkts angibt.

y1Ctrl Ein Gleitkommawert, der die *y*-Koordinate des ersten Steuerpunkts angibt.

x2Ctrl Ein Gleitkommawert, der die *x*-Koordinate des mittleren Steuerpunkts angibt.

y2Ctrl Ein Gleitkommawert, der die *y*-Koordinate des mittleren Steuerpunkts angibt.

xEnd Ein Gleitkommawert, der die *x*-Koordinate des letzten Steuerpunkts angibt.

yEnd Ein Gleitkommawert, der die *y*-Koordinate des letzten Steuerpunkts angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; zeichnet von der aktuellen Stiftposition aus eine kubische Kurve, wobei die Parameter als Koordinaten des kubischen Segments dienen. Diese Methode wird in der Regel nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Im folgenden Beispiel wird anhand der angegebenen Steuerpunkte eine kubische Kurve erstellt:

```
fl.drawingLayer.cubicCurveTo(0, 0, 1, 1, 2, 0);
```

drawingLayer.curveTo()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
drawingLayer.curveTo(xCtrl, yCtrl, xEnd, yEnd)
```

Parameter

xCtrl Ein Gleitkommawert, der die *x*-Koordinate des Steuerpunkts angibt.

yCtrl Ein Gleitkommawert, der die *y*-Koordinate des Steuerpunkts angibt.

xEnd Ein Gleitkommawert, der die *x*-Koordinate des letzten Steuerpunkts angibt.

yEnd Ein Gleitkommawert, der die *y*-Koordinate des letzten Steuerpunkts angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; zeichnet ein quadratisches Kurvensegment, das an der aktuellen Zeichnungsposition beginnt und an dem angegebenen Punkt endet. Diese Methode wird in der Regel nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Im folgenden Beispiel wird anhand der angegebenen Steuerpunkte eine quadratische Kurve erstellt:

```
fl.drawingLayer.curveTo(0, 0, 2, 0);
```

drawingLayer.drawPath()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
drawingLayer.drawPath(path)
```

Parameter

path Ein zu zeichnendes [Path-Objekt](#).

Rückgabewerte

Keine.

Beschreibung

Methode; zeichnet den von dem Parameter *path* angegebenen Pfad. Diese Methode wird in der Regel nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Im folgenden Beispiel wird ein von dem Path-Objekt `gamePath` näher bezeichneter Pfad gezeichnet:

```
fl.drawingLayer.drawPath(gamePath);
```

drawingLayer.endDraw()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
drawingLayer.endDraw()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; beendet den Zeichenmodus. Der Zeichenmodus dient zum vorübergehenden Zeichnen, während die Maustaste gedrückt ist. Diese Methode wird in der Regel nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Im folgenden Beispiel wird der Zeichenmodus beendet:

```
fl.drawingLayer.endDraw();
```

drawingLayer.endFrame()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
drawingLayer.endFrame()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; signalisiert das Ende einer Zeichenbefehlsgruppe. Eine Zeichenbefehlsgruppe bezieht sich auf alle Elemente, die zwischen `drawingLayer.beginFrame()` und `drawingLayer.endFrame()` erstellt werden. Beim nächsten Aufruf an `drawingLayer.beginFrame()` werden alle in dieser Zeichenbefehlsgruppe erstellten Elemente gelöscht. Diese Methode wird in der Regel nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

drawingLayer.lineTo()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
drawingLayer.lineTo(x, y)
```

Parameter

x Ein Gleitkommawert, der die *x*-Koordinate des Endpunkts der zu zeichnenden Linie angibt.

y Ein Gleitkommawert, der die *y*-Koordinate des Endpunkts der zu zeichnenden Linie angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; zeichnet eine Linie, die von der aktuellen Zeichnungsposition zu dem Punkt mit den Koordinaten (x,y) reicht. Diese Methode wird in der Regel nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Im folgenden Beispiel wird eine Linie gezeichnet, die von der aktuellen Zeichnungsposition zu dem Punkt mit den Koordinaten (20,30) reicht:

```
fl.drawingLayer.lineTo(20, 30);
```

drawingLayer.moveTo()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
drawingLayer.moveTo(x, y)
```

Parameter

x Ein Gleitkommawert, der die x-Koordinate der Stelle angibt, an der die Zeichnung beginnen soll.

y Ein Gleitkommawert, der die y-Koordinate der Stelle angibt, an der die Zeichnung beginnen soll.

Rückgabewerte

Keine.

Beschreibung

Methode; legt die aktuelle Zeichnungsposition fest. Diese Methode wird in der Regel nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Im folgenden Beispiel wird die aktuelle Zeichnungsposition auf den Punkt mit den Koordinaten (10,15) festgesetzt:

```
fl.drawingLayer.moveTo(10, 15);
```

drawingLayer.newPath()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
drawingLayer.newPath()
```

Parameter

Keine.

Rückgabewerte

Ein Path-Objekt.

Beschreibung

Methode; gibt ein neues Path-Objekt zurück. Diese Methode wird in der Regel nur beim Erstellen erweiterbarer Werkzeuge eingesetzt. Siehe [Path-Objekt](#).

Beispiel

Im folgenden Beispiel wird ein neues Path-Objekt zurückgegeben:

```
fl.drawingLayer.newPath();
```

drawingLayer.setColor()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
drawingLayer.setColor(color)
```

Parameter

color Die Farbe der nachfolgend gezeichneten Daten. Sie wird in einem der folgenden Formate angegeben:

- als String im Format "#RRGGBB" oder "#RRGGBBAA"
- als Hexadezimalzahl im Format "0xRRGGBB"
- als Ganzzahl, die die dezimale Entsprechung einer hexadezimalen Zahl darstellt

Rückgabewerte

Keine.

Beschreibung

Methode; legt die Farbe der nachfolgend gezeichneten Daten fest. Gilt nur für permanente Daten. Sie können diese Methode nur verwenden, wenn der an `drawingLayer.beginDraw()` übergebene Parameter den Wert `true` hat. Diese Methode wird in der Regel nur beim Erstellen erweiterbarer Werkzeuge eingesetzt. Siehe [drawingLayer.beginDraw\(\)](#).

Beispiel

Im folgenden Beispiel wird auf der Bühne eine rote Linie gezeichnet:

```
fl.drawingLayer.beginDraw( true );  
fl.drawingLayer.beginFrame();  
fl.drawingLayer.setColor( "#ff0000" );  
fl.drawingLayer.moveTo(0,0);  
fl.drawingLayer.lineTo(100,100);  
fl.drawingLayer.endFrame();  
fl.drawingLayer.endDraw();
```

drawingLayer.setFill()

Diese Methode steht nicht zur Verfügung.

drawingLayer.setStroke()

Diese Methode steht nicht zur Verfügung.

Kapitel 13: Edge-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Edge-Objekt stellt eine Kante einer Form auf der Bühne dar.

Übersicht über Methoden

Für das Edge-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>edge.getControl()</code>	Ruft ein Punktobjekt ab, das sich an der Position des angegebenen Steuerpunkts der Kante befindet.
<code>edge.getHalfEdge()</code>	Gibt ein HalfEdge-Objekt zurück.
<code>edge.setControl()</code>	Legt die Position des Steuerpunkts der Kante fest.
<code>edge.splitEdge()</code>	Zerlegt die Kante in zwei Teile.

Übersicht über Eigenschaften

Für das Edge-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>edge.cubicSegmentIndex</code>	Eine Ganzzahl, die den Indexwert eines kubischen Segments der Kante angibt.
<code>edge.id</code>	Schreibgeschützt; eine Ganzzahl, die einen eindeutigen Bezeichner für die Kante darstellt.
<code>edge.isLine</code>	Schreibgeschützt; eine Ganzzahl mit dem Wert 0 oder 1.
<code>edge.stroke</code>	Ein Stroke-Objekt .

edge.cubicSegmentIndex

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`edge.cubicSegmentIndex`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die den Indexwert eines kubischen Segments der Kante angibt (siehe [shape.getCubicSegmentPoints\(\)](#)).

Beispiel

Im folgenden Beispiel wird der Indexwert von allen kubischen Segmenten der angegebenen Kante angezeigt:

```
var theShape = fl.getDocumentDOM().selection[0];
var edgesArray = theShape.edges;
for (var i=0; i<edgesArray.length; i++) {
    fl.trace(edgesArray[i].cubicSegmentIndex);
}
```

edge.getControl()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
edge.getControl(i)
```

Parameter

i Eine Ganzzahl, die angibt, welcher Steuerpunkt der Kante zurückgegeben werden soll. Geben Sie 0 für den ersten, 1 für den mittleren oder 2 für den letzten Steuerpunkt an. Wenn die [edge.isLine](#)-Eigenschaft den Wert `true` hat, entspricht der mittlere Steuerpunkt dem Mittelpunkt des Segments, das den ersten und den letzten Steuerpunkt miteinander verbindet.

Rückgabewerte

Der angegebene Steuerpunkt.

Beschreibung

Methode; ruft ein Punktobjekt ab, das sich an der Position des angegebenen Steuerpunkts der Kante befindet.

Beispiel

Im folgenden Beispiel wird der erste Steuerpunkt der angegebenen Form in der Variablen `pt` gespeichert:

```
var shape = fl.getDocumentDOM().selection[0];
var pt = shape.edges[0].getControl(0);
```

edge.getHalfEdge()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
edge.getHalfEdge(index)
```

Parameter

index Eine Ganzzahl, die angibt, welche Halbkante zurückgegeben werden soll. Der Wert für *index* muss entweder 0 (erste Halbkante) oder 1 (zweite Halbkante) betragen.

Rückgabewerte

Ein `HalfEdge`-Objekt.

Beschreibung

Methode; gibt ein [HalfEdge-Objekt](#) zurück.

Beispiel

Im folgenden Beispiel werden die Halbkanten der angegebenen Kante in den Variablen `hEdge0` und `hEdge1` gespeichert:

```
var shape = fl.getDocumentDOM().selection[0];  
var edge = shape.edges[0];  
var hEdge0 = edge.getHalfEdge(0);  
var hEdge1 = edge.getHalfEdge(1);
```

edge.id

Verfügbarkeit

Flash MX 2004.

Verwendung

`edge.id`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die einen eindeutigen Bezeichner für die Kante darstellt.

Beispiel

Im folgenden Beispiel wird ein eindeutiger Bezeichner für die angegebene Kante in der Variablen `my_shape_id` gespeichert:

```
var shape = fl.getDocumentDOM().selection[0];  
var my_shape_id = shape.edges[0].id;
```

edge.isLine

Verfügbarkeit

Flash MX 2004.

Verwendung

`edge.isLine`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl mit dem Wert 0 oder 1. Der Wert 1 gibt an, dass es sich bei der Kante um eine gerade Linie handelt. In diesem Fall befindet sich der mittlere Steuerpunkt, der die beiden Endpunkte miteinander verbindet, genau in der Mitte der Linie.

Beispiel

Im folgenden Beispiel wird zunächst festgestellt, ob es sich bei der angegebenen Kante um eine gerade Linie handelt. Dann wird im Bedienfeld Ausgabe entweder der Wert 1 (gerade Linie) oder 0 (keine gerade Linie) angezeigt:


```
var shape = fl.getDocumentDOM().selection[0];  
fl.trace(shape.edges[0].isLine);
```

edge.setControl()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
edge.setControl(index, x, y)
```

Parameter

index Eine Ganzzahl, die angibt, welcher Steuerpunkt festgelegt werden soll. Der Wert 0 steht für den ersten, der Wert 1 für den mittleren und der Wert 2 für den letzten Steuerpunkt.

x Ein Gleitkommawert, der die horizontale Position des Steuerpunkts angibt. Wenn die Bühne sich im Modus „Bearbeiten“ oder „An Position bearbeiten“ befindet, wird die Punktkoordinate in Relation zu dem bearbeiteten Objekt angegeben, und andernfalls in Relation zur Bühne.

y Ein Gleitkommawert, der die vertikale Position des Steuerpunkts angibt. Wenn die Bühne sich im Modus „Bearbeiten“ oder „An Position bearbeiten“ befindet, wird die Punktkoordinate in Relation zu dem bearbeiteten Objekt angegeben, und andernfalls in Relation zur Bühne.

Rückgabewerte

Keine.

Beschreibung

Methode; legt die Position des Steuerpunkts der Kante fest. Sie müssen `shape.beginEdit()` aufrufen, bevor Sie diese Methode verwenden. Siehe [shape.beginEdit\(\)](#).

Beispiel

Im folgenden Beispiel wird der erste Steuerpunkt der angegebenen Kante an die Koordinaten (0, 1) gesetzt:

```
x = 0; y = 1;  
var shape = fl.getDocumentDOM().selection[0];  
shape.beginEdit();  
shape.edges[0].setControl(0, x, y);  
shape.endEdit();
```

edge.splitEdge()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
edge.splitEdge(t)
```

Parameter

t Ein Gleitkommawert zwischen 0 und 1, der angibt, wo die Kante geteilt werden soll. Der Wert 0 steht für den einen Endpunkt und der Wert 1 für den anderen. Wenn beispielsweise der Wert 0.5 übergeben wird, wird die Kante in der Mitte geteilt. Bei einer Linie befindet sich die Mitte genau im Mittelpunkt. Handelt es sich bei der Kante um eine Kurve, steht 0.5 für die parametrische Mitte der Kurve.

Rückgabewerte

Keine.

Beschreibung

Methode; zerlegt die Kante in zwei Teile. Sie müssen `shape.beginEdit()` aufrufen, bevor Sie diese Methode verwenden.

Beispiel

Im folgenden Beispiel wird die angegebene Kante zweigeteilt:

```
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit()
shape.edges[0].splitEdge( 0.5 );
shape.endEdit()
```

edge.stroke

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`edge.stroke`

Beschreibung

Eigenschaft; ein [Stroke-Objekt](#).

Beispiel

Im folgenden Beispiel wird die Strichfarbe der ersten Kante des ausgewählten Objekts angezeigt:

```
var shape = fl.getDocumentDOM().selection[0];
fl.trace(shape.edges[0].stroke.color);
```

Kapitel 14: Element-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Alles, was Sie auf der Bühne sehen, sind Element-Objekte. Mit dem folgenden Code können Sie ein Element auswählen:

```
var el = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];
```

Übersicht über Methoden

Für das Element-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>element.getPersistentData()</code>	Ruft den Wert der mit dem Parameter <i>name</i> bezeichneten Daten ab.
<code>element.getTransformationPoint()</code>	Ruft den Wert des Transformationspunkts des angegebenen Elements ab.
<code>element.hasPersistentData()</code>	Ermittelt, ob die angegebenen Daten mit dem angegebenen Element verbunden wurden.
<code>element.removePersistentData()</code>	Entfernt alle permanenten Daten mit dem angegebenen Namen, die mit dem Objekt verbunden wurden.
<code>element.setPersistentData()</code>	Speichert Daten mit einem Element.
<code>element.setTransformationPoint()</code>	Legt die Position des Transformationspunkts des Elements fest.

Übersicht über Eigenschaften

Für das Element-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>element.depth</code>	Schreibgeschützt; eine Ganzzahl mit einem Wert, der größer als 0 ist und die Tiefe des angezeigten Objekts angibt.
<code>element.elementType</code>	Schreibgeschützt; ein String, der den Typ des angegebenen Elements darstellt.
<code>element.height</code>	Ein Gleitkommawert, der die Höhe des Elements in Pixel angibt.
<code>element.layer</code>	Schreibgeschützt; stellt das Layer-Objekt dar, in dem sich das Element befindet.
<code>element.left</code>	Schreibgeschützt; ein Gleitkommawert, der die linke Elementseite darstellt.
<code>element.locked</code>	Der boolesche Wert <i>true</i> , wenn das Element gesperrt ist, ansonsten <i>false</i> .
<code>element.matrix</code>	Ein Matrix-Objekt . Die Matrix hat die Eigenschaften <i>a</i> , <i>b</i> , <i>c</i> , <i>d</i> , <i>tx</i> und <i>ty</i> . <i>a</i> , <i>b</i> , <i>c</i> und <i>d</i> sind Gleitkommawerte; <i>tx</i> und <i>ty</i> sind Koordinaten.
<code>element.name</code>	Ein String, der den Namen des Elements angibt. Man bezeichnet dies auch als Instanznamen.
<code>element.rotation</code>	Eine Ganzzahl oder eine Gleitkommazahl zwischen -180 und 180, die in Grad die Drehung des Objekts im Uhrzeigersinn angibt.

Eigenschaft	Beschreibung
<code>element.scaleX</code>	Ein Gleitkommawert, der den x-Wert für die Skalierung von Symbolen, Zeichenobjekten und Rechteck- und Ellipsengrundformen angibt.
<code>element.scaleY</code>	Ein Gleitkommawert, der den y-Wert für die Skalierung von Symbolen, Zeichenobjekten und Rechteck- und Ellipsengrundformen angibt.
<code>element.selected</code>	Ein boolescher Wert, der angibt, ob das Element ausgewählt ist.
<code>element.skewX</code>	Ein Gleitkommawert zwischen -180 und 180, der den x-Wert für die Neigung von Symbolen, Zeichenobjekten und Rechteck- und Ellipsengrundformen angibt.
<code>element.skewY</code>	Ein Gleitkommawert zwischen -180 und 180, der den y-Wert für die Neigung von Symbolen, Zeichenobjekten und Rechteck- und Ellipsengrundformen angibt.
<code>element.top</code>	Schreibgeschützt; die obere Elementseite.
<code>element.transformX</code>	Eine Gleitkommazahl, die den x-Wert des Transformationspunkts des ausgewählten Elements innerhalb des Koordinatensystems des übergeordneten Elements angibt.
<code>element.transformY</code>	Eine Gleitkommazahl, die den y-Wert des Transformationspunkts des ausgewählten Elements innerhalb des Koordinatensystems des übergeordneten Elements angibt.
<code>element.width</code>	Ein Gleitkommawert, der die Breite des Elements in Pixel angibt.
<code>element.x</code>	Ein Gleitkommawert, der den x-Wert des Registrierungspunkts des ausgewählten Elements angibt.
<code>element.y</code>	Ein Gleitkommawert, der den y-Wert des Registrierungspunkts des ausgewählten Elements angibt.

element.depth

Verfügbarkeit

Flash MX 2004.

Verwendung

`element.depth`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die größer als 0 ist und die Tiefe des angezeigten Objekts angibt. Die Reihenfolge, in der Objekte auf der Bühne gezeichnet werden, bestimmt die Reihenfolge, in der die Objekte übereinander gestapelt werden. Die Objektanordnung kann auch über den Befehl „Modifizieren“ > „Anordnen“ gesteuert werden.

Beispiel

Im folgenden Beispiel wird die Tiefe des angegebenen Elements im Bedienfeld „Ausgabe“ angezeigt:

```
// Select an object and run this script.  
fl.trace("Depth of selected object: " + fl.getDocumentDOM().selection[0].depth);
```

Siehe Beispiel zu [element.elementType](#).

element.elementType

Verfügbarkeit

Flash MX 2004.

Verwendung

`element.elementType`

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der den Typ des angegebenen Elements darstellt. Folgende Werte sind möglich: "shape", "text", "instance" oder "shapeObj". Ein "shapeObj" wird mit einem erweiterbaren Werkzeug erstellt.

Beispiel

Im folgenden Beispiel wird der Typ des ersten Elements in der Variablen `eType` gespeichert:

```
// In a new file, place a movie clip on first frame top layer, and
// then run this line of script.
var eType = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].elementType; //
eType = instance
```

Im folgenden Beispiel werden für sämtliche Elemente in der aktuellen Ebene bzw. im aktuellen Bild mehrere Eigenschaften angezeigt:

```
var tl = fl.getDocumentDOM().getTimeline()
var elts = tl.layers[tl.currentLayer].frames[tl.currentFrame].elements;
for (var x = 0; x < elts.length; x++) {
    var elt = elts[x];
    fl.trace("Element "+ x +" Name = " + elt.name + " Type = " + elt.elementType + " location
= " + elt.left + ", " + elt.top + " Depth = " + elt.depth);
}
```

element.getPersistentData()

Verfügbarkeit

Flash MX 2004.

Verwendung

`element.getPersistentData(name)`

Parameter

name Ein String, der die zurückzugebenden Daten identifiziert.

Rückgabewerte

Die mit dem Parameter *name* bezeichneten Daten bzw. 0, wenn die Daten nicht vorhanden sind.

Beschreibung

Methode; ruft den Wert der mit dem Parameter *name* bezeichneten Daten ab. Der Typ der Daten hängt von dem Typ der gespeicherten Daten ab (siehe [element.setPersistentData\(\)](#)). Nur Symbole und Bitmaps unterstützen permanente Daten.

Beispiel

Im folgenden Beispiel werden zuerst Daten für das angegebene Element festgelegt und abgerufen. Der Wert der Daten wird dann im Bedienfeld „Ausgabe“ angezeigt und zum Schluss werden die Daten entfernt:

```
// At least one symbol or bitmap is selected in the first layer, first frame.
var elt = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];
elt.setPersistentData("myData","integer", 12);
if (elt.hasPersistentData("myData")){
    fl.trace("myData = "+ elt.getPersistentData("myData"));
    elt.removePersistentData( "myData" );
    fl.trace("myData = "+ elt.getPersistentData("myData"));
}
```

element.getTransformationPoint()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
element.getTransformationPoint()
```

Parameter

Keine.

Rückgabewerte

Ein Punkt (zum Beispiel {x:10, y:20}, wobei x und y Gleitkommazahlen sind), der die Position des Transformationspunkts (auch *Bezugspunkt* oder *Nullpunkt* genannt) innerhalb des Koordinatensystems des Elements angibt.

Beschreibung

Methode; ruft den Wert des Transformationspunkts des angegebenen Elements ab.

Transformationspunkte stehen je nach Typ des ausgewählten Elements in Relation zu verschiedenen Positionen. Weitere Informationen finden Sie im Abschnitt zu [element.setTransformationPoint\(\)](#).

Beispiel

Im folgenden Beispiel wird der Transformationspunkt für das dritte Element im neunten Bild der ersten Ebene des Dokuments abgerufen. Die Eigenschaft `transPoint.x` gibt die x-Koordinate des Transformationspunkts an und die Eigenschaft `transPoint.y` die y-Koordinate.

```
var transPoint =
fl.getDocumentDOM().getTimeline().layers[0].frames[8].elements[2].getTransformationPoint();
```

Siehe auch

`document.getTransformationPoint()`, `element.setTransformationPoint()`, `element.transformX`,
`element.transformY`

element.hasPersistentData()

Verfügbarkeit

Flash MX 2004.

Verwendung

`element.hasPersistentData(name)`

Parameter

name Ein String, der den Namen des zu testenden Datenelements angibt.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die angegebenen Daten mit dem Objekt verbunden sind, ansonsten `false`.

Beschreibung

Methode; ermittelt, ob die angegebenen Daten mit dem angegebenen Element verbunden sind. Nur Symbole und Bitmaps unterstützen permanente Daten.

Beispiel

Siehe `element.getPersistentData()`.

element.height

Verfügbarkeit

Flash MX 2004.

Verwendung

`element.height`

Beschreibung

Eigenschaft; ein Gleitkommawert, der die Höhe des Elements in Pixel angibt.

Verwenden Sie diese Eigenschaft nicht, um die Größe eines Textfelds zu ändern. Wählen Sie stattdessen das Textfeld aus und verwenden Sie `document.setTextRectangle()`. Diese Eigenschaft bewirkt in einem Textfeld eine Skalierung des Textes.

Beispiel

Im folgenden Beispiel wird die Höhe des angegebenen Elements auf 100 festgesetzt:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].height = 100;
```

element.layer

Verfügbarkeit

Flash 8.

Verwendung

`element.layer`

Beschreibung

Schreibgeschützte Eigenschaft; stellt das [Layer-Objekt](#) dar, in dem sich das Element befindet.

Beispiel

Im folgenden Beispiel wird das Layer-Objekt, das das Element enthält, in der Variablen `theLayer` gespeichert:

```
var theLayer = element.layer;
```

element.left

Verfügbarkeit

Flash MX 2004.

Verwendung

`element.left`

Beschreibung

Schreibgeschützte Eigenschaft; ein Gleitkommawert, der die linke Elementseite darstellt. Der Wert für `element.left` wird für Elemente, die sich in einer Szene befinden, in Relation zur linken oberen Ecke der Bühne angegeben, und für Elemente, die in einem Symbol gespeichert sind, in Relation zum Registrierungspunkt (auch *Bezugspunkt* oder *Nullpunkt* genannt) des Symbols. Stellen Sie diese Eigenschaft mit `document.setSelectionBounds()` oder `document.moveSelectionBy()` ein.

Beispiel

Im folgenden Beispiel wird veranschaulicht, wie sich der Wert dieser Eigenschaft ändert, wenn ein Element verschoben wird:

```
// Select an element on the Stage and then run this script.  
var sel = fl.getDocumentDOM().selection[0];  
fl.trace("Left (before) = " + sel.left);  
fl.getDocumentDOM().moveSelectionBy({x:100, y:0});  
fl.trace("Left (after) = " + sel.left);
```

Siehe Beispiel zu [element.elementType](#).

element.locked

Verfügbarkeit

Flash MX 2004.

Verwendung

`element.locked`

Beschreibung

Eigenschaft; der boolesche Wert `true`, wenn das Element gesperrt ist, ansonsten `false`. Wenn der Wert für `element.elementType` "shape" ist, wird diese Eigenschaft ignoriert.

Beispiel

Im folgenden Beispiel wird das erste Element im ersten Bild in der obersten Ebene gesperrt:

```
// Similar to Modify > Arrange > Lock:  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].locked = true;
```

element.matrix

Verfügbarkeit

Flash MX 2004.

Verwendung

`element.matrix`

Beschreibung

Eigenschaft; ein Matrix-Objekt. Eine Matrix hat die Eigenschaften `a`, `b`, `c`, `d`, `tx` und `ty`. Die Eigenschaften `a`, `b`, `c` und `d` sind Gleitkommawerte. `tx` und `ty` sind Koordinaten. Siehe [Matrix-Objekt](#).

Beispiel

Im folgenden Beispiel wird das angegebene Element um 10 Pixel in der *x*-Achse und um 20 Pixel in der *y*-Achse verschoben:

```
var mat = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].matrix;  
mat.tx += 10;  
mat.ty += 20;  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].matrix = mat;
```

element.name

Verfügbarkeit

Flash MX 2004.

Verwendung

`element.name`

Beschreibung

Eigenschaft; ein String, der den Namen des Elements angibt. Man bezeichnet dies auch als Instanznamen. Wenn der Wert für `element.elementType` "shape" ist, wird diese Eigenschaft ignoriert. Siehe [element.elementType](#).

Beispiel

Im folgenden Beispiel wird der Instanzname des ersten Elements im ersten Bild in der obersten Ebene auf "clip_mc" festgelegt:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].name = "clip_mc";
```

Siehe Beispiel zu [element.elementType](#).

element.removePersistentData()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
element.removePersistentData(name)
```

Parameter

name Ein String, der den Namen der zu entfernenden Daten angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; entfernt alle permanenten Daten mit dem angegebenen Namen, die mit dem Objekt verbunden sind. Nur Symbole und Bitmaps unterstützen permanente Daten.

Beispiel

Siehe [element.getPersistentData\(\)](#).

element.rotation

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
element.rotation
```

Beschreibung

Eigenschaft; eine Ganzzahl oder eine Gleitkommazahl zwischen -180 und 180, die in Grad die Drehung des Objekts im Uhrzeigersinn angibt.

Beispiel

Im folgenden Beispiel wird die Drehung des zurzeit ausgewählten Elements auf 45 Grad festgelegt:

```
var element = fl.getDocumentDOM().selection[0];
fl.trace("Element rotation = " + element.rotation);
element.rotation = 45;
fl.trace("After setting rotation to 45: rotation = " + element.rotation);
```

element.scaleX

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`element.scaleX`

Beschreibung

Eigenschaft; ein Gleitkommawert, der den x -Wert für die Skalierung von Symbolen, Zeichenobjekten und Rechteck- und Ellipsengrundformen angibt. Der Wert 1 zeigt eine Skalierung von 100 Prozent an.

Beispiel

Im folgenden Beispiel wird der x -Wert der Skalierung für die aktuelle Auswahl auf 2 gesetzt, d. h. der Wert wird verdoppelt:

```
var element = fl.getDocumentDOM().selection[0];
element.scaleX = 2;
```

Siehe auch

[element.scaleY](#)

element.scaleY

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`element.scaleY`

Beschreibung

Eigenschaft; ein Gleitkommawert, der den y -Wert für die Skalierung von Symbolen, Zeichenobjekten und Rechteck- und Ellipsengrundformen angibt. Der Wert 1 zeigt eine Skalierung von 100 Prozent an.

Beispiel

Im folgenden Beispiel wird der y -Wert der Skalierung für die aktuelle Auswahl auf 2 gesetzt, d. h. der Wert wird verdoppelt:

```
var element = fl.getDocumentDOM().selection[0];  
element.scaleY = 2;
```

Siehe auch

[element.scaleX](#)

element.selected

Verfügbarkeit

Flash 8.

Verwendung

```
element.selected
```

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob das Element ausgewählt (`true`) oder nicht ausgewählt (`false`) ist.

Beispiel

Im folgenden Beispiel wird das Element ausgewählt:

```
element.selected = true;
```

element.setPersistentData()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
element.setPersistentData(name, type, value)
```

Parameter

name Ein String mit dem Namen, der mit den Daten verbunden werden soll. Die Daten werden anhand dieses Namens abgerufen.

type Ein String, der den Datentyp definiert. Die zulässigen Werte lauten "integer", "integerArray", "double", "doubleArray", "string" und "byteArray".

value Gibt den Wert an, der dem Objekt zugewiesen werden soll. Der Datentyp von *value* hängt von dem Wert des Parameters *type* ab. Der angegebene Wert sollte für den von dem Parameter *type* bezeichneten Datentyp angemessen sein.

Rückgabewerte

Keine.

Beschreibung

Methode; speichert Daten mit einem Element. Die Daten sind verfügbar, wenn die FLA-Datei, die das Element enthält, erneut geöffnet wird. Nur Symbole und Bitmaps unterstützen permanente Daten.

Beispiel

Siehe `element.getPersistentData()`.

element.setTransformationPoint()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
element.setTransformationPoint(transformationPoint)
```

Parameter

transformationPoint Ein Punkt (zum Beispiel {x:10, y:20}, wobei x und y Gleitkommazahlen sind), der Werte für den Transformationspunkt eines Elements oder einer Gruppe angibt:

- Formen: *transformationPoint* wird relativ zum Dokument festgelegt (0,0 ist die obere linke Ecke der Bühne).
- Symbole: *transformationPoint* wird relativ zum Registrierungspunkt des Symbols festgelegt (0,0 befindet sich am Registrierungspunkt).
- Text: *transformationPoint* wird relativ zum Textfeld festgelegt (0,0 ist die obere linke Ecke des Textfelds).
- Bitmaps/Videos: *transformationPoint* wird relativ zur Bitmap/zum Video festgelegt (0,0 ist die obere linke Ecke der Bitmap oder des Videos).
- Zeichenobjekte, Objekt-Grundformen sowie Gruppen: *transformationPoint* wird relativ zur Mitte des Elements oder der Gruppe festgelegt (0,0 ist der Mittelpunkt des Elements oder der Gruppe).

Rückgabewerte

Keine.

Beschreibung

Methode; legt die Position des Transformationspunkts des Elements fest.

Diese Methode ist fast identisch mit `document.setTransformationPoint()`. Der Unterschied besteht in Folgendem:

- Der Transformationspunkt für Zeichenobjekte, Objekt-Grundformen und Gruppen wird in Relation zum Mittelpunkt des Elements oder der Gruppe festgelegt, nicht in Relation zur Bühne.
- Sie können Transformationspunkte für Elemente festlegen, ohne sie zunächst auszuwählen.

Mit dieser Methode wird der Transformationspunkt, nicht jedoch das Element verschoben. Im Gegensatz dazu verschieben die Eigenschaften `element.transformX` und `element.transformY` das Element.

Beispiel

Im folgenden Beispiel wird der Transformationspunkt des dritten Elements auf der Bühne auf 100, 200 festgesetzt:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[2].setTransformationPoint({x:100, y:200});
```

Siehe auch

`document.setTransformationPoint()`, `element.getTransformationPoint()`, `element.transformX`, `element.transformY`

element.skewX

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`element.skewX`

Beschreibung

Eigenschaft; ein Gleitkommawert zwischen -180 und 180, der den *x*-Wert für die Neigung von Symbolen, Zeichenobjekten und Rechteck- und Ellipsengrundformen angibt.

Beispiel

Im folgenden Beispiel wird der *x*-Wert der Neigung für die aktuelle Auswahl auf 10 gesetzt:

```
var element = fl.getDocumentDOM().selection[0];  
element.skewX = 10;
```

Siehe auch

`document.setTransformationPoint()`, `element.skewY`

element.skewY

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`element.skewY`

Beschreibung

Eigenschaft; ein Gleitkommawert zwischen -180 und 180, der den *y*-Wert für die Neigung von Symbolen, Zeichenobjekten und Rechteck- und Ellipsengrundformen angibt.

Beispiel

Im folgenden Beispiel wird der *y*-Wert der Neigung für die aktuelle Auswahl auf 10 gesetzt:

```
var element = fl.getDocumentDOM().selection[0];  
element.skewY = 10;
```

Siehe auch

[document.setTransformationPoint\(\)](#), [element.skewX](#)

element.top

Verfügbarkeit

Flash MX 2004.

Verwendung

`element.top`

Beschreibung

Schreibgeschützte Eigenschaft; die obere Elementseite. Der Wert für `element.top` wird für Elemente, die sich in einer Szene befinden, in Relation zur linken oberen Ecke der Bühne angegeben, und für Elemente, die in einem Symbol gespeichert sind, in Relation zum Registrierungspunkt des Symbols. Stellen Sie diese Eigenschaft mit [document.setSelectionBounds\(\)](#) oder [document.moveSelectionBy\(\)](#) ein.

Beispiel

Im folgenden Beispiel wird veranschaulicht, wie sich der Wert dieser Eigenschaft ändert, wenn ein Element verschoben wird:

```
// Select an element on the Stage and then run this script.  
var sel = fl.getDocumentDOM().selection[0];  
fl.trace("Top (before) = " + sel.top);  
fl.getDocumentDOM().moveSelectionBy({x:0, y:100});  
fl.trace("Top (after) = " + sel.top);
```

Siehe Beispiel zu [element.elementType](#).

element.transformX

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`element.transformX`

Beschreibung

Eigenschaft; eine Gleitkommazahl, die den *x*-Wert des Transformationspunkts des ausgewählten Elements innerhalb des Koordinatensystems des übergeordneten Elements angibt. Wenn diese Eigenschaft auf einen neuen Wert gesetzt wird, wird das Element verschoben. Im Gegensatz dazu verschiebt die Methode [element.setTransformationPoint\(\)](#) den Transformationspunkt, nicht jedoch das Element.

Beispiel

Siehe auch

`element.getTransformationPoint()`, `element.setTransformationPoint()`, `element.transformY`

element.transformY

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`element.transformY`

Beschreibung

Eigenschaft; eine Gleitkommazahl, die den *y*-Wert des Transformationspunkts des ausgewählten Elements innerhalb des Koordinatensystems des übergeordneten Elements angibt. Wenn diese Eigenschaft auf einen neuen Wert gesetzt wird, wird das Element verschoben. Im Gegensatz dazu verschiebt die Methode

`element.setTransformationPoint()` den Transformationspunkt, nicht jedoch das Element.

Siehe auch

`element.getTransformationPoint()`, `element.setTransformationPoint()`, `element.transformX`

element.width

Verfügbarkeit

Flash MX 2004.

Verwendung

`element.width`

Beschreibung

Eigenschaft; ein Gleitkommawert, der die Breite des Elements in Pixel angibt.

Verwenden Sie diese Eigenschaft nicht, um die Größe eines Textfelds zu ändern. Wählen Sie stattdessen das Textfeld aus und verwenden Sie `document.setTextRectangle()`. Diese Eigenschaft bewirkt in einem Textfeld eine Skalierung des Textes.

Beispiel

Im folgenden Beispiel wird die Breite des angegebenen Elements auf 100 festgesetzt:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].width= 100;
```


element.x

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`element.x`

Beschreibung

Eigenschaft; ein Gleitkommawert, der den *x*-Wert des Registrierungspunkts des ausgewählten Elements angibt.

Beispiel

Im folgenden Beispiel wird der Wert des Registrierungspunkts des angegebenen Elements auf 100, 200 gesetzt:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].x= 100;  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].y= 200;
```

Siehe auch

[element.y](#)

element.y

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`element.y`

Beschreibung

Eigenschaft; ein Gleitkommawert, der den *y*-Wert des Registrierungspunkts des ausgewählten Elements angibt.

Beispiel

Siehe [element.x](#).

Kapitel 15: Fill-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Dieses Objekt enthält alle Eigenschaften der Einstellung „Füllfarbe“ des Bedienfelds „Werkzeuge“ bzw. einer ausgewählten Form. Fill-Objekte können mit `document.getCustomFill()` abgerufen werden.

Übersicht über Eigenschaften

Für das Fill-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>fill.bitmapIsClipped</code>	Ein boolescher Wert, der angibt, ob die Bitmap-Füllung einer Form, die größer als die Bitmap ist, beschnitten oder wiederholt wird.
<code>fill.bitmapPath</code>	Ein String, der den Pfad und Namen der Bitmap-Füllung in der Bibliothek angibt.
<code>fill.color</code>	Ein String, ein hexadezimaler Wert oder eine Ganzzahl zur Darstellung der Füllfarbe.
<code>fill.colorArray</code>	Ein Array von Farben in einem Farbverlauf.
<code>fill.focalPoint</code>	Eine Ganzzahl, die den horizontalen Abstand zwischen dem Farbverlaufsbrennpunkt und dem Transformationspunkt festlegt.
<code>fill.linearRGB</code>	Ein boolescher Wert, der angibt, ob die Füllung als linearer oder radialer RGB-Farbverlauf wiedergegeben werden soll.
<code>fill.matrix</code>	Ein Matrix-Objekt , das die Platzierung, Ausrichtung und Skalierung von Farbverlaufsfüllungen definiert.
<code>fill.overflow</code>	Ein String, der das Verhalten des Überlaufs eines Farbverlaufs angibt.
<code>fill.posArray</code>	Ein Array mit Ganzzahlen im Bereich von 0 bis 255, mit dem die Position der entsprechenden Farbe angegeben wird.
<code>fill.style</code>	Ein String, der den Füllstil angibt.

fill.bitmapsClipped

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`fill.bitmapIsClipped`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob die Bitmap-Füllung einer Form, die größer als die Bitmap ist, beschnitten (`true`) oder wiederholt (`false`) wird. Diese Eigenschaft ist nur verfügbar, wenn der Wert der `fill.style`-Eigenschaft "bitmap" lautet. Wenn die Form kleiner als die Bitmap ist, lautet der Wert `false`.

Beispiel

Im folgenden Beispiel wird, sofern zutreffend, im Bedienfeld „Ausgabe“ angezeigt, ob die Bitmap-Füllung beschnitten ist:

```
var fill = fl.getDocumentDOM().getCustomFill();  
if (fill.style == "bitmap")  
    fl.trace("Fill image is clipped: " + fill.bitmapIsClipped);
```

Siehe auch

[fill.bitmapPath](#)

fill.bitmapPath

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`fill.bitmapPath`

Beschreibung

Eigenschaft; ein String, der den Pfad und Namen der Bitmap-Füllung in der Bibliothek angibt. Diese Eigenschaft ist nur verfügbar, wenn der Wert der `fill.style`-Eigenschaft "bitmap" lautet.

Beispiel

Im folgenden Beispiel wird als Füllstil des angegebenen Elements eine in der Bibliothek enthaltene Bitmap festgelegt:

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.style = "bitmap";  
fill.bitmapPath = "myBitmap.jpg";  
fl.getDocumentDOM().setCustomFill(fill);
```

Siehe auch

[fill.bitmapIsClipped](#)

fill.color

Verfügbarkeit

Flash MX 2004.

Verwendung

`fill.color`

Beschreibung

Eigenschaft; die Farbe der Füllung. Sie wird in einem der folgenden Formate angegeben:

- als String im Format "#RRGGBB" oder "#RRGGBBAA"
- als Hexadezimalzahl im Format "0xRRGGBB"

- als Ganzzahl, die die dezimale Entsprechung einer hexadezimalen Zahl darstellt

Beispiel

Im folgenden Beispiel wird die Füllfarbe der aktuellen Auswahl festgelegt:

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.color = "#FFFFFF";  
fl.getDocumentDOM().setCustomFill( fill );
```

fill.colorArray

Verfügbarkeit

Flash MX 2004.

Verwendung

fill.colorArray

Beschreibung

Eigenschaft; ein Array von Farben in einem Farbverlauf. Die Farben werden als Ganzzahlen angegeben. Diese Eigenschaft ist nur verfügbar, wenn der Wert der fill.style-Eigenschaft entweder "radialGradient" oder "linearGradient" lautet. Siehe [fill.style](#)

Beispiel

Im folgenden Beispiel wird das Farbarray der aktuellen Auswahl im Bedienfeld „Ausgabe“ angezeigt, sofern zutreffend:

```
var fill = fl.getDocumentDOM().getCustomFill();  
if(fill.style == "linearGradient" || fill.style == "radialGradient")  
    alert(fill.colorArray);
```

Im folgenden Beispiel wird die Füllung auf den angegebenen linearen Farbverlauf festgelegt:

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.style = "linearGradient";  
fill.colorArray = ["#00ff00", "#ff00ff"];  
fill.posArray = [0, 255];  
fl.getDocumentDOM().setCustomFill(fill);
```

fill.focalPoint

Verfügbarkeit

Flash 8.

Verwendung

fill.focalPoint

Beschreibung

Eigenschaft; eine Ganzzahl, die den horizontalen Abstand zwischen dem Farbverlaufs-brennpunkt und dem Transformationspunkt festlegt. Bei dem Wert 10 beispielsweise wird der Brennpunkt an einer Stelle platziert, die 10/255stel des Abstands zwischen dem Transformationspunkt und der Kante des Farbverlaufs beträgt. Bei dem Wert -255 liegt der Brennpunkt auf der linken Begrenzung des Farbverlaufs. Der Standardwert ist 0.

Diese Eigenschaft ist nur verfügbar, wenn der Wert der `fill.style`-Eigenschaft "radialGradient" lautet.

Beispiel

Im folgenden Beispiel wird der Mittelpunkt eines radialen Farbverlaufs für die aktuelle Auswahl an einer Stelle platziert, die 100 Pixel rechts neben dem Mittelpunkt der Form liegt:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style = "radialGradient";
fill.colorArray = ["#00ff00", "#ff00ff"];
fill.posArray = [0, 255];
fill.focalPoint = 10100;
fl.getDocumentDOM().setCustomFill(fill);
```

fill.linearRGB

Verfügbarkeit

Flash 8.

Verwendung

`fill.linearRGB`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob die Füllung als linearer oder radialer RGB-Farbverlauf wiedergegeben werden soll. Bei dem Wert `true` wird ein Farbverlauf linear interpoliert und bei `false` radial. Der Standardwert lautet `false`.

Beispiel

Im folgenden Beispiel wird angegeben, dass der Farbverlauf der aktuellen Auswahl mit linearem RGB gerendert werden soll:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.linearRGB style = true"radialGradient";
fill.colorArray = ["#00ff00", "#ff00ff"];
fill.posArray = [0, 255];
fill.focalPoint = 100;
fill.linearRGB = true;
fl.getDocumentDOM().setCustomFill(fill);
```

fill.matrix

Verfügbarkeit

Flash MX 2004.

Verwendung

`fill.matrix`

Beschreibung

Eigenschaft; ein [Matrix-Objekt](#), das die Platzierung, Ausrichtung und Skalierung von Farbverlaufsfüllungen definiert.

Beispiel

Im folgenden Beispiel wird mit der `fill.matrix`-Eigenschaft eine Verlaufsfüllung für die aktuelle Auswahl festgelegt:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style = 'radialGradient';
fill.colorArray = ['#00ff00', '#ff00ff'];
fill.posArray = [0, 255];
fill.focalPoint = 100;
fill.linearRGB = false;
fill.overflow = 'repeat';
var mat = fl.getDocumentDOM().selection[0].matrix;
mat.a = 0.0167083740234375;
mat.b = -0.0096435546875;
mat.c = 0.0312957763671875;
mat.d = 0.05419921875;
mat.tx = 288.65;
mat.ty = 193.05;
for (i in mat) {
    fl.trace(i+' : '+mat[i]);
}
fl.getDocumentDOM().setCustomFill(fill);
```

fill.overflow

Verfügbarkeit

Flash 8.

Verwendung

`fill.overflow`

Beschreibung

Eigenschaft; ein String, der das Verhalten des Überlaufs eines Farbverlaufs angibt. Die zulässigen Werte lauten "extend", "repeat" und "reflect". Bei diesen Strings wird die Groß-/Kleinschreibung nicht beachtet. Der Standardwert lautet "extend".

Beispiel

Im folgenden Beispiel wird für das Verhalten des Überlaufs für die aktuelle Auswahl der Wert "extend" festgelegt:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.overflow = "extend";
fl.getDocumentDOM().setCustomFill(fill);
```

fill.posArray

Verfügbarkeit

Flash MX 2004.

Verwendung

fill.posArray

Beschreibung

Eigenschaft; ein Array mit Ganzzahlen im Bereich von 0 bis 255, mit dem die Position der entsprechenden Farbe angegeben wird. Diese Eigenschaft ist nur verfügbar, wenn der Wert der [fill.style](#)-Eigenschaft entweder "radialGradient" oder "linearGradient" lautet.

Beispiel

Im folgenden Beispiel werden die Farben für einen linearen Farbverlauf in der aktuellen Auswahl festgelegt:

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.style = "linearGradient";  
fill.colorArray = [ 0x00ff00, 0xff0000, 0x0000ff ];  
fill.posArray= [0,100, 200];  
fl.getDocumentDOM().setCustomFill( fill );
```

fill.style

Verfügbarkeit

Flash MX 2004. Der Wert "bitmap" wurde in Flash CS4 Professional hinzugefügt.

Verwendung

fill.style

Beschreibung

Eigenschaft; ein String, der den Füllstil angibt. Die zulässigen Werte lauten "bitmap", "solid", "linearGradient", "radialGradient" und "noFill".

Lautet dieser Wert "linearGradient" oder "radialGradient", sind auch die Eigenschaften [fill.colorArray](#) und [fill.posArray](#) verfügbar. Lautet dieser Wert "bitmap", sind auch die Eigenschaften [fill.bitmapIsClipped](#) und [fill.bitmapPath](#) verfügbar.

Beispiel

Im folgenden Beispiel werden die Farben für einen linearen Farbverlauf in der aktuellen Auswahl festgelegt:

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.style= "linearGradient";  
fill.colorArray = [ 0x00ff00, 0xff0000, 0x0000ff ];  
fill.posArray= [0,100, 200];  
fl.getDocumentDOM().setCustomFill( fill );
```

Kapitel 16: Filter-Objekt

Verfügbarkeit

Flash 8.

Beschreibung

Dieses Objekt enthält alle Eigenschaften für alle Filter. Die Eigenschaft `filter.name` gibt den Filtertyp an und bestimmt, welche Eigenschaften jeweils für einen Filter gelten. Siehe [filter.name](#).

Mit `document.getFilters()` wird die Filterliste für ein oder mehrere Objekte zurückgegeben. Mit `document.setFilters()` werden Filter auf ein oder mehrere Objekte angewendet. Siehe [document.getFilters\(\)](#) und [document.setFilters\(\)](#).

Übersicht über Eigenschaften

Für das Filter-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>filter.angle</code>	Ein Gleitkommawert, der den Winkel der Schatten- bzw. der Markierungsfarbe in Grad angibt.
<code>filter.blurX</code>	Ein Gleitkommawert, der den Umfang der Weichzeichnung in der x-Achse in Pixeln angibt.
<code>filter.blurY</code>	Ein Gleitkommawert, der den Umfang der Weichzeichnung in der y-Achse in Pixeln angibt.
<code>filter.brightness</code>	Ein Gleitkommawert, der die Helligkeit des Filters angibt.
<code>filter.color</code>	Ein String, ein hexadezimaler Wert oder eine Ganzzahl zur Darstellung der Filterfarbe.
<code>filter.contrast</code>	Ein Gleitkommawert, der den Kontrastwert des Filters angibt.
<code>filter.distance</code>	Ein Gleitkommawert, der den Abstand zwischen dem Filtereffekt und dem Objekt in Pixeln angibt.
<code>filter.enabled</code>	Ein boolescher Wert, der angibt, ob der angegebene Filter aktiviert ist.
<code>filter.hideObject</code>	Ein boolescher Wert, der angibt, ob das Quellbild ausgeblendet ist.
<code>filter.highlightColor</code>	Ein String, ein hexadezimaler Wert oder eine Ganzzahl zur Darstellung der Glanzlichtfarbe.
<code>filter.hue</code>	Ein Gleitkommawert, der den Farbton des Filters angibt.
<code>filter.inner</code>	Ein boolescher Wert, der angibt, ob es sich bei dem Schatten um einen inneren Schatten handelt.
<code>filter.knockout</code>	Ein boolescher Wert, der angibt, ob es sich bei dem Filter um einen Aussparungsfilter handelt.
<code>filter.name</code>	Schreibgeschützt; ein String, der den Typ des Filters angibt.
<code>filter.quality</code>	Ein String, der die Weichzeichnungsqualität angibt.
<code>filter.saturation</code>	Ein Gleitkommawert, der den Sättigungswert des Filters angibt.

Eigenschaft	Beschreibung
<code>filter.shadowColor</code>	Ein String, ein hexadezimaler Wert oder eine Ganzzahl zur Darstellung der Schattenfarbe.
<code>filter.strength</code>	Eine Ganzzahl, die die Stärke des Filters angibt.
<code>filter.type</code>	Ein String, den den Typ des Geschliffen- oder Glühen-Filters angibt.

filter.angle

Verfügbarkeit

Flash 8.

Verwendung

`filter.angle`

Beschreibung

Eigenschaft; ein Gleitkommawert, der den Winkel der Schatten- bzw. der Markierungsfarbe in Grad angibt. Zulässige Werte liegen zwischen 0 und 360. Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "bevelFilter", "dropShadowFilter", "gradientBevelFilter" oder "gradientGlowFilter" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird der Winkel der Abgeflacht-Filter bei den ausgewählten Objekten auf 120 eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++) {
    if(myFilters[i].name == 'bevelFilter'){
        myFilters[i].angle = 120;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

Siehe auch

`document.setFilterProperty()`

filter.blurX

Verfügbarkeit

Flash 8.

Verwendung

`filter.blurX`

Beschreibung

Eigenschaft; ein Gleitkommawert, der den Umfang der Weichzeichnung in der *x*-Achse in Pixeln angibt. Zulässige Werte liegen zwischen 0 und 255. Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "bevelFilter", "blurFilter", "dropShadowFilter", "glowFilter", "gradientBevelFilter" oder "gradientGlowFilter" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird der Wert `blurX` der Weichzeichnungsfilter bei den ausgewählten Objekten auf 30 und der Wert `blurY` auf 20 eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'blurFilter'){
        myFilters[i].blurX = 30;
        myFilters[i].blurY = 20;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

Siehe auch

`document.setFilterProperty(), filter.blurY`

filter.blurY

Verfügbarkeit

Flash 8.

Verwendung

`filter.blurY`

Beschreibung

Eigenschaft; ein Gleitkommawert, der den Umfang der Weichzeichnung in der *y*-Achse in Pixeln angibt. Zulässige Werte liegen zwischen 0 und 255. Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "bevelFilter", "blurFilter", "dropShadowFilter", "glowFilter", "gradientBevelFilter" oder "gradientGlowFilter" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Siehe `filter.blurX`.

Siehe auch

`document.setFilterProperty(), filter.blurX`

filter.brightness

Verfügbarkeit

Flash 8.

Verwendung

`filter.brightness`

Beschreibung

Eigenschaft; ein Gleitkommawert, der die Helligkeit des Filters angibt. Zulässige Werte liegen zwischen -100 und 100. Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "adjustColorFilter" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird die Helligkeit der Filter „Farbe anpassen“ bei den ausgewählten Objekten auf 30,5 eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].brightness = 30.5;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

filter.color

Verfügbarkeit

Flash 8.

Verwendung

`filter.color`

Beschreibung

Eigenschaft; die Farbe des Filters. Sie wird in einem der folgenden Formate angegeben:

- als String im Format "#RRGGBB" oder "#RRGGBBAA"
- als Hexadezimalzahl im Format "0xRRGGBB"
- als Ganzzahl, die die dezimale Entsprechung einer hexadezimalen Zahl darstellt

Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "dropShadowFilter" oder "glowFilter" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird die Farbe der Schlagschatten-Filter bei den ausgewählten Objekten auf "#ff00003e" eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'dropShadowFilter'){
        myFilters[i].color = '#ff00003e';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

Siehe auch

`document.setFilterProperty()`

filter.contrast

Verfügbarkeit

Flash 8.

Verwendung

`filter.contrast`

Beschreibung

Eigenschaft; ein Gleitkommawert, der den Kontrastwert des Filters angibt. Zulässige Werte liegen zwischen -100 und 100. Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "adjustColorFilter" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird der Kontrastwert der Filter „Farbe anpassen“ bei den ausgewählten Objekten auf -15,5 eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].contrast = -15.5;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

filter.distance

Verfügbarkeit

Flash 8.

Verwendung

`filter.distance`

Beschreibung

Eigenschaft; ein Gleitkommawert, der den Abstand zwischen dem Filtereffekt und dem Objekt in Pixel angibt. Zulässig sind Werte von -255 bis 255. Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "bevelFilter", "dropShadowFilter", "gradientBevelFilter" oder "gradientGlowFilter" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird der Wirkungsbereich der Schlagschatten-Filter bei den ausgewählten Objekten auf 10 Pixel eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'dropShadowFilter'){
        myFilters[i].distance = 10;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

Siehe auch

[document.setFilterProperty\(\)](#)

filter.enabled

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`filter.enabled`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob der angegebene Filter aktiviert (`true`) oder deaktiviert (`false`) ist.

Beispiel

Im folgenden Beispiel wird der Farbfiler für die ausgewählten Objekte deaktiviert:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].enabled = false;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

filter.hideObject

Verfügbarkeit

Flash 8.

Verwendung

`filter.hideObject`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob das Quellbild ausgeblendet (`true`) oder eingeblendet (`false`) ist. Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "dropShadowFilter" für die Eigenschaft [filter.name](#) aufweisen.

Beispiel

Im folgenden Beispiel wird der Wert `hideObject` der Schlagschatten-Filter bei den ausgewählten Objekten auf `true` eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'dropShadowFilter'){
        myFilters[i].hideObject = true;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

filter.highlightColor

Verfügbarkeit

Flash 8.

Verwendung

`filter.highlightColor`

Beschreibung

Eigenschaft; die Farbe des Glanzlichts. Sie wird in einem der folgenden Formate angegeben:

- als String im Format "`#RRGGBB`" oder "`#RRGGBBAA`"
- als Hexadezimalzahl im Format "`0xRRGGBB`"
- als Ganzzahl, die die dezimale Entsprechung einer hexadezimalen Zahl darstellt

Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "`adjustColorFilter`" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird die Glanzlichtfarbe der Abgeflacht-Filter bei den ausgewählten Objekten auf "`#ff00003e`" eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'bevelFilter'){
        myFilters[i].highlightColor = '#ff00003e';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

filter.hue

Verfügbarkeit

Flash 8.

Verwendung

`filter.hue`

Beschreibung

Eigenschaft; ein Gleitkommawert, der den Farbton des Filters angibt. Zulässige Werte liegen zwischen -180 und 180. Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "adjustColorFilter" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird der Farbton der Filter „Farbe anpassen“ bei den ausgewählten Objekten auf 120 eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].hue = 120;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

filter.inner

Verfügbarkeit

Flash 8.

Verwendung

`filter.inner`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob es sich bei dem Schatten um einen inneren Schatten handelt (`true`) oder nicht (`false`). Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "dropShadowFilter" oder "glowFilter" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird der Wert `inner` der Glühen-Filter bei den ausgewählten Objekten auf `true` eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'glowFilter'){
        myFilters[i].inner = true;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

Siehe auch

`document.setFilterProperty()`

filter.knockout

Verfügbarkeit

Flash 8.

Verwendung

`filter.knockout`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob es sich bei dem Filter um einen Aussparungsfilter handelt (`true`) oder nicht (`false`). Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert `"bevelFilter"`, `"dropShadowFilter"`, `"glowFilter"`, `"gradientBevelFilter"` oder `"gradientGlowFilter"` für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `knockout` der Glühen-Filter bei den ausgewählten Objekten auf `true` eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'glowFilter'){
        myFilters[i].knockout = true;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

Siehe auch

`document.setFilterProperty()`

filter.name

Verfügbarkeit

Flash 8.

Verwendung

`filter.name`

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der den Typ des Filters angibt. Es hängt vom Wert dieser Eigenschaft ab, welche anderen Eigenschaften des Filter-Objekts verfügbar sind. Folgende Werte sind möglich:

`"adjustColorFilter"`, `"bevelFilter"`, `"blurFilter"`, `"dropShadowFilter"`, `"glowFilter"`, `"gradientBevelFilter"` oder `"gradientGlowFilter"`.

Beispiel

Im folgenden Beispiel werden die Filternamen und Indexpositionen im Bedienfeld „Ausgabe“ angezeigt:


```
var myFilters = fl.getDocumentDOM().getFilters();
var traceStr = "";
for(i=0; i < myFilters.length; i++){
    traceStr = traceStr + " At index " + i + ": " + myFilters[i].name;
}
fl.trace(traceStr);
```

Siehe auch

[document.getFilters\(\)](#), [document.setFilterProperty\(\)](#)

filter.quality

Verfügbarkeit

Flash 8.

Verwendung

`filter.quality`

Beschreibung

Eigenschaft; ein String, der die Weichzeichnungsqualität angibt. Die zulässigen Werte lauten "low", "medium" und "high" ("high" ist mit einem Gaußschen Weichzeichner vergleichbar). Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "bevelFilter", "blurFilter", "dropShadowFilter", "glowFilter", "gradientGlowFilter" oder "gradientBevelFilter" für die Eigenschaft [filter.name](#) aufweisen.

Beispiel

Im folgenden Beispiel wird die Weichzeichnungsqualität der Glühen-Filter bei den ausgewählten Objekten auf "medium" eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'glowFilter'){
        myFilters[i].quality = 'medium';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

Siehe auch

[document.setFilterProperty\(\)](#)

filter.saturation

Verfügbarkeit

Flash 8.

Verwendung

`filter.saturation`

Beschreibung

Eigenschaft; ein Gleitkommawert, der den Sättigungswert des Filters angibt. Zulässig sind Werte von -100 bis 100. Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "adjustColorFilter" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird der Sättigungswert des Filters „Farbe anpassen“ bei den ausgewählten Objekten auf -100 (Graustufen) eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].saturation = 0-100;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

Siehe auch

`document.setFilterProperty()`

filter.shadowColor

Verfügbarkeit

Flash 8.

Verwendung

`filter.shadowColor`

Beschreibung

Eigenschaft; die Farbe des Schattens. Sie wird in einem der folgenden Formate angegeben:

- als String im Format "#RRGGBB" oder "#RRGGBBAA"
- als Hexadezimalzahl im Format "0xRRGGBB"
- als Ganzzahl, die die dezimale Entsprechung einer hexadezimalen Zahl darstellt

Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "adjustColorFilter" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird die Schattenfarbe der Abgeflacht-Filter bei den ausgewählten Objekten auf "#ff00003e" eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'bevelFilter'){
        myFilters[i].shadowColor = '#ff00003e';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

Siehe auch

`document.setFilterProperty()`

filter.strength

Verfügbarkeit

Flash 8.

Verwendung

`filter.strength`

Beschreibung

Eigenschaft; eine Ganzzahl, die die Stärke des Filters angibt. Zulässige Werte liegen zwischen 0 und 25,500. Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "bevelFilter", "dropShadowFilter", "glowFilter", "gradientGlowFilter" oder "gradientBevelFilter" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird die Stärke der Glühen-Filter bei den ausgewählten Objekten auf 50 eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'glowFilter'){
        myFilters[i].strength = 50;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

Siehe auch

`document.setFilterProperty()`

filter.type

Verfügbarkeit

Flash 8.

Verwendung

`filter.type`

Beschreibung

Eigenschaft; ein String, den den Typ des Geschliffen- oder Glühen-Filters angibt. Die zulässigen Werte lauten "inner", "outer" und "full". Diese Eigenschaft wird für Filter-Objekte definiert, die den Wert "bevelFilter", "gradientGlowFilter" oder "gradientBevelFilter" für die Eigenschaft `filter.name` aufweisen.

Beispiel

Im folgenden Beispiel wird der Typ aller Abgeflacht-Filter bei den ausgewählten Objekten auf "full" eingestellt:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'bevelFilter'){
        myFilters[i].type = 'full';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

Siehe auch

[document.setFilterProperty\(\)](#)

Kapitel 17: flash-Objekt (fl)

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das flash-Objekt stellt die Flash-Anwendung dar. Sie können mit `flash` oder `fl` auf dieses Objekt verweisen. In dieser Dokumentation wird in den Codebeispielen ausschließlich `fl` verwendet.

Übersicht über Methoden

Für das flash-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>fl.addEventListener()</code>	Registriert eine Funktion, die aufgerufen werden soll, wenn ein bestimmtes Ereignis empfangen wird.
<code>fl.browseForFileURL()</code>	Öffnet ein systemspezifisches Dialogfeld „Datei öffnen“ bzw. „Datei speichern“ und lässt den Benutzer eine zu öffnende oder zu speichernde Datei auswählen.
<code>fl.browseForFolderURL()</code>	Zeigt ein Dialogfeld zur Ordnersuche an und lässt den Benutzer einen Ordner auswählen.
<code>„fl.clearPublishCache()“</code> auf Seite 229	Leert den Veröffentlichungscache.
<code>fl.clipCopyString()</code>	Kopiert den angegebenen String in die Zwischenablage.
<code>fl.closeAll()</code>	Schließt alle geöffneten Dokumente und zeigt für alle bisher noch nicht gespeicherten Dokumente das Dialogfeld „Speichern unter“ an.
<code>fl.closeAllPlayerDocuments()</code>	Schließt alle SWF-Dateien, die mit „Steuerung“ > „Film testen“ geöffnet wurden.
<code>fl.closeDocument()</code>	Schließt das angegebene Dokument.
<code>fl.createDocument()</code>	Öffnet ein neues Dokument und wählt es aus.
<code>„fl.exportPublishProfileString()“</code> auf Seite 237	URI (Uniform Resource Identifier), von dem die Veröffentlichungseinstellungen exportiert werden sollen.
<code>fl.fileExists()</code>	Prüft, ob eine Datei bereits auf der Festplatte vorhanden ist.
<code>fl.findDocumentDOM()</code>	Ermöglicht die Angabe einer bestimmten Datei durch ihren eindeutigen Kennzeichner.
<code>fl.findDocumentIndex()</code>	Gibt ein Array mit Ganzzahlen zurück, mit denen die Position eines Dokuments im Array <code>fl.documents</code> angegeben wird.
<code>fl.findObjectInDocByName()</code>	Macht Elemente mit Instanznamen, die bestimmten Text enthalten, sichtbar.
<code>fl.findObjectInDocByType()</code>	Macht Elemente eines bestimmten Elementtyps in einem Dokument sichtbar.
<code>fl.getAppMemoryInfo()</code>	Gibt eine Ganzzahl zurück, die die Anzahl der Byte darstellt, die von einem bestimmten Bereich des Flash.exe-Speichers belegt sind.
<code>fl.getDocumentDOM()</code>	Ruft das DOM (Document-Objekt) des gegenwärtig aktiven Dokuments ab.

Methode	Beschreibung
<code>fl.getSwfPanel()</code>	Gibt das SWFPanel-Objekt auf Grundlage des lokalisierten Namens oder des SWF-Dateinamens des SWF-Bedienfeldes zurück.
<code>fl.isFontInstalled()</code>	Stellt fest, ob eine bestimmte Schriftart installiert ist.
<code>fl.mapPlayerURL()</code>	Konvertiert eine Unicode-URL mit Escape-Sequenz in eine UTF-8- oder MBCS-URL.
<code>fl.openDocument()</code>	Öffnet ein Flash-Dokument (FLA-Datei) zur Bearbeitung in einem neuen Flash-Dokumentfenster und verlagert den Fokus auf dieses Dokument.
<code>fl.openScript()</code>	Öffnet eine Skriptdatei (JSFL, AS, ASC) oder eine andere Datei (XML, TXT) im Flash-Texteditor.
<code>fl.quit()</code>	Beendet Flash und fordert zuvor den Benutzer auf, alle geänderten Dokumente zu speichern.
<code>fl.reloadTools()</code>	Baut das Bedienfeld „Werkzeuge“ anhand der Datei „toolconfig.xml“ neu auf. Wird nur beim Erstellen erweiterbarer Werkzeuge verwendet.
<code>fl.removeEventListener()</code>	Hebt die Registrierung einer Funktion auf, die mit <code>fl.addEventListener()</code> registriert wurde.
<code>fl.resetAS3PackagePaths()</code>	Setzt die globale Klassenpfad-Einstellung im Dialogfeld „ActionScript 3.0-Einstellungen“ auf den Standardwert zurück.
<code>fl.resetPackagePaths()</code>	Setzt die globale Klassenpfad-Einstellung im Dialogfeld „ActionScript 2.0-Einstellungen“ auf den Standardwert zurück.
<code>fl.runScript()</code>	Führt eine JavaScript-Datei aus.
<code>fl.saveAll()</code>	Speichert alle geöffneten Dokumente und zeigt für alle bisher noch nicht gespeicherten Dokumente das Dialogfeld „Speichern unter“ an.
<code>fl.saveDocument()</code>	Speichert das angegebene Dokument als FLA-Dokument.
<code>fl.saveDocumentAs()</code>	Zeigt das Dialogfeld „Speichern unter“ für das angegebene Dokument an.
<code>fl.selectElement()</code>	Ermöglicht die Auswahl oder Bearbeitung eines Elements.
<code>fl.selectTool()</code>	Wählen Sie im Bedienfeld „Werkzeuge“ das angegebene Werkzeug aus.
<code>fl.setActiveWindow()</code>	Legt fest, dass das angegebene Dokument das aktive Fenster sein soll.
<code>fl.showIdleMessage()</code>	Dient zum Deaktivieren der Warnmeldung bei zu lange dauernden Skripts.
<code>fl.toggleBreakpoint()</code>	Schaltet einen Haltepunkt für die angegebene .as-Datei in der angegebenen Zeile um.
<code>fl.trace()</code>	Sendet einen Textstring an das Bedienfeld „Ausgabe“.

Übersicht über Eigenschaften

Die folgenden Eigenschaften können mit dem flash-Objekt verwendet werden.

Eigenschaft	Beschreibung
<code>fl.actionsPanel</code>	Schreibgeschützt; ein <code>actionsPanel</code> -Objekt.
<code>fl.as3PackagePaths</code>	Ein String, welcher der globalen Klassenpfad-Einstellung im Dialogfeld „ActionScript 3.0-Einstellungen“ entspricht.
<code>fl.compilerErrors</code>	Schreibgeschützt, ein <code>compilerErrors</code> -Objekt.
<code>fl.componentsPanel</code>	Schreibgeschützt; ein <code>componentsPanel</code> -Objekt, das das Bedienfeld „Komponenten“ darstellt.
<code>fl.configDirectory</code>	Schreibgeschützt; ein String, der den vollständigen Pfad für das Verzeichnis „Configuration“ des lokalen Benutzers in einem plattformspezifischen Format angibt.
<code>fl.configURI</code>	Schreibgeschützt; ein String, der den vollständigen Pfad für das Verzeichnis „Configuration“ des lokalen Benutzers als file:///URI angibt.
<code>fl.contactSensitiveSelection</code>	Ein boolescher Wert, der angibt, ob der Auswahlmodus „Kontaktempfindlich“ aktiviert ist oder nicht.
<code>fl.createNewDocList</code>	Schreibgeschützt; ein Array von Strings für die verschiedenen Dokumenttypen, die erstellt werden können.
<code>fl.createNewDocListType</code>	Schreibgeschützt; ein Array von Strings für die Dateierweiterungen der verschiedenen Dokumenttypen, die erstellt werden können.
<code>fl.createNewTemplateList</code>	Schreibgeschützt; ein Array von Strings für die verschiedenen Vorlagentypen, die erstellt werden können.
<code>fl.documents</code>	Schreibgeschützt; ein Array von Document-Objekten (siehe <code>Document</code> -Objekt), die die gegenwärtig zur Bearbeitung geöffneten Dokumente (FLA-Dateien) darstellen.
<code>fl.drawingLayer</code>	Schreibgeschützt; <code>drawingLayer</code> -Objekt, das ein erweiterbares Werkzeug verwenden sollte, wenn der Benutzer während des Ziehens vorübergehend zeichnen möchte.
<code>fl.externalLibraryPath</code>	Ein String, der eine Liste von Elementen enthält, die sich im globalen Pfad der externen ActionScript 3.0-Bibliothek befinden, und dieser Pfad wiederum gibt den Speicherort von SWC-Dateien an, die zur Laufzeit als gemeinsame Bibliotheken verwendet werden.
<code>fl.flexSDKPath</code>	Ein String, der den Pfad des Ordners „Flex SDK“ angibt, der u. a. die Ordner „bin“, „frameworks“ und „lib“ enthält.
<code>fl.installedPlayers</code>	Gibt ein Array generischer Objekte entsprechend der Liste der installierten Flash Player im Eigenschafteninspektor des Dokuments zurück.
<code>fl.languageCode</code>	Gibt den fünfstelligen Code zurück, der das Gebietsschema der Benutzeroberfläche der Anwendung bestimmt.
<code>fl.libraryPath</code>	Ein String, der eine Liste von Elementen enthält, die sich im globalen Pfad der ActionScript 3.0-Bibliothek befinden, und dieser Pfad wiederum gibt den Speicherort von SWC-Dateien oder SWC-Dateien enthaltenden Ordnern an.
<code>fl.Math</code>	Schreibgeschützt; das <code>Math</code> -Objekt, das Methoden für Matrix- und Punktberechnungen bereitstellt.
<code>fl.mruRecentFileList</code>	Schreibgeschützt; ein Array der vollständigen Dateinamen in der MRU-Liste (Most Recently Used – Zuletzt verwendet), die vom Flash-Authoring-Tool verwaltet wird.
<code>fl.mruRecentFileListType</code>	Schreibgeschützt; ein Array der Dateitypen in der MRU-Liste, die vom Flash-Authoring-Tool verwaltet wird.

Eigenschaft	Beschreibung
fl.packagePaths	Ein String, welcher der globalen Klassenpfad-Einstellung im Dialogfeld „ActionScript 2.0-Einstellungen“ entspricht.
fl.publishCacheDiskSizeMax	Eine Ganzzahl für die Voreinstellung für das Limit des Festplattencache.
fl.publishCacheEnabled	Ein boolescher Wert, der festlegt, ob der Veröffentlichungscache aktiviert ist.
fl.publishCacheMemoryEntrySizeLimit	Eine Eigenschaft in Form einer Ganzzahl, die die Voreinstellung für die maximale Größe des Arbeitsspeichercacheeintrags festlegt.
fl.publishCacheMemorySizeMax	Eine Ganzzahl für die Voreinstellung für das Limit des Arbeitsspeichercache.
fl.objectDrawingMode	Eine Ganzzahl, die den aktivierten Objektzeichnungsmodus darstellt.
fl.outputPanel	Schreibgeschützt; Verweis auf das outputPanel-Objekt .
fl.presetPanel	Schreibgeschützt; ein presetPanel-Objekt .
fl.scriptURI	Schreibgeschützt; ein String im Format einer file:/// -URI, der den Pfad des zurzeit ausgeführten JSFL-Skripts angibt.
fl.sourcePath	Ein String, der eine Liste von Elementen enthält, die sich im globalen ActionScript 3.0-Quellpfad befinden, und dieser Pfad wiederum gibt den Speicherort von Dateien der ActionScript-Klasse an.
fl.swfPanels	Ein Array von registrierten swfPanel-Objekten (siehe swfPanel-Objekt).
fl.tools	Schreibgeschützt; ein Array von Tools-Objekten.
fl.version	Schreibgeschützt; der lange Versionsstring des Flash-Authoring-Tools, einschließlich der Plattform.
fl.xmlui	Schreibgeschützt; ein XMLUI-Objekt .

fl.actionsPanel

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`fl.actionsPanel`

Beschreibung

Schreibgeschützte Eigenschaft; ein actionsPanel-Objekt, welches das zurzeit angezeigte Bedienfeld „Aktionen“ darstellt. Weitere Informationen zur Verwendung dieser Eigenschaft finden Sie unter [actionsPanel-Objekt](#).

fl.addEventListener()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`fl.addEventListener(eventType, callbackFunction)`

Parameter

eventType Ein String, der den Ereignistyp angibt, der an diese Rückruffunktion übergeben werden soll. Zulässige Werte sind "documentNew", "documentOpened", "documentClosed", "mouseMove", "documentChanged", "layerChanged" und "frameChanged".

Der Wert `documentChanged` besagt nicht, dass der Inhalt eines Dokuments geändert wurde, sondern dass sich jetzt ein anderes Dokument im Vordergrund befindet. Dies bedeutet, dass `fl.getDocumentDOM()` einen anderen Wert zurückgibt als vor dem Auftreten dieses Ereignisses.

callbackFunction Der Name der Funktion, die Sie bei jedem Eintreten des Ereignisses ausführen möchten.

Rückgabewerte

Keine.

Beschreibung

Methode; registriert eine Funktion, die aufgerufen werden soll, wenn ein bestimmtes Ereignis eintritt.

Wenn Sie diese Methode verwenden, berücksichtigen Sie, dass Ihre Anwendung möglicherweise abstürzt oder dass ein Fehler auftritt, wenn das Ereignis häufig eintritt (wie zum Beispiel `mouseMove`) und die Funktion längere Zeit zum Ausführen benötigt.

Beispiel

Im folgenden Beispiel wird im Bedienfeld „Ausgabe“ eine Meldung angezeigt, wenn ein Dokument geschlossen wird.

```
myFunction = function () {  
    fl.trace('document was closed'); }  
fl.addEventListener("documentClosed", myFunction);
```

Siehe auch

[fl.removeEventListener\(\)](#)

fl.as3PackagePaths

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`fl.as3PackagePaths`

Beschreibung

Eigenschaft; ein String, welcher der globalen Klassenpfad-Einstellung im Dialogfeld „ActionScript 3.0-Einstellungen“ entspricht. Elemente innerhalb des Strings sind durch Semikolons getrennt. Um die ActionScript 2.0-Klassenpfad-Einstellungen anzuzeigen oder zu ändern, verwenden Sie [fl.packagePaths](#).

Beispiel

Im folgenden Beispiel wird das Ändern der ActionScript 3.0-Klassenpfad-Einstellungen veranschaulicht.

```
fl.trace(fl.as3PackagePaths);  
// Output (assuming started with default value)  
// .;$ (AppConfig)/ActionScript 3.0/Classes  
fl.as3PackagePaths="buying;selling";  
fl.trace(fl.as3PackagePaths);  
// Output  
// buying; selling
```

Siehe auch

[fl.resetAS3PackagePaths\(\)](#)

fl.browseForFileURL()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.browseForFileURL(browseType [, title [, previewArea]])
```

Parameter

browseType Ein String, der den Typ des Dateisuchvorgangs angibt. Die zulässigen Werte lauten "open", "select" und "save". Mit den Werten "open" und "select" wird das systemspezifische Dialogfeld „Datei öffnen“ geöffnet. Diese Werte werden bereitgestellt, um die Kompatibilität mit Dreamweaver zu gewährleisten. Mit dem Wert "save" wird ein systemspezifisches Dialogfeld „Datei öffnen“ geöffnet.

title Ein String, der den Titel des Dialogfelds zum Öffnen oder Speichern einer Datei angibt. Wenn dieser Parameter fehlt, wird ein Standardwert verwendet. Dieser Parameter ist optional.

previewArea Ein optionaler Parameter, der von Flash und Fireworks ignoriert wird. Er ist lediglich vorhanden, um die Kompatibilität mit Dreamweaver zu gewährleisten.

Rückgabewerte

Die URL der Datei im Format file:/// URI. Wenn der Benutzer das Dialogfeld durch „Abbrechen“ schließt, wird null zurückgegeben.

Beschreibung

Methode; öffnet ein systemspezifisches Dialogfeld „Datei öffnen“ bzw. „Datei speichern“ und lässt den Benutzer eine zu öffnende oder zu speichernde Datei auswählen.

Beispiel

Im folgenden Beispiel wird der Benutzer aufgefordert, eine FLA-Datei zu wählen, die er öffnen möchte. Diese Datei wird dann geöffnet: (Die Methode `fl.browseForFileURL()` kann zwar beliebige Dateitypen suchen, aber `fl.openDocument()` kann nur FLA-Dateien öffnen.)

```
var fileURL = fl.browseForFileURL("open", "Select file");  
var doc = fl.openDocument(fileURL);
```

Siehe auch

[fl.browseForFolderURL\(\)](#)

fl.browseForFolderURL()

Verfügbarkeit

Flash 8.

Verwendung

```
fl.browseForFolderURL([description])
```

Parameter

description Ein optionaler String mit der Beschreibung für das Dialogfeld „Ordner suchen“. Wenn dieser Parameter fehlt, bleibt der Beschreibungsbereich leer.

Rückgabewerte

Die URL des Ordners im Format file:/// URI. Wenn der Benutzer das Dialogfeld durch „Abbrechen“ schließt, wird null zurückgegeben.

Beschreibung

Methode; zeigt ein Dialogfeld zur Ordnersuche an und lässt den Benutzer einen Ordner auswählen.

Hinweis: Der Titel des Dialogfelds lautet stets „Ordner suchen“. Mit dem Parameter *description* können Sie dem Beschreibungsbereich detailliertere Anweisungen hinzufügen, z. B. „Wählen Sie einen Ordner aus“ oder „Wählen Sie den Pfad mit den zu importierenden Klassendefinitionsdateien (.as) aus“.

Beispiel

Im folgenden Beispiel wird der Benutzer zur Auswahl eines Ordners aufgefordert und anschließend wird eine Liste der in diesem Ordner enthaltenen Dateien angezeigt:

```
var folderURI = fl.browseForFolderURL("Select a folder.");  
var folderContents = FLfile.listFolder(folderURI);
```

Siehe auch

[fl.browseForFileURL\(\)](#), [FLfile-Objekt](#)

fl.clearPublishCache()

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

```
fl.clearPublishCache()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; leert den Veröffentlichungscache.

Beispiel

Mit dem folgenden Code wird der Veröffentlichungscache geleert:

```
fl.clearPublishCache()
```

Siehe auch

[fl.publishCacheDiskSizeMax](#), [fl.publishCacheEnabled](#), „[fl.publishCacheMemoryEntrySizeLimit](#)“ auf [Seite 254](#), „[fl.publishCacheMemorySizeMax](#)“ auf [Seite 254](#)

fl.clipCopyString()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
fl.clipCopyString(string)
```

Parameter

string Ein String, der in die Zwischenablage kopiert werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; kopiert den angegebenen String in die Zwischenablage.

Um die aktuelle Auswahl in die Zwischenablage zu kopieren, verwenden Sie [document.clipCopy\(\)](#).

Beispiel

Im folgenden Beispiel wird der Pfad des aktuellen Dokuments in die Zwischenablage kopiert:

```
var documentPath = fl.getDocumentDOM().path;  
fl.clipCopyString(documentPath);
```

fl.closeAll()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.closeAll([bPromptToSave])
```

Parameter

bPromptToSave Ein optionaler boolescher Wert, der angibt, ob das Dialogfeld „Speichern“ – für Dateien, die seit dem letzten Speichervorgang geändert wurden – oder das Dialogfeld „Speichern unter“ – für Dateien, die bisher noch gar nicht gespeichert wurden – angezeigt wird. Der Standardwert lautet `true`.

Rückgabewerte

Keine.

Beschreibung

Methode; schließt alle geöffneten Dateien (FLA-, SWF-, JSFL-Dateien usw.). Um alle geöffneten Dateien zu schließen, ohne Änderungen zu speichern, übergeben Sie den Parameter `false` an die Funktion `bPromptToSave`. Mit dieser Methode wird die Anwendung nicht beendet.

Beispiel

Der nachstehende Code bewirkt, dass alle geöffneten Dateien geschlossen werden und der Benutzer zum Speichern neuer oder geänderter Dateien aufgefordert wird.

```
fl.closeAll();
```

Siehe auch

`fl.closeAllPlayerDocuments()`, `fl.closeDocument()`

fl.closeAllPlayerDocuments()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
fl.closeAllPlayerDocuments()
```

Parameter

Keine.

Rückgabewerte

Ein boolescher Wert; `true`, wenn ein oder mehrere Filmfenster geöffnet waren; andernfalls `false`.

Beschreibung

Methode; schließt alle SWF-Dateien, die mit „Steuerung“ > „Film testen“ geöffnet wurden.

Beispiel

Im folgenden Beispiel werden alle SWF-Dateien, die mit „Steuerung“ > „Film testen“ geöffnet wurden, geschlossen.

```
fl.closeAllPlayerDocuments();
```

Siehe auch

`fl.closeAll()`, `fl.closeDocument()`

fl.closeDocument()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.closeDocument (documentObject [, bPromptToSaveChanges])
```

Parameter

documentObject Ein [Document-Objekt](#). Wenn die Methode *documentObject* sich auf das aktive Dokument bezieht, wird das Dokumentfenster möglicherweise erst nach Beendigung des aufrufenden Skripts geschlossen.

bPromptToSaveChanges Ein boolescher Wert. Wenn *bPromptToSaveChanges* den Wert `false` hat, wird der Benutzer nicht zur Eingabe aufgefordert, wenn das Dokument ungespeicherte Änderungen enthält, d. h. die Datei wird geschlossen und die Änderungen werden verworfen. Wenn *bPromptToSaveChanges* den Wert `true` hat und das Dokument ungespeicherte Änderungen enthält, wird der Benutzer mit einem standardmäßigen Dialogfeld (in dem „Ja“ und „Nein“ zur Auswahl stehen) zur Eingabe aufgefordert. Der Standardwert lautet `true`. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; schließt das angegebene Dokument.

Beispiel

Im folgenden Beispiel werden zwei Möglichkeiten zum Schließen eines Dokuments demonstriert:

```
// Closes the specified document and prompts to save changes.
fl.closeDocument (fl.documents[0]);
fl.closeDocument (fl.documents[0] , true); // Use of true is optional.
// Closes the specified document without prompting to save changes.
fl.closeDocument (fl.documents[0] , false);
```

Siehe auch

[fl.closeAll\(\)](#)

fl.compilerErrors

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
fl.compilerErrors
```

Beschreibung

Schreibgeschützte Eigenschaft; ein `compilerErrors`-Objekt, welches das Bedienfeld „Fehler“ darstellt. Weitere Informationen zur Verwendung dieser Eigenschaft finden Sie unter [compilerErrors-Objekt](#).

fl.componentsPanel

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.componentsPanel
```

Beschreibung

Schreibgeschützte Eigenschaft; ein [componentsPanel-Objekt](#), das das Bedienfeld „Komponenten“ darstellt.

Beispiel

Im folgenden Beispiel wird ein `componentsPanel`-Objekt in der Variablen `comPanel` gespeichert:

```
var comPanel = fl.componentsPanel;
```

fl.configDirectory

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.configDirectory
```

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der den vollständigen Pfad für das Verzeichnis „Configuration“ des lokalen Benutzers in einem plattformspezifischen Format angibt. Wenn Sie den Pfad in dem nicht plattformspezifischen Format `file:///` URI angeben möchten, verwenden Sie [fl.configURI](#).

Beispiel

Im folgenden Beispiel wird das Verzeichnis „Configuration“ im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace("My local configuration directory is " + fl.configDirectory);
```

fl.configURI

Verfügbarkeit

Flash MX 2004.

Verwendung

`fl.configURI`

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der den vollständigen Pfad für das Verzeichnis „Configuration“ des lokalen Benutzers im Format `file:///` URI angibt. Siehe auch [fl.configDirectory](#).

Beispiel

Im folgenden Beispiel wird ein angegebenes Skript ausgeführt: Mit `fl.configURI` können Sie den Speicherort des Skripts angeben, auch wenn Sie nicht wissen, auf welcher Plattform das Skript ausgeführt wird.

```
// To run a command in your commands menu, change "Test.jsfl"
// to the command you want to run in the line below.
fl.runScript( fl.configURI + "Commands/Test.jsfl" );
```

fl.contactSensitiveSelection

Verfügbarkeit

Flash 8.

Verwendung

`fl.contactSensitiveSelection`

Beschreibung

Ein boolescher Wert, der angibt, ob der Auswahlmodus „Kontaktempfindlich“ aktiviert ist (`true`) oder nicht (`false`).

Beispiel

Mit dem folgenden Beispiel wird veranschaulicht, wie der Auswahlmodus „Kontaktempfindlich“ vor einer Auswahl deaktiviert und nach der Auswahl wieder auf seinen ursprünglichen Wert zurückgesetzt wird:

```
var contact = fl.contactSensitiveSelection;
fl.contactSensitiveSelection = false;
// Insert selection code here.
fl.contactSensitiveSelection = contact;
```

fl.createDocument()

Verfügbarkeit

Flash MX 2004.

Verwendung

`fl.createDocument ([docType])`

Parameter

docType Ein String, der den zu erstellenden Dokumenttyp angibt. Die gültigen Werte lauten `"timeline"`, `"presentation"` und `"application"`. Der Standardwert lautet `"timeline"` und hat dieselbe Wirkung wie die Auswahl von „Datei“ > „Neu“ > „Flash-Datei (ActionScript 3.0)“. Dieser Parameter ist optional.

Rückgabewerte

Das Document-Objekt für das neu erstellte Dokument, wenn die Methode erfolgreich ist. Wenn ein Fehler auftritt, lautet der zurückgegebene Wert `undefined`.

Beschreibung

Methode; öffnet ein neues Dokument und wählt es aus. Die Werte für die Größe, Auflösung und Farbe entsprechen den aktuellen Standardwerten.

Beispiel

Im folgenden Beispiel werden unterschiedliche Dokumenttypen erstellt:

```
// Create two Timeline-based Flash documents.
fl.createDocument();
fl.createDocument("timeline");
// Create a Slide Presentation document.
fl.createDocument("presentation");
// Create a Form Application document.
fl.createDocument("application");
```

fl.createNewDocList

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.createNewDocList
```

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von Strings für die verschiedenen Dokumenttypen, die erstellt werden können.

Beispiel

Im folgenden Beispiel werden die Dokumenttypen, die erstellt werden können, im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace("Number of choices " + fl.createNewDocList.length);
for (i = 0; i < fl.createNewDocList.length; i++)
    fl.trace("choice: " + fl.createNewDocList[i]);
```

fl.createNewDocListType

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.createNewDocListType
```

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von Strings für die Dateierweiterungen der verschiedenen Dokumenttypen, die erstellt werden können. Die Einträge in dem Array entsprechen direkt (nach Index) den Einträgen in dem `fl.createNewDocList`-Array.

Beispiel

Im folgenden Beispiel werden die Erweiterungen der Dokumenttypen, die erstellt werden können, im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace("Number of types " + fl.createNewDocListType.length);  
for (i = 0; i < fl.createNewDocListType.length; i++) fl.trace("type: " +  
fl.createNewDocListType[i]);
```

fl.createNewTemplateList

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.createNewTemplateList
```

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von Strings für die verschiedenen Vorlagentypen, die erstellt werden können.

Beispiel

Im folgenden Beispiel werden die Vorlagentypen, die erstellt werden können, im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace("Number of template types: " + fl.createNewTemplateList.length); for (i = 0; i <  
fl.createNewTemplateList.length; i++) fl.trace("type: " + fl.createNewTemplateList[i]);
```

fl.documents

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.documents
```

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von Document-Objekten (siehe [Document-Objekt](#)), die die gegenwärtig zur Bearbeitung geöffneten Dokumente (FLA-Dateien) darstellen.

Beispiel

Im folgenden Beispiel wird ein Array von geöffneten Dokumenten in der Variablen `docs` gespeichert:

```
var docs = fl.documents;
```

Im folgenden Beispiel werden die Namen der zurzeit geöffneten Dokumente im Bedienfeld „Ausgabe“ angezeigt:

```
for (doc in fl.documents) {  
    fl.trace(fl.documents[doc].name);  
}
```

fl.drawingLayer

Verfügbarkeit

Flash MX 2004.

Verwendung

`fl.drawingLayer`

Beschreibung

Schreibgeschützte Eigenschaft; das [drawingLayer-Objekt](#), das ein erweiterbares Werkzeug verwenden sollte, wenn der Benutzer während des Ziehens vorübergehend zeichnen möchte (z. B. beim Erstellen eines Auswahlrahmens).

Beispiel

Siehe `drawingLayer.setColor()`.

fl.exportPublishProfileString()

Verfügbarkeit

Flash Professional CS5.

Verwendung

```
fl.exportPublishProfileString( ucURI [, profileName] )
```

Parameter

ucURI Ein String, der die Datei-URI (Uniform Resource Identifier) angibt, von der die Veröffentlichungseinstellungen exportiert werden sollen.

profileName Ein String, der den zu exportierenden Profilnamen angibt. Dieser Parameter ist optional.

Rückgabewerte

String.

Beschreibung

Gibt das Veröffentlichungsprofil eines bestimmten Dokuments zurück, ohne dass die Datei geöffnet werden muss. Wahlweise kann auch das Veröffentlichungsprofil angegeben werden.

Beispiel

Im folgenden Beispiel wird der String des Veröffentlichungsprofils gelesen:

```
var ppXML = "";
var ucfURI = fl.browseForFileURL("open", "select a FLA");
if (ucfURI && ucfURI.length > 0)
ppXML = fl.exportPublishProfileString(ucfURI);
fl.trace(ppXML);
```

fl.externalLibraryPath

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
fl.externalLibraryPath
```

Beschreibung

Eigenschaft; ein String, der eine Liste von Elementen enthält, die sich im globalen Pfad der externen ActionScript 3.0-Bibliothek befinden, und dieser Pfad wiederum gibt den Speicherort von SWC-Dateien an, die zur Laufzeit als gemeinsame Bibliotheken verwendet werden. Elemente innerhalb des Strings sind durch Semikolons getrennt. Zur Angabe der Elemente wählen Sie im Authoring-Tool „Bearbeiten“ > „Voreinstellungen“ > „ActionScript“ > „ActionScript 3.0-Einstellungen“.

Beispiel

Im folgenden Beispiel wird der Ordner „/SWC_runtime“ dem globalen Pfad der externen ActionScript 3.0-Bibliothek hinzugefügt:

```
fl.trace(fl.externalLibraryPath);
fl.externalLibraryPath = "/SWC_runtime;" + fl.externalLibraryPath;
fl.trace(fl.externalLibraryPath);
```

Siehe auch

[fl.flexSDKPath](#), [fl.libraryPath](#), [fl.sourcePath](#), [document.externalLibraryPath](#)

fl.fileExists()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.fileExists(fileURI)
```

Parameter

fileURI Ein String im Format file:/// URI, den Pfad der Datei angibt.

Rückgabewerte

Ein boolescher Wert; *true*, wenn die Datei auf der Festplatte vorhanden ist, ansonsten *false*.

Beschreibung

Methode; prüft, ob eine Datei bereits auf der Festplatte vorhanden ist.

Beispiel

Im folgenden Beispiel wird im Bedienfeld „Ausgabe“ für jede der angegebenen Dateien bei Vorhandensein `true`, ansonsten `false` angezeigt:

```
alert(fl.fileExists("file:///C:/example fla"));  
alert(fl.fileExists("file:///C:/example.jsfl"));  
alert(fl.fileExists(""));
```

fl.findDocumentDOM()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
fl.findDocumentDOM(id)
```

Parameter

id Eine Ganzzahl, die einen eindeutigen Kennzeichner für ein Dokument darstellt.

Rückgabewerte

Ein Document-Objekt oder `null`, wenn kein Dokument mit der angegebenen *id* vorhanden ist.

Beschreibung

Methode; ermöglicht die Angabe einer bestimmten Datei durch ihren eindeutigen Kennzeichner (anstatt z. B. durch ihren Indexwert). Verwenden Sie diese Methode zusammen mit `document.id`.

Beispiel

Im folgenden Beispiel wird das Lesen einer Dokument-ID und deren Verwendung zum Verweis auf das Dokument veranschaulicht:

```
var originalDocID = fl.getDocumentDOM().id;  
// other code here, maybe working in different files  
var targetDoc = fl.findDocumentDOM(originalDocID);  
// Set the height of the Stage in the original document to 400 pixels.  
targetDoc.height = 400;
```

Siehe auch

[fl.findDocumentIndex\(\)](#)

fl.findDocumentIndex()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.findDocumentIndex(name)
```

Parameter

name Der Dokumentname, dessen Index Sie suchen möchten. Das Dokument muss geöffnet sein.

Rückgabewerte

Gibt ein Array mit Ganzzahlen zurück, mit denen die Position des Dokuments *name* im Array `fl.documents` angegeben wird.

Beschreibung

Methode; gibt ein Array mit Ganzzahlen zurück, mit denen die Position des Dokuments *name* im Array `fl.documents` angegeben wird. Es ist möglich, dass mehr als ein Dokument mit dem gleichen Namen geöffnet ist, sofern die betreffenden Dokumente in unterschiedlichen Ordnern gespeichert sind.

Beispiel

Mit dem folgenden Beispiel wird veranschaulicht, wie die Indexpositionen mehrerer offener Dateien mit dem Namen „test.fl“ im Bedienfeld „Ausgabe“ angegeben werden:

```
var filename = "test.fl"
var docIndex = fl.findDocumentIndex(filename);
for (var index in docIndex)
    fl.trace(filename + " is open at index " + docIndex[index]);
```

Siehe auch

[fl.documents](#), [fl.findDocumentDOM\(\)](#)

fl.findObjectInDocByName()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
fl.findObjectInDocByName(instanceName, document)
```

Parameter

instanceName Ein String, der den Instanznamen eines Elements im angegebenen Dokument angibt.

document Das [Document-Objekt](#), in dem nach dem angegebenen Element gesucht werden soll.

Rückgabewerte

Ein Array von generischen Objekten. Verwenden Sie die Eigenschaft `.obj` der einzelnen Elemente im Array, um das Objekt abzurufen. Das Objekt hat die folgenden Eigenschaften: `keyframe`, `layer`, `timeline` und `parent`. Sie können diese Eigenschaften für den Zugriff auf die Hierarchie des Objekts verwenden. Weitere Informationen zu diesen Eigenschaften und zum Zugriff darauf finden Sie unter [fl.findObjectInDocByType\(\)](#).

Sie können auch auf Methoden und Eigenschaften für die Werte `layer` und `timeline` zugreifen; sie entsprechen dem [Layer-Objekt](#) bzw. [Timeline-Objekt](#).

Beschreibung

Methode; macht Elemente mit Instanznamen, die mit dem angegebenen Text übereinstimmen, in einem Dokument sichtbar.

Hinweis: In bestimmten Fällen funktioniert diese Methode nur, wenn sie als Befehl aus einer FLA-Datei ausgeführt wird, jedoch nicht, wenn Sie gerade die JSFL-Datei anzeigen oder bearbeiten.

Beispiel

Im folgenden Beispiel sucht das aktuelle Dokument nach Elementen mit dem Namen "instance01".

```
var nameToSearchFor = "instance01";
var doc = fl.getDocumentDOM();
var results = fl.findObjectInDocByName(nameToSearchFor, doc);
if (results.length > 0) {
    alert("success, found " + results.length + " objects");
}
else {
    alert("failed, no objects named " + nameToSearchFor + " found");
}
```

Siehe auch

[fl.findObjectInDocByType\(\)](#)

fl.findObjectInDocByType()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
fl.findObjectInDocByType(elementType, document)
```

Parameter

elementType Ein String, der den Typ des zu suchenden Elements darstellt. Informationen zu zulässigen Werten finden Sie unter [element.elementType](#).

document Das [Document-Objekt](#), in dem nach dem angegebenen Element gesucht werden soll.

Rückgabewerte

Ein Array von generischen Objekten. Verwenden Sie die Eigenschaft `.obj` der einzelnen Elemente im Array, um das Elementobjekt abzurufen. Jedes Objekt hat die folgenden Eigenschaften: `keyframe`, `layer`, `timeline` und `parent`. Sie können diese Eigenschaften für den Zugriff auf die Hierarchie des Objekts verwenden.

Sie können auch auf Methoden und Eigenschaften für die Werte `layer` und `timeline` zugreifen; sie entsprechen dem [Layer-Objekt](#) bzw. [Timeline-Objekt](#).

Das zweite und das dritte Beispiel veranschaulichen den Zugriff auf diese Eigenschaften.

Beschreibung

Methode; macht Elemente eines bestimmten Elementtyps in einem Dokument sichtbar.

Hinweis: In bestimmten Fällen funktioniert diese Methode nur, wenn sie als Befehl aus einer FLA-Datei ausgeführt wird, jedoch nicht, wenn Sie gerade die JSFL-Datei anzeigen oder bearbeiten.

Beispiel

Im folgenden Beispiel wird im aktuellen Dokument nach Textfeldern gesucht. Dann wird der Inhalt der Textfelder geändert:

```
var doc = fl.getDocumentDOM();
var typeToSearchFor = "text";
var results = fl.findObjectInDocByType(typeToSearchFor, doc);
if (results.length > 0) {
    for (var i = 0; i < results.length; i++) {
        results[i].obj.setTextString("new text");
    }
    alert("success, found " + results.length + " objects");
}
else {
    alert("failed, no objects of type " + typeToSearchFor + " found");
}
```

Im folgenden Beispiel wird der Zugriff auf die Eigenschaften des Objekts, die von dieser Methode zurückgegeben werden, beschrieben:

```
var doc = fl.getDocumentDOM();
var resultsArray = fl.findObjectInDocByType("text", doc);
if (resultsArray.length > 0)
{
    var firstItem = resultsArray[0];

    // firstItem.obj- This is the element object that was found.

    // You can access the following properties of this object:
    // firstItem.keyframe- The keyframe that the element is on.
    // firstItem.layer- The layer that the keyframe is on.
    // firstItem.timeline- The timeline that the layer is on.
    // firstItem.parent- The parent of the timeline. For example,
    // the timeline might be in a symbol instance.
}
```

Im folgenden Beispiel wird veranschaulicht, wie das DOM gesichert werden kann, um den Namen der Ebene, auf der ein Textfeld gefunden wurde, mit dem resultArray.obj-Objekt zu finden:

```
var doc = fl.getDocumentDOM();
var typeToSearchFor = "text";
var resultsArray = fl.findObjectInDocByType(typeToSearchFor, doc);
if (resultsArray.length > 0) {
    for (var i = 0; i < resultsArray.length; i++) {
        resultsArray[i].obj.setTextString("new text");
        var firstItem = resultsArray[0];
        firstItemObj = firstItem.obj;
        fl.trace(firstItemObj.layer.name+"layerName");
    }
}
else {
    alert("failed, no objects of type " + typeToSearchFor + " found");
}
```


Siehe auch

`fl.findObjectInDocByName()`

fl.flexSDKPath

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`fl.flexSDKPath`

Beschreibung

Eigenschaft; ein String, der den Pfad des Ordners „Flex SDK“ angibt, der u. a. die Ordner „bin“, „frameworks“ und „lib“ enthält. Zur Angabe der Elemente wählen Sie im Authoring-Tool „Bearbeiten“ > „Voreinstellungen“ > „ActionScript“ > „ActionScript 3.0-Einstellungen“.

Beispiel

Im folgenden Beispiel wird im Bedienfeld „Ausgabe“ der Pfad zum Ordner „Flex SDK“ angezeigt:

```
fl.trace(fl.flexSDKPath);
```

Siehe auch

`fl.externalLibraryPath`, `fl.libraryPath`, `fl.sourcePath`

fl.getAppMemoryInfo()

Verfügbarkeit

Flash 8 (nur Windows)

Verwendung

`fl.getAppMemoryInfo(memType)`

Parameter

memType Eine Ganzzahl, die den abzufragenden Speichernutzungsbereich angibt. Eine Liste der zulässigen Werte finden Sie in der folgenden Beschreibung.

Rückgabewerte

Eine Ganzzahl, die die Anzahl der Byte darstellt, die von einem bestimmten Bereich des Flash.exe-Speichers belegt sind.

Beschreibung

Methode (nur Windows); gibt eine Ganzzahl zurück, die die Anzahl der Byte darstellt, die von einem bestimmten Bereich des Flash.exe-Speichers belegt sind. Die folgende Tabelle enthält die Werte, die Sie für *memType* übergeben können:

memType	Ressourcendaten
0	PAGEFAULTCOUNT
1	PEAKWORKINGSETSIZE
2	WORKINGSETSIZE
3	QUOTAPEAKPAGEDPOOLUSAGE
4	QUOTAPAGEDPOOLUSAGE
5	QUOTAPEAKNONPAGEDPOOLUSAGE
6	QUOTANONPAGEDPOOLUSAGE
7	PAGEFILEUSAGE
8	PEAKPAGEFILEUSAGE

Beispiel

Im folgenden Beispiel wird die aktuelle Arbeitsspeicherbelegung abgerufen:

```
var memsize = fl.getAppMemoryInfo(2);  
fl.trace("Flash current memory consumption is " + memsize + " bytes or " + memsize/1024 + "  
KB");
```

fl.getDocumentDOM()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.getDocumentDOM()
```

Parameter

Keine.

Rückgabewerte

Ein Document-Objekt bzw. `null`, wenn keine Dokumente geöffnet sind.

Beschreibung

Methode; ruft das DOM ([Document-Objekt](#)) des gegenwärtig aktiven Dokuments (FLA-Datei) ab. Wenn ein oder mehrere Dokumente geöffnet sind, aber keines der Dokumente gegenwärtig das aktive Dokument ist (z. B. wenn eine JSFL-Datei aktiv ist), wird das DOM des zuletzt aktiven Dokuments abgerufen.

Beispiel

Im folgenden Beispiel wird der Name des gegenwärtig oder zuletzt aktiven Dokuments im Bedienfeld „Ausgabe“ angezeigt:

```
var currentDoc = fl.getDocumentDOM();  
fl.trace(currentDoc.name);
```

fl.getSwfPanel()

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

```
fl.getSwfPanel (panelName, [useLocalizedPanelName])
```

Parameter

panelName Der lokalisierte Name des Bedienfeldes oder der Stammdateiname der SWF-Datei des Bedienfeldes. Übergeben Sie im letzteren Fall „false“ als zweiten Parameter.

useLocalizedPanelName Optional. Der Standardwert lautet „true“. Beim Wert „false“ wird für den panelName-Parameter der englische (nicht lokalisierte) Name des Bedienfeldes angenommen, der dem Namen der SWF-Datei ohne Dateierweiterung entspricht.

Rückgabewerte

SWFPanel-Objekt.

Beschreibung

Methode; gibt das SWFPanel-Objekt auf Grundlage des lokalisierten Namens des Bedienfeldes oder des SWF-Dateinamens (ohne Dateinamenerweiterung) zurück.

Beispiel

Im folgenden Beispiel wird der Name des Bedienfelds namens „Projekt“ im Bedienfeld „Ausgabe“ angezeigt.

```
fl.trace('name of panel is: ' + fl.getSwfPanel('Project').name);
```

fl.installedPlayers

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

```
fl.installedPlayers()
```

Parameter

Keine.

Rückgabewerte

Ein Array generischer Objekte entsprechend der Liste der installierten Flash Player im Eigenschafteninspektor des Dokuments.

Beschreibung

Schreibgeschützte Eigenschaft; das Array generischer Objekte entsprechend der Liste der installierten Flash Player im Eigenschafteninspektor des Dokuments.

Jedes Objekt im Array enthält die folgenden Eigenschaften:

name Der Stringname des Dokuments.

version Kann verwendet werden, um den aktuellen Player für ein Dokument über die `Document.setPlayerVersion()`-Funktion festzulegen.

minASVersion Die ActionScript-Version, die mindestens für das Dokument erforderlich ist. Eine Ganzzahl zwischen „minASVersion“ und „maxASVersion“ kann verwendet werden, um die ActionScript-Version des Dokuments über die `Document.asVersion`-Eigenschaft festzulegen.

maxASVersion Die höchste ActionScript-Version, die vom Dokument unterstützt wird.

stageWidth Die Standardbreite der Bühne in Pixeln für das angegebene Ziel. Für das iPhone beträgt die Standardgröße beispielsweise 320 x 480 Pixel. Für Android ist die Standardgröße 480 x 800.

stageHeight Die Standardhöhe der Bühne in Pixeln für das angegebene Ziel. Für das iPhone beträgt die Standardgröße beispielsweise 320 x 480 Pixel. Für Android ist die Standardgröße 480 x 800.

Beispiel

Das folgende Beispiel erfasst die Eigenschaften aller Objekte im `installedPlayers`-Array im Ausgabefenster:

```
var arr = fl.installedPlayers;
for (var i in arr) fl.trace("name: " + arr[i].name + " version: " + arr[i].version + "
minASVersion: " + arr[i].minASVersion + " maxASVersion: " + arr[i].maxASVersion + " stageWidth:
" + arr[i].stageWidth + " stageHeight: " + arr[i].stageHeight + " ");
```

fl.isFontInstalled()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
fl.isFontInstalled(fontName)
```

Parameter

fontName Ein String, der den Namen einer Geräteschriftart angibt.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die angegebene Schriftart installiert ist; andernfalls `false`.

Beschreibung

Methode; stellt fest, ob eine angegebene Schriftart installiert ist.

Beispiel

Im folgenden Beispiel wird „true“ im Bedienfeld „Ausgabe“ angezeigt, wenn die Schriftart „Times“ installiert ist.

```
fl.trace(fl.isFontInstalled("Times"));
```

fl.languageCode

Verfügbarkeit

Flash CS5 Professional.

Verwendung

`fl.languageCode`

Beschreibung

Eigenschaft; ein String, der den fünfstelligen Code zurückgibt, der das Gebietsschema der Benutzeroberfläche der Anwendung bestimmt.

Beispiel

Das folgende Beispiel gibt den fünfstelligen Sprachcode zurück, der von der lokalisierten Benutzeroberfläche der Flash-Anwendung angegeben wird:

```
locConfigURI = fl.applicationURI + fl.languageCode + "/Configuration";
```

fl.libraryPath

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`fl.libraryPath`

Beschreibung

Eigenschaft; ein String, der eine Liste von Elementen enthält, die sich im globalen Pfad der ActionScript 3.0-Bibliothek befinden, und dieser Pfad wiederum gibt den Speicherort von SWC-Dateien oder SWC-Dateien enthaltenden Ordnern an. Elemente innerhalb des Strings sind durch Semikolons getrennt. Zur Angabe der Elemente wählen Sie im Authoring-Tool „Bearbeiten“ > „Voreinstellungen“ > „ActionScript“ > „ActionScript 3.0-Einstellungen“.

Beispiel

Im folgenden Beispiel wird der Ordner „/SWC“ dem globalen Pfad der ActionScript 3.0-Bibliothek hinzugefügt:

```
fl.trace(fl.libraryPath);  
fl.libraryPath = "/SWC;" + fl.libraryPath;  
fl.trace(fl.libraryPath);
```

Siehe auch

[fl.externalLibraryPath](#), [fl.flexSDKPath](#), [fl.sourcePath](#), [document.libraryPath](#)

fl.mapPlayerURL()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.mapPlayerURL(URI [, returnMBCS])
```

Parameter

URI Ein String, der die auskommentierte Unicode-URL enthält, die zugeordnet werden soll.

returnMBCS Ein boolescher Wert, der `true` lauten muss, wenn ein auskommentierter MBCS-Pfad zurückgegeben werden soll. Andernfalls gibt die Methode „UTF-8“ zurück. Der Standardwert lautet `false`. Dieser Parameter ist optional.

Rückgabewerte

Ein String mit der konvertierten URL.

Beschreibung

Methode; konvertiert eine Unicode-URL mit Escape-Sequenz in eine UTF-8- oder MBCS-URL. Verwenden Sie diese Methode, wenn der String in ActionScript zum Zugriff auf eine externe Ressource eingesetzt werden soll. Diese Methode ist erforderlich, wenn Multibyte-Zeichen verarbeitet werden müssen.

Beispiel

Im folgenden Beispiel wird eine URL in UTF-8 konvertiert, damit der Player sie laden kann:

```
var url = MMExecute( "fl.mapPlayerURL(" + myURL + ", false);" );  
mc.loadMovie( url );
```

fl.Math

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.Math
```

Beschreibung

Schreibgeschützte Eigenschaft; das [Math-Objekt](#) stellt Methoden für Matrix- und Punktberechnungen bereit.

Beispiel

Im folgenden Beispiel werden die Transformationsmatrix des ausgewählten Objekts und ihre Umkehrung angezeigt:

```
// Select an element on the Stage and then run this script.
var mat = fl.getDocumentDOM().selection[0].matrix;
for(var prop in mat){
fl.trace("mat."+prop+" = " + mat[prop]);
}
var invMat = fl.Math.invertMatrix( mat );
for(var prop in invMat) {
fl.trace("invMat."+prop+" = " + invMat[prop]);
}
```

fl.mruRecentFileList

Verfügbarkeit

Flash MX 2004.

Verwendung

fl.mruRecentFileList

Beschreibung

Schreibgeschützte Eigenschaft; ein Array der vollständigen Dateinamen in der MRU-Liste (Most Recently Used - Zuletzt verwendet), die vom Flash-Authoring-Tool verwaltet wird.

Beispiel

Im folgenden Beispiel werden die Anzahl der zuletzt geöffneten Dateien und die Namen der einzelnen Dateien im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace("Number of recently opened files: " + fl.mruRecentFileList.length);
for (i = 0; i < fl.mruRecentFileList.length; i++) fl.trace("file: " + fl.mruRecentFileList[i]);
```

fl.mruRecentFileType

Verfügbarkeit

Flash MX 2004.

Verwendung

fl.mruRecentFileType

Beschreibung

Schreibgeschützte Eigenschaft; ein Array der Dateitypen in der MRU-Liste, die vom Flash-Authoring-Tool verwaltet wird. Dieses Array entspricht dem Array in der Eigenschaft [fl.mruRecentFileList](#).

Beispiel

Im folgenden Beispiel werden die Anzahl der zuletzt geöffneten Dateien und der Typ der einzelnen Dateien im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace("Number of recently opened files: " + fl.mruRecentFileType.length);
for (i = 0; i < fl.mruRecentFileType.length; i++) fl.trace("type: " +
fl.mruRecentFileType[i]);
```

fl.objectDrawingMode

Verfügbarkeit

Flash 8.

Verwendung

`fl.objectDrawingMode`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob der Objektzeichnungsmodus (`true`) oder der Zeichnungsverbindungsmodus (`false`) aktiviert ist.

Beispiel

Im folgenden Beispiel wird der Status des Objektzeichnungsmodus umgeschaltet:

```
var toggleMode = fl.objectDrawingMode;
if (toggleMode) {
    fl.objectDrawingMode = false;
} else {
    fl.objectDrawingMode = true;
}
```

fl.openDocument()

Verfügbarkeit

Flash MX 2004.

Verwendung

`fl.openDocument (fileURI)`

Parameter

fileURI Ein String im Format `file:///` URI, der den Namen der zu öffnenden Datei angibt.

Rückgabewerte

Das [Document-Objekt](#) für das neu geöffnete Dokument, wenn die Methode erfolgreich ist. Wenn die Datei nicht gefunden wird oder es sich bei der Datei nicht um eine gültige FLA-Datei handelt, wird ein Fehler gemeldet und das Skript wird abgebrochen.

Beschreibung

Methode; öffnet ein Flash-Dokument (FLA-Datei) zur Bearbeitung in einem neuen Flash-Dokumentfenster und verlagert den Fokus auf dieses Dokument. Den gleichen Effekt erzielen Sie, wenn Sie den Befehl „Datei“ > „Öffnen“ wählen und dann eine Datei auswählen. Wenn die angegebene Datei bereits geöffnet ist, wird das zugehörige Dokumentfenster in den Vordergrund gestellt. Dieses Fenster wird zum aktuell ausgewählten Dokument.

Beispiel

Im folgenden Beispiel wird eine Datei mit dem Namen „Document fla“ geöffnet, die im Stammverzeichnis auf Laufwerk C gespeichert ist. Der Code speichert ein Document-Objekt, welches das Dokument in der `doc`-Variablen darstellt, und legt das Dokument als das zurzeit ausgewählte Dokument fest. Dies bedeutet, dass `fl.getDocumentDOM()` auf dieses Dokument verweist, bis der Fokus geändert wird.

```
var doc = fl.openDocument("file:///c:/Document fla");
```

fl.openScript()

Verfügbarkeit

Flash MX 2004. Optionale Parameter wurden in Flash Professional CS5 eingeführt.

Verwendung

```
fl.openScript(fileURI [, createExtension, className])
```

Parameter

fileURI Ein String im Format einer file:/// -URI, der den Pfad der JSFL-, AS-, ASC-, XML- oder TXT-Datei bzw. einer anderen Datei angibt, die in den Flash-Texteditor geladen werden soll. Dieser Parameter kann den Wert Null haben. Wenn der Wert Null lautet, öffnet diese Methode ein neues Skript mit dem Typ, der vom `createExtension`-Parameter angegeben wird.

createExtension Ein String, der bestimmt, welcher Dokumenttyp erstellt werden soll, wenn `fileURI` auf Null eingestellt ist. Der Standardwert lautet AS; zulässige Werte sind JSFL, AS, ASC, XML, TXT, AS3_CLASS und AS3_INTERFACE. Dieser Parameter wurde in Flash Professional CS5 eingeführt.

className Ein String, der den vollständig qualifizierten Klassennamen angibt, wenn eine Klasse oder eine Schnittstelle erstellt wird (wie vom `createExtension`-Parameter angegeben). Dieser Parameter wurde in Flash Professional CS5 eingeführt.

Rückgabewerte

Keine.

Beschreibung

Methode; öffnet eine vorhandene Datei oder erstellt ein neues Skript (JSFL, AS, ASC) oder eine andere Datei (XML, TXT) im Flash-Texteditor.

Beispiel

Im folgenden Beispiel wird eine Datei mit dem Namen „my_test.jsfl“ geöffnet, die im Verzeichnis „/temp“ auf Laufwerk C gespeichert ist:

```
fl.openScript("file:///c:/temp/my_test.jsfl");
```

Beispiel

Im folgenden Beispiel wird eine neue .as-Datei mit einer leeren AS3-Klassendefinition erstellt:

```
fl.openScript(null, 'AS3_CLASS');
```

fl.outputPanel

Verfügbarkeit

Flash MX 2004.

Verwendung

`fl.outputPanel`

Beschreibung

Schreibgeschützte Eigenschaft; Verweis auf das [outputPanel-Objekt](#).

Beispiel

Siehe [outputPanel-Objekt](#).

fl.packagePaths

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`fl.packagePaths`

Beschreibung

Eigenschaft; ein String, welcher der globalen Klassenpfad-Einstellung im Dialogfeld „ActionScript 2.0-Einstellungen“ entspricht. Klassenpfade innerhalb des Strings sind durch Semikolons (;) getrennt. Um die ActionScript 3.0-Klassenpfad-Einstellungen anzuzeigen oder zu ändern, verwenden Sie [fl.as3PackagePaths](#).

Beispiel

Im folgenden Beispiel wird das Ändern der ActionScript 2.0-Klassenpfad-Einstellungen veranschaulicht:

```
fl.trace(fl.packagePaths);  
// Output (assuming started with default value)  
// .;$ (LocalData)/Classes  
fl.packagePaths="buying;selling";  
fl.trace(fl.packagePaths);  
// Output  
// buying; selling
```

Siehe auch

[fl.resetPackagePaths\(\)](#)

fl.presetPanel

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`fl.presetPanel`

Beschreibung

Schreibgeschützte Eigenschaft; ein [presetPanel-Objekt](#).

fl.publishCacheDiskSizeMax

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

`fl.publishCacheDiskSizeMax`

Beschreibung

Eigenschaft; eine Ganzzahl, die die maximale Größe des Veröffentlichungscache auf der Festplatte in MB festlegt.

Beispiel

Mit dem folgenden Code wird die maximale Größe des Veröffentlichungscache auf der Festplatte auf 1 MB eingestellt:

```
fl.publishCacheDiskSizeMax = 1
```

Siehe auch

[fl.clearPublishCache\(\)](#), [fl.publishCacheEnabled](#), „[fl.publishCacheMemoryEntrySizeLimit](#)“ auf Seite 254, „[fl.publishCacheMemorySizeMax](#)“ auf Seite 254

fl.publishCacheEnabled

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

`fl.publishCacheEnabled`

Beschreibung

Eigenschaft; ein boolescher Wert, der festlegt, ob der Veröffentlichungscache aktiviert ist.

Beispiel

Der folgende Code zeigt, ob der Veröffentlichungscache im Ausgabefenster aktiviert ist.

```
fl.trace(fl.publishCacheEnabled);
```

Siehe auch

[fl.publishCacheDiskSizeMax](#), [fl.clearPublishCache\(\)](#), „[fl.publishCacheMemoryEntrySizeLimit](#)“ auf Seite 254, „[fl.publishCacheMemorySizeMax](#)“ auf Seite 254

fl.publishCacheMemoryEntrySizeLimit

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

```
fl.publishCacheMemoryEntrySizeLimit
```

Beschreibung

Eigenschaft; eine Ganzzahl, die die maximale Größe der Einträge, die dem Veröffentlichungscache im Arbeitsspeicher hinzugefügt werden können, in KB festlegt. Einträge kleiner oder gleich diesem Wert werden im Arbeitsspeicher behalten, größere Einträge werden auf die Festplatte geschrieben.

Wenn Sie über viel Arbeitsspeicher verfügen, können Sie diesen Wert erhöhen, um die Leistung zu verbessern. Ist der Arbeitsspeicher knapp, können Sie einen niedrigeren Wert angeben, um zu verhindern, dass der Veröffentlichungscache zu viel Arbeitsspeicher belegt.

Beispiel

Im folgenden Code wird die maximale Größe für Einträge im Veröffentlichungscache des Arbeitsspeichers auf 100 KB eingestellt:

```
fl.publishCacheMemoryEntrySizeLimit = 100
```

Siehe auch

[fl.publishCacheDiskSizeMax](#), [fl.publishCacheEnabled](#), „[fl.clearPublishCache\(\)](#)“ auf Seite 229, „[fl.publishCacheMemorySizeMax](#)“ auf Seite 254

fl.publishCacheMemorySizeMax

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

```
fl.publishCacheMemorySizeMax
```

Beschreibung

Eigenschaft; eine Ganzzahl, die die maximale Größe des Veröffentlichungscache im Arbeitsspeicher in MB festlegt.

Beispiel

Mit dem folgenden Code wird die maximale Größe des Veröffentlichungscache im Arbeitsspeicher auf 1 MB eingestellt:

```
fl.publishCacheMemorySizeMax = 1
```

Siehe auch

[fl.publishCacheDiskSizeMax](#), [fl.publishCacheEnabled](#), „[fl.publishCacheMemoryEntrySizeLimit](#)“ auf Seite 254, „[fl.clearPublishCache\(\)](#)“ auf Seite 229

fl.publishDocument()

Verfügbarkeit

Flash CS5 Professional.

Verwendung

```
fl.publishDocument( flaURI [, publishProfile] )
```

Parameter

flaURI Ein String, der als file:/// -URI ausgedrückt wird und den Pfad der FLA-Datei angibt, die automatisch veröffentlicht werden soll.

publishProfile Ein String, der das beim Veröffentlichen zu verwendende Veröffentlichungsprofil angibt. Wenn dieser Parameter weggelassen wird, wird das standardmäßige Veröffentlichungsprofil verwendet.

Rückgabewerte

Boolean

Beschreibung

Methode; veröffentlicht eine FLA-Datei, ohne sie zu öffnen. Diese API öffnet die FLA-Datei im automatischen Modus und veröffentlicht die SWF-Datei (oder berücksichtigt die Einstellung des Profils). Der zweite Parameter (publishProfile) ist optional. Der Rückgabewert ist ein boolescher Wert, der angibt, ob das Profil gefunden wurde oder nicht. Wenn der zweite Parameter nicht angegeben wird, ist der Rückgabewert immer „true“.

Beispiel

Im folgenden Beispiel wird der Benutzer zur Auswahl einer FLA-Datei aufgefordert, die dann automatisch mit dem standardmäßigen Veröffentlichungsprofil (Default) veröffentlicht wird:

```
var uri = fl.browseForFileURL("select", "select a FLA file to publish");  
var publishProfileName = "Default";  
fl.publishDocument(uri, publishProfileName);
```

fl.quit()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.quit( [bPromptIfNeeded] )
```

Parameter

bPromptIfNeeded Ein boolescher Wert, der `true` (Standard) lautet, wenn der Benutzer zum Speichern aller modifizierten Dokumente aufgefordert werden soll. Stellen Sie diesen Parameter auf `false` ein, wenn der Benutzer nicht zum Speichern der geänderten Dokumente aufgefordert werden soll. In letzterem Fall werden alle Änderungen in den geöffneten Dokumenten verworfen und die Anwendung wird sofort beendet. Diese Methode ist zwar bei der Stapelverarbeitung überaus nützlich, sollte aber mit Vorsicht eingesetzt werden. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; beendet Flash und fordert zuvor den Benutzer auf, alle geänderten Dokumente zu speichern.

Beispiel

Im folgenden Beispiel wird die Anwendung mit und ohne Aufforderung zum Speichern der modifizierten Dokumente beendet:

```
// Quit with prompt to save any modified documents.  
fl.quit();  
fl.quit(true); // True is optional.  
// Quit without saving any files.  
fl.quit(false);
```

fl.reloadEffects()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.reloadEffects()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; lädt alle Effektdeskriptoren neu, die im Ordner „Configuration Effects“ des Benutzers definiert sind. Dies ermöglicht es Ihnen, Skripts während der Entwicklung schnell zu ändern und Effekte ohne einen Anwendungsneustart zu verbessern. Diese Methode sollte idealerweise in einen Befehl integriert werden, den Sie im Ordner „Commands“ ablegen.

Beispiel

Das folgende Beispiel ist ein einzeliges Skript, das Sie im Ordner „Commands“ ablegen können. Wenn Sie Effekte neu laden müssen, führen Sie das Skript einfach über das Menü „Befehle“ aus.

```
fl.reloadEffects();
```

fl.reloadTools()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.reloadTools()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; baut das Bedienfeld „Werkzeuge“ anhand der Datei „toolconfig.xml“ neu auf. Diese Methode wird nur beim Erstellen erweiterbarer Werkzeuge eingesetzt. Sie kann verwendet werden, wenn das Bedienfeld „Werkzeuge“ erneut geladen werden muss, z. B. nach der Änderung einer JSFL-Datei, die bereits in diesem Bedienfeld enthalten ist.

Beispiel

Das folgende Beispiel ist ein einzeliges Skript, das Sie im Ordner „Commands“ ablegen können. Wenn Sie das Bedienfeld „Werkzeuge“ erneut laden müssen, führen Sie dieses Skript im Menü „Befehle“ aus.

```
fl.reloadTools();
```

fl.removeEventListener()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
fl.removeEventListener(eventType)
```

Parameter

eventType Ein String, der den Ereignistyp angibt, der von dieser Rückruffunktion entfernt werden soll. Zulässige Werte sind "documentNew", "documentOpened", "documentClosed", "mouseMove", "documentChanged", "layerChanged" und "frameChanged".

Rückgabewerte

Der boolesche Wert `true`, wenn der Ereignis-Listener erfolgreich entfernt wurde; `false`, wenn die Funktion nie mit der `fl.addEventListener()`-Methode zur Liste hinzugefügt wurde.

Beschreibung

Hebt die Registrierung einer Funktion auf, die mit `fl.addEventListener()` registriert wurde.

Beispiel

Im folgenden Beispiel wird der Ereignis-Listener, der dem Ereignis `documentClosed` zugeordnet ist, entfernt:

```
fl.removeEventListener("documentClosed");
```

Siehe auch

[fl.addEventListener\(\)](#)

fl.resetAS3PackagePaths()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
fl.resetAS3PackagePaths()
```

Parameter

Keine.

Beschreibung

Methode; setzt die globale Klassenpfad-Einstellung im Dialogfeld „ActionScript 3.0-Einstellungen“ auf den Standardwert zurück. Um den globalen ActionScript 2.0-Klassenpfad zurückzusetzen, verwenden Sie [fl.resetPackagePaths\(\)](#).

Beispiel

Im folgenden Beispiel wird der ActionScript 3.0-Klassenpfad auf den Standardwert zurückgesetzt:

```
fl.resetAS3PackagePaths();
```

Siehe auch

[fl.as3PackagePaths](#)

fl.resetPackagePaths()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
fl.resetPackagePaths()
```

Parameter

Keine.

Beschreibung

Methode; setzt die globale Klassenpfad-Einstellung im Dialogfeld „ActionScript 2.0-Einstellungen“ auf den Standardwert zurück. Um den globalen ActionScript 3.0-Klassenpfad zurückzusetzen, verwenden Sie [fl.resetAS3PackagePaths\(\)](#).

Beispiel

Im folgenden Beispiel wird der ActionScript 2.0-Klassenpfad auf den Standardwert zurückgesetzt:

```
fl.resetPackagePaths();
```

Siehe auch

[fl.packagePaths](#)

fl.revertDocument()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.revertDocument (documentObject)
```

Parameter

documentObject Ein [Document-Objekt](#). Wenn die Methode *documentObject* sich auf das aktive Dokument bezieht, wird das Dokumentfenster möglicherweise erst nach Beendigung des aufrufenden Skripts wiederhergestellt.

Rückgabewerte

Ein boolescher Wert; `true`, wenn der Wiederherstellen-Vorgang erfolgreich war, ansonsten `false`.

Beschreibung

Methode; stellt die zuletzt gespeicherte Version des angegebenen FLA-Dokuments wieder her. Im Gegensatz zur Menüoption „Datei“ > „Wiederherstellen“ blendet diese Methode keinen Warnhinweis ein, in dem der Benutzer zur Bestätigung des Vorgangs aufgefordert wird. Siehe auch [document.revert\(\)](#) und [document.canRevert\(\)](#).

Beispiel

Im folgenden Beispiel wird die zuletzt gespeicherte Version des aktuellen FLA-Dokuments wiederhergestellt. Alle Änderungen, die seit der letzten Speicherung vorgenommen wurden, gehen dabei verloren.

```
fl.revertDocument (fl.getDocumentDOM());
```

fl.runScript()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.runScript (fileURI [, funcName [, arg1, arg2, ...]])
```

Parameter

fileURI Ein String im Format `file:///` URI, der den Namen der auszuführenden Skriptdatei angibt.

funcName Ein String, der eine auszuführende Funktion in der JSFL-Datei angibt, die in *fileURI* festgelegt wurde. Dieser Parameter ist optional.

arg Ein optionaler Parameter, der ein oder mehrere Argumente angibt, die an *funcName* übergeben werden sollen.

Rückgabewerte

Das Ergebnis der Funktion als String, wenn *funcName* angegeben wurde, ansonsten wird nichts zurückgegeben.

Beschreibung

Methode; führt eine JavaScript-Datei aus. Wenn eine Funktion als eines der Argumente angegeben wurde, führt die Methode nicht nur die Funktion aus, sondern auch den Code in dem Skript, der sich nicht in der Funktion befindet. Der restliche Code in dem Skript wird vor der Funktion ausgeführt.

Beispiel

Im folgenden Beispiel befindet sich eine Skriptdatei mit dem Namen „testScript.jsfl“ mit folgendem Inhalt im Stammverzeichnis von Laufwerk C:

```
function testFunc(num, minNum) {  
    fl.trace("in testFunc: 1st arg: " + num + " 2nd arg: " + minNum);  
}  
for (i=0; i<2; i++) {  
    fl.trace("in for loop i=" + i);  
}  
fl.trace("end of for loop");  
// End of testScript.jsfl
```

Wenn Sie den folgenden Befehl eingeben,

```
fl.runScript("file:///C:/testScript.jsfl", "testFunc", 10, 1);
```

wird im Bedienfeld „Ausgabe“ Folgendes angezeigt:

```
in for loop i=0  
in for loop i=1  
end of for loop  
in testFunc: 1st arg: 10 2nd arg: 1
```

Sie können „testScript.jsfl“ auch aufrufen, ohne eine Funktion auszuführen:

```
fl.runScript("file:///C:/testScript.jsfl");
```

Im Bedienfeld „Ausgabe“ wird dann Folgendes angezeigt:

```
in for loop i=0  
in for loop i=1  
end of for loop
```

fl.saveAll()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.saveAll()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; speichert alle geöffneten Dokumente.

Falls eine Datei noch nicht gespeichert bzw. seit dem letzten Speichern nicht geändert wurde, wird diese Datei nicht gespeichert. Wenn Sie zulassen möchten, dass eine nicht gespeicherte bzw. nicht geänderte Datei gespeichert werden soll, verwenden Sie `fl.saveDocumentAs()`.

Beispiel

Im folgenden Beispiel werden alle geöffneten Dokumente gespeichert, die zuvor gespeichert und nach dem letzten Speichern geändert wurden:

```
fl.saveAll();
```

Siehe auch

`document.save()`, `document.saveAndCompact()`, `fl.saveDocument()`, `fl.saveDocumentAs()`

fl.saveDocument()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.saveDocument(document [, fileURI])
```

Parameter

document Ein [Document-Objekt](#), das das zu speichernde Dokument angibt. Wenn der Wert für *document* gleich `null` ist, wird das aktive Dokument gespeichert.

fileURI Ein String im Format `file:/// URI`, der den Namen der exportierten Datei angibt. Wenn der *fileURI*-Parameter `null` ist oder weggelassen wird, wird das Dokument unter seinem aktuellen Namen gespeichert. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn der Speichervorgang erfolgreich war, ansonsten `false`.

Falls die Datei noch nicht gespeichert bzw. seit dem letzten Speichern nicht geändert wurde, wird die Datei nicht gespeichert und `false` zurückgegeben. Wenn Sie zulassen möchten, dass eine nicht gespeicherte bzw. nicht geänderte Datei gespeichert werden soll, verwenden Sie `fl.saveDocumentAs()`.

Beschreibung

Methode; speichert das angegebene Dokument als FLA-Dokument.

Beispiel

Im folgenden Beispiel werden das aktuelle Dokument und zwei angegebene Dokumente gespeichert:

```
// Save the current document.
alert(fl.saveDocument(fl.getDocumentDOM()));
// Save the specified documents.
alert(fl.saveDocument(fl.documents[0], "file:///C:/example1 fla"));
alert(fl.saveDocument(fl.documents[1], "file:///C:/example2 fla"));
```

Siehe auch

`document.save()`, `document.saveAndCompact()`, `fl.saveAll()`, `fl.saveDocumentAs()`

fl.saveDocumentAs()

Verfügbarkeit

Flash MX 2004.

Verwendung

`fl.saveDocumentAs(document)`

Parameter

document Ein [Document-Objekt](#), das das zu speichernde Dokument angibt. Wenn der Wert für *document* gleich `null` ist, wird das aktive Dokument gespeichert.

Rückgabewerte

Ein boolescher Wert; `true`, wenn der Vorgang „Speichern unter“ erfolgreich war, ansonsten `false`.

Beschreibung

Methode; zeigt das Dialogfeld „Speichern unter“ für das angegebene Dokument an.

Beispiel

Mit dem folgenden Beispiel wird ein Benutzer aufgefordert, das angegebene Dokument zu speichern. Daraufhin wird eine Warnmeldung angezeigt, die angibt, ob das Dokument gespeichert wurde oder nicht:

```
alert(fl.saveDocumentAs(fl.documents[1]));
```

Siehe auch

`document.save()`, `document.saveAndCompact()`, `fl.saveAll()`, `fl.saveDocument()`

fl.scriptURI

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`fl.scriptURI`

Beschreibung

Schreibgeschützte Eigenschaft; ein String im Format `file:/// URI`, der den Pfad des zurzeit ausgeführten JSFL-Skripts angibt. Falls das Skript von `fl.runScript()` aufgerufen wurde, stellt diese Eigenschaft den Pfad des unmittelbar übergeordneten Skripts dar. Das bedeutet, es werden nicht mehrere Aufrufe von `fl.runScript()` durchlaufen, um den Pfad des ursprünglich aufrufenden Skripts zu finden.

Beispiel

Im folgenden Beispiel wird der Pfad des zurzeit ausgeführten JSFI-Skripts im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace(fl.scriptURI);
```

Siehe auch

[fl.runScript\(\)](#)

fl.selectElement()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
fl.selectElement(elementObject, editMode)
```

Parameter

elementObject Das [Element-Objekt](#), das Sie auswählen möchten.

editMode Ein boolescher Wert, der angibt, ob Sie das Element bearbeiten möchten (`true`) oder ob Sie es nur auswählen möchten (`false`).

Rückgabewerte

Ein boolescher Wert; `true`, wenn das Element erfolgreich ausgewählt wurde; andernfalls `false`.

Beschreibung

Methode, ermöglicht die Auswahl oder Bearbeitung eines Elements. Im Allgemeinen werden Sie diese Methode für Objekte, die von [fl.findObjectInDocByName\(\)](#) oder [fl.findObjectInDocByType\(\)](#) zurückgegeben werden, verwenden.

Beispiel

Im folgenden Beispiel wird ein Element namens "second text field" ausgewählt, falls es im Dokument gefunden wird:

```
var nameToSearchFor = "second text field";
var doc = fl.getDocumentDOM();

// Start by viewing Scene 1 (index value of 0).
document.editScene(0);

// Search for element by name.
var results = fl.findObjectInDocByName(nameToSearchFor, doc);
if (results.length > 0) {
    // Select the first element found.
    // Pass false, so the symbolInstance you are searching for is selected.
    // If you pass true, the symbol instance will switch to edit mode.
    fl.selectElement(results[0], false);
    alert("success, found " + results.length + " objects")
}
else {
    alert("failed, no objects with name " + nameToSearchFor + " found");
}
```

Siehe auch

[fl.findObjectInDocByName\(\)](#), [fl.findObjectInDocByType\(\)](#)

fl.selectTool()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
fl.selectTool(toolName)
```

Parameter

toolName Ein String, der den Namen des auszuwählenden Werkzeugs angibt. Informationen zu zulässigen Werten für diesen Parameter finden Sie unter „Beschreibung“ weiter unten.

Beschreibung

Methode, wählt das angegebene Werkzeug im Bedienfeld „Werkzeuge“ aus. Die zulässigen Werte für *toolName* lauten "arrow", "bezierSelect", "freeXform", "fillXform", "lasso", "pen", "penplus", "penminus", "penmodify", "text", "line", "rect", "oval", "rectPrimitive", "ovalPrimitive", "polystar", "pencil", "brush", "inkBottle", "bucket", "eyeDropper", "eraser", "hand" und "magnifier".

Wenn Sie oder andere Benutzer benutzerdefinierte Werkzeuge erstellen, können die Namen dieser Werkzeuge ebenfalls als *toolName*-Parameter übergeben werden. Eine Liste der Werkzeugnamen finden Sie in der folgenden Datei:

- Windows Vista:

Startlaufwerk\Benutzer\Benutzername\Lokale Einstellungen\Anwendungsdaten\Adobe\Flash
CS3\Sprache\Configuration\Tools\toolConfig.xml

- Windows XP:

*Startlaufwerk\Dokumente und Einstellungen\Benutzername\Lokale
Einstellungen\Anwendungsdaten\Adobe\Flash CS3\Sprache\Configuration\Tools\toolConfig.xml*

- Mac OS X:

*Macintosh HD/Users/Benutzername/Library/Application Support/Adobe/Flash
CS3/Sprache/Configuration/Tools/toolConfig.xml*

Beispiel

Im folgenden Beispiel wird das Stiftwerkzeug ausgewählt:

```
fl.selectTool("pen");
```

Siehe auch

[Tools-Objekt](#), [ToolObj-Objekt](#)

fl.setActiveWindow()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.setActiveWindow(document [, bActivateFrame])
```

Parameter

document Ein [Document-Objekt](#), das das als aktive Fenster auszuwählende Dokument angibt.

bActivateFrame Ein optionaler Parameter, der von Flash und Fireworks ignoriert wird. Er ist lediglich vorhanden, um die Kompatibilität mit Dreamweaver zu gewährleisten.

Rückgabewerte

Keine.

Beschreibung

Methode; legt fest, dass das angegebene Dokument das aktive Fenster sein soll. Diese Methode wird auch von Dreamweaver und Fireworks unterstützt. Wenn das Dokument mehrere Ansichten hat (von „Fenster“ > „Fenster duplizieren“ erstellt), wird die erste Ansicht ausgewählt.

Beispiel

Im folgenden Beispiel werden zwei Möglichkeiten zum Aktivieren eines angegebenen Dokuments demonstriert:

```
fl.setActiveWindow(fl.documents[0]);  
  
var theIndex = fl.findDocumentIndex("myFile fla");  
fl.setActiveWindow(fl.documents[theIndex]);
```

fl.showIdleMessage()

Verfügbarkeit

Flash 8.

Verwendung

```
fl.showIdleMessage(show)
```

Parameter

show Ein boolescher Wert, der angibt, ob Warnmeldungen bei zu lange dauernden Skripts aktiviert oder deaktiviert werden sollen.

Rückgabewerte

Keine.

Beschreibung

Methode; dient zum Deaktivieren der Warnmeldung bei zu lange dauernden Skripts (Sie müssen `false` für `show` übergeben). Dies bietet sich etwa bei langwierigen Stapelverarbeitungsvorgängen an. Wenn Sie die Warnmeldung wieder aktivieren möchten, geben Sie den Befehl erneut ein und übergeben Sie dabei `true` für `show`.

Beispiel

Im folgenden Beispiel wird demonstriert, wie man den Warnhinweis bei zu lange dauernden Skripts deaktiviert und reaktiviert:

```
fl.showIdleMessage(false);  
var result = timeConsumingFunction();  
fl.showIdleMessage(true); ;  
var result = timeConsumingFunction();
```

fl.sourcePath

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
fl.sourcePath
```

Beschreibung

Eigenschaft; ein String, der eine Liste von Elementen enthält, die sich im globalen ActionScript 3.0-Quellpfad befinden, und dieser Pfad wiederum gibt den Speicherort von Dateien der ActionScript-Klasse an. Elemente innerhalb des Strings sind durch Semikolons getrennt. Zur Angabe der Elemente wählen Sie im Authoring-Tool „Bearbeiten“ > „Voreinstellungen“ > „ActionScript“ > „ActionScript 3.0-Einstellungen“.

Beispiel

Im folgenden Beispiel wird der Ordner „/Classes“ dem globalen ActionScript 3.0-Quellpfad hinzugefügt:


```
fl.trace(fl.sourcePath);  
fl.sourcePath = "/Classes;" + fl.sourcePath;  
fl.trace(fl.sourcePath);
```

Siehe auch

[fl.flexSDKPath](#), [fl.externalLibraryPath](#), [fl.libraryPath](#), [document.sourcePath](#)

fl.swfPanels

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`fl.swfPanels`

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von registrierten swfPanel-Objekten (siehe [swfPanel-Objekt](#)). Ein swfPanel-Objekt ist registriert, wenn es mindestens einmal geöffnet wurde.

Die Position des Bedienfelds im Array gibt die Reihenfolge an, in der es geöffnet wurde. Wenn das als erstes geöffnete Bedienfeld „TraceBitmap“ und das als zweites geöffnete Bedienfeld „AnotherFunction“ heißt, dann handelt es sich bei `fl.swfPanels[0]` um das swfPanel-Objekt „TraceBitmap“ und bei `fl.swfPanels[1]` um das swfPanel-Objekt „AnotherFunction“ usw.

Beispiel

Im folgenden Beispiel werden im Bedienfeld „Ausgabe“ die Namen und Pfade sämtlicher registrierter Window SWF-Bedienfelder angezeigt:

```
if(fl.swfPanels.length > 0){  
    for(x = 0; x < fl.swfPanels.length; x++){  
        fl.trace("Panel: " + fl.swfPanels[x].name + " -- Path: " + fl.swfPanels[x].path);  
    }  
}
```

fl.toggleBreakpoint()

Verfügbarkeit

Flash Professional CS5.

Verwendung

`fl.toggleBreakPoint(String fileURI, int line, Boolean enable)`

Parameter

fileURI String; die URI der AS-Datei, in der der Haltepunkt umgeschaltet werden soll.

line Ganzzahl; die Nummer der Zeile, an der der Haltepunkt umgeschaltet werden soll.

enable Boolescher Wert; wenn dieser Wert „true“ lautet, ist der Haltepunkt aktiviert. Beim Wert „false“ ist der Haltepunkt deaktiviert.

Beschreibung

Schaltet einen Haltepunkt für die angegebene .as-Datei in der angegebenen Zeile um. Wenn **enable** den Wert „false“ hat, wird der Haltepunkt, der sich zurzeit in dieser Zeile befindet, gelöscht.

Beispiel

Das folgende Beispiel aktiviert einen Haltepunkt in Zeile 10 der AS-Datei im Pfad C:\AS\breakpointTest.as:

```
fl.toggleBreakPoint("file:///C:/AS/breakpointTest.as", 10, 1);
```

fl.tools

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.tools
```

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von Tools-Objekten (siehe [Tools-Objekt](#)). Diese Eigenschaft wird nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

fl.trace()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.trace(message)
```

Parameter

message Ein String, der im Bedienfeld „Ausgabe“ angezeigt wird.

Rückgabewerte

Keine.

Beschreibung

Methode; sendet einen durch einen Zeilenumbruch abgeschlossenen Textstring an das Bedienfeld „Ausgabe“ und zeigt das Bedienfeld „Ausgabe“ an, sofern es nicht bereits sichtbar ist. Diese Methode ist mit [outputPanel.trace\(\)](#) identisch und funktioniert genau wie die Anweisung `trace()` in ActionScript.

Wenn Sie eine leere Zeile senden möchten, verwenden Sie `fl.trace("")` oder `fl.trace("\n")`. Sie können letzteren Befehl inline verwenden, indem Sie `\n` in den String *message* aufnehmen.

Beispiel

Im folgenden Beispiel werden mehrere Textzeilen im Bedienfeld „Ausgabe“ angezeigt:

```
fl.outputPanel.clear();  
fl.trace("Hello World!!!");  
var myPet = "cat";  
fl.trace("\nI have a " + myPet);  
fl.trace("");  
fl.trace("I love my " + myPet);  
fl.trace("Do you have a " + myPet + "?");
```

fl.version

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.version
```

Beschreibung

Schreibgeschützte Eigenschaft; der lange Versionsstring des Flash-Authoring-Tools, einschließlich der Plattform.

Beispiel

Im folgenden Beispiel wird die Version des Flash-Authoring-Tools im Bedienfeld „Ausgabe“ angezeigt:

```
alert(fl.version); // For example, WIN 10,0,0,540
```

fl.xmlui

Verfügbarkeit

Flash MX 2004.

Verwendung

```
fl.xmlui
```

Beschreibung

Schreibgeschützte Eigenschaft; ein [XMLUI-Objekt](#). Mit dieser Eigenschaft können Sie XMLUI-Eigenschaften in einem XMLUI-Dialogfeld abrufen und einstellen sowie das Dialogfeld programmgesteuert akzeptieren oder abbrechen.

Beispiel

Siehe [XMLUI-Objekt](#).

Kapitel 18: FLfile-Objekt

Verfügbarkeit

Flash MX 2004 7.2.

Beschreibung

Mit dem FLfile-Objekt können Sie Flash-Erweiterungen schreiben, die im lokalen Dateisystem auf Dateien und Ordner zugreifen und diese modifizieren und entfernen können. Die FLfile-API wird der JavaScript-API in Form einer Erweiterung bereitgestellt. Diese Erweiterung, die man als *gemeinsame Bibliothek* bezeichnet, befindet sich in folgendem Ordner:

- Windows Vista:

Startlaufwerk\Benutzer*Benutzername*\Lokale Einstellungen\Anwendungsdaten\Adobe\Flash CS3*Sprache*\Configuration\External Libraries\FLfile.dll

- Windows XP:

Startlaufwerk\Dokumente und Einstellungen*Benutzername*\Lokale Einstellungen\Anwendungsdaten\Adobe\Flash CS3*Sprache*\Configuration\External Libraries\FLfile.dll

- Mac OS X:

Macintosh HD/Users/*Benutzername*/Library/Application Support/Adobe/Flash CS3/*Sprache*/Configuration/External Libraries/FLfile.dll

Hinweis: Verwechseln Sie die gemeinsamen Bibliotheken in Flash-Dokumenten, die Symbole enthalten, nicht mit den gemeinsamen JavaScript-API-Bibliotheken. Sie haben nichts miteinander zu tun.

Die FLfile-Methoden wirken sich auf Dateien oder Ordner (Verzeichnisse) auf der Festplatte aus. Jede Methode erwartet daher einen oder mehrere Parameter, der den Speicherort einer Datei oder eines Ordners angibt. Der Speicherort einer Datei oder eines Ordners wird in Form eines Strings angegeben, der ähnlich wie eine Website-URL aussieht. Man bezeichnet diesen String als Datei-URI (Uniform Resource Identifier). Er sieht folgendermaßen aus (einschließlich der Anführungszeichen):

```
"file:///drive|/folder 1/folder 2/.../filename"
```

Wenn Sie zum Beispiel auf dem Laufwerk C einen Ordner mit dem Namen „config“ erstellen und im Ordner „Program Files/MyApp“ ablegen möchten, geben Sie Folgendes ein:

```
FLfile.createFolder("file:///C:/Program Files/MyApp/config");
```

Wenn Sie dann eine Datei mit der Bezeichnung „config.ini“ in diesem Ordner ablegen möchten, geben Sie folgenden Befehl ein:

```
FLfile.write("file:///C:/Program Files/MyApp/config/config.ini", "");
```

Um einen Ordner auf dem Macintosh zu erstellen, können Sie einen Befehl eingeben, der beispielsweise folgendermaßen lautet:

```
FLfile.createFolder("file:///Macintosh/MyApp/config");
```

Übersicht über Methoden

Für das FLfile-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>FLfile.copy()</code>	Kopiert eine Datei.
<code>FLfile.createFolder()</code>	Erstellt einen oder mehrere Ordner.
<code>FLfile.exists()</code>	Stellt fest, ob eine Datei oder ein Ordner vorhanden ist.
<code>FLfile.getAttributes()</code>	Stellt fest, ob eine Datei schreibbar, schreibgeschützt, versteckt, sichtbar oder ein Systemordner ist.
<code>FLfile.getCreationDate()</code>	Gibt an, wie viele Sekunden seit dem 1. Januar 1970 bis zur Erstellung der Datei bzw. des Ordners vergangen sind.
<code>FLfile.getCreationDateObj()</code>	Ruft das Datum ab, an dem eine Datei bzw. ein Ordner erstellt wurde.
<code>FLfile.getModificationDate()</code>	Gibt an, wie viele Sekunden seit dem 1. Januar 1970 bis zur letzten Änderung der Datei bzw. des Ordners vergangen sind.
<code>FLfile.getModificationDateObj()</code>	Ruft das Datum ab, an dem eine Datei bzw. ein Ordner zuletzt geändert wurde.
<code>FLfile.getSize()</code>	Ruft die Größe einer Datei ab.
<code>FLfile.listFolder()</code>	Listet den Inhalt eines Ordners auf.
<code>FLfile.platformPathToURI()</code>	Konvertiert einen Dateinamen in das plattformspezifische Format einer file:/// -URI.
<code>FLfile.read()</code>	Liest den Inhalt einer Datei.
<code>FLfile.remove()</code>	Löscht eine Datei oder einen Ordner.
<code>FLfile.setAttributes()</code>	Legt fest, dass eine Datei oder ein Ordner schreibgeschützt, schreibbar, versteckt oder sichtbar sein soll.
<code>FLfile.uriToPlatformPath()</code>	Konvertiert einen Dateinamen als file:/// -URI in ein plattformspezifisches Format.
<code>FLfile.write()</code>	Erstellt oder ergänzt eine Datei bzw. schreibt in eine Datei.

FLfile.copy()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

```
FLfile.copy(fileURI, copyURI)
```

Parameter

fileURI Ein String im Format file:/// URI, der die zu kopierende Datei angibt.

copyURI Ein String im Format file:/// URI, der den Speicherort und den Namen der kopierten Datei angibt.

Rückgabewerte

Der boolesche Wert `true`, wenn erfolgreich, ansonsten `false`.

Beschreibung

Methode; kopiert eine Datei von einem Speicherort an einen anderen. Diese Methode gibt `false` zurück, wenn `copyURI` bereits vorhanden ist.

Beispiel

Im folgenden Beispiel wird von einer Konfigurationsdatei mit dem Namen „config.ini“ eine Sicherungskopie erstellt und unter einem neuen Namen in demselben Ordner wie die Originaldatei abgelegt:

```
var originalFileURI="file:///C:/Program Files/MyApp/config.ini";
var newFileURI="file:///C:/Program Files/MyApp/config_backup.ini";
FLfile.copy(originalFileURI, newFileURI);
```

Auf Wunsch lässt sich dieser Vorgang auch mit einem einzigen Befehl ausführen:

```
FLfile.copy("file:///C:/Program Files/MyApp/config.ini", file:///C:/Program
Files/MyApp/config_backup.ini");
```

FLfile.createFolder()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

```
FLfile.createFolder(folderURI)
```

Parameter

folderURI Eine Ordner-URI, die die zu erstellende Ordnerstruktur angibt.

Rückgabewerte

Der boolesche Wert `true`, wenn erfolgreich und `false`, wenn `folderURI` bereits vorhanden ist.

Beschreibung

Methode; erstellt einen bzw. mehrere Ordner am angegebenen Speicherort.

Sie können mehrere Ordner auf einmal erstellen. So erstellt beispielsweise der folgende Befehl die beiden Ordner „MyData“ und „TempData“, sofern diese noch nicht vorhanden sind:

```
FLfile.createFolder("file:///c:/MyData/TempData")
```

Beispiel

Mit dem folgenden Beispiel werden im Konfigurationsordner ein Verzeichnis und ein Unterverzeichnisse erstellt ([fl.configURI](#)):

```
fl.trace(FLfile.createFolder(fl.configURI+"folder01/subfolder01"));
```

Im folgenden Beispiel wird versucht, auf der Stammebene des Laufwerks C: einen Ordner mit dem Namen „tempFolder“ zu erstellen. Anschließend wird eine Meldung eingeblendet, aus der hervorgeht, ob der Vorgang erfolgreich war:

```
var folderURI = "file:///c:/tempFolder";
if (FLfile.createFolder(folderURI)) {
    alert("Created " + folderURI);
}
else {
    alert(folderURI + " already exists");
}
```

Siehe auch

[FLfile.remove\(\)](#), [FLfile.write\(\)](#)

FLfile.exists()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

`FLfile.exists(fileURI)`

Parameter

fileURI Ein String im Format file:/// URI, der die zu prüfende Datei angibt.

Rückgabewerte

Der boolesche Wert `true`, wenn erfolgreich, ansonsten `false`.

Beschreibung

Methode; stellt fest, ob eine angegebene Datei vorhanden ist. Wenn Sie einen Ordner und einen Dateinamen angeben, muss der Ordner bereits vorhanden sein. Wenn Sie also zunächst einen Ordner erstellen möchten, informieren Sie sich unter [FLfile.createFolder\(\)](#).

Beispiele

Im folgenden Beispiel wird festgestellt, ob im Ordner „temp“ eine Datei mit dem Namen „mydata.txt“ vorhanden ist, und dann wird eine entsprechende Meldung angezeigt:

```
var fileURI = "file:///c:/temp/mydata.txt";
if (FLfile.exists(fileURI)) {
    alert( fileURI + " exists.");
}
else {
    alert( fileURI + " does not exist.");
}
```

Im folgenden Beispiel wird geprüft, ob im Ordner „MyApplication“ eine erforderliche Konfigurationsdatei vorhanden ist. Sollte die Datei nicht existieren, wird sie erstellt.

```
var configFile = "file:///C:/MyApplication/config.ini";
if (!FLfile.exists(configFile)) {
    FLfile.write(configFile, "");
}
```

Siehe auch

[FLfile.write\(\)](#)

FLfile.getAttributes()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

```
FLfile.getAttributes(fileOrFolderURI)
```

Parameter

fileOrFolderURI Ein String im Format file:/// URI, der die Datei bzw. den Ordner angibt, deren oder dessen Attribute Sie abrufen möchten.

Rückgabewerte

Ein String, der die Attribute der angegebenen Datei bzw. des angegebenen Ordners darstellt.

Sollte die Datei bzw. der Ordner nicht vorhanden sein, können unerwartete Ergebnisse auftreten. Sie sollten daher [FLfile.exists\(\)](#) aufrufen, bevor Sie diese Methode verwenden.

Beschreibung

Methode; gibt einen String zurück, der die Attribute der angegebenen Datei bzw. des Ordners darstellt. Falls die Datei über keine speziellen Attribute verfügt (also beispielsweise nicht schreibgeschützt oder versteckt ist), wird ein leerer String zurückgegeben. Bevor Sie diese Methode verwenden, sollten Sie immer mithilfe von [FLfile.exists\(\)](#) prüfen, ob die Datei bzw. der Ordner vorhanden ist.

In dem zurückgegebenen String werden die Attribute durch folgende Zeichen angegeben:

- R (Read-only) – *fileOrFolderURI* ist schreibgeschützt
- D (Directory) – *fileOrFolderURI* ist ein Ordner (bzw. ein Verzeichnis)
- H (Hidden) – *fileOrFolderURI* ist versteckt (nur bei Windows)
- S (System) – *fileOrFolderURI* ist eine Systemdatei bzw. ein Systemordner (nur bei Windows)
- A (Ready for archiving) – *fileOrFolderURI* kann archiviert werden (nur bei Windows)

Wenn *fileOrFolderURI* z. B. ein versteckter Ordner ist, wird der String "DH" zurückgegeben.

Beispiel

Im folgenden Beispiel werden die Attribute der Datei „mydata.txt“ abgerufen und wenn die Datei schreibgeschützt ist, wird eine Warnmeldung angezeigt:

```
var URI = "file:///c:/temp/mydata.txt";
if (FLfile.exists(URI)) {
    var attr = FLfile.getAttributes(URI);
    if (attr && (attr.indexOf("R") != -1)) { // Returned string contains R.
        alert(URI + " is read only!");
    }
}
```


Siehe auch

`FLfile.setAttributes()`

FLfile.getCreationDate()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

`FLfile.getCreationDate(fileOrFolderURI)`

Parameter

fileOrFolderURI Ein String im Format file:/// URI, der die Datei bzw. den Ordner angibt, deren bzw. dessen Erstellungsdatum und -uhrzeit Sie in Form eines hexadezimalen Strings abrufen möchten.

Rückgabewerte

Ein String mit einer hexadezimalen Zahl, die die Anzahl der Sekunden angibt, die seit dem 1. Januar 1970 bis zur Erstellung der Datei bzw. des Ordners vergangen sind. Falls die Datei bzw. der Ordner nicht vorhanden ist, wird "00000000" zurückgegeben.

Beschreibung

Methode; gibt an, wie viele Sekunden seit dem 1. Januar 1970 bis zur Erstellung der Datei bzw. des Ordners vergangen sind. Mithilfe dieser Methode werden in erster Linie die Erstellungs- bzw. Änderungsdaten von Dateien oder Ordnern verglichen.

Beispiel

Im folgenden Beispiel wird festgestellt, ob eine Datei seit ihrer Erstellung geändert wurde:

```
// Make sure the specified file exists
var fileURI = "file:///C:/MyApplication/MyApp.fla";
var creationTime = FLfile.getCreationDate(fileURI);
var modificationTime = FLfile.getModificationDate(fileURI);
if ( modificationTime > creationTime ) {
    alert("The file has been modified since it was created.");
}
else {
    alert("The file has not been modified since it was created.");
}
```

Siehe auch

`FLfile.getCreationDateObj()`, `FLfile.getModificationDate()`

FLfile.getCreationDateObj()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

```
FLfile.getCreationDateObj(fileOrFolderURI)
```

Parameter

fileOrFolderURI Ein String im Format file:/// URI, der die Datei bzw. den Ordner angibt, deren bzw. dessen Erstellungsdatum und -uhrzeit Sie in Form eines JavaScript Date-Objekts abrufen möchten.

Rückgabewerte

Ein JavaScript Date-Objekt mit dem das Datum und die Uhrzeit der Erstellung der angegebenen Datei bzw. des angegebenen Ordners dargestellt werden. Falls die Datei nicht vorhanden ist, enthält das Objekt die Angabe, dass die Datei bzw. der Ordner am 31. Dezember 1969 um 0.00 Uhr GMT erstellt wurde.

Beschreibung

Methode; gibt ein JavaScript Date-Objekt zurück, mit dem das Datum und die Uhrzeit der Erstellung der angegebenen Datei bzw. des angegebenen Ordners dargestellt werden.

Beispiel

Mit dem folgenden Beispiel wird im Bedienfeld „Ausgabe“ in normal lesbarer Form das Erstellungsdatum einer Datei angezeigt:

```
// Make sure the specified file exists.  
var file1Date = FLfile.getCreationDateObj("file:///c:/temp/file1.txt");  
fl.trace(file1Date);
```

Siehe auch

[FLfile.getCreationDate\(\)](#), [FLfile.getModificationDateObj\(\)](#)

FLfile.getModificationDate()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

```
FLfile.getModificationDate(fileOrFolderURI)
```

Parameter

fileOrFolderURI Ein String im Format file:/// URI, der die Datei angibt, deren Änderungsdatum und -uhrzeit Sie in Form eines hexadezimalen Strings abrufen möchten.

Rückgabewerte

Ein String mit einer hexadezimalen Zahl, die die Anzahl der Sekunden angibt, die seit dem 1. Januar 1970 bis zur letzten Änderung der Datei bzw. des Ordners vergangen sind. Falls die Datei bzw. der Ordner nicht vorhanden ist, wird "00000000" zurückgegeben.

Beschreibung

Methode; gibt an, wie viele Sekunden seit dem 1. Januar 1970 bis zur letzten Änderung der Datei bzw. des Ordners vergangen sind. Mithilfe dieser Methode werden in erster Linie die Erstellungs- bzw. Änderungsdaten von Dateien oder Ordnern verglichen.

Beispiel

Im folgenden Beispiel werden die Änderungsdaten zweier Dateien miteinander verglichen, um festzustellen, welche Datei zuletzt geändert wurde:

```
// Make sure the specified files exist.
file1 = "file:///C:/MyApplication/MyApp fla";
file2 = "file:///C:/MyApplication/MyApp.as";
modificationTime1 = FLfile.getModificationDate(file1);
modificationTime2 = FLfile.getModificationDate(file2) ;
if(modificationTime1 > modificationTime2) {
    alert("File 2 is older than File 1") ;
}
else if(modificationTime1 < modificationTime2) {
    alert("File 1 is older than File 2") ;
}
else {
    alert("File 1 and File 2 were saved at the same time") ;
}
```

Siehe auch

[FLfile.getCreationDate\(\)](#), [FLfile.getModificationDateObj\(\)](#)

FLfile.getModificationDateObj()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

`FLfile.getModificationDateObj(fileOrFolderURI)`

Parameter

fileOrFolderURI Ein String im Format file:/// URI, der die Datei bzw. den Ordner angibt, deren bzw. dessen Änderungsdatum und -uhrzeit Sie in Form eines JavaScript Date-Objekts abrufen möchten.

Rückgabewerte

Ein JavaScript Date-Objekt mit dem das Datum und die Uhrzeit der letzten Änderung der angegebenen Datei bzw. des angegebenen Ordners dargestellt werden. Falls die Datei oder der Ordner nicht vorhanden ist, enthält das Objekt die Angabe, dass die Datei bzw. der Ordner am 31. Dezember 1969 um 0.00 Uhr GMT erstellt wurde.

Beschreibung

Methode; gibt ein JavaScript Date-Objekt zurück, mit dem das Datum und die Uhrzeit der letzten Änderung der angegebenen Datei bzw. des angegebenen Ordners dargestellt werden.

Beispiel

Mit dem folgenden Beispiel wird im Bedienfeld „Ausgabe“ in normal lesbarer Form das Datum der letzten Änderung einer Datei angezeigt:

```
// Make sure the specified file exists.  
var file1Date = FLfile.getModificationDateObj("file:///c:/temp/file1.txt");  
trace(file1Date);
```

Siehe auch

[FLfile.getCreationDateObj\(\)](#), [FLfile.getModificationDate\(\)](#)

FLfile.getSize()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

```
FLfile.getSize(fileURI)
```

Parameter

fileURI Ein String im Format file:/// URI, der die Datei angibt, deren Größe Sie abrufen möchten.

Rückgabewerte

Gibt eine Ganzzahl zurück, die die Größe der angegebenen Datei in Byte darstellt, bzw. 0, wenn die Datei nicht vorhanden ist.

Beschreibung

Methode; gibt eine Ganzzahl zurück, die die Größe der angegebenen Datei in Byte darstellt, bzw. 0, wenn die Datei nicht vorhanden ist. Falls der Rückgabewert 0 lautet, können Sie mithilfe von [FLfile.exists\(\)](#) feststellen, ob es sich bei der Datei um eine Null-Byte-Datei handelt oder diese Datei gar nicht vorhanden ist.

Diese Methode gibt nur dann die richtigen Werte für die Dateigröße zurück, wenn die Datei nicht größer als 2 GB ist.

Beispiel

Mit dem folgenden Beispiel wird die Größe der Datei „mydata.txt“ in der Variablen `fileSize` gespeichert:

```
var URL = "file:///c:/temp/mydata.txt";  
var fileSize = FLfile.getSize(URL);
```

FLfile.listFolder()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

```
FLfile.listFolder(folderURI [, filesOrDirectories])
```

Parameter

folderURI Ein String im Format file:/// URI, der den Ordner angibt, dessen Inhalte Sie abrufen möchten. Die *folderURI* kann eine Platzhaltermaske enthalten. Die gültigen Platzhalter sind * (für ein oder mehrere Zeichen) und ? (für ein Zeichen).

filesOrDirectories Ein optionaler String, der angibt, ob nur Dateinamen oder nur Ordnernamen (Verzeichnisnamen) zurückgegeben werden sollen. Wenn dieser Parameter fehlt, werden sowohl Dateinamen als auch Ordnernamen zurückgegeben. Die zulässigen Werte lauten "files" und "directories".

Rückgabewerte

Ein Array von Strings, die den Inhalt des Ordners darstellen. Wenn der Ordner nicht vorhanden ist oder wenn keine Dateien oder Ordner dem angegebenen Kriterium entsprechen, wird ein leeres Array zurückgegeben.

Beschreibung

Methode; gibt ein Array von Strings zurück, die den Inhalt des Ordners darstellen.

Beispiele

Im folgenden Beispiel werden drei Arrays zurückgegeben. Das erste repräsentiert alle Dateien im Ordner „C:\temp“, das zweite repräsentiert alle Ordner im Ordner „C:\temp“ und das dritte repräsentiert alle Dateien und Ordner im Ordner „C:\temp“:

```
var fileURI = "file:///C:/temp/" ;
var folderURI = "file:///C:/temp" ;
var fileList1 = FLfile.listFolder(fileURI, "files"); // files
var fileList2 = FLfile.listFolder(folderURI, "directories"); //folders
var fileList3 = FLfile.listFolder(folderURI); //files and folders
fl.trace("Files: " + fileList1);
fl.trace("");
fl.trace("Folders:  " + fileList2);
fl.trace("");
fl.trace("Files and folders: " + fileList3);
```

Im folgenden Beispiel wird ein Array aller Textdateien (.txt) zurückgegeben, die sich im Ordner „temp“ befinden, und die Dateiliste wird in einem Hinweisfeld angezeigt:

```
var folderURI = "file:///c:/temp";
var fileMask = "*.txt";
var list = FLfile.listFolder(folderURI + "/" + fileMask, "files");
if (list) {
    alert(folderURI + " contains: " + list.join(" "));
}
```

Im folgenden Beispiel wird in der angegebenen *folderURI* eine Dateimaske verwendet, damit die Namen aller ausführbaren Dateien zurückgegeben werden, die sich im Windows-Anwendungsordner befinden:

```
var executables = FLfile.listFolder("file:///C:/WINDOWS/*.exe", "files");
alert(executables.join("\n"));
```

FLfile.platformPathToURI()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
FLfile.platformPathToURI(fileName)
```

Parameter

fileName Ein String in einem plattformspezifischen Format, der den zu konvertierenden Dateinamen angibt.

Rückgabewerte

Ein String im Format file:/// URI.

Beschreibung

Methode; konvertiert einen Dateinamen in einem plattformspezifischen Format in das Format file:/// URI.

Beispiel

Im folgenden Beispiel wird ein Dateiname aus einem plattformspezifischen Format in eine file:/// -URI konvertiert, die dann an `outputPanel.save()` übergeben wird:

```
var myFilename = "C:\\outputPanel.txt";  
var myURI=FLfile.platformPathToURI(myFilename);  
fl.outputPanel.save(myURI);
```

Siehe auch

[FLfile.uriToPlatformPath\(\)](#)

FLfile.read()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

```
FLfile.read()
```

Parameter

fileOrFolderURI Ein String im Format file:/// URI, der die Datei bzw. den Ordner angibt, deren oder dessen Attribute Sie abrufen möchten.

Rückgabewerte

Der Inhalt der angegebenen Datei in Form eines Strings bzw. `null`, wenn der Lesevorgang fehlschlägt.

Beschreibung

Methode; gibt den Inhalt der angegebenen Datei in Form eines Strings zurück. Wenn der Lesevorgang fehlschlägt, wird `null` zurückgegeben.

Beispiele

Im folgenden Beispiel wird die Datei „mydata.txt“ gelesen und bei einem erfolgreichen Lesevorgang ein Hinweisfeld mit dem Inhalt der Datei angezeigt:

```
var fileURI = "file:///c:/temp/mydata.txt";
var str = FLfile.read( fileURI);
if (str) {
    alert( fileURL + " contains: " + str);
}
```

Im folgenden Beispiel wird der ActionScript-Code aus einer Klassendatei gelesen und in der Variablen `code` gespeichert:

```
var classFileURI = "file:///C:/MyApplication/TextCarousel.as";
var code = FLfile.read(classFileURI);
```

FLfile.remove()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

```
FLfile.remove(fileOrFolderURI)
```

Parameter

fileOrFolderURI Ein String im Format file:/// URI, der die Datei oder den Ordner angibt, die bzw. den Sie entfernen (löschen) möchten.

Rückgabewerte

Der boolesche Wert `true`, wenn erfolgreich, ansonsten `false`.

Beschreibung

Methode; löscht die angegebene Datei oder den angegebenen Ordner. Wenn der Ordner Dateien enthält, werden diese Dateien ebenfalls gelöscht. Dateien mit dem Attribut R (schreibgeschützt) können nicht entfernt werden.

Beispiele

Im folgenden Beispiel wird ein Warnhinweis angezeigt, wenn eine Datei vorhanden ist. Die Datei wird gelöscht, wenn der Benutzer dies wünscht:

```
var fileURI = prompt ("Enter file/folder to be deleted: ", "file:///c:/temp/delete.txt");
if (FLfile.exists(fileURI)) {
    var confirm = prompt("File exists. Delete it? (y/n)", "y");
    if (confirm == "y" || confirm == "Y") {
        if(FLfile.remove(fileURI)) {
            alert(fileURI + " is deleted.");
        }
        else {
            alert("fail to delete " + fileURI);
        }
    }
}
else {
    alert(fileURI + " does not exist");
}
```

Im folgenden Beispiel wird eine von einer Anwendung erstellte Konfigurationsdatei gelöscht:

```
if (FLfile.remove("file:///C:/MyApplication/config.ini")) {  
    alert("Configuration file deleted");  
}
```

Im folgenden Beispiel wird der Ordner „Configuration“ mitsamt Inhalt gelöscht:

```
FLfile.remove("file:///C:/MyApplication/Configuration/");
```

Siehe auch

[FLfile.createFolder\(\)](#), [FLfile.getAttributes\(\)](#)

FLfile.setAttributes()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

```
FLfile.setAttributes(fileURI, strAttrs)
```

Parameter

fileURI Ein String im Format file:/// URI, der die Datei angibt, deren Attribute Sie festlegen möchten.

strAttrs Ein String, der die Werte der festzulegenden Attribute angibt. Eine Liste der für *strAttrs* zulässigen Werte finden Sie weiter unten im Abschnitt „Beschreibung“.

Rückgabewerte

Der boolesche Wert `true`, wenn erfolgreich.

Hinweis: Sollte die Datei bzw. der Ordner nicht vorhanden sein, können unerwartete Ergebnisse auftreten. Sie sollten daher [FLfile.exists\(\)](#) aufrufen, bevor Sie diese Methode verwenden.

Beschreibung

Methode; legt für die angegebene Datei systembezogene Attribute fest.

Folgende Werte sind für *strAttrs* zulässig:

- **N** (No specific attributes) – keine besonderen Attribute (nicht schreibgeschützt, nicht versteckt usw.)
- **A** (Ready for archiving) – kann archiviert werden (nur bei Windows)
- **R** (Read-only) – ist schreibgeschützt (bei Macintosh ist „schreibgeschützt“ gleichbedeutend mit „gesperrt“)
- **W** (Writable) – überschreibbar (setzt **R** außer Kraft)
- **H** (Hidden) – ist versteckt (nur bei Windows)
- **V** (Visible) – sichtbar (setzt **H** außer Kraft, nur bei Windows)

Wenn Sie sowohl **R** und **W** für *strAttrs* angeben, wird **R** ignoriert und die Datei als überschreibbar definiert. Wenn Sie in ähnlicher Weise gleichzeitig **H** und **V** übergeben, wird **H** ignoriert und die Datei als sichtbar definiert.

Falls Sie sicherstellen möchten, dass bei einer Datei keine Attribute gesetzt sind, verwenden Sie diesen Befehl zuerst mit dem Parameter **N**, bevor Sie die gewünschten Attribute vergeben. Allerdings gibt es keinen unmittelbaren Gegenparameter für **A**, mit dem das Archiv-Attribut deaktiviert werden könnte.

Beispiele

Im folgenden Beispiel wird festgelegt, dass die Datei „mydata.txt“ schreibgeschützt und versteckt sein soll. Das Archiv-Attribut bleibt hiervon unberührt.

```
var URI = "file:///c:/temp/mydata.txt";
if (FLfile.exists(URI)) {
    FLfile.setAttributes(URI, "RH");
}
```

Im folgenden Beispiel wird festgelegt, dass die Datei „mydata.txt“ schreibgeschützt und versteckt sein soll. Auch hierbei wird das Archiv-Attribut nicht verändert.

```
var URI = "file:///c:/temp/mydata.txt";

if (FLfile.exists(URI)) {
    FLfile.setAttributes(URI, "N");
    FLfile.setAttributes(URI, "RH");
}
```

Siehe auch

[FLfile.getAttributes\(\)](#)

FLfile.uriToPlatformPath()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
FLfile.uriToPlatformPath(fileURI)
```

Parameter

fileURI Ein String im Format file:/// URI, der den zu konvertierenden Dateinamen angibt.

Rückgabewerte

Ein String, der einen plattformspezifischen Pfad darstellt.

Beschreibung

Methode; konvertiert einen Dateinamen im Format file:/// URI in ein plattformspezifisches Format.

Beispiel

Im folgenden Beispiel wird eine file:/// -URI in ein plattformspezifisches Format konvertiert:

```
var dir = (fl.configDirectory);
var URI = FLfile.platformPathToURI(dir);
fl.trace(URI == fl.configURI); // displays "true"
```

Siehe auch

[FLfile.platformPathToURI\(\)](#)

FLfile.write()

Verfügbarkeit

Flash MX 2004 7.2.

Verwendung

```
FLfile.write(fileURI, textToWrite, [ , strAppendMode])
```

Parameter

fileURI Ein String im Format file:/// URI, der die Datei angibt, in die Sie Daten schreiben möchten.

textToWrite Ein String mit dem Text, den Sie in der Datei platzieren möchten.

strAppendMode Ein optionaler String mit dem Wert "append", der angibt, dass Sie *textToWrite* an die vorhandene Datei anhängen möchten. Fehlt dieser String, wird *fileURI* durch *textToWrite* überschrieben.

Rückgabewerte

Der boolesche Wert `true`, wenn erfolgreich, ansonsten `false`.

Beschreibung

Methode; fügt den angegebenen String in der angegebenen Datei ein (im UTF-8-Format). Wenn die Datei nicht vorhanden ist, wird sie erstellt. Allerdings muss der Ordner, in dem Sie die Datei speichern möchten, vorhanden sein, bevor Sie diese Methode verwenden können. Wenn Sie also zunächst einen Ordner erstellen möchten, verwenden Sie [FLfile.createFolder\(\)](#).

Beispiel

Im folgenden Beispiel wird versucht, den String "xxx" in die Datei „mydata.txt“ einzufügen. Wenn der Vorgang erfolgreich abgeschlossen wurde, wird eine Meldung angezeigt. Anschließend wird versucht, den String "aaa" an die Datei anzuhängen. Wenn der Vorgang erfolgreich abgeschlossen ist, wird eine zweite Meldung ausgegeben. Nach der Ausführung dieses Skripts enthält die Datei „mydata.txt“ lediglich den Text "xxxaaa".

```
var URI = "file:///c:/temp/mydata.txt";
if (FLfile.write(URI, "xxx")) {
    alert("Wrote xxx to " + URI);
}
if (FLfile.write(URI, "aaa", "append")) {
    alert("Appended aaa to " + fileURI);
}
```

Siehe auch

[FLfile.createFolder\(\)](#), [FLfile.exists\(\)](#)

Kapitel 19: folderItem-Objekt

Vererbung [Item-Objekt](#) > folderItem-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das folderItem-Objekt ist eine Unterklasse des Item-Objekts. Für folderItem gibt es keine spezifischen Methoden oder Eigenschaften. Siehe [Item-Objekt](#).

Kapitel 20: fontItem-Objekt

Vererbung [Item-Objekt](#) > fontItem-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das fontItem-Objekt ist eine Unterklasse des Item-Objekts (siehe [Item-Objekt](#)).

Übersicht über Eigenschaften

Abgesehen von den Eigenschaften des Item-Objekts stehen für das fontItem-Objekt die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
fontItem.bitmap	Gibt an, ob das Font-Element als Bitmap dargestellt wird.
fontItem.bold	Gibt an, ob das Font-Element fett formatiert ist.
fontItem.embeddedCharacters	Gibt die einzubettenden Zeichen an.
fontItem.embedRanges	Gibt die Elemente an, die im Dialogfeld für die Schriftarteinbettung ausgewählt werden können.
fontItem.embedVariantGlyphs	Gibt an, ob Glyphenvarianten bei der Veröffentlichung einer SWF-Datei in der Schrift ausgegeben werden sollten.
fontItem.font	Der Name der Geräteschriftart, die dem Font-Element zugewiesen ist.
fontItem.isDefineFont4Symbol	Gibt das Format der Schrift an, die bei der Veröffentlichung einer SWF-Datei ausgegeben wird.
fontItem.italic	Gibt an, ob das Font-Element kursiv formatiert ist.
fontItem.size	Die Größe des Font-Elements in Punkt.

fontItem.bitmap

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`fontItem.bitmap`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob das Font-Element als Bitmap dargestellt wird (`true`) oder nicht (`false`).

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Font-Element handelt, wird im folgenden Beispiel im Bedienfeld „Ausgabe“ `true` angezeigt, falls es als Bitmap dargestellt wird, andernfalls `false`:

```
var theItem = fl.getDocumentDOM().library.items[0];  
fl.trace("bitmap: " + theItem.bitmap);
```

fontItem.bold

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`fontItem.bold`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob das Font-Element fett dargestellt wird (`true`) oder nicht (`false`).

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Font-Element handelt, wird im folgenden Beispiel im Bedienfeld „Ausgabe“ `true` angezeigt, falls es fett dargestellt wird, andernfalls `false`; dann wird die Fettdarstellung festgelegt.

```
var theItem = fl.getDocumentDOM().library.items[0];  
fl.outputPanel.clear();  
fl.trace("bold: " + theItem.bold);  
theItem.bold=true;  
fl.trace("bold: " + theItem.bold);
```

fontItem.embeddedCharacters

Verfügbarkeit

Flash CS5 Professional.

Verwendung

`fontItem.embeddedCharacters`

Beschreibung

Eigenschaft; ein Stringwert, mit dem Sie die Zeichen angeben können, die in einer SWF-Datei eingebettet werden sollen, damit die Zeichen nicht auf den Geräten vorhanden sein müssen, auf denen die SWF-Datei später abgespielt wird. Diese Eigenschaft bietet dieselbe Funktionalität wie das Dialogfeld für die Schriftarteinbettung.

Diese Eigenschaft kann auch gelesen werden. So können Sie feststellen, welche Zeichen im Dialogfeld für die Schriftarteinbettung für ein bestimmtes Schriftartelement festgelegt wurden.

Beispiel

Unter der Voraussetzung, dass das erste Element in der Bibliothek ein Schriftartelement ist, werden mit dem folgenden Code die Zeichen a, b und c eingebettet.

```
fl.getDocumentDOM().library.items[0].embeddedCharacters = "abc";
```

fontItem.embedRanges

Verfügbarkeit

Flash CS5 Professional.

Verwendung

fontItem.embedRanges

Beschreibung

Eigenschaft; ein Stringwert, der sich aus Ganzzahlen mit Trennzeichen zusammensetzt. Diese Ganzzahlen entsprechen den Elementen, die im Dialogfeld für die Schriftarteinbettung ausgewählt werden können.

Diese Eigenschaft kann auch gelesen werden. So können Sie feststellen, welche Zeichen im Dialogfeld für die Schriftarteinbettung für ein bestimmtes Schriftartelement festgelegt wurden.

Hinweis: Die Bereichsnummern entsprechen der Datei „FontEmbedding/UnicodeTables.xml“ im Konfigurationsordner.

Beispiel

Unter der Voraussetzung, dass das erste Element in der Bibliothek ein Schriftartelement ist, werden mit dem folgenden Code die Bereiche eingebettet, die über die Ganzzahlen 1, 3 und 7 definiert werden.

```
fl.getDocumentDOM().library.items[0].embedRanges = "1|3|7";
```

Wenn das erste Element in der Bibliothek ein Schriftartelement ist, werden mit dem folgenden Code die einzubettenden Bereiche zurückgesetzt.

```
fl.getDocumentDOM().library.items[0].embedRanges = "";
```

fontItem.embedVariantGlyphs

Verfügbarkeit

Flash CS4 Professional.

Verwendung

fontItem.embedVariantGlyphs

Beschreibung

Hinweis: Diese Eigenschaft ist zwar in Flash CS5 Professional verfügbar, hat aber keine Auswirkungen, wenn sie auf TLF-Text (Text Layout Framework) angewendet wird. Ab Flash Professional CS5 werden Glyphenvarianten immer in Schriftarten eingebettet, die mit TLF-Text verwendet werden. Die unten genannte flash.text.engine-Komponente (FTE) ist nur in Flash Professional CS4 verfügbar.

Eigenschaft; ein boolescher Wert, der angibt, ob Glyphenvarianten bei der Veröffentlichung einer SWF-Datei in der Schrift ausgegeben werden sollten (`true`) oder nicht (`false`). Durch Einstellen dieses Wertes auf `true` nimmt die Größe der SWF-Datei zu. Der Standardwert lautet `false`.

In einigen Sprachen werden Glyphen dynamisch bei der Eingabe ersetzt (z. B. Thai, Arabisch, Hebräisch und Griechisch). Wenn Sie Text in einer solchen Sprache anordnen oder eingeben, empfiehlt es sich, diese Eigenschaft auf `true` einzustellen.

Beispiele

Schriftsymbole, die mit flash.text-APIs kompatibel sind, werden in der Bibliothek angezeigt und können vom Benutzer direkt verwaltet werden. Dagegen werden Schriftsymbole, die mit flash.text.engine-APIs (FTE), nicht in der Bibliothek angezeigt und müssen daher manuell verwaltet werden. Mithilfe der folgenden Funktion wird der Bibliothek eine neue Schrift hinzugefügt, die mit FTE-APIs verwendet werden kann.

```
function embedFontSymbol(symbolName, fontName, includeVariants) {
    var doc = fl.getDocumentDOM();
    if (doc) {
        // look up the item. if it exists, delete it.
        var index = doc.library.findItemIndex(symbolName);
        if (index > -1)
            doc.library.deleteItem(symbolName);

        // make a new font symbol in the library
        doc.library.addNewItem('font', symbolName);

        // look up the symbol by its name
        var index = doc.library.findItemIndex(symbolName);
        if (index > -1) {
            // get the item from the library and set the attributes of interest
            var fontObj = doc.library.items[index];
            fontObj.isDefineFont4Symbol = true;
            fontObj.font = fontName;
            fontObj.bold = false;
            fontObj.italic = false;
            fontObj.embedVariantGlyphs = includeVariants;
            // this is what forces the font into the SWF stream
            fontObj.linkageExportForAS = true;
            fontObj.linkageExportInFirstFrame = true;
        }
    }
}
```

Mithilfe der folgenden Funktion werden alle Schriftsymbole im Bedienfeld „Ausgabe“ angezeigt.

```
function dumpFontSymbols()
{
    var doc = fl.getDocumentDOM();
    if (doc) {
        var items = doc.library.items;
        fl.trace("items length = " + items.length);
        var i;
        for(i=0; i<items.length; i++) {
            var item = items[i];
            fl.trace("itemType = " + item.itemType);
            if (item.itemType == 'font') {
                fl.trace("name = " + item.name);
                fl.trace("DF4 symbol = " + item.isDefineFont4Symbol);
                fl.trace("font = " + item.font);
            }
        }
    }
}
```

Siehe auch

`fontItem.isDefineFont4Symbol`, `text.embedVariantGlyphs`

fontItem.font

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`fontItem.font`

Beschreibung

Eigenschaft; ein String, der den Namen der Geräteschriftart angibt, die mit dem Font-Element verknüpft ist. Wenn Sie einen String eingeben, der keiner installierten Geräteschriftart entspricht, wird eine Fehlermeldung angezeigt. Um festzustellen, ob eine Schrift im System vorhanden ist, verwenden Sie `fl.isFontInstalled()`.

Hinweis: Der Eigenschaftswert, der sich aus der Festlegung dieses Werts ergibt, unterscheidet sich möglicherweise vom eingegebenen String. Dies wird im folgenden Beispiel verdeutlicht.

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Font-Element handelt, wird im folgenden Beispiel der Name der Geräteschriftart angezeigt, die mit dem Font-Element verknüpft ist; dann wird die Schriftart in Times geändert:

```
fl.outputPanel.clear();
var theItem = fl.getDocumentDOM().library.items[0];
fl.trace(theItem.font);
theItem.font = "Times";
// depending on your system, the following may display something like "Times-Roman"
fl.trace(theItem.font);
```

fontItem.isDefineFont4Symbol

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`fontItem.isDefineFont4Symbol`

Beschreibung

Eigenschaft; ein boolescher Wert, der das Format der Schrift angibt, die bei der Veröffentlichung einer SWF-Datei ausgegeben wird. Wenn dieser Wert `true` lautet, gibt Flash eine Schrift aus, die mit flash.text.engine-APIs (FTE) verwendet werden kann. Wenn dieser Wert `false` lautet, kann die Schrift mit flash.text-APIs, einschließlich Textfeldern, verwendet werden. Der Standardwert lautet `false`.

Beispiel

Siehe [fontItem.embedVariantGlyphs](#).

fontItem.italic

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`fontItem.italic`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob das Font-Element kursiv dargestellt wird (`true`) oder nicht (`false`).

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Font-Element handelt, wird im folgenden Beispiel im Bedienfeld „Ausgabe“ `true` angezeigt, falls es kursiv dargestellt wird, andernfalls `false`; dann wird die Kursivdarstellung festgelegt.

```
var theItem = fl.getDocumentDOM().library.items[0];
fl.outputPanel.clear();
fl.trace("italic: " + theItem.italic);
theItem.italic=true;
fl.trace("italic: " + theItem.italic);
```

fontItem.size

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`fontItem.size`

Beschreibung

Eigenschaft; eine Ganzzahl, die die Größe des Font-Elements in Punkt angibt.

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Font-Element handelt, wird im folgenden Beispiel im Bedienfeld „Ausgabe“ die Punktgröße des Elements angezeigt, die dann auf 24 festgelegt wird.

```
var theItem = fl.getDocumentDOM().library.items[0];
fl.outputPanel.clear();
fl.trace("font size: " + theItem.size);
theItem.size=24;
fl.trace("font size: " + theItem.size);
```

Kapitel 21: Frame-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Frame-Objekt stellt Bilder in der Ebene dar.

Übersicht über Methoden

Für das Frame-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>frame.convertMotionObjectTo2D()</code>	Konvertiert das ausgewählte Bewegungsobjekt in ein 2D-Bewegungsobjekt.
<code>frame.convertMotionObjectTo3D()</code>	Konvertiert das ausgewählte Bewegungsobjekt in ein 3D-Bewegungsobjekt.
<code>frame.getCustomEase()</code>	Gibt ein Array von JavaScript-Objekten zurück. Jedes dieser Objekte hat eine <i>x</i> - und eine <i>y</i> -Eigenschaft.
<code>frame.getMotionObjectXML()</code>	Gibt den XML-Bewegungscode vom ausgewählten Bewegungsobjekt zurück.
<code>frame.hasMotionPath()</code>	Gibt Aufschluss, ob die aktuelle Auswahl ein Bewegungs-Tween enthält oder nicht.
<code>frame.is3DMotionObject()</code>	Gibt Aufschluss, ob es sich bei der aktuellen Auswahl um ein 3D-Bewegungsobjekt handelt oder nicht.
<code>frame.isMotionObject()</code>	Gibt Aufschluss, ob es sich bei der aktuellen Auswahl um ein Bewegungsobjekt handelt oder nicht.
<code>frame.selectMotionPath()</code>	Wählt den Bewegungspfad des aktuellen Bewegungsobjekts aus oder hebt die Auswahl des Bewegungspfads auf.
<code>frame.setCustomEase()</code>	Gibt eine kubische Bézierkurve an, die als benutzerdefinierte Beschleunigungskurve verwendet werden soll.
<code>frame.setMotionObjectDuration()</code>	Gibt die Dauer (Tween-Länge) des derzeit ausgewählten Bewegungsobjekts an.
<code>frame.setMotionObjectXML()</code>	Wendet den angegebenen XML-Bewegungscode auf das ausgewählte Bewegungsobjekt an.

Übersicht über Eigenschaften

Die folgenden Eigenschaften können mit dem Frame-Objekt verwendet werden:

Eigenschaft	Beschreibung
<code>frame.actionScript</code>	Ein String, der ActionScript-Code darstellt.
<code>frame.duration</code>	Schreibgeschützt; eine Ganzzahl, die die Anzahl der Bilder in einer Bildsequenz angibt.
<code>frame.elements</code>	Schreibgeschützt; ein Array von Element-Objekten (siehe Element-Objekt).
<code>frame.hasCustomEase</code>	Ein boolescher Wert, der angibt, ob die Beschleunigungsdaten des Bilds von der benutzerdefinierten Beschleunigungskurve bereitgestellt werden.
<code>frame.labelType</code>	Ein String, der den Typ des Bildnamens angibt.

Eigenschaft	Beschreibung
<code>frame.motionTweenOrientToPath</code>	Ein boolescher Wert, der angibt, ob das getweente Element während der Bewegung an einem Pfad entlang gedreht wird, um seinen Winkel im Verhältnis zu den einzelnen Pfadpunkten zu erhalten.
<code>frame.motionTweenRotate</code>	Ein String, der angibt, wie das getweente Element gedreht wird.
<code>frame.motionTweenRotateTimes</code>	Eine Ganzzahl, die angibt, wie oft das getweente Element zwischen dem ersten und dem nächsten Schlüsselbild gedreht wird.
<code>frame.motionTweenScale</code>	Ein boolescher Wert, der angibt, ob das getweente Element auf die Größe des Objekts im folgenden Schlüsselbild skaliert wird und dabei seine Größe in jedem Bild des Tweens zunimmt (<code>true</code>) oder ob keine Skalierung erfolgt (<code>false</code>).
<code>frame.motionTweenSnap</code>	Ein boolescher Wert, der angibt, ob das getweente Element automatisch am nächsten Punkt auf der mit der Ebene dieses Bilds verknüpften Pfadebene ausgerichtet wird (<code>true</code>) oder nicht (<code>false</code>).
<code>frame.motionTweenSync</code>	Ein boolescher Wert; wenn dieser Wert <code>true</code> lautet, wird die Animation des getweenten Objekts mit der Hauptzeitleiste synchronisiert.
<code>frame.name</code>	Ein String, der den Namen des Bilds angibt.
<code>frame.shapeTweenBlend</code>	Ein String, der angibt, wie zwischen der Form im Schlüsselbild am Anfang des Tweens und der Form im folgenden Schlüsselbild ein Form-Tween erstellt wird.
<code>frame.soundEffect</code>	Ein String, der Effekte für einen Sound angibt, der direkt mit einem Bild verknüpft ist (<code>frame.soundLibraryItem</code>).
<code>frame.soundLibraryItem</code>	Ein Bibliothekselement, anhand dessen ein Sound erstellt wird (siehe SoundItem-Objekt).
<code>frame.soundLoop</code>	Ein Ganzzahlenwert, der angibt, wie oft ein Sound abgespielt wird, der direkt mit einem Bild verknüpft ist (<code>frame.soundLibraryItem</code>).
<code>frame.soundLoopMode</code>	Ein String, der angibt, ob ein Sound, der direkt mit einem Bild verknüpft ist (<code>frame.soundLibraryItem</code>), nur einige Male oder unendlich oft abgespielt werden soll.
<code>frame.soundName</code>	Ein String, der den in der Bibliothek gespeicherten Namen eines Sounds angibt, der einem Bild direkt zugeordnet ist (<code>frame.soundLibraryItem</code>).
<code>frame.soundSync</code>	Ein String, der das Synchronisierungsverhalten eines Sounds angibt, der direkt mit einem Bild verknüpft ist (<code>frame.soundLibraryItem</code>).
<code>frame.startFrame</code>	Schreibgeschützt; der Index des ersten Bilds in einer Sequenz.
<code>frame.tweenEasing</code>	Eine Ganzzahl, die die Stärke der Abbremsfunktion angibt, die auf das getweente Objekt angewendet werden soll.
<code>frame.tweenInstanceName</code>	Weist dem angegebenen Bewegungsobjekt einen Instanznamen zu.
<code>frame.tweenType</code>	Ein String, der den Typ des Tweens angibt.
<code>frame.useSingleEaseCurve</code>	Ein boolescher Wert, der angibt, ob die Beschleunigungsdaten sämtlicher Eigenschaften auf ein und derselben benutzerdefinierten Beschleunigungskurve beruhen.

frame.convertMotionObjectTo2D()

Verfügbarkeit

Flash Professional CS5.

Verwendung

```
frame.convertMotionObjectTo2D()
```

Beschreibung

Methode; konvertiert das ausgewählte Bewegungsobjekt in ein 2D-Bewegungsobjekt.

Beispiel

Das folgende Beispiel konvertiert das ausgewählte Bewegungsobjekt in ein 2D-Bewegungsobjekt.

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function() {
var layer = my_tl.layers[my_tl.currentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject() && theFrame.is2DMotionObject()){
theFrame.convertMotionObjectTo2D();
}else{
fl.trace("It isn't motion or it's already a 2D motion");
}
```

frame.convertMotionObjectTo3D()

Verfügbarkeit

Flash Professional CS5.

Verwendung

```
frame.convertMotionObjectTo3D()
```

Beschreibung

Methode; konvertiert das ausgewählte Bewegungsobjekt in ein 3D-Bewegungsobjekt.

Beispiel

Das folgende Beispiel konvertiert das ausgewählte Bewegungsobjekt in ein 3D-Bewegungsobjekt.

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function() {
var layer = my_tl.layers[my_tl.currentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject() && !theFrame.is3DMotionObject()){
theFrame.convertMotionObjectTo3D();
}else{
fl.trace("It isn't motion or it's already a 3D motion");
}
```

frame.actionScript

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.actionScript`

Beschreibung

Eigenschaft; ein String, der ActionScript-Code darstellt. Wenn Sie einen Zeilenumbruch einfügen möchten, verwenden Sie `"\n"`.

Beispiel

Im folgenden Beispiel wird der Aktion im ersten Bild in der obersten Ebene der Wert `stop()` zugewiesen:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].actionScript = 'stop();';
```

frame.duration

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.duration`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die die Anzahl der Bilder in einer Bildsequenz angibt.

Beispiel

Im folgenden Beispiel wird die Anzahl der Bilder in einer Bildsequenz, die bei dem ersten Bild in der obersten Ebene beginnt, in der Variablen `frameSpan` gespeichert:

```
var frameSpan = fl.getDocumentDOM().getTimeline().layers[0].frames[0].duration;
```

frame.elements

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.elements`

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von Element-Objekten (siehe [Element-Objekt](#)). Die Reihenfolge der Elemente entspricht der Reihenfolge, in der sie in der FLA-Datei gespeichert sind. Wenn es auf der Bühne mehrere Formen gibt und keine dieser Formen gruppiert ist, werden die Formen von Flash als ein einzelnes Element behandelt. Wenn die einzelnen Formen gruppiert sind und es auf der Bühne folglich mehrere Gruppen gibt, werden die Formen von Flash als separate Elemente behandelt. Flash behandelt also unformatierte, ungruppierte Formen als ein einzelnes Element; dabei ist es gleichgültig, wie viele separate Formen sich auf der Bühne befinden. Wenn ein Bild beispielsweise drei unformatierte, ungruppierte Formen enthält, gibt `elements.length` in diesem Bild den Wert 1 zurück. Sie können dieses Problem vermeiden, indem Sie jede Form einzeln auswählen und gruppieren.

Beispiel

Im folgenden Beispiel wird ein Array der aktuellen Elemente, die sich im ersten Bild in der obersten Ebene befinden, in der Variablen `myElements` gespeichert:

```
var myElements = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements;
```

frame.getCustomEase()

Verfügbarkeit

Flash 8.

Verwendung

```
Frame.getCustomEase ( [property] )
```

Parameter

property Ein optionaler String, der die Eigenschaft angibt, für die der benutzerdefinierte Beschleunigungswert zurückgegeben werden soll. Die zulässigen Werte lauten "all", "position", "rotation", "scale", "color" und "filters". Der Standardwert lautet "all".

Rückgabewerte

Gibt ein Array von JavaScript-Objekten zurück. Jedes dieser Objekte hat eine *x*- und eine *y*-Eigenschaft.

Beschreibung

Methode; gibt ein Array von Objekten zurück, die die Steuerpunkte der kubischen Bézierkurve darstellen, die wiederum die Beschleunigungskurve definiert.

Beispiel

Im folgenden Beispiel wird der benutzerdefinierte Beschleunigungswert der Eigenschaft `position` für das erste Bild in der obersten Ebene zurückgegeben:

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]
var easeArray = theFrame.getCustomEase("position");
```

Siehe auch

[frame.hasCustomEase](#), [frame.setCustomEase\(\)](#), [frame.useSingleEaseCurve](#)

frame.getMotionObjectXML()

Verfügbarkeit

Flash Professional CS5.

Verwendung

```
Frame.getMotionObjectXML()
```

Beschreibung

Gibt einen String des XML-Bewegungscode vom ausgewählten Bewegungsobjekt zurück.

Beispiel

Das folgende Beispiel gibt den XML-Bewegungscode vom ausgewählten Bewegungsobjekt zurück.

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function() {
    var layer = my_tl.layers[my_tl.currentLayer];
    var frame = layer.frames[my_tl.currentFrame];
    return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject()) {
    //fl.trace(theFrame.getMotionObjectXML());
}else{
    fl.trace("It is not motion.");
}
```

frame.hasCustomEase

Verfügbarkeit

Flash 8.

Verwendung

```
frame.hasCustomEase
```

Beschreibung

Eigenschaft; ein boolescher Wert. Wenn der Wert `true` lautet, werden die Beschleunigungsdaten des Bilds aus der benutzerdefinierten Beschleunigungskurve abgerufen, bei dem Wert `false` aus dem Beschleunigungswert.

Beispiel

Im folgenden Beispiel wird angegeben, dass die Beschleunigungsdaten des ersten Bilds in der obersten Ebene nicht aus der benutzerdefinierten Beschleunigungskurve, sondern aus dem Beschleunigungswert stammen sollen:

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]
theFrame.hasCustomEase = false;
```

Siehe auch

[frame.getCustomEase\(\)](#), [frame.setCustomEase\(\)](#), [frame.useSingleEaseCurve](#)

frame.hasMotionPath()

Verfügbarkeit

Flash Professional CS5.

Verwendung

`Frame.hasMotionPath()`

Beschreibung

Methode; ein boolescher Wert. Gibt Aufschluss, ob die aktuelle Auswahl einen Bewegungspfad enthält.

Beispiel

Das folgende Beispiel gibt eine Trace-Anweisung zurück, die Aufschluss darüber gibt, ob die aktuelle Auswahl einen Bewegungspfad enthält.

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function(){
    var layer = my_tl.layers[my_tl.currentLayer];
    var frame = layer.frames[my_tl.currentFrame];
    return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject()){
    if (theFrame.hasMotionPath()){
        fl.trace("There is a motion path");
    }else{
        fl.trace("There is no motion path");
    }
}
```

frame.is3DMotionObject()

Verfügbarkeit

Flash Professional CS5.

Verwendung

`Frame.is3DMotionObject()`

Beschreibung

Methode; ein boolescher Wert. Gibt Aufschluss, ob es sich bei der aktuellen Auswahl um ein 3D-Bewegungsobjekt handelt.

Beispiel

Das folgende Beispiel gibt eine Trace-Anweisung zurück, die Aufschluss darüber gibt, ob es sich bei der aktuellen Auswahl um ein 3D-Bewegungsobjekt handelt oder nicht.


```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function() {
    var layer = my_tl.layers[my_tl.currentLayer];
    var frame = layer.frames[my_tl.currentFrame];
    return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject() && theFrame.is3DMotionObject()) {
    fl.trace("This selection is 3D Motion");
} else {
    fl.trace("This selection is not 3D motion");
}
```

frame.isMotionObject()

Verfügbarkeit

Flash Professional CS5.

Verwendung

Frame.isMotionObject()

Beschreibung

Methode; ein boolescher Wert. Gibt Aufschluss, ob es sich bei der aktuellen Auswahl um ein Bewegungsobjekt handelt.

Beispiel

Das folgende Beispiel gibt eine Trace-Anweisung zurück, die Aufschluss darüber gibt, ob es sich bei der aktuellen Auswahl um ein Bewegungsobjekt handelt oder nicht.

```
var my_tl = doc.getTimeline();
this.getCurrentFrame = function() {
    var layer = my_tl.layers[my_tl.currentLayer];
    var frame = layer.frames[my_tl.currentFrame];
    return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject()) {
    fl.trace("This selection is motion.");
} else {
    fl.trace("This selection is not motion.");
}
```

frame.labelType

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.labelType`

Beschreibung

Eigenschaft; ein String, der den Typ des Bildnamens angibt. Die zulässigen Werte lauten "none", "name", "comment" und "anchor". Wird eine Bezeichnung auf "none" eingestellt, wird die Eigenschaft `frame.name` gelöscht.

Beispiel

Im folgenden Beispiel erhält das erste Bild in der obersten Ebene den Namen "First Frame" und dann die Bezeichnung "comment":

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].name = 'First Frame';  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].labelType = 'comment';
```

frame.motionTweenOrientToPath

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.motionTweenOrientToPath`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob das getweente Element während der Bewegung an einem Pfad entlang gedreht wird, um seinen Winkel im Verhältnis zu den einzelnen Pfadpunkten zu erhalten (`true`), oder ob das Element nicht gedreht wird (`false`).

Wenn Sie für diese Eigenschaft einen Wert angeben möchten, sollten Sie `frame.motionTweenRotate` auf "none" einstellen.

frame.motionTweenRotate

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.motionTweenRotate`

Beschreibung

Eigenschaft; ein String, der angibt, wie das getweente Element gedreht wird. Die zulässigen Werte lauten "none", "auto", "clockwise" und "counter-clockwise". Bei dem Wert "auto" wird das Objekt in die Richtung gedreht, bei der sich die Drehung mit einem minimalen Bewegungsaufwand an die Drehung des Objekts im folgenden Schlüsselbild anpassen lässt.

Wenn Sie für `frame.motionTweenOrientToPath` einen Wert angeben möchten, sollten Sie diese Eigenschaft auf "none" einstellen.

Beispiel

Siehe `frame.motionTweenRotateTimes`.

frame.motionTweenRotateTimes

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.motionTweenRotateTimes`

Beschreibung

Eigenschaft; eine Ganzzahl, die angibt, wie oft das getweente Element zwischen dem ersten und dem nächsten Schlüsselbild gedreht wird.

Beispiel

Im folgenden Beispiel wird das Element in diesem Bild dreimal gegen den Uhrzeigersinn gedreht, bevor es das nächste Schlüsselbild erreicht:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenRotate = "counter-clockwise";  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenRotateTimes = 3;
```

frame.motionTweenScale

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.motionTweenScale`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob das getweente Element auf die Größe des Objekts im folgenden Schlüsselbild skaliert wird und dabei seine Größe in jedem Bild des Tweens zunimmt (`true`) oder ob keine Skalierung erfolgt (`false`).

Beispiel

Im folgenden Beispiel wird festgelegt, dass das getweente Element auf die Größe des Objekts im folgenden Schlüsselbild skaliert werden und dabei seine Größe in jedem Bild des Tweens zunehmen soll:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenScale = true;
```

frame.motionTweenSnap

Verfügbarkeit

Flash MX 2004.

Verwendung

```
frame.motionTweenSnap
```

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob das getweente Element automatisch an dem nächsten Punkt auf der mit der Ebene dieses Bilds verknüpften Pfadebene ausgerichtet wird (`true`) oder nicht (`false`).

frame.motionTweenSync

Verfügbarkeit

Flash MX 2004.

Verwendung

```
frame.motionTweenSync
```

Beschreibung

Eigenschaft; ein boolescher Wert. Lautet der Wert `true`, wird die Animation des getweenten Objekts mit der Hauptzeitleiste synchronisiert.

Beispiel

Im folgenden Beispiel wird angegeben, dass das getweente Objekt mit der Zeitleiste synchronisiert werden soll:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenSync = true;
```

frame.name

Verfügbarkeit

Flash MX 2004.

Verwendung

```
frame.name
```

Beschreibung

Eigenschaft; ein String, der den Namen des Bilds angibt.

Beispiel

Im folgenden Beispiel wird dem ersten Bild in der obersten Ebene der Name "First Frame" zugewiesen und dann der Wert `name` in der Variablen `frameLabel` gespeichert:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].name = 'First Frame';  
var frameLabel = fl.getDocumentDOM().getTimeline().layers[0].frames[0].name;
```

frame.selectMotionPath()

Verfügbarkeit

Flash Professional CS5.

Verwendung

```
Frame.selectMotionPath()
```

Beschreibung

Methode; ein boolescher Wert. Wählt den Bewegungspfad des aktuellen Bewegungsobjekts aus (true) oder hebt die Auswahl des Bewegungspfads auf (false).

Beispiel

Dieses Beispiel wählt den Bewegungspfad des aktuellen Bewegungsobjekts aus oder hebt die Auswahl des Bewegungspfads auf.

```
var doc = fl.getDocumentDOM();  
var my_tl = doc.getTimeline();  
t    his.getCurrentFrame = function(){  
var layer = my_tl.layers[my_tl.c u rrentLayer];  
var frame = layer.frames[my_tl.currentFrame];  
return frame;  
}  
var theFrame = getCurrentFrame();  
if(theFrame.isMotionObject()){  
    if (theFrame.hasMotionPath()){  
        theFrame.selectMotionPath(true);  
    }  
    else{  
        fl.trace("There is no motion path");  
    }  
}else{  
    fl.trace("It is no motion");  
}
```

frame.setCustomEase()

Verfügbarkeit

Flash 8.

Verwendung

```
frame.setCustomEase(property, easeCurve)
```

Parameter

property Ein String, der angibt, für welche Eigenschaft die Beschleunigungskurve verwendet werden soll. Die zulässigen Werte lauten "all", "position", "rotation", "scale", "color" und "filters".

easeCurve Ein Array von Objekten, das die Beschleunigungskurve definiert. Alle Array-Elemente müssen JavaScript-Objekte mit den Eigenschaften *x* und *y* sein.

Rückgabewerte

Keine.

Beschreibung

Methode; legt ein Array von Koordinaten für Steuerpunkte und Tangentenendpunkte fest, die eine als benutzerdefinierte Beschleunigungskurve zu verwendende kubische Bézierkurve beschreiben. Dieses Array wird aus der horizontalen Lage (ordinal: von links nach rechts) der Steuerpunkte und der Tangentenendpunkte aufgebaut.

Beispiel

Im folgenden Beispiel wird die Beschleunigungskurve für alle Eigenschaften des ersten Bilds in der ersten Ebene auf die Bézierkurve eingestellt, die mit dem `easeCurve`-Array angegeben wird:

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0];
var easeCurve = [ {x:0,y:0}, {x:.3,y:.3}, {x:.7,y:.7}, {x:1,y:1} ];
theFrame.setCustomEase( "all", easeCurve );
```

Siehe auch

[frame.getCustomEase\(\)](#), [frame.hasCustomEase](#), [frame.useSingleEaseCurve](#)

frame.setMotionObjectDuration()

Verfügbarkeit

Flash Professional CS5.

Verwendung

```
Frame.setMotionObjectDuration( duration [, stretchExistingKeyframes] )
```

Parameter

duration Gibt die Anzahl der Bilder für die Tween-Länge des ausgewählten Bewegungsobjekts an.

stretchExistingKeyframes Ein boolescher Wert, der angibt, ob die Tween-Länge gestreckt wird oder ob nach dem letzten Bild weitere Bilder hinzugefügt werden.

Beschreibung

Methode; legt die Dauer (Tween-Länge) des derzeit ausgewählten Bewegungsobjekts fest.

Beispiel

Das folgende Beispiel legt eine Dauer von 11 Bildern für das ausgewählte Bewegungsobjekt fest.

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function(){
var layer = my_tl.layers[my_tl.currentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject()){
theFrame.setMotionObjectDuration(11);
}else{
fl.trace("It isn't motion");
}
```

frame.setMotionObjectXML()

Verfügbarkeit

Flash Professional CS5.

Verwendung

```
Frame.setMotionObjectXML( xmlstr [, endAtCurrentLocation] )
```

Parameter

xmlstr Ein Stringwert, der den XML-String angibt.

endAtCurrentLocation Ein boolescher Wert, der angibt, ob das Tweening an der aktuellen Position startet oder endet.

Beschreibung

Methode; wendet den angegebenen XML-Bewegungscode auf das ausgewählte Bewegungsobjekt an.

Beispiel

Dieses Beispiel gibt an, dass der durch `myMotionXML` gekennzeichnete XML-Bewegungscode auf das ausgewählte Bewegungsobjekt angewendet wird.

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function(){
var layer = my_tl.layers[my_tl.currentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
theFrame.setMotionObjectXML(myMotionXML.toString(), false);
```

frame.shapeTweenBlend

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.shapeTweenBlend`

Beschreibung

Eigenschaft; ein String, der angibt, wie zwischen der Form im Schlüsselbild am Anfang des Tweens und der Form im folgenden Schlüsselbild ein Form-Tween erstellt wird. Die zulässigen Werte lauten "distributive" und "angular".

frame.soundEffect

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.soundEffect`

Beschreibung

Eigenschaft; ein String, der Effekte für einen Sound angibt, der direkt mit einem Bild verknüpft ist ([frame.soundLibraryItem](#)). Die zulässigen Werte lauten "none", "left channel", "right channel", "fade left to right", "fade right to left", "fade in", "fade out" und "custom".

Beispiel

Im folgenden Beispiel wird angegeben, dass der mit dem ersten Bild verknüpfte Sound eingeblendet werden soll:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundEffect = "fade in";
```

frame.soundLibraryItem

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.soundLibraryItem`

Beschreibung

Eigenschaft; ein Bibliothekselement, anhand dessen ein Sound erstellt wird (siehe [SoundItem-Objekt](#)). Der Sound ist direkt mit dem Bild verknüpft.

Beispiel

Im folgenden Beispiel wird das erste Element in der Bibliothek der Eigenschaft `soundLibraryItem` des ersten Bilds zugewiesen:

```
// The first item in the library must be a sound object.  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundLibraryItem  
=fl.getDocumentDOM().library.items[0];
```


frame.soundLoop

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.soundLoop`

Beschreibung

Eigenschaft; ein Ganzzahlwert, der angibt, wie oft ein Sound abgespielt wird, der direkt mit einem Bild verknüpft ist ([frame.soundLibraryItem](#)). Wenn Sie für diese Eigenschaft einen Wert angeben möchten, stellen Sie [frame.soundLoopMode](#) auf "repeat" ein.

Beispiel

Siehe [frame.soundLoopMode](#).

frame.soundLoopMode

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.soundLoopMode`

Beschreibung

Eigenschaft; ein String, der angibt, ob ein Sound, der direkt mit einem Bild verknüpft ist ([frame.soundLibraryItem](#)), nur einige Male oder unendlich oft abgespielt werden soll. Die gültigen Werte lauten "repeat" und "loop". Um anzugeben, wie oft der Sound abgespielt werden soll, legen Sie einen Wert für [frame.soundLoop](#) fest.

Beispiel

Im folgenden Beispiel wird festgelegt, dass ein Sound zweimal abgespielt werden soll:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundLoopMode = "repeat";  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundLoop = 2;
```

frame.soundName

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.soundName`

Beschreibung

Eigenschaft; ein String, der den in der Bibliothek gespeicherten Namen eines Sounds angibt, der einem Bild direkt zugeordnet ist ([frame.soundLibraryItem](#)).

Beispiel

Im folgenden Beispiel wird die Eigenschaft `soundName` des ersten Bilds in `"song1.mp3"` abgeändert; „song1.mp3“ muss in der Bibliothek vorhanden sein:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundName = "song1.mp3";
```

frame.soundSync

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.soundSync`

Beschreibung

Eigenschaft; ein String, der das Synchronisierungsverhalten für einen Sound angibt, der direkt mit einem Bild verknüpft ist ([frame.soundLibraryItem](#)). Die zulässigen Werte lauten `"event"`, `"stop"`, `"start"` und `"stream"`.

Beispiel

Im folgenden Beispiel wird festgelegt, dass ein Sound gestreamt werden soll:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundSync = 'stream';
```

frame.startFrame

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.startFrame`

Beschreibung

Schreibgeschützte Eigenschaft; der Index des ersten Bilds in einer Sequenz.

Beispiel

Im folgenden Beispiel ist `stFrame` der Index des ersten Bilds in der Bildsequenz. In diesem Beispiel umfasst eine Bildsequenz die sechs Bilder von Bild 5 bis Bild 10. Aus diesem Grund hat `stFrame` für jedes Bild zwischen Bild 5 und Bild 10 den Wert 4 (hierbei ist zu berücksichtigen, dass sich die Indexwerte von den Bildnummern unterscheiden).

```
var stFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[4].startFrame;  
fl.trace(stFrame); // 4  
var stFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[9].startFrame;  
fl.trace(stFrame); // 4
```

frame.tweenEasing

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.tweenEasing`

Beschreibung

Eigenschaft; eine Ganzzahl, die die Stärke der Abbremsfunktion angibt, die auf das getweente Objekt angewendet werden soll. Die zulässigen Werte liegen zwischen -100 und 100. Geben Sie einen Wert zwischen -1 und -100 ein, um das Bewegungs-Tweening langsam zu beginnen und gegen Ende der Animation zu beschleunigen. Geben Sie einen Wert zwischen 1 und 100 ein, um das Bewegungs-Tweening schnell zu beginnen und gegen Ende der Animation zu verlangsamen.

Beispiel

Im folgenden Beispiel wird festgelegt, dass die Bewegung des getweenten Objekts zunächst ziemlich schnell ist und gegen Ende der Animation langsamer wird:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].tweenEasing = 50;
```

frame.tweenInstanceName

Verfügbarkeit

Flash Professional CS5.

Verwendung

`Frame.tweenInstanceName()`

Beschreibung

Eigenschaft; ein String, der dem ausgewählten Bewegungsobjekt einen Instanznamen zuweist.

Beispiel

Das folgende Beispiel weist dem angegebenen Bewegungsobjekt den Instanznamen `MyMotionTween` zu.

```
theFrame.tweenInstanceName = "MyMotionTween";
```

frame.tweenType

Verfügbarkeit

Flash MX 2004.

Verwendung

`frame.tweenType`

Beschreibung

Eigenschaft; ein String, der den Tween-Typ angibt; die zulässigen Werte lauten "motion", "shape" und "none". Bei dem Wert "none" wird der Bewegungs-Tween entfernt. Bewegungs-Tweens werden mit der Methode `timeline.createMotionTween()` erstellt.

Wenn Sie den Wert "motion" eingeben, muss es sich bei dem Objekt in dem Bild um ein Symbol, ein Textfeld oder ein gruppiertes Objekt handeln. Der Bewegungs-Tween beginnt an der Objektposition im aktuellen Schlüsselbild und endet an der entsprechenden Position im folgenden Schlüsselbild.

Wenn Sie den Wert "shape" angeben, muss es sich bei dem Objekt in dem Bild um eine Form handeln. Der Form-Tween beginnt bei der Objektform im aktuellen Schlüsselbild und endet bei der Form im folgenden Schlüsselbild.

Beispiel

Im folgenden Beispiel wird angegeben, dass das Objekt ein Bewegungs-Tween ist und dass das Tweening von der Objektposition im aktuellen Schlüsselbild bis zur entsprechenden Position im nächsten Schlüsselbild erfolgen soll:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].tweenType = "motion";
```

frame.useSingleEaseCurve

Verfügbarkeit

Flash 8.

Verwendung

```
frame.useSingleEaseCurve
```

Beschreibung

Eigenschaft; ein boolescher Wert. Wenn der Wert `true` lautet, beruhen die Beschleunigungsdaten sämtlicher Eigenschaften auf ein und derselben benutzerdefinierten Beschleunigungskurve. Bei `false` hat jede Eigenschaft eine eigene Beschleunigungskurve.

Diese Eigenschaft wird ignoriert, wenn auf das Bild kein benutzerdefiniertes Abbremsen angewendet wurde.

Beispiel

Im folgenden Beispiel wird festgelegt, dass für alle Eigenschaften des ersten Bilds in der ersten Ebene ein und dieselbe benutzerdefinierte Beschleunigungskurve verwendet werden soll:

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]  
theFrame.useSingleEaseCurve = true;
```

Siehe auch

`frame.getCustomEase()`, `frame.hasCustomEase`, `frame.setCustomEase()`

Kapitel 22: HalfEdge-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das HalfEdge-Objekt ist die gerichtete Hälfte der Kante eines [Shape-Objekt](#). Eine Kante setzt sich aus zwei Halbkanten zusammen. Wenn Sie diese Halbkanten nacheinander „abschreiten“, können Sie die Konturen einer Form erfassen. Beispielsweise können Sie ausgehend von einer Halbkante nach und nach alle anderen Halbkanten anzeigen, die die Kontur einer Form umgeben, und schließlich zur ursprünglichen Halbkante zurückkehren.

Halbkanten sind in einer bestimmten Reihenfolge angeordnet. Eine Halbkante stellt eine Hälfte einer Kante dar und eine zweite Halbkante die andere Hälfte.

Übersicht über Methoden

Für das HalfEdge-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>halfEdge.getEdge()</code>	Ruft das Edge-Objekt für das HalfEdge-Objekt ab.
<code>halfEdge.getNext()</code>	Ruft die nächste Halbkante auf der aktuellen Kontur ab.
<code>halfEdge.getOppositeHalfEdge()</code>	Ruft das HalfEdge-Objekt auf der gegenüberliegenden Seite der Kante ab.
<code>halfEdge.getPrev()</code>	Ruft das vorherige HalfEdge-Objekt auf der aktuellen Kontur ab.
<code>halfEdge.getVertex()</code>	Ruft das Vertex-Objekt am Anfang des HalfEdge-Objekts ab.

Übersicht über Eigenschaften

Für das HalfEdge-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>halfEdge.id</code>	Schreibgeschützt; eine eindeutige Ganzzahl, die das HalfEdge-Objekt bezeichnet.
<code>halfEdge.index</code>	Ein ganzzahliger Wert von 0 bzw. 1, der den Index für dieses HalfEdge-Objekt in der übergeordneten Kante angibt.

halfEdge.getEdge()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
halfEdge.getEdge()
```

Parameter

Keine.

Rückgabewerte

Ein [Edge-Objekt](#).

Beschreibung

Methode; ruft das Edge-Objekt für das HalfEdge-Objekt ab. Siehe [Edge-Objekt](#).

Beispiel

Im folgenden Beispiel werden eine Kante und eine Halbkante für die angegebene Form abgerufen:

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge(0);  
var edge = hEdge.getEdge();
```

halfEdge.getNext()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
halfEdge.getNext()
```

Parameter

Keine.

Rückgabewerte

Ein HalfEdge-Objekt.

Beschreibung

Methode; ruft die nächste Halbkante auf der aktuellen Kontur ab.

Hinweis: Halbkanten sind gerichtet (d. h. sie haben eine Orientierung) und in einer bestimmten Reihenfolge angeordnet, Kanten jedoch nicht.

Beispiel

Im folgenden Beispiel wird die nächste Halbkante der angegebenen Form in der Variablen nextHalfEdge gespeichert:

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge( 0 );  
var nextHalfEdge = hEdge.getNext();
```

halfEdge.getOppositeHalfEdge()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
halfEdge.getOppositeHalfEdge()
```

Parameter

Keine.

Rückgabewerte

Ein HalfEdge-Objekt.

Beschreibung

Methode; ruft das HalfEdge-Objekt auf der gegenüberliegenden Seite der Kante ab.

Beispiel

Im folgenden Beispiel wird die Halbkante, die `hEdge` gegenüberliegt, in der Variablen „otherHalfEdge“ gespeichert:

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge(0);  
var otherHalfEdge = hEdge.getOppositeHalfEdge();
```

halfEdge.getPrev()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
halfEdge.getPrev()
```

Parameter

Keine.

Rückgabewerte

Ein HalfEdge-Objekt.

Beschreibung

Methode; ruft das vorherige HalfEdge-Objekt auf der aktuellen Kontur ab.

Hinweis: Halbkanten sind gerichtet (d. h. sie haben eine Orientierung) und in einer bestimmten Reihenfolge angeordnet, Kanten jedoch nicht.

Beispiel

Im folgenden Beispiel wird die vorherige Halbkante der angegebenen Kontur in der Variablen `prevHalfEdge` gespeichert:

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge( 0 );  
var prevHalfEdge = hEdge.getPrev();
```

halfEdge.getVertex()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
halfEdge.getVertex()
```

Parameter

Keine.

Rückgabewerte

Ein [Vertex-Objekt](#).

Beschreibung

Methode; ruft das Vertex-Objekt am Anfang des HalfEdge-Objekts ab. Siehe [Vertex-Objekt](#).

Beispiel

Im folgenden Beispiel wird das Vertex-Objekt am Anfang von hEdge in der Variablen vertex gespeichert:

```
var shape = fl.getDocumentDOM().selection[0];  
var edge = shape.edges[0];  
var hEdge = edge.getHalfEdge(0);  
var vertex = hEdge.getVertex();
```

halfEdge.id

Verfügbarkeit

Flash MX 2004.

Verwendung

```
halfEdge.id
```

Beschreibung

Schreibgeschützte Eigenschaft; eine eindeutige Ganzzahl, die das HalfEdge-Objekt bezeichnet.

Beispiel

Im folgenden Beispiel wird für die angegebene Halbkante im Bedienfeld „Ausgabe“ ein eindeutiger Bezeichner angezeigt:

```
var shape = fl.getDocumentDOM().selection[0];  
alert(shape.contours[0].getHalfEdge().id);
```


halfEdge.index

Verfügbarkeit

Flash MX 2004.

Verwendung

`halfEdge.index`

Beschreibung

Schreibgeschützte Eigenschaft; ein ganzzahliger Wert von 0 bzw. 1, der den Index für dieses HalfEdge-Objekt in der übergeordneten Kante angibt.

Beispiel

Im folgenden Beispiel wird der Indexwert der angegebenen Halbkante im Bedienfeld „Ausgabe“ angezeigt:

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge(0);  
var heIndex = hEdge.index;
```

Kapitel 23: Instance-Objekt

Vererbung [Element-Objekt](#) > Instance-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Instance-Objekt ist eine Unterklasse aus dem [Element-Objekt](#).

Übersicht über Eigenschaften

Abgesehen von den Eigenschaften des Element-Objekts stehen für Instance-Objekt die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
instance.instanceType	Schreibgeschützt; ein String, der den Typ der Instanz angibt.
instance.libraryItem	Bibliothekselement, das zur Instanziierung dieser Instanz dient.

instance.instanceType

Verfügbarkeit

Flash MX 2004; in Flash 8 wurde "video" als möglicher Wert hinzugefügt.

Verwendung

```
instance.instanceType
```

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der den Typ der Instanz angibt. Die möglichen Werte lauten "symbol", "bitmap", "embedded video", "linked video", "video" und "compiled clip".

In Flash MX 2004 lautet der Wert für `instance.instanceType` eines mithilfe von `library.addNewItem("video")` zur Bibliothek hinzugefügten Objekts "embedded_video". In Flash 8 und neueren Versionen lautet der Wert "video". Siehe [library.addNewItem\(\)](#).

Beispiel

Im folgenden Beispiel wird angezeigt, dass der Instanztyp eines Movieclips `symbol` ist:

```
// Select a movie clip and then run this script.
var type = fl.getDocumentDOM().selection[0].instanceType;
fl.trace("This instance type is " + type);
```

instance.libraryItem

Verfügbarkeit

Flash MX 2004.

Verwendung

`instance.libraryItem`

Beschreibung

Eigenschaft; ein Bibliothekselement, das zur Instanziierung dieser Instanz dient. Sie können diese Eigenschaft nur in ein anderes Bibliothekselement desselben Typs abändern (eine `symbol`-Instanz kann beispielsweise nicht auf eine Bitmap verweisen). Siehe [library-Objekt](#).

Beispiel

Im folgenden Beispiel wird das ausgewählte Symbol so geändert, dass es auf das erste Element in der Bibliothek verweist:

```
fl.getDocumentDOM().selection[0].libraryItem = fl.getDocumentDOM().library.items[0];
```

Kapitel 24: Item-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Item-Objekt ist eine abstrakte Basisklasse. Alle Bibliotheksinhalte sind von dem Item-Objekt abgeleitet. Siehe auch [library-Objekt](#).

Übersicht über Methoden

Für das Item-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>item.addData()</code>	Fügt einem Bibliothekselement bestimmte Daten hinzu.
<code>item.getData()</code>	Ruft den Wert der angegebenen Daten ab.
<code>item.hasData()</code>	Ermittelt, ob sich die benannten Daten in dem Bibliothekselement befinden.
<code>item.removeData()</code>	Entfernt permanente Daten aus dem Bibliothekselement.

Übersicht über Eigenschaften

Für das Item-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>item.itemType</code>	Schreibgeschützt; ein String, der den Typ des Elements angibt.
<code>item.linkageBaseClass</code>	Ein String, der die ActionScript 3.0-Klasse angibt, die mit dem Symbol verknüpft wird.
<code>item.linkageClassName</code>	Ein String, der die ActionScript 2.0-Klasse angibt, die mit dem Symbol verknüpft wird.
<code>item.linkageExportForAS</code>	Ein boolescher Wert. Wenn der Wert <code>true</code> lautet, wird das Element für ActionScript exportiert.
<code>item.linkageExportForRS</code>	Ein boolescher Wert. Wenn der Wert <code>true</code> lautet, wird das Element für die gemeinsame Nutzung zur Laufzeit exportiert.
<code>item.linkageExportInFirstFrame</code>	Ein boolescher Wert. Wenn der Wert <code>true</code> lautet, wird das Element in das erste Bild exportiert.
<code>item.linkageIdentifier</code>	Ein String mit dem Namen, anhand dessen Flash den Bestand beim Verknüpfen mit der SWF-Zieldatei identifiziert.
<code>item.linkageImportForRS</code>	Ein boolescher Wert. Wenn der Wert <code>true</code> lautet, wird das Element für die gemeinsame Nutzung zur Laufzeit importiert.
<code>item.linkageURL</code>	Ein String mit der URL der SWF-Datei, die den gemeinsamen Bestand enthält.
<code>item.name</code>	Ein String mit dem Namen des Bibliothekselements, der die Ordnerstruktur einschließt.

item.addData()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
item.addData(name, type, data)
```

Parameter

name Ein String, der den Namen der Daten angibt.

type Ein String, der den Datentyp angibt. Die zulässigen Werte lauten "integer", "integerArray", "double", "doubleArray", "string" und "byteArray".

data Die Daten, die dem angegebenen Bibliothekselement hinzugefügt werden sollen. Der Datentyp hängt von dem Wert des type-Parameters ab. Wenn der Parameter z. B. den Wert "integer" hat, muss es sich bei dem Datenwert um eine Ganzzahl handeln.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt einem Bibliothekselement bestimmte Daten hinzu.

Beispiel

Im folgenden Beispiel werden dem ersten Element in der Bibliothek die Daten mit dem Namen `myData` und dem Ganzzahlenwert 12 hinzugefügt:

```
fl.getDocumentDOM().library.items[0].addData("myData", "integer", 12);
```

item.getData()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
item.getData(name)
```

Parameter

name Ein String, der den Namen der abzurufenden Daten angibt.

Rückgabewerte

Die von dem Parameter *name* angegebenen Daten. Der Typ der zurückgegebenen Daten hängt von dem Typ der gespeicherten Daten ab.

Beschreibung

Methode; ruft den Wert der angegebenen Daten ab.

Beispiel

Im folgenden Beispiel wird der Wert der Daten mit dem Namen `myData` aus dem ersten Element in der Bibliothek abgerufen und in der Variablen `libData` gespeichert:

```
var libData = fl.getDocumentDOM().library.items[0].getData("myData");
```

item.hasData()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
item.hasData(name)
```

Parameter

name Ein String, der den Namen der Daten angibt, auf deren Vorhandensein das Bibliothekselement hin überprüft werden soll.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die angegebenen Daten vorhanden sind, ansonsten `false`.

Beschreibung

Methode; ermittelt, ob sich die benannten Daten in dem Bibliothekselement befinden.

Beispiel

Im folgenden Beispiel wird im Bedienfeld „Ausgabe“ eine Meldung angezeigt, wenn das erste Element in der Bibliothek die Daten mit dem Namen `myData` enthält:

```
if (fl.getDocumentDOM().library.items[0].hasData("myData")) {  
    fl.trace("Yep, it's there!");  
}
```

item.itemType

Verfügbarkeit

Flash MX 2004.

Verwendung

```
item.itemType
```

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der den Typ des Elements angibt. Folgende Werte sind möglich:

"undefined", "component", "movie clip", "graphic", "button", "folder", "font", "sound", "bitmap", "compiled clip", "screen" oder "video". Wenn es sich bei dieser Eigenschaft um "video" handelt, können Sie den Videotyp bestimmen. Weitere Hinweise finden Sie unter [videoItem.videoType](#).

Beispiel

Im folgenden Beispiel wird der Typ des angegebenen Bibliothekselements im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace(fl.getDocumentDOM().library.items[0].itemType);
```

item.linkageBaseClass

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`item.linkageBaseClass`

Beschreibung

Eigenschaft; ein String, der die ActionScript 3.0-Klasse angibt, die mit dem Symbol verknüpft wird. Der hier angegebene Wert erscheint in der Authoring-Umgebung im Dialogfeld „Verknüpfung“ und in anderen Dialogfeldern, die über die Steuerungen des Dialogfeld „Verknüpfung“ verfügen (z. B. im Dialogfeld „Symboleigenschaften“). (Um diesen Wert für eine ActionScript 2.0-Klasse anzugeben, verwenden Sie [item.linkageClassName](#).)

Wenn die Basisklasse der Standard für den Symboltyp ist (zum Beispiel "flash.display.MovieClip" für Movieclips, "flash.display.SimpleButton" für Schaltflächen usw.), ist diese Eigenschaft ein leerer String (""). Wenn Sie ein Element zur Standardbasisklasse machen möchten, verwenden Sie für diesen Wert einen leeren String.

Wenn Sie diesen Wert festlegen, wird keine der Überprüfungen, die vom Dialogfeld „Verknüpfungen“ ausgeführt werden, ausgeführt und es werden keine Fehler ausgegeben, falls Flash die Basisklassen nicht auf den angegebenen Wert setzen kann. Wird dieser Wert im Dialogfeld „Verknüpfungen“ festgelegt, wird zum Beispiel überprüft, dass die Basisklasse im Klassenpfad der FLA-Datei zu finden ist. Es wird sichergestellt, dass im Dialogfeld „Einstellungen für Veröffentlichungen“ auf der Registerkarte „Flash“ die Option „ActionScript 3.0“ ausgewählt ist usw. Diese Überprüfungen werden nicht ausgeführt, wenn Sie diese Eigenschaft in einem Skript festlegen.

Beispiel

Die folgenden Codezeilen zeigen einige Möglichkeiten, diese Eigenschaft zu verwenden:

```
// sets the library item base class to "Sprite"
fl.getDocumentDOM().library.items[0].linkageBaseClass = "flash.display.Sprite";
// sets the library item base class to the default for that item type
fl.getDocumentDOM().library.items[0].linkageBaseClass = "";
// finds and displays the library item's base class
fl.trace(fl.getDocumentDOM().library.items[0].linkageBaseClass);
```

Siehe auch

[document.docClass](#)

item.linkageClassName

Verfügbarkeit

Flash MX 2004.

Verwendung

`item.linkageClassName`

Beschreibung

Eigenschaft; ein String, der die ActionScript 2.0-Klasse angibt, die mit dem Symbol verknüpft wird. (Um diesen Wert für eine ActionScript 3.0-Klasse anzugeben, verwenden Sie `item.linkageBaseClass`.)

Um diese Eigenschaft zu definieren, muss die Eigenschaft `item.linkageExportForAS` und/oder `item.linkageExportForRS` auf `true` festgelegt werden und die Eigenschaft `item.linkageImportForRS` muss auf `false` festgelegt werden.

Beispiel

Im folgenden Beispiel wird festgelegt, dass der mit dem ersten Element in der Bibliothek verknüpfte ActionScript 2.0-Klassenname `myClass` lautet:

```
fl.getDocumentDOM().library.items[0].linkageClassName = "myClass";
```

item.linkageExportForAS

Verfügbarkeit

Flash MX 2004.

Verwendung

`item.linkageExportForAS`

Beschreibung

Eigenschaft; ein boolescher Wert. Wenn diese Eigenschaft `true` lautet, wird das Element für ActionScript exportiert. Sie können auch den Eigenschaften `item.linkageExportForRS` und `item.linkageExportInFirstFrame` den Wert `true` zuweisen.

Wenn Sie diese Eigenschaft auf „true“ setzen, müssen Sie `item.linkageImportForRS` auf `false` einstellen. Außerdem müssen Sie einen Bezeichner (`item.linkageIdentifier`) und eine URL (`item.linkageURL`) angeben.

Beispiel

Im folgenden Beispiel wird diese Eigenschaft für das angegebene Bibliothekselement festgelegt:

```
fl.getDocumentDOM().library.items[0].linkageExportForAS = true;
```

item.linkageExportForRS

Verfügbarkeit

Flash MX 2004.

Verwendung

`item.linkageExportForRS`

Beschreibung

Eigenschaft; ein boolescher Wert. Wenn diese Eigenschaft `true` lautet, wird das Element für die gemeinsame Nutzung zur Laufzeit exportiert. Sie können auch den Eigenschaften `item.linkageExportForAS` und `item.linkageExportInFirstFrame` den Wert `true` zuweisen.

Wenn Sie diese Eigenschaft auf „true“ setzen, müssen Sie `item.linkageImportForRS` auf `false` einstellen. Außerdem müssen Sie einen Bezeichner (`item.linkageIdentifier`) und eine URL (`item.linkageURL`) angeben.

Beispiel

Im folgenden Beispiel wird diese Eigenschaft für das angegebene Bibliothekselement festgelegt:

```
fl.getDocumentDOM().library.items[0].linkageExportForRS = true;
```

item.linkageExportInFirstFrame

Verfügbarkeit

Flash MX 2004.

Verwendung

`item.linkageExportInFirstFrame`

Beschreibung

Eigenschaft; ein boolescher Wert. Wenn der Wert `true` lautet, wird das Element in das erste Bild exportiert; bei `false` wird das Element in das Bild der ersten Instanz exportiert. Wenn das Element nicht auf der Bühne angezeigt wird, wird es nicht exportiert.

Für diese Eigenschaft kann der Wert `true` nur festgelegt werden, wenn `item.linkageExportForAS` und/oder `item.linkageExportForRS` ebenfalls den Wert `true` aufweisen.

Beispiel

Im folgenden Beispiel wird festgelegt, dass das angegebene Bibliothekselement in das erste Bild exportiert wird:

```
fl.getDocumentDOM().library.items[0].linkageExportInFirstFrame = true;
```

item.linkageIdentifier

Verfügbarkeit

Flash MX 2004.

Verwendung

`item.linkageIdentifier`

Beschreibung

Eigenschaft; ein String mit dem Namen, anhand dessen Flash den Bestand beim Verknüpfen mit der SWF-Zielfile identifiziert. Flash ignoriert diese Eigenschaft, wenn `item.linkageImportForRS`, `item.linkageExportForAS` und `item.linkageExportForRS` auf `false` gesetzt sind. Auf der anderen Seite muss diese Eigenschaft immer dann eingestellt werden, wenn mindestens eine der vorgenannten Eigenschaften `true` lautet.

Beispiel

Im folgenden Beispiel wird festgelegt, dass der String `my_mc` zur Identifizierung des Bibliothekselements dient, wenn das Element mit der SWF-Zieldatei verknüpft wird, in die es exportiert werden soll:

```
fl.getDocumentDOM().library.items[0].linkageIdentifier = "my_mc";
```

Siehe auch

[item.linkageURL](#)

item.linkageImportForRS

Verfügbarkeit

Flash MX 2004.

Verwendung

`item.linkageImportForRS`

Beschreibung

Eigenschaft; wenn dieser boolesche Wert `true` lautet, wird das Element für die gemeinsame Nutzung zur Laufzeit importiert. Wenn diese Eigenschaft den Wert `true` hat, müssen sowohl [item.linkageExportForAS](#) als auch [item.linkageExportForRS](#) auf `false` eingestellt werden. Außerdem müssen Sie einen Bezeichner ([item.linkageIdentifier](#)) und eine URL ([item.linkageURL](#)) angeben.

Beispiel

Im folgenden Beispiel wird diese Eigenschaft für das angegebene Bibliothekselement auf `true` eingestellt:

```
fl.getDocumentDOM().library.items[0].linkageImportForRS = true;
```

item.linkageURL

Verfügbarkeit

Flash MX 2004.

Verwendung

`item.linkageURL`

Beschreibung

Eigenschaft; ein String mit der URL der SWF-Datei, die den gemeinsamen Bestand enthält. Flash ignoriert diese Eigenschaft, wenn [item.linkageImportForRS](#), [item.linkageExportForAS](#) und [item.linkageExportForRS](#) auf `false` gesetzt sind. Auf der anderen Seite muss diese Eigenschaft immer dann eingestellt werden, wenn mindestens eine der vorgenannten Eigenschaften `true` lautet. Sie können eine Web-URL oder einen Dateinamen in einem plattformabhängigen Format angeben, d. h. je nach Plattform mit rechts- bzw. linksgeneigten Schrägstrichen ([/] bzw. [\\]).

Beispiel

Im folgenden Beispiel wird eine Verknüpfungs-URL für das angegebene Bibliothekselement angegeben:

```
fl.getDocumentDOM().library.items[0].linkageURL = "theShareSWF.swf";
```

Siehe auch

[item.linkageIdentifier](#)

item.name

Verfügbarkeit

Flash MX 2004.

Verwendung

`item.name`

Beschreibung

Methode; ein String mit dem Namen des Bibliothekselements, der die Ordnerstruktur einschließt. Wenn „Symbol_1“ sich beispielsweise in einem Ordner mit dem Namen „Folder_1“ befindet, lautet die Eigenschaft `name` von „Symbol_1“ `"Folder_1/Symbol_1"`.

Beispiel

Im folgenden Beispiel wird der Name des angegebenen Bibliothekselements im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace(fl.getDocumentDOM().library.items[0].name);
```

item.removeData()

Verfügbarkeit

Flash MX 2004.

Verwendung

`item.removeData(name)`

Parameter

name Gibt den Namen der Daten an, die aus dem Bibliothekselement entfernt werden sollen.

Rückgabewerte

Keine.

Beschreibung

Eigenschaft; entfernt permanente Daten aus dem Bibliothekselement.

Beispiel

Im folgenden Beispiel werden die Daten mit dem Namen `myData` aus dem ersten Element in der Bibliothek entfernt:

```
fl.getDocumentDOM().library.items[0].removeData("myData");
```

Kapitel 25: Layer-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Layer-Objekt stellt eine Ebene in der Zeitleiste dar. Die Eigenschaft `timeline.layers` enthält ein Array von Layer-Objekten, auf die Sie über `fl.getDocumentDOM().getTimeline().layers` zugreifen können.

Übersicht über Eigenschaften

Für das Layer-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>layer.color</code>	Ein String, ein hexadezimaler Wert oder eine Ganzzahl, der bzw. die angibt, welche Farbe zur Konturierung der Ebene eingesetzt wird.
<code>layer.frameCount</code>	Schreibgeschützt; eine Ganzzahl, die angibt, wie viele Bilder die Ebene enthält.
<code>layer.frames</code>	Schreibgeschützt; ein Array von Frame-Objekten.
<code>layer.height</code>	Eine Ganzzahl, die die Höhe der Ebene als Prozentwert angibt. Entspricht dem Wert „Höhe der Ebene“ im Dialogfeld „Ebeneneigenschaften“.
<code>layer.layerType</code>	Ein String, der die aktuelle Verwendung der Ebene angibt. Entspricht dem Wert „Stil“ im Dialogfeld „Ebeneneigenschaften“.
<code>layer.locked</code>	Ein boolescher Wert, der angibt, ob die Ebene gesperrt ist.
<code>layer.name</code>	Ein String, der den Namen der Ebene angibt.
<code>layer.outline</code>	Ein boolescher Wert, der angibt, ob alle in der Ebene enthaltenen Objekte mit Konturen angezeigt werden.
<code>layer.parentLayer</code>	Ein Layer-Objekt, das den übergeordneten Ordner, die Führungsebene oder die Maskenebene der Ebene darstellt.
<code>layer.visible</code>	Ein boolescher Wert, der angibt, ob die Ebenenobjekte auf der Bühne ein- oder ausgeblendet sind.

layer.color

Verfügbarkeit

Flash MX 2004.

Verwendung

`layer.color`

Beschreibung

Eigenschaft; die zur Ebenenkoturierung verwendete Farbe. Sie wird in einem der folgenden Formate angegeben:

- als String im Format `"#RRGGBB"` oder `"#RRGGBBAA"`
- als Hexadezimalzahl im Format `"0xRRGGBB"`

- als Ganzzahl, die die dezimale Entsprechung einer hexadezimalen Zahl darstellt

Diese Eigenschaft entspricht der Einstellung „Konturfarbe“ im Dialogfeld „Ebeneneinstellungen“.

Beispiel

Im folgenden Beispiel wird der Wert der ersten Ebene in der Variablen `colorValue` gespeichert:

```
var colorValue = fl.getDocumentDOM().getTimeline().layers[0].color;
```

Im folgenden Beispiel wird demonstriert, wie man der ersten Ebene auf drei verschiedene Arten eine rote Farbe zuweisen kann:

```
fl.getDocumentDOM().getTimeline().layers[0].color=16711680;  
fl.getDocumentDOM().getTimeline().layers[0].color="#ff0000";  
fl.getDocumentDOM().getTimeline().layers[0].color=0xFF0000;
```

layer.frameCount

Verfügbarkeit

Flash MX 2004.

Verwendung

`layer.frameCount`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die angibt, wie viele Bilder die Ebene enthält.

Beispiel

Im folgenden Beispiel wird die Anzahl der Bilder, die sich in der ersten Ebene befinden, in der Variablen `fcNum` gespeichert:

```
var fcNum = fl.getDocumentDOM().getTimeline().layers[0].frameCount;
```

layer.frames

Verfügbarkeit

Flash MX 2004.

Verwendung

`layer.frames`

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von Frame-Objekten (siehe [Frame-Objekt](#)).

Beispiel

Im folgenden Beispiel wird die Variable `frameArray` auf das Array der Frame-Objekte für die Bilder im aktuellen Dokument festgesetzt:

```
var frameArray = fl.getDocumentDOM().getTimeline().layers[0].frames;
```

Im folgenden Beispiel wird geprüft, ob die Eigenschaft `frame.startFrame` mit dem Array-Index übereinstimmt, um festzustellen, ob ein Bild ein Schlüsselbild ist:

```
var frameArray = fl.getDocumentDOM().getTimeline().layers[0].frames;
var n = frameArray.length;
for (i=0; i<n; i++) {
    if (i==frameArray[i].startFrame) {
        alert("Keyframe at: " + i);
    }
}
```

layer.height

Verfügbarkeit

Flash MX 2004.

Verwendung

`layer.height`

Beschreibung

Eigenschaft; eine Ganzzahl, die die Höhe der Ebene als Prozentwert angibt. Entspricht dem Wert „Höhe der Ebene“ im Dialogfeld „Ebeneneigenschaften“. Zulässige Werte sind Prozentangaben der Standardhöhe: 100, 200 oder 300.

Beispiel

Im folgenden Beispiel wird die Höheneinstellung der ersten Ebene als Prozentwert gespeichert:

```
var layerHeight = fl.getDocumentDOM().getTimeline().layers[0].height;
```

Im folgenden Beispiel wird die Höhe der ersten Ebene auf 300 Prozent festgesetzt:

```
fl.getDocumentDOM().getTimeline().layers[0].height = 300;
```

layer.layerType

Verfügbarkeit

Flash MX 2004.

Verwendung

`layer.layerType`

Beschreibung

Eigenschaft; ein String, der die aktuelle Verwendung der Ebene angibt. Entspricht dem Wert „Stil“ im Dialogfeld „Ebeneneigenschaften“. Die zulässigen Werte lauten "normal", "guide", "guided", "mask", "masked" und "folder".

Beispiel

Im folgenden Beispiel wird der ersten Ebene in der Zeitleiste der Typ `folder` zugewiesen:

```
fl.getDocumentDOM().getTimeline().layers[0].layerType = "folder";
```

layer.locked

Verfügbarkeit

Flash MX 2004.

Verwendung

`layer.locked`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob die Ebene gesperrt ist. Wenn der Wert `true` lautet, ist die Ebene gesperrt. Der Standardwert lautet `false`.

Beispiel

Im folgenden Beispiel wird der boolesche Wert für den Status (gesperrt oder nicht gesperrt) der ersten Ebene in der Variablen `lockStatus` gespeichert:

```
var lockStatus = fl.getDocumentDOM().getTimeline().layers[0].locked;
```

Im folgenden Beispiel wird der ersten Ebene der Status „nicht gesperrt“ zugewiesen:

```
fl.getDocumentDOM().getTimeline().layers[0].locked = false;
```

layer.name

Verfügbarkeit

Flash MX 2004.

Verwendung

`layer.name`

Beschreibung

Eigenschaft; ein String, der den Namen der Ebene angibt.

Beispiel

Im folgenden Beispiel wird der ersten Ebene im aktuellen Dokument der Name `foreground` zugewiesen:

```
fl.getDocumentDOM().getTimeline().layers[0].name = "foreground";
```

layer.outline

Verfügbarkeit

Flash MX 2004.

Verwendung

`layer.outline`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob alle in der Ebene enthaltenen Objekte mit Konturen angezeigt werden. Wenn der Wert `true` lautet, werden sämtliche Objekte in der Ebene nur mit Konturen angezeigt. Bei `false` werden die Objekte in ihrer ursprünglichen Form angezeigt.

Beispiel

Im folgenden Beispiel wird festgelegt, dass alle Objekte auf der ersten Ebene nur mit Konturen angezeigt werden:

```
fl.getDocumentDOM().getTimeline().layers[0].outline = true;
```

layer.parentLayer

Verfügbarkeit

Flash MX 2004.

Verwendung

`layer.parentLayer`

Beschreibung

Eigenschaft; ein Layer-Objekt, das den übergeordneten Ordner, die Führungsebene oder die Maskenebene der Ebene darstellt. Als übergeordnete Ebene kommen in Frage: ein Ordner, eine Führungsebene oder eine Maskenebene, der bzw. die der Ebene vorausgeht, sowie die übergeordnete Ebene `parentLayer` der vorausgehenden oder nachfolgenden Ebene. Wenn Sie die übergeordnete Ebene `parentLayer` der Ebene festlegen, wird die Ebene innerhalb der Liste nicht an eine andere Position verschoben. Es ist nicht möglich, die übergeordnete Ebene `parentLayer` einer Ebene auf eine Ebene festzusetzen, wenn dies bedeutet, dass die Ebene verschoben werden müsste. Die oberste Ebene in einer Ebenenhierarchie hat den Wert `null`.

Beispiel

Im folgenden Beispiel werden zwei Ebenen verwendet, die sich in derselben Hierarchieebene und in derselben Zeitleiste befinden. Die erste Ebene (`layers[0]`) wird in einen Ordner konvertiert und dann der zweiten Ebene (`layers[1]`) als übergeordneter Ordner zugewiesen. Dadurch wird die zweite Ebene in die erste Ebene verschoben.

```
var parLayer = fl.getDocumentDOM().getTimeline().layers[0];  
parLayer.layerType = "folder";  
fl.getDocumentDOM().getTimeline().layers[1].parentLayer = parLayer;
```

layer.visible

Verfügbarkeit

Flash MX 2004.

Verwendung

`layer.visible`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob die Ebenenobjekte auf der Bühne ein- oder ausgeblendet sind. Wenn der Wert `true` lautet, sind alle Objekte in der Ebene sichtbar; bei `false` sind sie verborgen. Der Standardwert lautet `true`.

Beispiel

Im folgenden Beispiel wird festgelegt, dass alle Objekte in der ersten Ebene unsichtbar sind:

```
fl.getDocumentDOM().getTimeline().layers[0].visible = false;
```

Kapitel 26: library-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das library-Objekt stellt das Bedienfeld „Bibliothek“ dar und ist eine Eigenschaft des Document-Objekts (siehe `document.library`). Der Zugriff auf das library-Objekt kann über `fl.getDocumentDOM().library` erfolgen.

Das library-Objekt enthält ein Array von verschiedenartigen Elementen; dazu gehören unter anderem Symbole, Bitmaps, Sounds und Video.

Übersicht über Methoden

Für das library-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>library.addItemToDocument()</code>	Fügt das aktuelle oder angegebene Element an einer bestimmten Stelle der Bühne hinzu.
<code>library.addNewItem()</code>	Erstellt im Bedienfeld „Bibliothek“ ein neues Element des angegebenen Typs und wählt dieses Element aus.
<code>library.deleteItem()</code>	Löscht die aktuellen Elemente bzw. ein angegebenes Element aus dem Bedienfeld „Bibliothek“.
<code>library.duplicateItem()</code>	Erstellt von dem gegenwärtig ausgewählten oder angegebenen Element eine Kopie.
<code>library.editItem()</code>	Öffnet das gegenwärtig ausgewählte oder angegebene Element im Bearbeitungsmodus.
<code>library.expandFolder()</code>	Erweitert oder reduziert den in der Bibliothek gegenwärtig ausgewählten oder angegebenen Ordner.
<code>library.findItemIndex()</code>	Gibt den Indexwert (auf Null basierend) des Bibliothekselements zurück.
<code>library.getItemProperty()</code>	Ruft die Eigenschaft des ausgewählten Elements ab.
<code>library.getItemType()</code>	Ruft den Typ des Objekts ab, das gegenwärtig ausgewählt oder durch einen Bibliothekspfad näher bezeichnet ist.
<code>library.getSelectedItems()</code>	Ruft das Array aller in der Bibliothek gegenwärtig ausgewählten Elemente ab.
<code>library.importEmbeddedSWF()</code>	Importiert eine SWF-Datei als kompilierten Clip in die Bibliothek.
<code>library.itemExists()</code>	Prüft, ob ein angegebenes Element in der Bibliothek vorhanden ist.
<code>library.moveToFolder()</code>	Verschiebt das gegenwärtig ausgewählte oder angegebene Bibliothekselement in einen bestimmten Ordner.
<code>library.newFolder()</code>	Erstellt im gegenwärtig ausgewählten Ordner einen neuen Ordner. Dem neuen Ordner wird entweder der angegebene Name zugewiesen oder ein Standardname ("Unbenannter Ordner Nummer", falls der Parameter <code>folderName</code> fehlt).
<code>library.renameItem()</code>	Benennt das im Bedienfeld „Bibliothek“ gegenwärtig ausgewählte Bibliothekselement um.
<code>library.selectAll()</code>	Wählt alle Elemente in der Bibliothek aus bzw. hebt die Elementauswahl auf.

Methode	Beschreibung
<code>library.selectItem()</code>	Wählt ein bestimmtes Bibliothekselement aus.
<code>library.selectNone()</code>	Hebt die Auswahl für alle Bibliothekselemente auf.
<code>library.setItemProperty()</code>	Legt die Eigenschaft für alle ausgewählten Bibliothekselemente fest (Ordner werden dabei ignoriert).
<code>library.updateItem()</code>	Aktualisiert das angegebene Element.

Übersicht: Eigenschaften des library-Objekts

Für das library-Objekt ist die folgende Eigenschaft verfügbar:

Eigenschaft	Beschreibung
<code>library.items</code>	Ein Array von Item-Objekten, die in der Bibliothek vorhanden sind.

library.addItemToDocument()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.addItemToDocument(position [, namePath])
```

Parameter

position Ein Punkt, der die *x,y*-Koordinaten des Elementmittelpunkts auf der Bühne angibt.

namePath Ein String, der den Namen des Elements angibt. Wenn sich das Element in einem Ordner befindet, können Sie seinen Namen und Pfad anhand von Schrägstrichnotation angeben. Wenn *namePath* fehlt, wird die aktuelle Bibliotheksauswahl verwendet. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; *true*, wenn das Element dem Dokument erfolgreich hinzugefügt wurde, andernfalls *false*.

Beschreibung

Methode; fügt das aktuelle oder angegebene Element an einer bestimmten Stelle der Bühne hinzu.

Beispiel

Im folgenden Beispiel wird das gegenwärtig ausgewählte Element der Bühne an der durch die Koordinaten (3, 60) bezeichneten Stelle hinzugefügt:

```
fl.getDocumentDOM().library.addItemToDocument({x:3, y:60});
```

Im folgenden Beispiel wird das Element *Symbol1*, das sich im Ordner „folder1“ der Bibliothek befindet, der Bühne an der durch die Koordinaten (550, 485) bezeichneten Stelle hinzugefügt:

```
fl.getDocumentDOM().library.addItemToDocument({x:550.0, y:485.0}, "folder1/Symbol1");
```

library.addNewItem()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.addNewItem(type [, namePath])
```

Parameter

type Ein String, der den zu erstellenden Elementtyp angibt. Für *type* sind nur die Werte "video", "movie clip", "button", "graphic", "bitmap", "screen" und "folder" zulässig (Sie können der Bibliothek mit dieser Methode also beispielsweise keine Sounds hinzufügen). Durch Angabe eines Ordnerpfads erzielen Sie dasselbe Ergebnis wie durch Verwendung von `library.newFolder()` vor dem Aufruf dieser Methode.

namePath Ein String, der den Namen des hinzuzufügenden Elements angibt. Wenn sich das Element in einem Ordner befindet, geben Sie seinen Namen und Pfad anhand von Schrägstrichnotation an. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn das Element erfolgreich hinzugefügt wurde, ansonsten `false`.

Beschreibung

Methode; erstellt im Bedienfeld „Bibliothek“ ein neues Element des angegebenen Typs und wählt dieses Element aus. Weitere Informationen zum Importieren von Objekten in die Bibliothek, einschließlich solcher Objekte wie Sounds, finden Sie unter `document.importFile()`.

Beispiel

Im folgenden Beispiel wird ein neues Schaltflächenelement mit dem Namen `start` in einem neuen Ordner mit dem Namen `folderTwo` erstellt:

```
fl.getDocumentDOM().library.addNewItem("button", "folderTwo/start");
```

library.deleteItem()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.deleteItem([namePath])
```

Parameter

namePath Ein String, der den Namen des zu löschenden Elements angibt. Wenn sich das Element in einem Ordner befindet, können Sie seinen Namen und Pfad anhand von Schrägstrichnotation angeben. Wenn Sie einen Ordernamen übergeben, werden der Ordner und alle darin enthaltenen Elemente gelöscht. Wird kein Name angegeben, löscht Flash das gegenwärtig ausgewählte Element bzw. die gegenwärtig ausgewählten Elemente. Wenn Sie alle Elemente aus dem Bedienfeld „Bibliothek“ löschen möchten, wählen Sie zuerst alle Elemente aus und verwenden Sie dann diese Methode. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die Elemente erfolgreich gelöscht wurden, ansonsten `false`.

Beschreibung

Methode; löscht die aktuellen Elemente bzw. ein angegebenes Element aus dem Bedienfeld „Bibliothek“. Diese Methode kann sich auf mehrere Elemente auswirken, sofern mehrere Elemente ausgewählt sind.

Beispiel

Im folgenden Beispiel wird das gegenwärtig ausgewählte Element gelöscht:

```
fl.getDocumentDOM().library.deleteItem();
```

Im folgenden Beispiel wird das Element `Symbol_1` aus dem Bibliotheksordner `Folder_1` gelöscht:

```
fl.getDocumentDOM().library.deleteItem("Folder_1/Symbol_1");
```

library.duplicateItem()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.duplicateItem( [ namePath ] )
```

Parameter

namePath Ein String, der den Namen des zu duplizierenden Elements angibt. Wenn sich das Element in einem Ordner befindet, können Sie seinen Namen und Pfad anhand von Schrägstrichnotation angeben. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn das Element erfolgreich dupliziert wurde, ansonsten `false`. Wenn Sie mehrere Elemente ausgewählt haben, gibt Flash `false` zurück.

Beschreibung

Methode; erstellt von dem gegenwärtig ausgewählten oder angegebenen Element eine Kopie. Das neue Element hat einen Standardnamen (z. B. `Element-Kopie`) und wird ausgewählt. Wenn mehrere Elemente ausgewählt wurden, schlägt der Befehl fehl.

Beispiel

Im folgenden Beispiel wird von dem Element `square`, das sich im Bibliotheksordner `test` befindet, eine Kopie erstellt:

```
fl.getDocumentDOM().library.duplicateItem("test/square");
```

library.editItem()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.editItem([namePath])
```

Parameter

namePath Ein String, der den Namen des Elements angibt. Wenn sich das Element in einem Ordner befindet, können Sie seinen Namen und Pfad anhand von Schrägstrichnotation angeben. Wenn *namePath* fehlt, wird das ausgewählte Bibliothekselement im Bearbeitungsmodus geöffnet. Wenn in der Bibliothek gegenwärtig keine oder mehrere Elemente ausgewählt sind, wird die erste Szene in der Hauptzeitleiste zur Bearbeitung angezeigt. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; *true*, wenn das angegebene Element vorhanden ist und bearbeitet werden kann, ansonsten *false*.

Beschreibung

Methode; öffnet das gegenwärtig ausgewählte oder angegebene Element im Bearbeitungsmodus.

Beispiel

Im folgenden Beispiel wird das Element *circle* im Bibliotheksordner *test* zur Bearbeitung geöffnet:

```
fl.getDocumentDOM().library.editItem("test/circle");
```

library.expandFolder()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.expandFolder(bExpand [, bRecurseNestedParents [, namePath]])
```

Parameter

bExpand Ein boolescher Wert; wenn der Wert *true* lautet, wurde der Ordner erweitert; bei *false* (Standardwert) ist er reduziert.

bRecurseNestedParents Ein boolescher Wert; wenn der Wert *true* lautet, wurden alle Ordner innerhalb des angegebenen Ordners je nach dem Wert von *bExpand* erweitert oder reduziert. Der Standardwert lautet *false*. Dieser Parameter ist optional.

namePath Ein String, der den Namen und optional auch den Pfad des zu erweiternden bzw. reduzierenden Ordners angibt. Wenn dieser Parameter fehlt, gilt die Methode für den gegenwärtig ausgewählten Ordner. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; *true*, wenn das Element erfolgreich erweitert oder reduziert wurde; *false*, wenn der Vorgang nicht erfolgreich war oder wenn das angegebene Element kein Ordner ist.

Beschreibung

Methode; erweitert oder reduziert den in der Bibliothek gegenwärtig ausgewählten oder angegebenen Ordner.

Beispiel

Im folgenden Beispiel werden der Bibliotheksordner „test“ und alle darin eventuell vorhandenen Ordner reduziert:

```
fl.getDocumentDOM().library.expandFolder(false, true, "test");
```

library.findItemIndex()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.findItemIndex(namePath)
```

Parameter

namePath Ein String, der den Namen des Elements angibt. Wenn sich das Element in einem Ordner befindet, können Sie seinen Namen und Pfad anhand von Schrägstrichnotation angeben.

Rückgabewerte

Ein Ganzzahlenwert, der den Indexwert (auf Null basierend) des Elements angibt.

Beschreibung

Methode; gibt den Indexwert (auf Null basierend) des Bibliothekselements zurück. Der Bibliotheksindex ist flach, d. h. alle Ordner gehören dem Hauptindex an. Verschachtelte Elemente können anhand von Ordnerpfaden angegeben werden.

Beispiel

Im folgenden Beispiel wird der auf Null basierende Indexwert des Bibliothekselements „square“, das sich im Ordner „test“ befindet, in der Variablen `sqIndex` gespeichert. Anschließend wird der Indexwert in einem Dialogfeld angezeigt:

```
var sqIndex = fl.getDocumentDOM().library.findItemIndex("test/square");  
alert(sqIndex);
```

library.getItemProperty()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.getItemProperty(property)
```

Parameter

property Ein String. Eine Liste der Werte, die als *property*-Parameter verwendet werden können, finden Sie in der Übersicht über die Eigenschaften für das [Item-Objekt](#) und in den Eigenschaftsübersichten der zugehörigen Unterklassen.

Rückgabewerte

Ein Stringwert für die Eigenschaft.

Beschreibung

Methode; ruft die Eigenschaft des ausgewählten Elements ab.

Beispiel

Im folgenden Beispiel wird ein Dialogfeld mit dem Wert des Verknüpfungsbezeichners des Symbols angezeigt, wenn über ActionScript bzw. zum Runtime Sharing (gemeinsame Nutzung zur Laufzeit) darauf verwiesen wird:

```
alert(fl.getDocumentDOM().library.getItemProperty("linkageIdentifier"));
```

library.getItemType()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.getItemType([namePath])
```

Parameter

namePath Ein String, der den Namen des Elements angibt. Wenn sich das Element in einem Ordner befindet, geben Sie seinen Namen und Pfad anhand von Schrägstrichnotation an. Wenn *namePath* fehlt, gibt Flash den Typ der aktuellen Auswahl zurück. Wenn mehrere Elemente ausgewählt wurden und *namePath* fehlt, wird der Befehl von Flash ignoriert. Dieser Parameter ist optional.

Rückgabewerte

Ein Stringwert, der den Typ des Objekts angibt. Informationen zu möglichen Rückgabewerten finden Sie unter [item.itemType](#).

Beschreibung

Methode; ruft den Typ des Objekts ab, das gegenwärtig ausgewählt oder durch einen Bibliothekspfad näher bezeichnet ist.

Beispiel

Im folgenden Beispiel wird in einem Dialogfeld der Elementtyp *Symbol_1*, der sich im Ordner *Folder_1/Folder_2* befindet, angezeigt:

```
alert(fl.getDocumentDOM().library.getItemType("Folder_1/Folder_2/Symbol_1"));
```

library.getSelectedItems()

Verfügbarkeit

Flash MX 2004.

Parameter

Keine.

Rückgabewerte

Ein Array der Werte aller in der Bibliothek gegenwärtig ausgewählten Elemente.

Beschreibung

Methode; ruft das Array aller in der Bibliothek gegenwärtig ausgewählten Elemente ab.

Beispiel

Im folgenden Beispiel wird das Array der gegenwärtig ausgewählten Bibliothekselemente (in diesem Fall mehrere Audiodateien) in der Variablen `selItems` gespeichert. Anschließend wird im Array die Eigenschaft `sampleRate` der ersten Audiodatei in 11 kHz geändert:

```
var selItems = fl.getDocumentDOM().library.getSelectedItems();  
selItems[0].sampleRate = "11 kHz";
```

library.importEmbeddedSWF()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.importEmbeddedSWF(linkageName, swfData [, libName])
```

Parameter

linkageName Ein String mit dem Namen der SWF-Verknüpfung des Stamm-Movieclips.

swfData Ein Array von binären SWF-Daten, die aus einer externen Bibliothek oder DLL stammen.

libName Ein String, der den Bibliotheksnamen für das erstellte Element angibt. Wenn der Name bereits in Gebrauch ist, erstellt die Methode einen anderen Namen. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; importiert eine SWF-Datei als kompilierten Clip in die Bibliothek. Anders als bei dem Befehl „Datei“ > „Importieren“ > „SWF“ können Sie bei dieser Methode eine kompilierte SWF-Datei in die Bibliothek einbetten. Es gibt keine entsprechende Benutzeroberflächenfunktion. Diese Methode muss in Verbindung mit einer externen Bibliothek oder DLL verwendet werden (siehe „[C-Level-Erweiterbarkeit](#)“ auf Seite 539).

Die zu importierende SWF-Datei muss über einen Movieclip auf der obersten Ebene verfügen, der den gesamten Inhalt enthält. Deren Verknüpfungsbezeichner sollte auf den gleichen Wert eingestellt sein, mit dem der Parameter *linkageName* an diese Methode übergeben wird.

Beispiel

Im folgenden Beispiel wird der Bibliothek die SWF-Datei mit dem *linkageName*-Wert `MyMovie` als kompilierter Clip mit dem Namen `Intro` hinzugefügt:

```
fl.getDocumentDOM().library.importEmbeddedSWF("MyMovie", swfData, "Intro");
```

library.itemExists()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.itemExists(namePath)
```

Parameter

namePath Ein String, der den Namen des Elements angibt. Wenn sich das Element in einem Ordner befindet, geben Sie seinen Namen und Pfad anhand von Schrägstrichnotation an.

Rückgabewerte

Ein boolescher Wert; `true`, wenn das angegebene Element in der Bibliothek vorhanden ist, andernfalls `false`.

Beschreibung

Methode; prüft, ob ein angegebenes Element in der Bibliothek vorhanden ist.

Beispiel

Im folgenden Beispiel wird in einem Dialogfeld `true` oder `false` angezeigt, je nachdem, ob das Element „Symbol_1“ in dem Bibliotheksordner „Folder_1“ vorhanden ist:

```
alert(fl.getDocumentDOM().library.itemExists('Folder_1/Symbol_1'));
```

library.items

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.items
```

Beschreibung

Eigenschaft; ein Array von item-Objekten, die in der Bibliothek vorhanden sind.

Beispiel

Im folgenden Beispiel wird das Array aller Bibliothekselemente in der Variablen `itemArray` gespeichert:

```
var itemArray = fl.getDocumentDOM().library.items;
```

library.moveToFolder()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.moveToFolder(folderPath [, itemToMove [, bReplace]])
```

Parameter

folderPath Ein String, der den Pfad zu dem Ordner anhand von "FolderName" oder "FolderName/FolderName" angibt. Um ein Element auf die oberste Ebene zu verschieben, geben Sie für *folderPath* einen leeren String ("") ein.

itemToMove Ein String, der den Namen des zu verschiebenden Elements angibt. Wenn *itemToMove* fehlt, werden die gegenwärtig ausgewählten Elemente verschoben. Dieser Parameter ist optional.

bReplace Ein boolescher Wert. Wenn bereits ein gleichnamiges Element vorhanden ist und Sie den Wert `true` dem Parameter *bReplace* zuweisen, wird das vorhandene Element durch das verschobene Element ersetzt. Bei `false` wird der Name des verschobenen Elements durch einen eindeutigen Namen ersetzt. Der Standardwert lautet `false`. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn das Element erfolgreich verschoben wurde, ansonsten `false`.

Beschreibung

Methode; verschiebt das gegenwärtig ausgewählte oder angegebene Bibliothekselement in einen bestimmten Ordner. Wenn der Parameter *folderPath* leer ist, werden die Elemente auf die oberste Ebene verschoben.

Beispiel

Im folgenden Beispiel wird das Element `Symbol_1` in den Bibliotheksordner `new` verschoben. Dabei ersetzt das neue Element das gleichnamige Element in diesem Ordner:

```
fl.getDocumentDOM().library.moveToFolder("new", "Symbol_1", true);
```

library.newFolder()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.newFolder([folderPath])
```

Parameter

folderPath Ein String, der den Namen des zu erstellenden Ordners angibt. Wenn der Ordner als Pfad angegeben wird und dieser Pfad nicht vorhanden ist, wird der Pfad erstellt. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn der Ordner erfolgreich erstellt wurde, ansonsten `false`.

Beschreibung

Methode; erstellt im gegenwärtig ausgewählten Ordner einen neuen Ordner. Dem neuen Ordner wird entweder der angegebene Name zugewiesen oder ein Standardname ("Unbenannter Ordner Nummer", falls der Parameter *folderName* fehlt).

Beispiel

Im folgenden Beispiel werden zwei neue Bibliotheksordner erstellt. Der zweite Ordner ist ein Unterordner des ersten Ordners:

```
fl.getDocumentDOM().library.newFolder("first/second");
```

library.renameItem()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.renameItem(name)
```

Parameter

name Ein String, der für das Bibliothekselement einen neuen Namen angibt.

Rückgabewerte

Der boolesche Wert `true`, wenn der Name des Elements geändert wurde, andernfalls `false`. Wenn mehrere Elemente ausgewählt wurden, bleiben alle Namen unverändert. In diesem Fall wird (entsprechend dem Verhalten der Benutzeroberfläche) der Wert `false` zurückgegeben.

Beschreibung

Methode; benennt das im Bedienfeld „Bibliothek“ gegenwärtig ausgewählte Bibliothekselement um.

Beispiel

Im folgenden Beispiel wird das gegenwärtig ausgewählte Bibliothekselement in `new name` umbenannt:

```
fl.getDocumentDOM().library.renameItem("new name");
```

library.selectAll()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.selectAll([bSelectAll])
```

Parameter

bSelectAll Ein boolescher Wert, der angibt, ob in der Bibliothek sämtliche Elemente ausgewählt werden sollen bzw. ob die Elementauswahl aufgehoben werden soll. Lassen Sie diesen Parameter weg oder verwenden Sie den Standardwert `true`, um alle Elemente in der Bibliothek auszuwählen; mit `false` wird die Elementauswahl in der Bibliothek aufgehoben. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; wählt alle Elemente in der Bibliothek aus bzw. hebt die Elementauswahl auf.

Beispiel

Im folgenden Beispiel werden alle Elemente in der Bibliothek ausgewählt:

```
fl.getDocumentDOM().library.selectAll();  
fl.getDocumentDOM().library.selectAll(true);
```

Im folgenden Beispiel wird die Elementauswahl in der Bibliothek vollständig aufgehoben:

```
fl.getDocumentDOM().library.selectAll(false);  
fl.getDocumentDOM().library.selectNone();
```

library.selectItem()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.selectItem(namePath [, bReplaceCurrentSelection [, bSelect]])
```

Parameter

namePath Ein String, der den Namen des Elements angibt. Wenn sich das Element in einem Ordner befindet, können Sie seinen Namen und Pfad anhand von Schrägstrichnotation angeben.

bReplaceCurrentSelection Ein boolescher Wert, der angibt, ob die aktuelle Auswahl durch das Element ersetzt oder erweitert werden soll. Der Standardwert lautet `true` (aktuelle Auswahl ersetzen). Dieser Parameter ist optional.

bSelect Ein boolescher Wert, der angibt, ob ein Element ausgewählt oder die Elementauswahl aufgehoben werden soll. Der Standardwert lautet `true` (auswählen). Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn das angegebene Element vorhanden sind, ansonsten `false`.

Beschreibung

Methode; wählt ein angegebenes Bibliothekselement aus.

Beispiel

Im folgenden Beispiel wird die aktuelle Auswahl in der Bibliothek durch `Symbol 1` im Ordner `untitled folder 1` ersetzt:

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1");
```

Im folgenden Beispiel wird die aktuelle Auswahl in der Bibliothek auf `Symbol 1` im Ordner `untitled folder 1` ausgeweitet:

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1", false);
```

Im folgenden Beispiel wird die Auswahl von `Symbol 1` in `untitled folder 1` aufgehoben. Alle anderen ausgewählten Elemente bleiben weiterhin ausgewählt:

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1", true, false);
```

library.selectNone()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.selectNone()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; hebt die Elementauswahl in der Bibliothek auf.

Beispiel

Im folgenden Beispiel wird die Elementauswahl in der Bibliothek vollständig aufgehoben:

```
fl.getDocumentDOM().library.selectNone();  
fl.getDocumentDOM().library.selectAll(false);
```

library.setItemProperty()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.setItemProperty(property, value)
```

Parameter

property Ein String mit dem Namen der festzulegenden Eigenschaft. Eine Liste der Eigenschaften finden Sie unter [Item-Objekt](#) und in den Eigenschaftsübersichten der zugehörigen Unterklassen. Wenn Sie wissen möchten, welche Objekte Unterklassen des Item-Objekts sind, lesen Sie die „[Übersicht über die DOM-Struktur](#)“ auf Seite 12.

value Der Wert, der der angegebenen Eigenschaft zugewiesen werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; legt die Eigenschaft für alle ausgewählten Bibliothekselemente fest (Ordner werden dabei ignoriert).

Beispiel

Im folgenden Beispiel wird der Eigenschaft `symbolType` des ausgewählten Bibliothekselements bzw. der ausgewählten Bibliothekselemente der Wert „button“ zugewiesen. In diesem Fall muss es sich bei dem Element um ein [SymbolItem-Objekt](#) handeln; `symbolType` ist eine gültige Eigenschaft für `SymbolItem`-Objekte.

```
fl.getDocumentDOM().library.setItemProperty("symbolType", "button");
```

library.updateItem()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
library.updateItem([namePath])
```

Parameter

namePath Ein String, der den Namen des Elements angibt. Wenn sich das Element in einem Ordner befindet, geben Sie seinen Namen und Pfad anhand von Schrägstrichnotation an. Die gleiche Wirkung erzielen Sie, wenn Sie in der Benutzeroberfläche mit der rechten Maustaste auf ein Element klicken und in dem Kontextmenü „Aktualisieren“ auswählen. Wenn kein Name angegeben wurde, wird die aktuelle Auswahl aktualisiert. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn Flash das Element erfolgreich aktualisiert hat, ansonsten `false`.

Beschreibung

Methode; aktualisiert das angegebene Element.

Beispiel

Im folgenden Beispiel wird ein Dialogfeld angezeigt, das angibt, ob das gegenwärtig ausgewählte Element aktualisiert (`true`) oder nicht aktualisiert (`false`) wurde:

```
alert(fl.getDocumentDOM().library.updateItem());
```

Kapitel 27: Math-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Math-Objekt ist als schreibgeschützte Eigenschaft des flash-Objekts verfügbar (siehe [fl.Math](#)). Dieses Objekt bietet Methoden, die gängige Rechengänge ausführen.

Übersicht über Methoden

Für das Math-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
Math.concatMatrix()	Führt eine Matrixverkettung durch und gibt das Ergebnis zurück.
Math.invertMatrix()	Gibt die Umkehrung der angegebenen Matrix zurück.
Math.pointDistance()	Berechnet den Abstand zwischen zwei Punkten.

Math.concatMatrix()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
Math.concatMatrix(mat1, mat2)
```

Parameter

mat1, mat2 Geben die zu verkettenden Matrix-Objekte an (siehe [Matrix-Objekt](#)). Bei den Parametern muss es sich jeweils um ein Objekt mit den Feldern `a`, `b`, `c`, `d`, `tx` und `ty` handeln.

Rückgabewerte

Eine verkettete Objektmatrix.

Beschreibung

Methode; führt eine Matrixverkettung durch und gibt das Ergebnis zurück.

Beispiel

Im folgenden Beispiel wird das gegenwärtig ausgewählte Objekt in der Variablen `elt` gespeichert, die Objektmatrix mit der Ansichtsmatrix multipliziert und das Ergebnis in der Variablen `mat` gespeichert:

```
var elt = fl.getDocumentDOM().selection[0];
var mat = fl.Math.concatMatrix( elt.matrix , fl.getDocumentDOM().viewMatrix );
```


Math.invertMatrix()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
Math.invertMatrix(mat)
```

Parameter

mat Gibt das umzukehrende Matrix-Objekt an (siehe [Matrix-Objekt](#)). Die folgenden Felder müssen angegeben werden: a, b, c, d, tx und ty.

Rückgabewerte

Ein Matrix-Objekt, das die Umkehrung der Originalmatrix darstellt.

Beschreibung

Methode; gibt die Umkehrung der angegebenen Matrix zurück.

Beispiel

Im folgenden Beispiel wird das gegenwärtig ausgewählte Objekt in der Variablen `elt` gespeichert, diese Matrix der Variablen `mat` zugewiesen und die Umkehrung der Matrix in der Variablen `inv` gespeichert:

```
var elt = fl.getDocumentDOM().selection[0];  
var mat = elt.matrix;  
var inv = fl.Math.invertMatrix( mat );
```

Math.pointDistance()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
Math.pointDistance(pt1, pt2)
```

Parameter

pt1, pt2 Gibt die beiden Punkte an, deren Abstand gemessen wird.

Rückgabewerte

Ein Gleitkommawert, der den Abstand zwischen den Punkten angibt.

Beschreibung

Methode; berechnet den Abstand zwischen zwei Punkten.

Beispiel

Im folgenden Beispiel wird der Wert für den Abstand zwischen *pt1* und *pt2* in der Variablen *dist* gespeichert:

```
var pt1 = {x:10, y:20}  
var pt2 = {x:100, y:200}  
var dist = fl.Math.pointDistance(pt1, pt2);
```

Kapitel 28: Matrix-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Matrix-Objekt stellt eine Transformationsmatrix dar.

Übersicht über Eigenschaften

Für das Matrix-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>matrix.a</code>	Ein Gleitkommawert, der das Element (0,0) in der Transformationsmatrix repräsentiert.
<code>matrix.b</code>	Ein Gleitkommawert, der das Element (0,1) in der Matrix repräsentiert.
<code>matrix.c</code>	Ein Gleitkommawert, der das Element (1,0) in der Matrix repräsentiert.
<code>matrix.d</code>	Ein Gleitkommawert, der das Element (1,1) in der Matrix repräsentiert.
<code>matrix.tx</code>	Ein Gleitkommawert, der die Lage des Registrierungspunkts eines Symbols in der x-Achse oder den Mittelpunkt einer Form repräsentiert.
<code>matrix.ty</code>	Ein Gleitkommawert, der die Lage des Registrierungspunkts eines Symbols in der y-Achse oder den Mittelpunkt einer Form repräsentiert.

matrix.a

Verfügbarkeit

Flash MX 2004.

Verwendung

`matrix.a`

Beschreibung

Eigenschaft; ein Gleitkommawert, der das Element (0,0) in der Transformationsmatrix repräsentiert. Dieser Wert ist der Skalierungsfaktor der x-Achse des Objekts.

Beispiel

Die Eigenschaften a und d in einer Matrix geben die Skalierung an. Im folgenden Beispiel werden diesen Eigenschaften die Werte 2 bzw. 3 zugewiesen, um die Breite des ausgewählten Objekts zu verdoppeln und die Höhe zu verdreifachen:

```
var mat = fl.getDocumentDOM().selection[0].matrix;
mat.a = 2;
mat.d = 3;
fl.getDocumentDOM().selection[0].matrix = mat;
```

Sie können ein Objekt drehen, indem Sie die Matriceigenschaften a, b, c und d im Verhältnis zueinander festlegen, wobei $a = d$ und $b = -c$. Mit den Werten 0,5, 0,8, -0,8 und 0,5 wird das Objekt beispielsweise um 60° gedreht:

```
var mat = fl.getDocumentDOM().selection[0].matrix;  
mat.a = 0.5;  
mat.b = 0.8;  
mat.c = 0.8*(-1);  
mat.d = 0.5;  
fl.getDocumentDOM().selection[0].matrix = mat;
```

Mit $a = d = 1$ und $c = b = 0$ wird die ursprüngliche Form des Objekts wiederhergestellt.

matrix.b

Verfügbarkeit

Flash MX 2004.

Verwendung

`matrix.b`

Beschreibung

Eigenschaft; ein Gleitkommawert, der das Element (0,1) in der Matrix repräsentiert. Dieser Wert gibt die vertikale Neigung einer Form an. Er bewirkt, dass Flash die rechte Kante der Form in der vertikalen Achse verschiebt.

Die Eigenschaften `matrix.b` und `matrix.c` in einer Matrix geben die Neigung an (siehe [matrix.c](#)).

Beispiel

Im folgenden Beispiel werden `b` und `c` die Werte `-1` und `0` zugewiesen. Dadurch wird das Objekt vertikal um 45° geneigt:

```
var mat = fl.getDocumentDOM().selection[0].matrix;  
mat.b = -1;  
mat.c = 0;  
fl.getDocumentDOM().selection[0].matrix = mat;
```

Um die ursprüngliche Objektform durch eine erneute Neigung wiederherzustellen, weisen Sie `b` und `c` den Wert `0` zu.

Siehe auch das Beispiel zu [matrix.a](#).

matrix.c

Verfügbarkeit

Flash MX 2004.

Verwendung

`matrix.c`

Beschreibung

Eigenschaft; ein Gleitkommawert, der das Element (1,0) in der Matrix repräsentiert. Dieser Wert bewirkt, dass Flash das Objekt durch Verschieben der unteren Kante in der horizontalen Achse neigt.

Die Eigenschaften `matrix.b` und `matrix.c` in einer Matrix geben die Neigung an.

Beispiel

Siehe das Beispiel zu [matrix.b](#).

matrix.d

Verfügbarkeit

Flash MX 2004.

Verwendung

`matrix.d`

Beschreibung

Eigenschaft; ein Gleitkommawert, der das Element (1,1) in der Matrix repräsentiert. Dieser Wert ist der Skalierungsfaktor der *y*-Achse des Objekts.

Beispiel

Siehe das Beispiel zu [matrix.a](#).

matrix.tx

Verfügbarkeit

Flash MX 2004.

Verwendung

`matrix.tx`

Beschreibung

Eigenschaft; ein Gleitkommawert, der die Lage des Registrierungspunkts eines Symbols (auch *Bezugspunkt* oder *Nullpunkt* genannt) in der *x*-Achse oder den Mittelpunkt einer Form repräsentiert. Sie definiert die *x*-Umsetzung der Transformation.

Ein Objekt kann durch Einstellen der Eigenschaften `matrix.tx` und `matrix.ty` verschoben werden (siehe [matrix.ty](#)).

Beispiel

Im folgenden Beispiel wird `tx` und `ty` der Wert 0 zugewiesen. Der Registrierungspunkt des Objekts wird dadurch in dem Dokument zum Punkt 0,0 verschoben:

```
var mat = fl.getDocumentDOM().selection[0].matrix;  
mat.tx = 0;  
mat.ty = 0;  
fl.getDocumentDOM().selection[0].matrix = mat;
```

matrix.ty

Verfügbarkeit

Flash MX 2004.

Verwendung

`matrix.ty`

Beschreibung

Eigenschaft; ein Gleitkommawert, der die Lage des Registrierungspunkts eines Symbols in der *y*-Achse oder den Mittelpunkt einer Form repräsentiert. Sie definiert die *y*-Umsetzung der Transformation.

Ein Objekt kann durch Einstellen der Eigenschaften `matrix.tx` und `matrix.ty` verschoben werden.

Beispiel

Siehe das Beispiel zu [matrix.tx](#).

Kapitel 29: outputPanel-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Dieses Objekt stellt das Bedienfeld „Ausgabe“ dar, in dem Informationen zur Behebung von Fehlern wie z. B. Syntaxfehlern angezeigt werden. Der Zugriff auf dieses Objekt erfolgt über `fl.outputPanel` (oder `flash.outputPanel`). Siehe [fl.outputPanel](#).

Übersicht über Methoden

Für das outputPanel-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
outputPanel.clear()	Löscht den Inhalt des Bedienfelds „Ausgabe“.
outputPanel.save()	Speichert den Inhalt des Bedienfelds „Ausgabe“ in einer lokalen Textdatei.
outputPanel.trace()	Fügt dem Inhalt des Bedienfelds „Ausgabe“ eine Zeile hinzu, die durch einen Zeilenumbruch abgeschlossen wird.

outputPanel.clear()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
outputPanel.clear()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; löscht den Inhalt des Bedienfelds „Ausgabe“. Mit dieser Methode können Sie in einer Stapelverarbeitungsanwendung eine Fehlerliste löschen. Wenn Sie sie mit [outputPanel.save\(\)](#) kombinieren, können Sie die Fehler inkrementell speichern.

Beispiel

Im folgenden Beispiel wird der aktuelle Inhalt des Bedienfelds „Ausgabe“ gelöscht:

```
fl.outputPanel.clear();
```

outputPanel.save()

Verfügbarkeit

Flash MX 2004; Parameter *bUseSystemEncoding* ist neu in Flash 8.

Verwendung

```
outputPanel.save(fileURI [, bAppendToFile [ , bUseSystemEncoding]])
```

Parameter

fileURI Ein String im Format file:/// URI, der die lokale Datei angibt, in der der Inhalt des Bedienfelds „Ausgabe“ gespeichert werden soll.

bAppendToFile Ein optionaler boolescher Wert. Bei dem Wert `true` wird der Inhalt des Bedienfelds „Ausgabe“ der Ausgabedatei hinzugefügt. Bei `false` überschreibt die Methode die Ausgabedatei, sofern diese bereits vorhanden ist. Der Standardwert lautet `false`.

bUseSystemEncoding Ein optionaler boolescher Wert. Lautet der Wert `true`, wird der Text des Bedienfelds „Ausgabe“ anhand der Textkodierung des Betriebssystems gespeichert. Bei `false` wird der Text des Bedienfelds „Ausgabe“ mit UTF-8-Kodierung und BOM (Byte Order Mark)-Zeichen am Anfang des Texts gespeichert. Der Standardwert lautet `false`.

Rückgabewerte

Keine.

Beschreibung

Methode; speichert die Inhalte des Bedienfelds „Ausgabe“ in einer lokalen Textdatei. Dabei wird eine vorhandene Datei entweder überschrieben oder der Text an die Datei angehängt.

Wenn *fileURI* ungültig ist oder nicht angegeben wurde, wird ein Fehler gemeldet.

Diese Methode ist bei der Stapelverarbeitung nützlich. Sie können beispielsweise eine JSFL-Datei erstellen, die mehrere Komponenten kompiliert. Alle Kompilierungsfehler werden im Bedienfeld „Ausgabe“ angezeigt. Mit dieser Methode können Sie die Fehler in einer Textdatei speichern, die wiederum automatisch von dem verwendeten Buildsystem analysiert werden kann.

Beispiel

Im folgenden Beispiel wird der Inhalt des Bedienfelds „Ausgabe“ in der Datei „batch.log“ im Ordner „/tests“ gespeichert. Dabei wird die Datei „batch.log“ überschrieben, sofern sie bereits vorhanden ist:

```
fl.outputPanel.save("file:///c:/tests/batch.log");
```

outputPanel.trace()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
outputPanel.trace(message)
```


Parameter

message Ein String mit dem Text, der dem Bedienfeld „Ausgabe“ hinzugefügt werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; sendet einen durch einen Zeilenumbruch abgeschlossenen Textstring an das Bedienfeld „Ausgabe“ und zeigt das Bedienfeld „Ausgabe“ an, sofern es nicht bereits sichtbar ist. Diese Methode ist mit `fl.trace()` identisch und funktioniert genau wie die Anweisung `trace()` in ActionScript.

Wenn Sie eine leere Zeile senden möchten, verwenden Sie `outputPanel.trace("")` oder `outputPanel.trace("\n")`. Sie können letzteren Befehl inline verwenden, indem Sie `\n` in den String *message* aufnehmen.

Beispiel

Im folgenden Beispiel werden mehrere Textzeilen im Bedienfeld „Ausgabe“ angezeigt:

```
fl.outputPanel.clear();  
fl.outputPanel.trace("Hello World!!!");  
var myPet = "cat";  
fl.outputPanel.trace("\nI have a " + myPet);  
fl.outputPanel.trace("");  
fl.outputPanel.trace("I love my " + myPet);  
fl.outputPanel.trace("Do you have a " + myPet + "?");
```

Kapitel 30: Oval-Objekt

Vererbung [Element-Objekt](#) > [Shape-Objekt](#) > Oval-Objekt

Verfügbarkeit

Flash CS3 Professional.

Beschreibung

Das Oval-Objekt ist eine Form, die mit dem Werkzeug für Ellipsengrundform gezeichnet wird. Um festzustellen, ob ein Element ein Oval-Objekt ist, verwenden Sie [shape.isOvalObject](#).

Übersicht über Eigenschaften

Abgesehen von den Eigenschaften für das [Shape-Objekt](#) stehen für das Oval-Objekt die folgenden Eigenschaften zur Auswahl. Um die Eigenschaften für ein Oval-Objekt festzulegen, verwenden Sie [document.setOvalObjectProperty\(\)](#).

Eigenschaft	Beschreibung
OvalObject.closePath	Schreibgeschützt; ein boolescher Wert, der festlegt, ob das Kontrollkästchen „Pfad schließen“ im Eigenschafteninspektor ausgewählt ist.
OvalObject.endAngle	Schreibgeschützt; ein Gleitkommawert, der den Endwinkel des Oval-Objekts festlegt.
OvalObject.innerRadius	Schreibgeschützt; ein Gleitkommawert, der den inneren Radius des Oval-Objekts als Prozentwert festlegt.
OvalObject.startAngle	Schreibgeschützt; ein Gleitkommawert, der den Anfangswinkel des Oval-Objekts festlegt.

OvalObject.closePath

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`OvalObject.closePath`

Beschreibung

Schreibgeschützte Eigenschaft; ein boolescher Wert, der festlegt, ob das Kontrollkästchen „Pfad schließen“ im Eigenschafteninspektor ausgewählt ist. Wenn die Werte für den Anfangswinkel und den Endwinkel des Objekts identisch sind, hat das Festlegen dieser Eigenschaft keine Auswirkungen, bis die Werte geändert werden.

Um diesen Wert einzustellen, verwenden Sie [document.setOvalObjectProperty\(\)](#).

Beispiel

Im folgenden Beispiel wird die Auswahl der `OvalObject.closePath`-Eigenschaft aufgehoben:

```
fl.getDocumentDOM().setOvalObjectProperty("closePath", false);
```

Siehe auch

`document.setOvalObjectProperty()`, `shape.isOvalObject`

OvalObject.endAngle

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`OvalObject.endAngle`

Beschreibung

Schreibgeschützte Eigenschaft; ein Gleitkommawert, der den Endwinkel des Oval-Objekts festlegt. Es sind Werte von 0 bis 360 zulässig.

Um diesen Wert einzustellen, verwenden Sie `document.setOvalObjectProperty()`.

Beispiel

Im folgenden Beispiel wird der Endwinkel des ausgewählten Oval-Objekts auf 270 festgelegt:

```
fl.getDocumentDOM().setOvalObjectProperty("endAngle",270);
```

Siehe auch

`document.setOvalObjectProperty()`, `OvalObject.startAngle`, `shape.isOvalObject`

OvalObject.innerRadius

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`OvalObject.innerRadius`

Beschreibung

Schreibgeschützte Eigenschaft; ein Gleitkommawert, der den inneren Radius des Oval-Objekts als Prozentwert festlegt. Es sind Werte von 0 bis 99 zulässig.

Um diesen Wert einzustellen, verwenden Sie `document.setOvalObjectProperty()`.

Beispiel

Im folgenden Beispiel wird der innere Radius des ausgewählten Oval-Objekts auf 50 Prozent festgelegt:

```
fl.getDocumentDOM().setOvalObjectProperty("innerRadius",50);
```

Siehe auch

`document.setOvalObjectProperty()`, `shape.isOvalObject`

OvalObject.startAngle

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`OvalObject.startAngle`

Beschreibung

Schreibgeschützt Eigenschaft; ein Gleitkommawert, der den Anfangswinkel des Oval-Objekts festlegt. Es sind Werte von 0 bis 360 zulässig.

Um diesen Wert einzustellen, verwenden Sie `document.setOvalObjectProperty()`.

Beispiel

Im folgenden Beispiel wird der Anfangswinkel des ausgewählten Oval-Objekts auf 270 festgelegt:

```
fl.getDocumentDOM().setOvalObjectProperty("startAngle", 270);
```

Siehe auch

`document.setOvalObjectProperty()`, `OvalObject.endAngle`, `shape.isOvalObject`

Kapitel 31: Parameter-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Der Zugriff auf den Parameter-Objekttyp erfolgt vom `componentInstance.parameters`-Array aus (dies entspricht dem Eigenschafteninspektor für Komponenten im Authoring-Tool).

Übersicht über Methoden

Für das Parameter-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>parameter.insertItem()</code>	Fügt ein Element in eine Liste, ein Objekt oder ein Array ein.
<code>parameter.removeItem()</code>	Entfernt ein Element des Typs Liste, Objekt oder Array aus einem Bildschirm- oder Komponentenparameter.

Übersicht über Eigenschaften

Für das Parameter-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>parameter.category</code>	Ein String, der die Eigenschaft <code>category</code> für den Parameter <code>screen</code> oder <code>componentInstance</code> angibt.
<code>parameter.listIndex</code>	Eine Ganzzahl, die den Wert des ausgewählten Listenelements angibt.
<code>parameter.name</code>	Schreibgeschützt; ein String, der den Namen des Parameters angibt.
<code>parameter.value</code>	Entspricht dem Feld „Wert“ auf der Registerkarte „Parameter“ des Komponenteninspektors, auf der Registerkarte „Parameter“ des Eigenschafteninspektors oder im Eigenschafteninspektor für Bildschirme.
<code>parameter.valueType</code>	Schreibgeschützt; ein String, der den Typ des Bildschirm- oder Komponentenparameters angibt.
<code>parameter.verbose</code>	Gibt an, wo der Parameter angezeigt wird.

parameter.category

Verfügbarkeit

Flash MX 2004.

Verwendung

`parameter.category`

Beschreibung

Eigenschaft; ein String, der die Eigenschaft `category` für den Parameter `screen` oder `componentInstance` angibt. Diese Eigenschaft stellt eine alternative Möglichkeit zur Anzeige einer Parameterliste dar. Diese Funktionalität ist nicht in der Flash-Benutzeroberfläche verfügbar.

parameter.insertItem()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
parameter.insertItem(index, name, value, type)
```

Parameter

index Ein auf Null basierender Ganzzahlindex, der angibt, wo das Element in der Liste, in dem Objekt oder in dem Array eingefügt wird. Wenn der Indexwert 0 ist, wird das Element am Anfang der Liste eingefügt. Ist der Indexwert höher als die Listengröße, wird das neue Element am Ende des Arrays eingefügt.

name Ein String, der den Namen des zu einzufügenden Elements angibt. Dieser Parameter ist für Objektparameter erforderlich.

value Ein String, der den Wert des zu einzufügenden Elements angibt.

type Ein String, der den einzufügenden Elementtyp angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt ein Element in eine Liste, ein Objekt oder ein Array ein. Wenn es sich bei einem Parameter um eine Liste, ein Objekt oder ein Array handelt, ist die Eigenschaft *value* ein Array.

Beispiel

Im folgenden Beispiel wird der Wert für `New Value` in den Parameter `labelPlacement` eingefügt:

```
// Select an instance of a Button component on the Stage.
var parms = fl.getDocumentDOM().selection[0].parameters;
parms[2].insertItem(0, "name", "New Value", "String");
var values = parms[2].value;
for(var prop in values){
    fl.trace("labelPlacement parameter value = " + values[prop].value);
}
```

parameter.listIndex

Verfügbarkeit

Flash MX 2004.

Verwendung

`parameter.listIndex`

Beschreibung

Eigenschaft; der Wert des ausgewählten Listenelements. Diese Eigenschaft ist nur gültig, wenn der `parameter.valueType` den Wert "List" aufweist.

Beispiel

Im folgenden Beispiel wird der erste Parameter (`autoKeyNav`) einer Folie festgelegt. Um dem Parameter einen seiner zulässigen Werte zuzuweisen (`true`, `false` oder `inherit`), wird `parameter.listIndex` auf die Indexposition des Elements in der Liste eingestellt (0 steht für `true`, 1 für `false`, 2 für `inherit`).

```
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters;  
parms[0].listIndex = 1;
```

parameter.name

Verfügbarkeit

Flash MX 2004.

Verwendung

`parameter.name`

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der den Namen des Parameters angibt.

Beispiel

Im folgenden Beispiel wird der Name des fünften Parameters für die ausgewählte Komponente angezeigt:

```
var parms = fl.getDocumentDOM().selection[0].parameters;  
fl.trace("name: " + parms[4].name);
```

Im folgenden Beispiel wird der Name des fünften Parameters für den angegebenen Bildschirm angezeigt:

```
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters; fl.trace("name: " +  
parms[4].name);
```

parameter.removeItem()

Verfügbarkeit

Flash MX 2004.

Verwendung

`parameter.removeItem(index)`

Parameter

index Der auf Null basierende Ganzzahlindex des Elements, das aus der Bildschirm- oder Komponenteneigenschaft entfernt werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; entfernt ein Element des Typs Liste, Objekt oder Array aus einem Bildschirm- oder Komponentenparameter.

Beispiel

Im folgenden Beispiel wird das Element an Indexposition 1 aus dem Parameter `labelPlacement` einer Komponente entfernt:

```
// Select an instance of a Button component on the Stage.
var parms = fl.getDocumentDOM().selection[0].parameters;
var values = parms[2].value;
fl.trace("--Original--");
for(var prop in values){
    fl.trace("labelPlacement value = " + values[prop].value);
}
parms[2].removeItem(1);

var newValues = parms[2].value;
fl.trace("--After Removing Item--");
for(var prop in newValues){
    fl.trace("labelPlacement value = " + newValues[prop].value);
}
```

Im folgenden Beispiel wird das Element an Indexposition 1 aus dem Parameter `autoKeyNav` eines Bildschirms entfernt:

```
// Open a presentation document.
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters;
var values = parms[0].value;
fl.trace("--Original--");
for(var prop in values){
    fl.trace("autoKeyNav value = " + values[prop].value);
}
parms[0].removeItem(1);

var newValues = parms[0].value;
fl.trace("--After Removing Item--");
for(var prop in newValues){
    fl.trace("autoKeyNav value = " + newValues[prop].value);
}
```

parameter.value

Verfügbarkeit

Flash MX 2004.

Verwendung

`parameter.value`

Beschreibung

Eigenschaft; entspricht dem Feld „Wert“ auf der Registerkarte „Parameter“ des Komponenteninspektors, auf der Registerkarte „Parameter“ des Eigenschafteninspektors oder im Eigenschafteninspektor für Bildschirme. Der Typ der Eigenschaft `value` hängt von der Eigenschaft `valueType` des Parameters ab (siehe [parameter.valueType](#)).

parameter.valueType

Verfügbarkeit

Flash MX 2004.

Verwendung

`parameter.valueType`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die den Typ des Bildschirm- oder Komponentenparameters angibt. Dieser Typ kann einen der folgenden Werte enthalten: "Default", "Array", "Object", "List", "String", "Number", "Boolean", "Font Name", "Color", "Collection", "Web Service URL" oder "Web Service Operation".

Siehe auch

[parameter.value](#)

parameter.verbose

Verfügbarkeit

Flash MX 2004.

Verwendung

`parameter.verbose`

Beschreibung

Eigenschaft; gibt an, wo der Parameter angezeigt wird. Falls der Wert dieser Eigenschaft 0 lautet (ohne Anzeige), wird der Parameter nur im Komponenteninspektor angezeigt. Bei 1 (mit Anzeige) wird der Parameter im Komponenteninspektor und auf der Registerkarte „Parameter“ des Eigenschafteninspektors angezeigt.

Kapitel 32: Path-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Path-Objekt definiert eine Sequenz von Liniensegmenten (gerade, gebogen oder beides), die in der Regel beim Erstellen erweiterbarer Werkzeuge eingesetzt werden. Im folgenden Beispiel wird eine Instanz eines Path-Objekts von dem flash-Objekt zurückgegeben:

```
path = fl.drawingLayer.newPath();
```

Siehe auch das Beispiel zu [drawingLayer-Objekt](#).

Übersicht über Methoden

Für das Path-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>path.addCubicCurve()</code>	Fügt dem Pfad ein kubisches Bézierkurvensegment hinzu.
<code>path.addCurve()</code>	Fügt dem Pfad ein quadratisches Bézierkurvensegment hinzu.
<code>path.addPoint()</code>	Fügt dem Pfad einen Punkt hinzu.
<code>path.clear()</code>	Entfernt alle Punkte aus dem Pfad.
<code>path.close()</code>	Fügt einen Punkt an der Stelle des Pfads hinzu, an der sich der erste Pfadpunkt befindet. Erweitert dann den Pfad bis zu diesem Punkt, wodurch der Pfad geschlossen wird.
<code>path.makeShape()</code>	Erstellt anhand der aktuellen Strich- und Füllungseinstellungen eine Form auf der Bühne.
<code>path.newContour()</code>	Beginnt in dem Pfad eine neue Kontur.

Übersicht über Eigenschaften

Für das Path-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>path.nPts</code>	Schreibgeschützt; eine Ganzzahl, die angibt, wie viele Punkte der Pfad enthält.

path.addCubicCurve()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
path.addCubicCurve(xAnchor, yAnchor, x2, y2, x3, y3, x4, y4)
```

Parameter

xAnchor Ein Gleitkommawert, der die *x*-Koordinate des ersten Steuerpunkts angibt.

yAnchor Ein Gleitkommawert, der die *y*-Koordinate des ersten Steuerpunkts angibt.

x2 Ein Gleitkommawert, der die *x*-Koordinate des zweiten Steuerpunkts angibt.

y2 Ein Gleitkommawert, der die *y*-Koordinate des zweiten Steuerpunkts angibt.

x3 Ein Gleitkommawert, der die *x*-Koordinate des dritten Steuerpunkts angibt.

y3 Ein Gleitkommawert, der die *y*-Koordinate des dritten Steuerpunkts angibt.

x4 Ein Gleitkommawert, der die *x*-Koordinate des vierten Steuerpunkts angibt.

y4 Ein Gleitkommawert, der die *y*-Koordinate des vierten Steuerpunkts angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt dem Pfad ein kubisches Bézierkurvensegment hinzu.

Beispiel

Im folgenden Beispiel wird ein neuer Pfad erstellt und in der Variablen `myPath` gespeichert. Anschließend wird die Kurve dem Pfad zugewiesen:

```
var myPath = fl.drawingLayer.newPath();  
myPath.addCubicCurve(0, 0, 10, 20, 20, 20, 30, 0);
```

path.addCurve()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
path.addCurve(xAnchor, yAnchor, x2, y2, x3, y3)
```

Parameter

xAnchor Ein Gleitkommawert, der die *x*-Koordinate des ersten Steuerpunkts angibt.

yAnchor Ein Gleitkommawert, der die *y*-Koordinate des ersten Steuerpunkts angibt.

x2 Ein Gleitkommawert, der die *x*-Koordinate des zweiten Steuerpunkts angibt.

y2 Ein Gleitkommawert, der die *y*-Koordinate des zweiten Steuerpunkts angibt.

x3 Ein Gleitkommawert, der die *x*-Koordinate des dritten Steuerpunkts angibt.

y3 Ein Gleitkommawert, der die *y*-Koordinate des dritten Steuerpunkts angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt dem Pfad ein quadratisches Bézierkurvensegment hinzu.

Beispiel

Im folgenden Beispiel wird ein neuer Pfad erstellt und in der Variablen `myPath` gespeichert. Anschließend wird die Kurve dem Pfad zugewiesen:

```
var myPath = fl.drawingLayer.newPath();  
myPath.addCurve(0, 0, 10, 20, 20, 0);
```

path.addPoint()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
path.addPoint(x, y)
```

Parameter

x Ein Gleitkommawert, der die *x*-Koordinate des Punkts angibt.

y Ein Gleitkommawert, der die *y*-Koordinate des Punkts angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt dem Pfad einen Punkt hinzu.

Beispiel

Im folgenden Beispiel wird ein neuer Pfad erstellt und in der Variablen `myPath` gespeichert. Anschließend wird der neue Punkt dem Pfad zugewiesen:

```
var myPath = fl.drawingLayer.newPath();  
myPath.addPoint(10, 100);
```

path.clear()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
path.clear()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; entfernt alle Punkte aus dem Pfad.

Beispiel

Im folgenden Beispiel werden alle Punkte aus einem in der Variablen `myPath` gespeicherten Pfad entfernt:

```
var myPath = fl.drawingLayer.newPath();  
myPath.clear();
```

path.close()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
path.close()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt einen Punkt an der Stelle des Pfads hinzu, an der sich der erste Pfadpunkt befindet. Erweitert dann den Pfad bis zu diesem Punkt, wodurch der Pfad geschlossen wird. Wenn der Pfad keine Punkte hat, werden keine Punkte hinzugefügt.

Beispiel

Im folgenden Beispiel wird ein geschlossener Pfad erstellt:

```
var myPath = fl.drawingLayer.newPath();  
myPath.close();
```

path.makeShape()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
path.makeShape([bSupressFill [, bSupressStroke]])
```

Parameter

bSupressFill Ein boolescher Wert; bei dem Wert `true` wird die Füllung unterdrückt, die ansonsten auf die Form angewendet würde. Der Standardwert lautet `false`. Dieser Parameter ist optional.

bSuppressStroke Ein boolescher Wert; bei dem Wert `true` wird der Strich unterdrückt, der ansonsten auf die Form angewendet würde. Der Standardwert lautet `false`. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; erstellt anhand der aktuellen Strich- und Füllungseinstellungen eine Form auf der Bühne. Der Pfad wird gelöscht, nachdem die Form erstellt wurde. Diese Methode hat zwei optionale Parameter, mit denen die Füllung und der Strich des resultierenden Formobjekts unterdrückt werden können. Wenn Sie diese Parameter weglassen oder ihnen den Wert `false` zuweisen, werden die aktuellen Füllungs- und Strichwerte verwendet.

Beispiel

Im folgenden Beispiel wird eine Form mit der aktuellen Füllung, aber ohne Strich erstellt:

```
var myPath = fl.drawingLayer.newPath();  
myPath.makeShape(false, true);
```

path.newContour()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
path.newContour()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; beginnt in dem Pfad eine neue Kontur.

Beispiel

Im folgenden Beispiel wird ein leeres Quadrat erstellt:

```
var myPath = fl.drawingLayer.newPath();
myPath.addPoint(0, 0);
myPath.addPoint(0, 30);
myPath.addPoint(30, 30);
myPath.addPoint(30, 0);
myPath.addPoint(0, 0);

myPath.newContour();
myPath.addPoint(10, 10);
myPath.addPoint(10, 20);
myPath.addPoint(20, 20);
myPath.addPoint(20, 10);
myPath.addPoint(10, 10);

myPath.makeShape();
```

path.nPts

Verfügbarkeit

Flash MX 2004.

Verwendung

path.nPts

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die angibt, wie viele Punkte der Pfad enthält. Ein neuer Pfad hat 0 Punkte.

Beispiel

Im folgenden Beispiel wird im Bedienfeld „Ausgabe“ angegeben, auf wie viele Punkte die Variable `myPath` in dem Pfad verweist:

```
var myPath = fl.drawingLayer.newPath();
var numOfPoints = myPath.nPts;
fl.trace("Number of points in the path: " + numOfPoints);
// Displays: Number of points in the path: 0
```

Kapitel 33: presetItem-Objekt

Verfügbarkeit

Flash CS4 Professional.

Beschreibung

Das presetItem-Objekt repräsentiert ein Element (Voreinstellung oder Ordner) im Bedienfeld „Bewegungsvoreinstellungen“ („Fenster“ > „Bewegungsvoreinstellungen“). Das Array von presetItem-Objekten ist eine Eigenschaft des presetPanel-Objekts ([presetPanel.items](#)).

Alle Eigenschaften des presetItem-Objekts sind schreibgeschützt. Um Aufgaben wie das Löschen, Umbenennen oder Verschieben von Elementen auszuführen, verwenden Sie die Methoden des [presetPanel-Objekt](#).

Übersicht über Eigenschaften

Sie können die folgenden Eigenschaften mit dem presetItem-Objekt verwenden:

Eigenschaft	Beschreibung
presetItem.isDefault	Gibt an, ob das Element zusammen mit Flash installiert wird oder ob es sich um ein benutzerdefiniertes Element handelt, das von Ihnen oder einer anderen Person installiert wurde.
presetItem.isFolder	Gibt an, ob es sich bei dem Element im Bedienfeld „Bewegungsvoreinstellungen“ um einen Ordner oder um eine Voreinstellung handelt.
presetItem.level	Die Ebene des Elements in der Ordnerstruktur des Bedienfelds „Bewegungsvoreinstellungen“.
presetItem.name	Der Name der Voreinstellung oder des Ordners ohne Pfadinformationen.
presetItem.open	Gibt an, ob ein Ordner im Bedienfeld „Bewegungsvoreinstellungen“ zurzeit erweitert (eingebildet) ist.
presetItem.path	Der Pfad zum Element in der Ordnerstruktur des Bedienfelds „Bewegungsvoreinstellungen“ und der Name des Elements.

presetItem.isDefault

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`presetItem.isDefault`

Beschreibung

Schreibgeschützte Eigenschaft; ein boolescher Wert, der angibt, ob das Element zusammen mit Flash installiert wird (`true`) oder ob es sich um ein benutzerdefiniertes Element handelt, dass von Ihnen oder einem anderen Benutzer erstellt wurde (`false`). Lautet dieser Wert `true`, kann es als „schreibgeschütztes“ Element betrachtet werden; es kann weder verschoben oder gelöscht noch können andere ähnliche Vorgänge darauf angewendet werden.

Beispiel

Im folgenden Beispiel wird der Inhalt des Bedienfelds „Bewegungsvoreinstellungen“ angezeigt und es wird angegeben, ob ein Element zusammen mit Flash installiert wird:

```
fl.outputPanel.clear();
var presetItemArray=fl.presetPanel.items;
for (i=0;i<presetItemArray.length; i++){
    var presetItem = presetItemArray[i];
    fl.trace(presetItem.name +", default =" + presetItem.isDefault);
}
```

presetItem.isFolder

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`presetItem.isFolder`

Beschreibung

Schreibgeschützte Eigenschaft; ein boolescher Wert, der angibt, ob es sich bei dem Element im Bedienfeld „Bewegungsvoreinstellungen“ um einen Ordner (`true`) oder eine Voreinstellung (`false`) handelt.

Beispiel

Im folgenden Beispiel wird gezeigt, dass es sich beim ersten Element im Bedienfeld „Bewegungsvoreinstellungen“ um einen Ordner und beim zweiten um eine Voreinstellung handelt:

```
var presetItemArray=fl.presetPanel.items;
fl.trace(presetItemArray[0].isFolder);
fl.trace(presetItemArray[1].isFolder);
```

presetItem.level

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`presetItem.level`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die die Ebene des Elements in der Ordnerstruktur im Bedienfeld „Bewegungsvoreinstellungen“ angibt. Die Ordner „Default Folder“ und „Custom Presets“ befinden sich auf Ebene 0.

Beispiel

Im folgenden Beispiel wird gezeigt, dass sich das erste Element im Bedienfeld „Bewegungsvoreinstellungen“ auf Ebene 0 und das zweite auf Ebene 1 befindet:

```
var presetItemArray=fl.presetPanel.items;  
fl.trace(presetItemArray[0].level);  
fl.trace(presetItemArray[1].level);
```

presetItem.name

Verfügbarkeit

Flash CS4 Professional.

Verwendung

presetItem.name

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der den Voreinstellungs- oder Ordnernamen ohne Pfadangaben angibt.

Beispiel

Siehe [presetItem.path](#).

presetItem.open

Verfügbarkeit

Flash CS4 Professional.

Verwendung

presetItem.open

Beschreibung

Schreibgeschützte Eigenschaft; gibt an, ob ein Ordner im Bedienfeld „Bewegungsvoreinstellungen“ erweitert ist (true) oder nicht (false).

Der Wert dieser Eigenschaft lautet true, wenn das Element kein Ordner ist. Um festzustellen, ob ein Element ein Ordner oder eine Voreinstellung ist, verwenden Sie [presetItem.isFolder](#).

Beispiel

Im folgenden Beispiel wird angezeigt, ob Ordner im Bedienfeld „Bewegungsvoreinstellungen“ erweitert oder reduziert sind:

```
fl.outputPanel.clear();  
var presetItemArray=fl.presetPanel.items;  
for (i=0;i<presetItemArray.length; i++){  
    var presetItem = presetItemArray[i];  
    if (presetItem.isFolder) {  
        var status = presetItem.open ? "Open" : "Closed"  
        fl.trace(presetItem.level + "-" + presetItem.name + " folder is " + status);  
    }  
}
```

presetItem.path

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`presetItem.path`

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der den Pfad des Elements in der Ordnerstruktur des Bedienfelds „Bewegungsvoreinstellungen“ und den Elementnamen angibt.

Beispiel

Im folgenden Beispiel werden die unterschiedlichen Werte von `presetItem.name` und `presetItem.path` veranschaulicht.

```
fl.outputPanel.clear();
var presetItemArray=fl.presetPanel.items;
for (i=0;i<presetItemArray.length; i++){
    var presetItem = presetItemArray[i];
    fl.trace("Name: " + presetItem.name + "\n" + "Path: " + presetItem.path);
    fl.trace("");
}
```

Kapitel 34: presetPanel-Objekt

Verfügbarkeit

Flash CS4 Professional.

Beschreibung

Das presetPanel-Objekt repräsentiert das Bedienfeld „Bewegungsvoreinstellungen“ („Fenster“ > „Bewegungsvoreinstellungen“). Es ist eine Eigenschaft des flash-Objekts ([fl.presetPanel](#)).

Übersicht über Methoden

Mit dem presetPanel-Objekt können Sie die folgenden Methoden verwenden:

Methode	Beschreibung
presetPanel.addItem()	Wenn auf der Bühne ein einzelnes Bewegungs-Tween ausgewählt ist, wird diese Bewegung dem Bedienfeld „Bewegungsvoreinstellungen“ hinzugefügt.
presetPanel.applyPreset()	Wendet die angegebene oder ausgewählte Voreinstellung auf das auf der Bühne ausgewählte Element an.
presetPanel.deleteFolder()	Löscht den angegebenen Ordner und dessen Unterordner aus der Ordnerstruktur im Bedienfeld „Bewegungsvoreinstellungen“.
presetPanel.deleteItem()	Löscht die angegebene Voreinstellung aus dem Bedienfeld „Bewegungsvoreinstellungen“.
presetPanel.expandFolder()	Erweitert oder reduziert den oder die im Bedienfeld „Bewegungsvoreinstellungen“ gegenwärtig ausgewählten Ordner.
presetPanel.exportItem()	Exportiert die gegenwärtig ausgewählte oder angegebene Voreinstellung in eine XML-Datei.
presetPanel.findItemIndex()	Gibt eine Ganzzahl zurück, die die Indexposition eines Elements im Bedienfeld „Bewegungsvoreinstellungen“ angibt.
presetPanel.getSelectedItems()	Gibt einen Array von presetItem-Objekten zurück, die den gegenwärtig im Bedienfeld „Bewegungsvoreinstellungen“ ausgewählten Elementen entsprechen.
presetPanel.importItem()	Fügt anhand einer angegebenen XML-Datei eine Voreinstellung im Bedienfeld „Bewegungsvoreinstellungen“ hinzu.
presetPanel.moveToFolder()	Verschiebt das angegebene Element in den angegebenen Ordner.
presetPanel.newFolder()	Erstellt einen Ordner in der Ordnerstruktur des Bedienfelds „Bewegungsvoreinstellungen“.
presetPanel.renameItem()	Ändert den Namen der gegenwärtig ausgewählten Voreinstellung oder des Ordners in den angegebenen Namen.
presetPanel.selectItem()	Wählt ein Element im Bedienfeld „Bewegungsvoreinstellungen“ aus bzw. hebt dessen Auswahl auf.

Übersicht über Eigenschaften

Sie können die folgende Eigenschaft mit dem presetPanel-Objekt verwenden:

Eigenschaft	Beschreibung
<code>presetPanel.items</code>	Ein Array von <code>presetItem</code> -Objekten im Bedienfeld „Bewegungsvoreinstellungen“.

presetPanel.addNewItem()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
fl.presetPanel.addNewItem( [namePath] );
```

Parameter

namePath Ein String, der den Pfad und den Namen des Elements angibt, das dem Bedienfeld „Bewegungsvoreinstellungen“ hinzugefügt werden soll. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn das Element erfolgreich hinzugefügt wurde; andernfalls `false`.

Beschreibung

Methode; wenn momentan ein Bewegungs-Tween auf der Bühne ausgewählt ist, wird diese Bewegung dem Bedienfeld „Bewegungsvoreinstellungen“ im angegebenen Ordner unter dem angegebenen Namen hinzugefügt. Der in *namePath* angegebene Pfad muss im Bedienfeld vorhanden sein.

Wenn bereits eine mit *namePath* übereinstimmende Voreinstellung vorhanden ist, hat diese Methode keine Auswirkungen und gibt `false` zurück.

Wenn Sie keinen Wert für *namePath* übergeben, wird das Element zum Ordner „Custom Presets“ unter dem Namen „Custom preset *n*“ hinzugefügt, wobei *n* mit jedem auf diese Weise hinzugefügten Element schrittweise zunimmt.

Beispiel

Sofern ein Bewegungs-Tween auf der Bühne ausgewählt ist, wird im folgenden Beispiel eine Voreinstellung namens Bouncing Ball zum Ordner „Custom Presets“ hinzugefügt:

```
fl.presetPanel.addNewItem("Custom Presets/Bouncing Ball");
```

Siehe auch

[presetPanel.newFolder\(\)](#)

presetPanel.applyPreset()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
presetPanel.applyPreset( [presetPath] )
```

Parameter

presetPath Ein String, der den vollständigen Pfad und Namen der anzuwendenden Voreinstellung angibt, so wie sie im Bedienfeld „Bewegungsvoreinstellungen“ angezeigt werden. Dieser Parameter ist optional; wenn kein Wert übergeben wird, wird die aktuell ausgewählte Voreinstellung angewendet.

Rückgabewerte

Der boolescher Wert `true`, wenn die Voreinstellung erfolgreich angewendet wurde; andernfalls `false`.

Beschreibung

Methode; wendet die angegebene oder ausgewählte Voreinstellung auf das auf der Bühne ausgewählte Element an. Bei dem Element muss es sich um einen Bewegungs-Tween, ein Symbol oder ein Element, das in ein Symbol konvertiert werden kann, handeln. Ist das Element ein Bewegungs-Tween, wird die aktuelle Bewegung ohne Anforderung einer Benutzerbestätigung durch die ausgewählte Voreinstellung ersetzt.

Diese Methode schlägt in den folgenden Situationen fehl:

- Der unter *presetPath* angegebene Pfad existiert nicht.
- Sie übergeben keinen Wert für *presetPath* und Sie haben keine Voreinstellung ausgewählt.
- Sie übergeben keinen Wert für *presetPath* und Sie haben mehrere Voreinstellungen ausgewählt.
- Das auf der Bühne ausgewählte Element ist kein Symbol und es kann nicht in ein Symbol konvertiert werden.

Beispiel

Im folgenden Beispiel wird die Voreinstellung `aDribble` auf das auf der Bühne ausgewählte Element angewendet.

```
var result = fl.presetPanel.applyPreset("Custom Presets/Bounces/aDribble");  
fl.trace(result);
```

presetPanel.deleteFolder()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
presetPanel.deleteFolder( [folderPath] )
```

Parameter

folderPath Ein String, der den aus dem Bedienfeld „Bewegungsvoreinstellungen“ zu löschenden Ordner angibt. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn der oder die Ordner erfolgreich gelöscht wurden, ansonsten `false`.

Beschreibung

Methode; löscht den angegebenen Ordner und dessen Unterordner aus der Ordnerstruktur im Bedienfeld „Bewegungsvoreinstellungen“. Alle in den Ordnern enthaltenen Voreinstellungen werden ebenfalls gelöscht. Ordner im Ordner „Default Presets“ können nicht gelöscht werden.

Wenn Sie für *folderPath* keinen Wert übergeben, werden alle gegenwärtig ausgewählten Ordner gelöscht.

Hinweis: Ordner werden ohne Anforderung einer Benutzerbestätigung gelöscht und diese Aktion kann nicht rückgängig gemacht werden.

Beispiel

Im folgenden Beispiel wird im Ordner „Custom Presets“ ein Ordner namens `Bouncing` gelöscht; alle in `Bouncing` enthaltenen Ordner werden ebenfalls gelöscht:

```
fl.presetPanel.deleteFolder("Custom Presets/Bouncing");
```

Siehe auch

[presetPanel.deleteItem\(\)](#)

presetPanel.deleteItem()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
presetPanel.deleteItem( [namePath] )
```

Parameter

namePath Ein String, der den Pfad und den Namen des Elements angibt, das aus dem Bedienfeld „Bewegungsvoreinstellungen“ gelöscht werden soll. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn das Element oder die Elemente erfolgreich gelöscht wurden, ansonsten `false`.

Beschreibung

Methode; löscht die angegebene Voreinstellung aus dem Bedienfeld „Bewegungsvoreinstellungen“. Wenn Sie für *namePath* keinen Wert übergeben, werden alle gegenwärtig ausgewählten Voreinstellungen gelöscht. Elemente im Ordner „Default Presets“ können nicht gelöscht werden.

Hinweis: Elemente werden ohne Anforderung einer Benutzerbestätigung gelöscht und diese Aktion kann nicht rückgängig gemacht werden.

Beispiel

Im folgenden Beispiel wird eine Voreinstellung namens `aDribble` aus dem Ordner „Custom Presets“ gelöscht:

```
fl.presetPanel.deleteItem("Custom Presets/aDribble");
```

Siehe auch

[presetPanel.deleteFolder\(\)](#)

presetPanel.expandFolder()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
presetPanel.expandFolder( [bExpand [, bRecurse [, folderPath] ] ] )
```

Parameter

bExpand Ein boolescher Wert, der angibt, ob der Ordner erweitert (`true`) oder reduziert (`false`) werden soll. Dieser Parameter ist optional, sein Standardwert lautet `true`.

bRecurse Ein boolescher Wert, der angibt, ob die im Ordner enthaltenen Ordner erweitert (`true`) oder reduziert (`false`) werden sollen. Dieser Parameter ist optional, sein Standardwert lautet `false`.

folderPath Ein String, der den Pfad des Ordners angibt, der erweitert oder reduziert werden soll. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn der oder die Ordner erfolgreich erweitert oder reduziert wurden, ansonsten `false`.

Beschreibung

Methode; erweitert oder reduziert den oder die im Bedienfeld „Bewegungsvoreinstellungen“ gegenwärtig ausgewählten Ordner. Um andere als die momentan ausgewählten Ordner zu erweitern oder zu reduzieren, müssen Sie einen Wert für `folderPath` übergeben.

Beispiel

Im folgenden Beispiel wird der Ordner „Custom Presets“ erweitert, aber nicht dessen Unterordner:

```
fl.presetPanel.expandFolder(true, false, "Custom Presets");
```

Im folgenden Beispiel werden der Ordner „Custom Presets“ und alle seine Unterordner erweitert:

```
fl.presetPanel.expandFolder(true, true, "Custom Presets");
```

presetPanel.exportItem()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
presetPanel.exportItem(fileURI [, namePath] )
```

Parameter

fileURI Ein String im Format `file:///` URI, der den Pfad und optional einen Dateinamen der exportierten Datei angibt. Weitere Informationen finden Sie unter „Beschreibung“ weiter unten.

namePath Ein String, der den Pfad und den Namen des Elements angibt, das im Bedienfeld „Bewegungsvoreinstellungen“ ausgewählt werden soll. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die Voreinstellung erfolgreich exportiert wurde; andernfalls `false`.

Beschreibung

Methode; exportiert die gegenwärtig ausgewählte oder angegebene Voreinstellung in eine XML-Datei. Nur Voreinstellungen können exportiert werden; beim Versuch, einen Ordner zu exportieren, schlägt die Methode fehl. Diese Methode schlägt auch fehl, wenn versucht wird, eine Datei auf der Festplatte zu überschreiben.

Wenn Sie keinen Dateinamen als Teil von *fileURI* angeben (d. h. wenn es sich beim letzten Zeichen von *fileURI* um einen Schrägstrich (/) handelt), wird die exportierte Datei unter demselben Namen wie die exportierte Voreinstellung gespeichert. Wenn Sie für *namePath* keinen Wert angeben, wird die gegenwärtig ausgewählte Voreinstellung exportiert. Siehe dazu auch folgendes Beispiel.

Beispiel

Das folgende Beispiel zeigt, welche Dateien bei der Übergabe verschiedener Parameter an diese Methode erstellt werden und zeigt an, ob die angegebene Datei erfolgreich erstellt wurde. Wählen Sie vor Ausführung dieses Beispiels im Ordner „Default Presets“ die Voreinstellung „Hereinfliegen-links“ und erstellen Sie auf der Festplatte den Ordner My Presets.

```
//Exports fly-in-left to C:\My Presets\fly-in-left.xml
fl.presetPanel.exportItem("file:///C:/My Presets/");
//Exports fly-in-left to C:\My Presets\myFavoritePreset.xml
fl.presetPanel.exportItem("file:///C:/My Presets/myFavoritePreset.xml");
// Exports the "pulse" preset to C:\My Presets\pulse.xml
fl.presetPanel.exportItem("file:///C:/My Presets/", "Default Presets/pulse");
// Exports the "pulse" preset to C:\My Presets\thePulsePreset.xml
fl.presetPanel.exportItem("file:///C:/My Presets/thePulsePreset.xml", "Default Presets/pulse");
```

Siehe auch

[presetPanel.importItem\(\)](#)

presetPanel.findItemIndex()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
presetPanel.findItemIndex([presetName])
```

Parameter

presetName Ein String, der den Namen der Voreinstellung angibt, für die der Indexwert zurückgegeben werden soll. Dieser Parameter ist optional.

Rückgabewerte

Eine Ganzzahl, die den Index der angegebenen Voreinstellung im `presetPanel.items`-Array angibt. Wenn Sie für `presetName` keinen Wert übergeben, wird der Index der gegenwärtig ausgewählten Voreinstellung zurückgegeben. Diese Methode gibt in den folgenden Situationen -1 zurück:

- Sie übergeben keinen Wert für `presetName` und Sie haben keine Voreinstellung ausgewählt.
- Sie übergeben keinen Wert für `presetName` und Sie haben mehrere Voreinstellungen ausgewählt.
- Sie übergeben einen Wert für `presetName`, der mit keinem Element im Bedienfeld übereinstimmt.

Beschreibung

Methode; gibt eine Ganzzahl zurück, die die Indexposition eines Elements im Bedienfeld „Bewegungsvoreinstellungen“ angibt.

Beispiel

Im folgenden Beispiel werden der Indexwert und der vollständige Pfadname der gegenwärtig ausgewählten Voreinstellung angezeigt:

```
// Select one preset in the Motions Preset panel before running this code
var selectedPreset = fl.presetPanel.findItemIndex();
fl.trace(selectedPreset);
fl.trace(fl.presetPanel.items[selectedPreset].path);
```

presetPanel.getSelectedItems()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
presetPanel.getSelectedItems()
```

Parameter

Keine.

Rückgabewerte

Ein Array von `presetItem`-Objekten.

Beschreibung

Methode; gibt einen Array von `presetItem`-Objekten zurück, die den gegenwärtig im Bedienfeld „Bewegungsvoreinstellungen“ ausgewählten Elementen entsprechen (siehe [presetItem-Objekt](#)). Jedes Element im Array stellt entweder einen Ordner oder eine Voreinstellung dar.

Beispiel

Im folgenden Beispiel werden die vollständigen Pfadnamen der gegenwärtig im Bedienfeld „Bewegungsvoreinstellungen“ ausgewählten Elemente angezeigt:

```
var itemArray = fl.presetPanel.getSelectedItems();  
var length = itemArray.length  
for (x=0; x<length; x++) {  
    fl.trace(itemArray[x].path);  
}
```

Siehe auch

[presetPanel.items](#)

presetPanel.importItem()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
presetPanel.importItem(fileURI [,namePath ])
```

Parameter

fileURI Ein String im Format file:/// URI, der die XML-Datei angibt, die als Voreinstellung in das Bedienfeld „Bewegungsvoreinstellungen“ importiert werden soll.

namePath Ein String, der angibt in welchem Ordner und unter welchem Namen die importierte Datei gespeichert werden soll. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; *true*, wenn die Datei erfolgreich importiert wurde; andernfalls *false*.

Beschreibung

Methode; fügt anhand einer angegebenen XML-Datei eine Voreinstellung im Bedienfeld „Bewegungsvoreinstellungen“ hinzu. Der in *namePath* angegebene Pfad muss im Bedienfeld vorhanden sein.

Um XML-Dateien zu erstellen, die importiert werden können, verwenden Sie [presetPanel.exportItem\(\)](#).

Wenn Sie keinen Wert für *namePath* übergeben, wird die importierte Voreinstellung im Ordner „Custom Presets“ gespeichert und erhält denselben Namen wie die importierte Datei (ohne die Erweiterung .XML).

Beispiel

Im folgenden Beispiel wird eine Voreinstellung in den Ordner „Custom Presets/Pulse“ importiert und dort unter dem Namen *fastPulse* gespeichert.

```
fl.presetPanel.importItem("file:///C:/My Presets/thePulsePreset.xml", "Custom  
Presets/Pulse/fastPulse");
```

Siehe auch

[presetPanel.exportItem\(\)](#)

presetPanel.items

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`presetPanel.items`

Beschreibung

Eigenschaft; ein Array von `presetItem`-Objekten im Bedienfeld „Bewegungsvoreinstellungen“ (siehe [presetItem-Objekt](#)). Jedes Element im Array stellt entweder einen Ordner oder eine Voreinstellung dar.

Beispiel

Im folgenden Beispiel werden die vollständigen Pfadnamen der Elemente im Bedienfeld „Bewegungsvoreinstellungen“ angezeigt:

```
var itemArray = fl.presetPanel.items;
var length = itemArray.length
for (x=0; x<length; x++) {
    fl.trace(itemArray[x].path);
}
```

Siehe auch

[presetPanel.getSelectedItems\(\)](#)

presetPanel.moveToFolder()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`presetPanel.moveToFolder(folderPath [, namePath])`

Parameter

folderPath Ein String, der den Pfad des Ordners im Bedienfeld „Bewegungsvoreinstellungen“ angibt, in den das Element bzw. die Elemente verschoben werden.

namePath Ein String, der den Pfad und Namen des zu verschiebenden Elements angibt. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die Elemente erfolgreich verschoben wurden, ansonsten `false`.

Beschreibung

Methode; verschiebt das angegebene Element in den angegebenen Ordner.

Wenn Sie einen leeren String ("") für *folderPath* übergeben, werden die Elemente in den Ordner „Custom Presets“ verschoben. Wenn Sie für *namePath* keinen Wert übergeben, werden die gegenwärtig ausgewählten Elemente verschoben.

Sie können keine Elemente in den oder aus dem Ordner „Default Presets“ verschieben.

Beispiel

Im folgenden Beispiel werden die gegenwärtig ausgewählten Elemente in den Ordner „Custom Presets/Bouncing“ verschoben und anschließend wird die Voreinstellung „Fast Bounce“ in denselben Ordner verschoben:

```
fl.presetPanel.moveToFolder("Custom Presets/Bouncing");  
fl.presetPanel.moveToFolder("Custom Presets/Bouncing" , "Custom Presets/Fast Bounce");
```

presetPanel.newFolder()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
presetPanel.newFolder( [folderPath] )
```

Parameter

folderPath Ein String, der angibt, wo im Bedienfeld „Bewegungsvoreinstellungen“ ein neuer Ordner hinzugefügt wird und wie der neue Ordner heißt. Dieser Parameter ist optional.

Rückgabewerte

Ein boolescher Wert; `true`, wenn der Ordner erfolgreich hinzugefügt wurde; andernfalls `false`.

Beschreibung

Methode; erstellt einen Ordner in der Ordnerstruktur des Bedienfelds „Bewegungsvoreinstellungen“. Mit dieser Methode können Sie nur eine neue Ordnerstufe erstellen. Wenn Sie also beispielsweise „Custom Presets/My First Folder/My Second Folder“ für *folderPath* übergeben, muss „Custom Presets/My First Folder“ bereits in der Ordnerstruktur vorhanden sein.

Wenn Sie keinen Wert für *folderPath* übergeben, wird auf der ersten Ebene unter dem Ordner „Custom preset“ ein Ordner namens „Untitled folder *n*“ erstellt, wobei *n* mit jedem auf diese Weise hinzugefügten Ordner schrittweise zunimmt.

Hinweis: Dem Ordner „Default Presets“ können keine Ordner hinzugefügt werden.

Beispiel

Im folgenden Beispiel wird dem Ordner „Custom Presets“ ein Ordner namens `Bouncing` hinzugefügt:

```
fl.presetPanel.newFolder("Custom Presets/Bouncing");
```

Siehe auch

[presetPanel.addItem\(\)](#)

presetPanel.renameItem()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
presetPanel.renameItem(newName)
```

Parameter

newName Ein String, der den neuen Namen der Voreinstellung oder des Ordners angibt.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die Voreinstellung oder der Ordner erfolgreich umbenannt wurden, ansonsten `false`.

Beschreibung

Methode; ändert den Namen der gegenwärtig ausgewählten Voreinstellung oder des Ordners in den angegebenen Namen. Diese Methode wird nur dann erfolgreich ausgeführt, wenn im Ordner „Custom Presets“ eine Voreinstellung oder ein Ordner ausgewählt ist. Diese Methode schlägt in den folgenden Situationen fehl:

- Es ist kein Element ausgewählt.
- Es sind mehrere Elemente ausgewählt.
- Das ausgewählte Element befindet sich im Ordner „Default Presets“.
- Ein Element namens *newName* existiert am selben Speicherort wie das ausgewählte Element.

Beispiel

Im folgenden Beispiel wird die gegenwärtig im Ordner „Custom Presets“ ausgewählte Voreinstellung in *Bounce Faster* umbenannt.

```
var renamed = fl.presetPanel.renameItem("Bounce Faster");  
fl.trace(renamed);
```

presetPanel.selectItem()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
presetPanel.selectItem(namePath [, bReplaceCurrentSelection [, bSelect] ])
```

Parameter

namePath Ein String, der den Pfad und den Namen des Elements angibt, das im Bedienfeld „Bewegungsvoreinstellungen“ ausgewählt werden soll.

bReplaceCurrentSelection Ein boolescher Wert, der angibt, ob das angegebene Element die aktuelle Auswahl ersetzt (`true`) oder erweitert (`false`). Dieser Parameter ist optional, sein Standardwert lautet `true`.

bSelect Ein boolescher Wert, der angibt, ob das Element ausgewählt wird (`true`) oder nicht (`false`). Dieser Parameter ist optional, sein Standardwert lautet `true`. Wenn Sie `false` für *bSelect* übergeben, wird der Wert von *bReplaceCurrentSelection* ignoriert.

Rückgabewerte

Ein boolescher Wert; `true`, wenn das Element erfolgreich ausgewählt oder seine Auswahl aufgehoben wurde, ansonsten `false`.

Beschreibung

Methode; wählt im Bedienfeld „Bewegungsvoreinstellungen“ ein Element aus oder hebt eine Auswahl auf und optional können gegenwärtig ausgewählte Elemente ersetzt werden.

Beispiel

Im folgenden Beispiel wird den gegebenenfalls gegenwärtig im Bedienfeld „Bewegungsvoreinstellungen“ ausgewählten Voreinstellungen die Voreinstellung Hereinfliegen-Weichzeichnen-rechts hinzugefügt:

```
fl.presetPanel.selectItem("Default Presets/fly-in-blur-right", false);
```

Kapitel 35: Rectangle-Objekt

Vererbung [Element-Objekt](#) > [Shape-Objekt](#) > Rectangle-Objekt

Verfügbarkeit

Flash CS3 Professional.

Beschreibung

Das Rectangle-Objekt ist eine Form, die mit dem Werkzeug für Rechteckgrundform gezeichnet wird. Um festzustellen, ob ein Element ein Rectangle-Objekt ist, verwenden Sie [shape.isRectangleObject](#).

Übersicht über Eigenschaften

Abgesehen von den Eigenschaften für das [Shape-Objekt](#) stehen für das Rectangle-Objekt die folgenden Eigenschaften zur Auswahl: Um die Eigenschaften für ein Rectangle-Objekt festzulegen, verwenden Sie [document.setRectangleObjectProperty\(\)](#).

Eigenschaft	Beschreibung
RectangleObject.bottomLeftRadius	Schreibgeschützt; ein Gleitkommawert, der den Radius der unteren linken Ecke des Rectangle-Objekts festlegt.
RectangleObject.bottomRightRadius	Schreibgeschützt; ein Gleitkommawert, der den Radius der unteren rechten Ecke des Rectangle-Objekts festlegt.
RectangleObject.lockFlag	Schreibgeschützt; ein boolescher Wert, der festlegt, ob die verschiedenen Ecken des Rechtecks unterschiedliche Radiuswerte aufweisen können.
RectangleObject.topLeftRadius	Schreibgeschützt; ein Gleitkommawert, der den Radius aller Ecken des Rechtecks oder nur den Radius der oberen linken Ecke des Rechteckobjekts festlegt.
RectangleObject.topRightRadius	Schreibgeschützt; ein Gleitkommawert, der den Radius der oberen rechten Ecke des Rectangle-Objekts festlegt.

RectangleObject.bottomLeftRadius

Verfügbarkeit

Flash CS3 Professional.

Verwendung

[RectangleObject.bottomLeftRadius](#)

Beschreibung

Schreibgeschützte Eigenschaft; ein Gleitkommawert, der den Radius der unteren linken Ecke des Rectangle-Objekts festlegt. Wenn [RectangleObject.lockFlag](#) den Wert `true` aufweist, hat das Festlegen dieses Werts keine Auswirkungen.

Um diesen Wert einzustellen, verwenden Sie [document.setRectangleObjectProperty\(\)](#).

Siehe auch

`document.setRectangleObjectProperty()`, `RectangleObject.bottomRightRadius`,
`RectangleObject.lockFlag`, `RectangleObject.topLeftRadius`, `RectangleObject.topRightRadius`

RectangleObject.bottomRightRadius

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`RectangleObject.bottomRightRadius`

Beschreibung

Schreibgeschützte Eigenschaft; ein Gleitkommawert, der den Radius der unteren rechten Ecke des Rectangle-Objekts festlegt. Wenn `RectangleObject.lockFlag` den Wert „true“ aufweist, hat das Festlegen dieses Werts keine Auswirkungen.

Um diesen Wert einzustellen, verwenden Sie `document.setRectangleObjectProperty()`.

Siehe auch

`document.setRectangleObjectProperty()`, `RectangleObject.bottomLeftRadius`,
`RectangleObject.lockFlag`, `RectangleObject.topLeftRadius`, `RectangleObject.topRightRadius`

RectangleObject.lockFlag

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`RectangleObject.lockFlag`

Beschreibung

Schreibgeschützte Eigenschaft; ein boolescher Wert, der festlegt, ob die verschiedenen Ecken des Rechtecks unterschiedliche Radiuswerte aufweisen können. Wenn dieser Wert `true` lautet, haben alle Ecken den Wert, der festgelegt wurde für `RectangleObject.topLeftRadius`. Wenn dieser Wert `false` lautet, kann der Radius für jede Ecke separat festgelegt werden.

Um diesen Wert einzustellen, verwenden Sie `document.setRectangleObjectProperty()`.

Siehe auch

`document.setRectangleObjectProperty()`, `RectangleObject.bottomLeftRadius`,
`RectangleObject.bottomRightRadius`, `RectangleObject.topLeftRadius`,
`RectangleObject.topRightRadius`

RectangleObject.topLeftRadius

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`RectangleObject.topLeftRadius`

Beschreibung

Schreibgeschützte Eigenschaft; ein Gleitkommawert, der den Radius für alle Ecken des Rechtecks (falls `RectangleObject.lockFlag` den Wert `true` aufweist) oder den Wert der oberen linken Ecke (falls `RectangleObject.lockFlag` den Wert `false` aufweist) festlegt.

Um diesen Wert einzustellen, verwenden Sie `document.setRectangleObjectProperty()`.

Siehe auch

`document.setRectangleObjectProperty()`, `RectangleObject.bottomLeftRadius`,
`RectangleObject.bottomRightRadius`, `RectangleObject.lockFlag`, `RectangleObject.topRightRadius`

RectangleObject.topRightRadius

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`RectangleObject.topRightRadius`

Beschreibung

Schreibgeschützte Eigenschaft; ein Gleitkommawert, der den Radius der oberen rechten Ecke des Rectangle-Objekts festlegt. Wenn `RectangleObject.lockFlag` den Wert `true` aufweist, hat das Festlegen dieses Werts keine Auswirkungen.

Um diesen Wert einzustellen, verwenden Sie `document.setRectangleObjectProperty()`.

Siehe auch

`document.setRectangleObjectProperty()`, `RectangleObject.bottomLeftRadius`,
`RectangleObject.bottomRightRadius`, `RectangleObject.lockFlag`, `RectangleObject.topLeftRadius`

Kapitel 36: Shape-Objekt

Vererbung [Element-Objekt](#) > Shape-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Shape-Objekt ist eine Unterklasse des Element-Objekts. Das Shape-Objekt bietet bei der Bearbeitung und beim Erstellen von Formen auf der Bühne präzisere Steuerungsmöglichkeiten als die Zeichnungs-APIs. Eine präzise Steuerung ist die Voraussetzung dafür, dass Skripts nützliche Effekte und andere Zeichenbefehle erstellen können (siehe [Element-Objekt](#)).

Alle Shape-Methoden und -Eigenschaften, die eine Form oder einen untergeordneten Teil einer Form ändern, müssen zwischen `shape.beginEdit()` und `shape.endEdit()` platziert werden, da sie andernfalls nicht ordnungsgemäß funktionieren.

Übersicht über Methoden

Abgesehen von den Methoden des Element-Objekts stehen für das Shape-Objekt auch die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>shape.getCubicSegmentPoints()</code>	Gibt ein Array von Punkten zurück, die eine kubische Kurve definieren.
<code>shape.beginEdit()</code>	Definiert den Anfang einer Bearbeitungssitzung.
<code>shape.deleteEdge()</code>	Löscht die angegebene Kante.
<code>shape.endEdit()</code>	Definiert das Ende einer Bearbeitungssitzung für die Form.

Übersicht über Eigenschaften

Abgesehen von den Eigenschaften des Element-Objekts stehen für das Shape-Objekt die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>shape.contours</code>	Schreibgeschützt; ein Array von Contour-Objekten für die Form (siehe Contour-Objekt).
<code>shape.edges</code>	Schreibgeschützt; ein Array von Edge-Objekten (siehe Edge-Objekt).
<code>shape.isDrawingObject</code>	Schreibgeschützt; bei dem Wert true ist die Form ein Zeichenobjekt.
<code>shape.isGroup</code>	Schreibgeschützt; bei dem Wert true ist die Form eine Gruppe.
<code>shape.isOvalObject</code>	Schreibgeschützt; falls true, handelt es sich bei der Form um eine Oval-Objektgrundform (die mit dem Ellipsenwerkzeug erstellt wurde).
<code>shape.isRectangleObject</code>	Schreibgeschützt; falls true, handelt es sich bei der Form um eine Rectangle-Objektgrundform (die mit dem Rechteckwerkzeug erstellt wurde).

Eigenschaft	Beschreibung
shape.members	Ein Array von Objekten in der gegenwärtig ausgewählten Gruppe.
shape.numCubicSegments	Schreibgeschützt; die Anzahl der kubischen Segmente in der Form.
shape.vertices	Schreibgeschützt; ein Array von Vertex-Objekten (siehe Vertex-Objekt).

shape.beginEdit()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
shape.beginEdit()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; definiert den Anfang einer Bearbeitungssitzung. Diese Methode muss vor Befehlen verwendet werden, die das Shape-Objekt oder einen seiner untergeordneten Teile ändern.

Beispiel

Im folgenden Beispiel wird die erste Kante des Kanten-Arrays der gegenwärtig ausgewählten Form entfernt:

```
var shape = fl.getDocumentDOM().selection[0];  
shape.beginEdit();  
shape.deleteEdge(0);  
shape.endEdit();
```

shape.contours

Verfügbarkeit

Flash MX 2004.

Verwendung

```
shape.contours
```

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von Contour-Objekten für die Form (siehe [Contour-Objekt](#)).

Beispiel

Im folgenden Beispiel wird die erste Kontur des Konturen-Arrays in der Variablen *c* und dann das [HalfEdge-Objekt](#) dieser Kontur in der Variablen *he* gespeichert:

```
var c = fl.getDocumentDOM().selection[0].contours[0];  
var he = c.getHalfEdge();
```

shape.deleteEdge()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
shape.deleteEdge(index)
```

Parameter

index Ein auf Null basierender Index, der angibt, welche Kante aus dem [shape.edges](#)-Array gelöscht werden soll. Diese Methode ändert die Länge des *shape.edges*-Arrays.

Rückgabewerte

Keine.

Beschreibung

Methode; löscht die angegebene Kante. Sie müssen [shape.beginEdit\(\)](#) aufrufen, bevor Sie diese Methode verwenden.

Beispiel

Im folgenden Beispiel wird die erste Kante des Kanten-Arrays der gegenwärtig ausgewählten Form entfernt:

```
var shape = fl.getDocumentDOM().selection[0];  
shape.beginEdit();  
shape.deleteEdge(0);  
shape.endEdit();
```

shape.edges

Verfügbarkeit

Flash MX 2004.

Verwendung

```
shape.edges
```

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von Edge-Objekten (siehe [Edge-Objekt](#)).

shape.endEdit()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
shape.endEdit()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; definiert das Ende einer Bearbeitungssitzung für die Form. Alle an dem Shape-Objekt oder einem seiner untergeordneten Teile vorgenommen Änderungen werden auf die Form angewendet. Diese Methode muss nach Befehlen verwendet werden, die das Shape-Objekt oder einen seiner untergeordneten Teile ändern.

Beispiel

Im folgenden Beispiel wird die erste Kante des Kanten-Arrays der gegenwärtig ausgewählten Form entfernt:

```
var shape = fl.getDocumentDOM().selection[0];  
shape.beginEdit();  
shape.deleteEdge(0);  
shape.endEdit();
```

shape.getCubicSegmentPoints()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
shape.getCubicSegmentPoints(cubicSegmentIndex)
```

Parameter

cubicSegmentIndex Eine Ganzzahl, die das kubische Segment angibt, für das Punkte zurückgegeben werden.

Rückgabewerte

Ein Array aus Punkten, die eine kubische Kurve für das Edge-Objekt definieren, das dem angegebenen *cubicSegmentIndex* entspricht (siehe [edge.cubicSegmentIndex](#)).

Beschreibung

Methode; gibt ein Array von Punkten zurück, die eine kubische Kurve definieren.

Beispiel

Im folgenden Beispiel werden die *x*- und *y*-Werte für jeden Punkt auf der kubischen Kurve der ersten Kante der Auswahl angezeigt:

```
var elem = fl.getDocumentDOM().selection[0];
var index = elem.edges[0].cubicSegmentIndex;
var cubicPoints = elem.getCubicSegmentPoints(index);
for (i=0; i<cubicPoints.length; i++) {
    fl.trace("index " + i + " x: " + cubicPoints[i].x + " y: " + cubicPoints[i].y);
}
```

shape.isDrawingObject

Verfügbarkeit

Flash 8.

Verwendung

`shape.isDrawingObject`

Beschreibung

Schreibgeschützte Eigenschaft; bei dem Wert `true` ist die Form ein Zeichenobjekt.

Beispiel

Im folgenden Beispiel wird das zuerst ausgewählte Objekt in der `sel`-Variablen gespeichert. Dann wird anhand der Eigenschaften `element.elementType` und `shape.isDrawingObject` geprüft, ob das ausgewählte Element ein Zeichenobjekt ist:

```
var sel = fl.getDocumentDOM().selection[0];
var shapeDrawingObject = (sel.elementType == "shape") && sel.isDrawingObject;
fl.trace(shapeDrawingObject);
```

Siehe auch

[document.crop\(\)](#), [document.deleteEnvelope\(\)](#), [document.intersect\(\)](#), [document.punch\(\)](#),
[document.union\(\)](#), [shape.isGroup](#)

shape.isGroup

Verfügbarkeit

Flash MX 2004.

Verwendung

`shape.isGroup`

Beschreibung

Schreibgeschützte Eigenschaft; bei dem Wert `true` ist die Form eine Gruppe. Eine Gruppe kann Elemente verschiedener Typen enthalten, wie etwa Textelemente und Symbole. Die Gruppe selbst wird jedoch als Form angesehen und die Eigenschaft `shape.isGroup` kann unabhängig von den in der Gruppe enthaltenen Elementtypen angewendet werden.

Beispiel

Im folgenden Beispiel wird das zuerst ausgewählte Objekt in der `sel`-Variablen gespeichert. Dann wird anhand der Eigenschaften `element.elementType` und `shape.isGroup` geprüft, ob das ausgewählte Element eine Gruppe ist:

```
var sel = fl.getDocumentDOM().selection[0];  
var shapeGroup = (sel.elementType == "shape") && sel.isGroup;  
fl.trace(shapeGroup);
```

Siehe auch

[shape.isDrawingObject](#)

shape.isOvalObject

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`shape.isOvalObject`

Beschreibung

Schreibgeschützte Eigenschaft; falls `true`, handelt es sich bei der Form um eine Oval-Objektgrundform (die mit dem Werkzeug für Ellipsengrundform erstellt wurde).

Beispiel

Im folgenden Beispiel wird `"true"` angezeigt, wenn das erste ausgewählte Element eine Oval-Objektgrundform ist; andernfalls wird `"false"` angezeigt:

```
var sel = fl.getDocumentDOM().selection[0];  
fl.trace(sel.isOvalObject);
```

Siehe auch

[shape.isRectangleObject](#)

shape.isRectangleObject

Verfügbarkeit

Flash CS3 Professional.

Verwendung

`shape.isRectangleObject`

Beschreibung

Schreibgeschützte Eigenschaft; falls `true`, handelt es sich bei der Form um eine Rectangle-Objektgrundform (die mit dem Werkzeug für Rechteckgrundform erstellt wurde).

Beispiel

Im folgenden Beispiel wird `"true"` angezeigt, wenn das erste ausgewählte Element eine Rectangle-Objektgrundform ist; andernfalls wird `"false"` angezeigt:

```
var sel = fl.getDocumentDOM().selection[0];  
fl.trace(sel.isRectangleObject);
```

Siehe auch

[shape.isOvalObject](#)

shape.members

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`shape.members`

Beschreibung

Schreibgeschützt; ein Array von Objekten in der gegenwärtig ausgewählten Gruppe. Diese Eigenschaft ist nur verfügbar, wenn der Wert von `shape.isGroup` `true` lautet. Unformatierte Formen in der Gruppe werden nicht in das `shape.members`-Array einbezogen.

Wenn die Gruppe beispielsweise drei Zeichenobjekte und drei unformatierte Formen enthält, enthält das Array `shape.members` drei Einträge, einen für jedes Zeichenobjekt. Wenn die Gruppe nur unformatierte Formen enthält, ist das Array leer.

Beispiel

Im folgenden Beispiel wird die Anzahl der kubischen Segmente in jedem Zeichenobjekt in der gegenwärtig ausgewählten Gruppe angezeigt:

```
var shapesArray = fl.getDocumentDOM().selection[0].members;  
for (i=0; i<shapesArray.length; i++) {  
    fl.trace(shapesArray[i].numCubicSegments);  
}
```

Siehe auch

[shape.isGroup](#)

shape.numCubicSegments

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`shape.numCubicSegments`

Beschreibung

Schreibgeschützte Eigenschaft; die Anzahl der kubischen Segmente in der Form.

Beispiel

Sofern eine Quadrat- oder Rechteckform ausgewählt ist, wird im folgenden Beispiel „4“ im Bedienfeld „Ausgabe“ angezeigt:

```
var theShape = fl.getDocumentDOM().selection[0];  
fl.trace(theShape.numCubicSegments);
```

shape.vertices

Verfügbarkeit

Flash MX 2004.

Verwendung

`shape.vertices`

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von Vertex-Objekten (siehe [Vertex-Objekt](#)).

Beispiel

Im folgenden Beispiel wird das zuerst ausgewählte Objekt in der Variablen `someShape` gespeichert und dann die Anzahl der Scheitelpunkte dieses Objekts im Bedienfeld „Ausgabe“ angezeigt:

```
var someShape = fl.getDocumentDOM().selection[0];  
fl.trace("The shape has " + someShape.vertices.length + " vertices.");
```

Kapitel 37: SoundItem-Objekt

Vererbung [Item-Objekt](#) > SoundItem-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das SoundItem-Objekt ist eine Unterklasse des Item-Objekts. Es stellt ein Bibliothekselement dar, anhand dessen ein Sound erstellt wird. Siehe auch [frame.soundLibraryItem](#) und [Item-Objekt](#).

Übersicht über Methoden

Neben den Methoden des Item-Objekts steht für das SoundItem-Objekt die folgende Methode zur Verfügung:

Eigenschaft	Beschreibung
soundItem.exportToFile()	Exportiert das angegebene Element in eine QuickTime-Datei (Macintosh) bzw. eine WAV- oder QT-Datei (Windows).

Übersicht über Eigenschaften

Abgesehen von den Eigenschaften des Item-Objekts stehen für das SoundItem-Objekt die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
soundItem.bitRate	Ein String, der die Bitrate eines Sounds in der Bibliothek angibt. Nur für den Komprimierungstyp MP3 verfügbar.
soundItem.bits	Ein String, der den Bitwert eines Sounds in der Bibliothek angibt, der mit ADPCM komprimiert wurde.
soundItem.compressionType	Ein String, der den Komprimierungstyp eines Sounds in der Bibliothek angibt.
soundItem.convertStereoToMono	Ein boolescher Wert, der nur für die Komprimierungstypen „MP3“ und „Raw“ verfügbar ist.
soundItem.fileLastModifiedDate	Schreibgeschützt; ein String mit einer hexadezimalen Zahl, die die Anzahl der Sekunden angibt, die seit dem 1. Januar 1970 bis zum Änderungsdatum der ursprünglichen Datei zu dem Zeitpunkt vergangen sind, an dem die Datei in die Bibliothek importiert wurde.
soundItem.originalCompressionType	Schreibgeschützt; ein String, der angibt, ob das angegebene Element als MP3-Datei importiert wurde.
soundItem.quality	Ein String, der die Wiedergabequalität eines Sounds in der Bibliothek angibt. Nur für den Komprimierungstyp MP3 verfügbar.
soundItem.sampleRate	Ein String, der die Abtastrate des Audioclips festlegt.
soundItem.sourceFileExists	Schreibgeschützt; ein boolescher Wert der angibt, ob die in die Bibliothek importierte Datei noch an dem Speicherort existiert, von dem aus sie importiert wurde.

Eigenschaft	Beschreibung
<code>soundItem.sourceFileIsCurrent</code>	Schreibgeschützt; ein boolescher Wert, der angibt, ob das Dateiänderungsdatum des Bibliothekselements mit dem Änderungsdatum der importierten Datei auf der Festplatte identisch ist.
<code>soundItem.sourceFilePath</code>	Schreibgeschützt; ein String im Format file:/// URI, der den Pfad und den Namen der in die Bibliothek importierten Datei angibt.
<code>soundItem.useImportedMP3Quality</code>	Ein Boolescher Wert; mit dem Wert <code>true</code> werden alle anderen Eigenschaften ignoriert und die importierte MP3-Qualität wird verwendet.

soundItem.bitRate

Verfügbarkeit

Flash MX 2004.

Verwendung

`soundItem.bitRate`

Beschreibung

Eigenschaft; ein String, der die Bitrate eines Sounds in der Bibliothek angibt. Diese Eigenschaft ist nur für den Komprimierungstyp „MP3“ verfügbar. Die zulässigen Werte lauten "8 kbps", "16 kbps", "20 kbps", "24 kbps", "32 kbps", "48 kbps", "56 kbps", "64 kbps", "80 kbps", "112 kbps", "128 kbps" und "160 kbps". Mit 8 oder 16 KBit/s exportierte Stereosounds werden in Mono umgewandelt. Bei allen anderen Komprimierungstypen hat die Eigenschaft den Wert `undefined`.

Wenn Sie für diese Eigenschaft einen Wert angeben möchten, müssen Sie `soundItem.useImportedMP3Quality` auf `false` einstellen.

Beispiel

Im folgenden Beispiel wird der Wert `bitRate` im Bedienfeld „Ausgabe“ angezeigt, wenn das angegebene Bibliothekselement den Komprimierungstyp „MP3“ aufweist:

```
alert(fl.getDocumentDOM().library.items[0].bitRate);
```

Siehe auch

`soundItem.compressionType`, `soundItem.convertStereoToMono`

soundItem.bits

Verfügbarkeit

Flash MX 2004.

Verwendung

`soundItem.bits`

Beschreibung

Eigenschaft; ein String, der den Bitwert eines Sounds in der Bibliothek angibt, der mit „ADPCM“ komprimiert wurde. Die zulässigen Werte lauten "2 bit", "3 bit", "4 bit" und "5 bit".

Wenn Sie für diese Eigenschaft einen Wert angeben möchten, müssen Sie `soundItem.useImportedMP3Quality` auf `false` einstellen.

Beispiel

Im folgenden Beispiel wird der Bitwert im Bedienfeld „Ausgabe“ angezeigt, wenn das gegenwärtig ausgewählte Bibliothekselement den Komprimierungstyp „ADPCM“ aufweist:

```
alert(fl.getDocumentDOM().library.items[0].bits);
```

Siehe auch

[soundItem.compressionType](#)

soundItem.compressionType

Verfügbarkeit

Flash MX 2004.

Verwendung

`soundItem.compressionType`

Beschreibung

Eigenschaft; ein String, der den Komprimierungstyp eines Sounds in der Bibliothek angibt. Die zulässigen Werte lauten "Default", "ADPCM", "MP3", "Raw" und "Speech".

Wenn Sie für diese Eigenschaft einen Wert angeben möchten, müssen Sie `soundItem.useImportedMP3Quality` auf `false` einstellen.

Beispiel

Im folgenden Beispiel wird der Komprimierungstyp eines Bibliothekselements in `Raw` abgeändert:

```
fl.getDocumentDOM().library.items[0].compressionType = "Raw";
```

Im folgenden Beispiel wird der Komprimierungstyp der ausgewählten Bibliothekselemente in `Speech` geändert:

```
fl.getDocumentDOM().library.getSelectedItems().compressionType = "Speech";
```

Siehe auch

[soundItem.originalCompressionType](#)

soundItem.convertStereoToMono

Verfügbarkeit

Flash MX 2004.

Verwendung

```
soundItem.convertStereoToMono
```

Beschreibung

Eigenschaft; ein boolescher Wert, der nur für die Komprimierungstypen „MP3“ und „Raw“ verfügbar ist. Bei dem Wert `true` wird ein Stereosound in einen Monosound konvertiert; bei `false` bleibt der Sound ein Stereosound. Bei dem Komprimierungstyp „MP3“ wird diese Eigenschaft ignoriert und auf `true` eingestellt, wenn `soundItem.bitRate` weniger als 20 KBit/s beträgt (siehe [soundItem.bitRate](#)).

Wenn Sie für diese Eigenschaft einen Wert angeben möchten, müssen Sie [soundItem.useImportedMP3Quality](#) auf `false` einstellen.

Beispiel

Im folgenden Beispiel wird ein Bibliothekselement in Mono umgewandelt, sofern das Element den Komprimierungstyp „MP3“ oder „Raw“ aufweist:

```
fl.getDocumentDOM().library.items[0].convertStereoToMono = true;
```

Siehe auch

[soundItem.compressionType](#)

soundItem.exportToFile()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
soundItem.exportToFile(fileURI)
```

Parameter

fileURI Ein String im Format `file:///` URI, der den Namen der exportierten Datei angibt.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die Datei erfolgreich exportiert wurde; andernfalls `false`.

Beschreibung

Methode; exportiert das angegebene Element in eine QuickTime-Datei (Macintosh) bzw. eine WAV- oder QT-Datei (Windows). Die exportierten QuickTime- oder QT-Dateien enthalten nur Audio; Video wird nicht exportiert. Die Exporteinstellungen sind vom exportierten Element abhängig.

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Sound-Element handelt, wird es im folgenden Beispiel als WAV-Datei exportiert:

```
var soundFileURL = "file:///C:/out.wav";  
var libItem = fl.getDocumentDOM().library.items[0];  
libItem.exportToFile(soundFileURL);
```

soundItem.fileLastModifiedDate

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
soundItem.fileLastModifiedDate
```

Beschreibung

Schreibgeschützte Eigenschaft; ein String mit einer hexadezimalen Zahl, die die Anzahl der Sekunden angibt, die seit dem 1. Januar 1970 bis zum Änderungsdatum der ursprünglichen Datei zu dem Zeitpunkt vergangen sind, an dem die Datei in die Bibliothek importiert wurde. Sollte die Datei nicht mehr existieren, lautet dieser Wert "00000000".

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Sound-Element handelt, wird im folgenden Beispiel wie oben beschrieben eine hexadezimale Zahl angezeigt:

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("Mod date when imported = " + libItem.fileLastModifiedDate);
```

Siehe auch

[soundItem.sourceFileExists](#), [soundItem.sourceFileIsCurrent](#), [soundItem.sourceFilePath](#),
[FLfile.getModificationDate\(\)](#)

soundItem.originalCompressionType

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
soundItem.originalCompressionType
```

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der angibt, ob das angegebene Element als mp3-Datei importiert wurde. Die zulässigen Werte für diese Eigenschaft lauten „RAW“ und „MP3“.

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Sound-Element handelt, wird im folgenden Beispiel "MP3" angezeigt, wenn die Datei als mp3-Datei in die Bibliothek importiert wurde, andernfalls "RAW":

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("Imported compression type = " + libItem.originalCompressionType);
```

Siehe auch

[soundItem.compressionType](#)

soundItem.quality

Verfügbarkeit

Flash MX 2004.

Verwendung

`soundItem.quality`

Beschreibung

Eigenschaft; ein String, der die Wiedergabequalität eines Sounds in der Bibliothek angibt. Diese Eigenschaft ist nur für den Komprimierungstyp „MP3“ verfügbar. Die zulässigen Werte lauten "Fast", "Medium" und "Best".

Wenn Sie für diese Eigenschaft einen Wert angeben möchten, müssen Sie `soundItem.useImportedMP3Quality` auf `false` einstellen.

Beispiel

Im folgenden Beispiel wird die Wiedergabequalität eines Bibliothekselements auf `Best` eingestellt, sofern das Element den Komprimierungstyp „MP3“ aufweist:

```
fl.getDocumentDOM().library.items[0].quality = "Best";
```

Siehe auch

[soundItem.compressionType](#)

soundItem.sampleRate

Verfügbarkeit

Flash MX 2004.

Verwendung

`soundItem.sampleRate`

Beschreibung

Eigenschaft; ein String, der die Abtastrate des Audioclips festlegt. Diese Eigenschaft ist nur bei den Komprimierungstypen „ADPCM“, „Raw“ oder „Speech“ verfügbar. Die zulässigen Werte lauten "5 kHz", "11 kHz", "22 kHz" und "44 kHz".

Wenn Sie für diese Eigenschaft einen Wert angeben möchten, müssen Sie `soundItem.useImportedMP3Quality` auf `false` einstellen.

Beispiel

Im folgenden Beispiel wird die Abtastrate eines Bibliothekselements auf 5 kHz eingestellt, sofern das Element den Komprimierungstyp „ADPCM“, „Raw“ oder „Speech“ aufweist:

```
fl.getDocumentDOM().library.items[0].sampleRate = "5 kHz";
```

Siehe auch

[soundItem.compressionType](#)

soundItem.sourceFileExists

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
soundItem.sourceFileExists
```

Beschreibung

Schreibgeschützte Eigenschaft; der boolesche Wert `true`, wenn die in die Bibliothek importierte Datei noch an dem Speicherort existiert, von dem aus sie importiert wurde; andernfalls `false`.

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Sound-Element handelt, wird im folgenden Beispiel "true" angezeigt, wenn die in die Bibliothek importierte Datei noch existiert.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("sourceFileExists = "+ libItem.sourceFileExists);
```

Siehe auch

[soundItem.sourceFileIsCurrent](#), [soundItem.sourceFilePath](#)

soundItem.sourceFileIsCurrent

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
soundItem.sourceFileIsCurrent
```

Beschreibung

Schreibgeschützte Eigenschaft; der boolesche Wert `true`, wenn das Dateiänderungsdatum des Bibliothekselements mit dem Änderungsdatum der importierten Datei auf der Festplatte identisch ist; andernfalls `false`.

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Sound-Element handelt, wird im folgenden Beispiel "true" angezeigt, wenn die importierte Datei auf der Festplatte seit dem Import nicht geändert wurde.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("fileIsCurrent = "+ libItem.sourceFileIsCurrent);
```

Siehe auch

[soundItem.fileLastModifiedDate](#), [soundItem.sourceFilePath](#)

soundItem.sourceFilePath

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`soundItem.sourceFilePath`

Beschreibung

Schreibgeschützte Eigenschaft; ein String im Format `file:///` URI, der den Pfad und den Namen der in die Bibliothek importierten Datei angibt.

Beispiel

Im folgenden Beispiel werden die Namen und die Quelldateipfade aller Bibliothekselemente angezeigt, die dem Typ "sound" angehören:

```
for (idx in fl.getDocumentDOM().library.items) {  
    if (fl.getDocumentDOM().library.items[idx].itemType == "sound") {  
        var myItem = fl.getDocumentDOM().library.items[idx];  
        fl.trace(myItem.name + " source is " + myItem.sourceFilePath);  
    }  
}
```

Siehe auch

[soundItem.sourceFileExists](#)

soundItem.useImportedMP3Quality

Verfügbarkeit

Flash MX 2004.

Verwendung

`soundItem.useImportedMP3Quality`

Beschreibung

Eigenschaft; ein boolescher Wert. Bei dem Wert `true` werden alle anderen Eigenschaften ignoriert und die importierte MP3-Qualität wird verwendet.

Beispiel

Im folgenden Beispiel wird für ein Bibliothekselement festgelegt, dass es die importierte MP3-Qualität verwenden soll:

```
fl.getDocumentDOM().library.items[0].useImportedMP3Quality = true;
```

Siehe auch

[soundItem.compressionType](#)

Kapitel 38: Stroke-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Stroke-Objekt enthält sämtliche Einstellungen für einen Strich, einschließlich der benutzerdefinierten Einstellungen. Dieses Objekt stellt die im Eigenschafteninspektor enthaltenen Informationen dar. Wenn Sie das Stroke-Objekt mit der Methode `document.setCustomStroke()` kombinieren, können Sie die Stricheinstellungen des Bedienfelds „Werkzeuge“, des Eigenschafteninspektors und der aktuellen Auswahl ändern. Mit der Methode `document.getCustomStroke()` können Sie außerdem die Stricheinstellungen des Bedienfelds „Werkzeuge“, des Eigenschafteninspektors oder der aktuellen Auswahl abrufen.

Dieses Objekt hat die folgenden vier Eigenschaften: `style`, `thickness`, `color` und `breakAtCorners`. (In Flash CS3 wurde die Eigenschaft `breakAtCorners` nicht mehr verwendet und durch `stroke.joinType` ersetzt.) Je nach dem Wert der Eigenschaft `stroke.style` können auch andere Eigenschaften eingestellt werden.

Übersicht über Eigenschaften

Für das Stroke-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>stroke.breakAtCorners</code>	Ein boolescher Wert; entspricht der Einstellung „Spitze Ecken“ im Dialogfeld „Strichstil“.
<code>stroke.capType</code>	Ein String, der den Abschlusstyp des Strichs angibt.
<code>stroke.color</code>	Ein String, ein hexadezimaler Wert oder eine Ganzzahl zur Darstellung der Strichfarbe.
<code>stroke.curve</code>	Ein String, der den Schraffierungstyp für den Strich festlegt.
<code>stroke.dash1</code>	Eine Ganzzahl, die die Länge der Striche einer gestrichelten Linie angibt.
<code>stroke.dash2</code>	Eine Ganzzahl, die die Länge der Lücken einer gestrichelten Linie angibt.
<code>stroke.density</code>	Ein String, der die Dichte einer getupften Linie angibt.
<code>stroke.dotSize</code>	Ein String, der die Punktgröße einer getupften Linie angibt.
<code>stroke.dotSpace</code>	Eine Ganzzahl, die den Punktabstand in einer einfach gepunkteten Linie angibt.
<code>stroke.hatchThickness</code>	Ein String, der die Stärke einer schraffierten Linie angibt.
<code>stroke.jiggle</code>	Ein String, der die Verwackeln-Eigenschaft einer schraffierten Linie angibt.
<code>stroke.joinType</code>	Ein String, der den Verbindungstyp des Strichs angibt.
<code>stroke.length</code>	Ein String, der die Länge einer schraffierten Linie angibt.
<code>stroke.miterLimit</code>	Ein Gleitkommawert, der den Winkel angibt, bei dessen Überschreitung die Winkelspitze von einem Segment abgeschnitten wird.
<code>stroke.pattern</code>	Ein String, der das Muster einer ausgefransten Linie angibt.
<code>stroke.rotate</code>	Ein String, der die Drehung einer schraffierten Linie angibt.
<code>stroke.scaleType</code>	Ein String, der den auf den Strich anzuwendenden Skalierungstyp angibt.
<code>stroke.shapeFill</code>	Ein Fill-Objekt , das die Füllungseinstellungen des Strichs darstellt.

Eigenschaft	Beschreibung
<code>stroke.space</code>	Ein String, der die Abstände in einer schraffierten Linie angibt.
<code>stroke.strokeHinting</code>	Ein boolescher Wert, der angibt, ob für einen Strich Hinweise festgelegt wurden.
<code>stroke.style</code>	Ein String, der den Strichstil beschreibt.
<code>stroke.thickness</code>	Eine Ganzzahl, die die Strichgröße festlegt.
<code>stroke.variation</code>	Ein String, der die Variation einer getupften Linie angibt.
<code>stroke.waveHeight</code>	Ein String, der die Wellenhöhe einer ausgefranten Linie angibt.
<code>stroke.waveLength</code>	Ein String, der die Wellenlänge einer ausgefranten Linie angibt.

stroke.breakAtCorners

Verfügbarkeit

Flash MX 2004. Wird in Flash CS3 nicht mehr verwendet und durch `stroke.joinType` ersetzt.

Verwendung

`stroke.breakAtCorners`

Beschreibung

Eigenschaft; ein boolescher Wert. Diese Eigenschaft entspricht der Einstellung „Spitze Ecken“ im Dialogfeld „Strichstil“.

Beispiel

Im folgenden Beispiel wird der Eigenschaft `breakAtCorners` der Wert `true` zugewiesen:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.breakAtCorners = true;  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.capType

Verfügbarkeit

Flash 8.

Verwendung

`stroke.capType`

Beschreibung

Eigenschaft; ein String, der den Abschlusstyp des Strichs angibt. Die zulässigen Werte lauten "none", "round" und "square".

Beispiel

Im folgenden Beispiel wird festgelegt, dass für den Strich der Abschlusstyp `round` verwendet werden soll:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.capType = "round";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.color

Verfügbarkeit

Flash MX 2004. In Flash 8 und höher wurde diese Eigenschaften zugunsten von `stroke.shapeFill.color` aufgegeben.

Verwendung

`stroke.color`

Beschreibung

Eigenschaft; die Farbe des Strichs. Sie wird in einem der folgenden Formate angegeben:

- als String im Format "#RRGGBB" oder "#RRGGBBAA"
- als Hexadezimalzahl im Format "0xRRGGBB"
- als Ganzzahl, die die dezimale Entsprechung einer hexadezimalen Zahl darstellt

Beispiel

Im folgenden Beispiel wird die Strichfarbe festgelegt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.color = "#000000";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

Siehe auch

[stroke.shapeFill](#)

stroke.curve

Verfügbarkeit

Flash MX 2004.

Verwendung

`stroke.curve`

Beschreibung

Eigenschaft; ein String, der den Schraffierungstyp des Strichs festlegt. Diese Eigenschaft kann nur festgelegt werden, wenn die Eigenschaft `stroke.style` den Wert "hatched" (schraffiert) hat (siehe [stroke.style](#)). Die zulässigen Werte lauten "straight", "slight curve", "medium curve" und "very curved".

Beispiel

Im folgenden Beispiel werden für einen Strich mit dem Stil `hatched` die Eigenschaft „curve“ und andere Eigenschaften eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "hatched";  
myStroke.curve = "straight";  
myStroke.space = "close";  
myStroke.jiggle = "wild";  
myStroke.rotate = "free";  
myStroke.length = "slight";  
myStroke.hatchThickness = "thin";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.dash1

Verfügbarkeit

Flash MX 2004.

Verwendung

`stroke.dash1`

Beschreibung

Eigenschaft; eine Ganzzahl, die die Länge der Striche einer gestrichelten Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die `stroke.style` -Eigenschaft `dashed` lautet (siehe [stroke.style](#)).

Beispiel

Im folgenden Beispiel werden die Eigenschaften `dash1` und `dash2` des Strichstils `dashed` festgelegt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "dashed";  
myStroke.dash1 = 1;  
myStroke.dash2 = 2;  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.dash2

Verfügbarkeit

Flash MX 2004.

Verwendung

`stroke.dash2`

Beschreibung

Eigenschaft; eine Ganzzahl, die die Länge der Lücken einer gestrichelten Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die Eigenschaft `stroke.style` auf `dashed` (gestrichelt) eingestellt wurde (siehe [stroke.style](#)).

Beispiel

Siehe [stroke.dash1](#).

stroke.density

Verfügbarkeit

Flash MX 2004.

Verwendung

`stroke.density`

Beschreibung

Eigenschaft; ein String, der die Dichte einer getupften Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die Eigenschaft `stroke.style` auf `stipple` (getupft) eingestellt wurde (siehe [stroke.style](#)). Die zulässigen Werte lauten "very dense", "dense", "sparse" und "very sparse".

Beispiel

Im folgenden Beispiel wird die Eigenschaft „density“ des Strichstils `stipple` auf `sparse` eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "stipple";  
myStroke.dotSpace = 3;  
myStroke.variation = "random sizes";  
myStroke.density = "sparse";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.dotSize

Verfügbarkeit

Flash MX 2004.

Verwendung

`stroke.dotSize`

Beschreibung

Eigenschaft; ein String, der die Punktgröße einer getupften Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die Eigenschaft `stroke.style` auf `stipple` (getupft) eingestellt wurde (siehe [stroke.style](#)). Die zulässigen Werte lauten "tiny", "small", "medium" und "large".

Im folgenden Beispiel wird die Eigenschaft `dotSize` des Strichstils `stipple` auf `tiny` eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "stipple";  
myStroke.dotSpace = 3;  
myStroke.dotSize = "tiny";  
myStroke.variation = "random sizes";  
myStroke.density = "sparse";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.dotSpace

Verfügbarkeit

Flash MX 2004.

Verwendung

`stroke.dotSpace`

Beschreibung

Eigenschaft; eine Ganzzahl, die den Punktabstand in einer gepunkteten Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die Eigenschaft `stroke.style` auf `dotted` eingestellt wurde. Siehe [stroke.style](#).

Beispiel

Im folgenden Beispiel wird die Eigenschaft `dotSpace` des Strichstils `dotted` auf „3“ eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "dotted";  
myStroke.dotSpace = 3;  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.hatchThickness

Verfügbarkeit

Flash MX 2004.

Verwendung

`stroke.hatchThickness`

Beschreibung

Eigenschaft; ein String, der die Stärke einer schraffierten Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die Eigenschaft `stroke.style` auf `hatched` (schraffiert) eingestellt wurde (siehe [stroke.style](#)). Die zulässigen Werte lauten "hairline", "thin", "medium" und "thick".

Beispiel

Im folgenden Beispiel wird die Eigenschaft `hatchThickness` des Strichstils `hatched` auf `thin` eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "hatched";  
myStroke.curve = "straight";  
myStroke.space = "close";  
myStroke.jiggle = "wild";  
myStroke.rotate = "free";  
myStroke.length = "slight";  
myStroke.hatchThickness = "thin";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```


stroke.jiggle

Verfügbarkeit

Flash MX 2004.

Verwendung

`stroke.jiggle`

Beschreibung

Eigenschaft; ein String, der die Verwackeln-Eigenschaft einer schraffierten Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die Eigenschaft `stroke.style` auf `hatched` (schraffiert) eingestellt wurde (siehe [stroke.style](#)). Die zulässigen Werte lauten "none", "bounce", "loose" und "wild".

Beispiel

Im folgenden Beispiel wird die Eigenschaft `jiggle` des Strichstils `hatched` auf `wild` eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "hatched";  
myStroke.curve = "straight";  
myStroke.space = "close";  
myStroke.jiggle = "wild";  
myStroke.rotate = "free";  
myStroke.length = "slight";  
myStroke.hatchThickness = "thin";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.joinType

Verfügbarkeit

Flash 8.

Verwendung

`stroke.joinType`

Beschreibung

Eigenschaft; ein String, der den Verbindungstyp des Strichs angibt. Die zulässigen Werte lauten "miter", "round" und "bevel".

Siehe auch

[stroke.capType](#)

stroke.length

Verfügbarkeit

Flash MX 2004.

Verwendung

`stroke.length`

Beschreibung

Eigenschaft; ein String, der die Länge einer schraffierten Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die Eigenschaft `stroke.style` auf `hatched` (schraffiert) eingestellt wurde (siehe [stroke.style](#)). Zulässige Werte sind "equal", "slight variation", "medium variation" und "random". (Der Wert "random" wird "medium variation" zugeordnet.)

Beispiel

Im folgenden Beispiel wird die Eigenschaft `length` des Strichstils `hatched` auf `slight` eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "hatched";
myStroke.curve = "straight";
myStroke.space = "close";
myStroke.jiggle = "wild";
myStroke.rotate = "free";
myStroke.length = "slight variation";
myStroke.hatchThickness = "thin";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.miterLimit

Verfügbarkeit

Flash 8.

Verwendung

`stroke.miterLimit`

Beschreibung

Eigenschaft; ein Gleitkommawert, der den Winkel angibt, bei dessen Überschreitung die Winkelspitze von einem Segment abgeschnitten wird. Dies geschieht nur, wenn der Winkel größer als der Wert für `miterLimit` ist.

Beispiel

Im folgenden Beispiel wird die Gehrungsgrenze der Stricheinstellung auf 3 geändert: Wenn der Gehrungswinkel größer ist als 3, wird die Gehrung abgeschnitten.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.miterLimit = 3;
var myStroke = fl.getDocumentDOM().setCustomStroke();
```

stroke.pattern

Verfügbarkeit

Flash MX 2004.

Verwendung

`stroke.pattern`

Beschreibung

Eigenschaft; ein String, der das Muster einer ausgefranst Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die Eigenschaft `stroke.style` auf `ragged` (ausgefranst) eingestellt wurde (siehe [stroke.style](#)). Die zulässigen Werte lauten "solid", "simple", "random", "dotted", "random dotted", "triple dotted" und "random triple dotted".

Beispiel

Im folgenden Beispiel wird die Eigenschaft `pattern` des Strichstils `ragged` auf `random` eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "ragged";  
myStroke.pattern = "random";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.rotate

Verfügbarkeit

Flash MX 2004.

Verwendung

`stroke.rotate`

Beschreibung

Eigenschaft; ein String, der die Drehung einer schraffierten Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die Eigenschaft `stroke.style` auf `hatched` (schraffiert) eingestellt wurde (siehe [stroke.style](#)). Die zulässigen Werte lauten "none", "slight", "medium" und "free".

Beispiel

Im folgenden Beispiel wird die Eigenschaft `rotate` des Strichstils `hatched` auf `free` eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "hatched";  
myStroke.curve = "straight";  
myStroke.space = "close";  
myStroke.jiggle = "wild";  
myStroke.rotate = "free";  
myStroke.length = "slight";  
myStroke.hatchThickness = "thin";
```

stroke.scaleType

Verfügbarkeit

Flash 8.

Verwendung

`stroke.scaleType`

Beschreibung

Eigenschaft; ein String, der den auf den Strich anzuwendenden Skalierungstyp angibt. Die zulässigen Werte lauten "normal", "horizontal", "vertical" und "none".

Beispiel

Im folgenden Beispiel wird der Skalierungstyp des Strichs auf `horizontal` eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.scaleType = "horizontal";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.shapeFill

Verfügbarkeit

Flash 8.

Verwendung

`stroke.shapeFill`

Beschreibung

Eigenschaft; ein [Fill-Objekt](#), das die Füllungseinstellungen des Strichs darstellt.

Beispiel

Im folgenden Beispiel werden Füllungseinstellungen festgelegt und anschließend auf den Strich angewendet:

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.linearGradient = true;  
fill.colorArray = [ 00ff00, ff0000, fffff ];  
var stroke = fl.getDocumentDOM().getCustomStroke();  
stroke.shapeFill = fill;  
fl.getDocumentDOM().setCustomStroke(stroke);
```

stroke.space

Verfügbarkeit

Flash MX 2004.

Verwendung

`stroke.space`

Beschreibung

Eigenschaft; ein String, der die Abstände in einer schraffierten Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die Eigenschaft `stroke.style` auf `hatched` (schraffiert) eingestellt wurde (siehe [stroke.style](#)). Die zulässigen Werte lauten "very close", "close", "distant" und "very distant".

Beispiel

Im folgenden Beispiel wird die Eigenschaft `space` des Strichstils `hatched` auf `close` eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "hatched";  
myStroke.curve = "straight";  
myStroke.space = "close";  
myStroke.jiggle = "wild";  
myStroke.rotate = "free";  
myStroke.length = "slight";  
myStroke.hatchThickness = "thin";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.strokeHinting

Verfügbarkeit

Flash 8.

Verwendung

`stroke.strokeHinting`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob für einen Strich Hinweise festgelegt wurden.

Beispiel

Im folgenden Beispiel werden die Strichhinweise für den Strich aktiviert:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.strokeHinting = true;  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.style

Verfügbarkeit

Flash MX 2004.

Verwendung

`stroke.style`

Beschreibung

Eigenschaft; ein String, der den Strichstil beschreibt. Die zulässigen Werte lauten `noStroke`, `solid`, `dashed`, `dotted`, `ragged`, `stipple` und `hatched`. Einige diese Werte setzen die Einstellung weiterer Eigenschaften des Stroke-Objekts voraus, wie in der folgenden Liste beschrieben:

- Lautet der Wert `solid` oder `noStroke`, gibt es keine weiteren Eigenschaften.
- Lautet der Wert `dashed`, gibt es zwei zusätzliche Eigenschaften: `dash1` und `dash2`.
- Lautet der Wert `dotted`, gibt es eine zusätzliche Eigenschaft: `dotSpace`.

- Lautet der Wert "ragged", gibt es drei zusätzliche Eigenschaften: pattern, waveHeight und waveLength.
- Lautet der Wert "stipple", gibt es drei zusätzliche Eigenschaften: dotSize, variation und density.
- Lautet der Wert "hatched", gibt es sechs zusätzliche Eigenschaften: hatchThickness, space, jiggle, rotate, curve und length.

Beispiel

Im folgenden Beispiel wird der Strichstil auf ragged eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "ragged";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.thickness

Verfügbarkeit

Flash MX 2004.

Verwendung

stroke.thickness

Beschreibung

Eigenschaft; eine Ganzzahl, die die Strichgröße festlegt.

Beispiel

Im folgenden Beispiel wird der Eigenschaft thickness des Strichs der Wert 2 zugewiesen:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.thickness = 2;  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.variation

Verfügbarkeit

Flash MX 2004.

Verwendung

stroke.variation

Beschreibung

Eigenschaft; ein String, der die Variation einer getupften Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die Eigenschaft stroke.style auf stipple (getupft) eingestellt wurde (siehe [stroke.style](#)). Die zulässigen Werte lauten "one size", "small variation", "varied sizes" und "random sizes".

Beispiel

Im folgenden Beispiel wird die variation-Eigenschaft des Strichstils stipple auf random sizes eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "stipple";  
myStroke.dotSpace= 3;  
myStroke.variation = "random sizes";  
myStroke.density = "sparse";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.waveHeight

Verfügbarkeit

Flash MX 2004.

Verwendung

stroke.waveHeight

Beschreibung

Eigenschaft; ein String, der die Wellenhöhe einer ausgefranst Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die Eigenschaft `stroke.style` auf `ragged` (ausgefranst) eingestellt wurde (siehe [stroke.style](#)). Die zulässigen Werte lauten "flat", "wavy", "very wavy" und "wild".

Beispiel

Im folgenden Beispiel wird die Eigenschaft `waveHeight` des Strichstils `ragged` auf `flat` eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "ragged";  
myStroke.pattern = "random";  
myStroke.waveHeight = "flat";  
myStroke.waveLength = "short";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.waveLength

Verfügbarkeit

Flash MX 2004.

Verwendung

stroke.waveLength

Beschreibung

Eigenschaft; ein String, der die Wellenlänge einer ausgefranst Linie angibt. Diese Eigenschaft ist nur verfügbar, wenn die Eigenschaft `stroke.style` auf `ragged` (ausgefranst) eingestellt wurde (siehe [stroke.style](#)). Die zulässigen Werte lauten "very short", "short", "medium" und "long".

Beispiel

Im folgenden Beispiel wird die Eigenschaft `waveLength` des Strichstils `ragged` auf `short` eingestellt:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "ragged";  
myStroke.pattern = "random";  
myStroke.waveHeight = 'flat';  
myStroke.waveLength = "short";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```


Kapitel 39: swfPanel-Objekt

Verfügbarkeit

Flash CS4 Professional.

Beschreibung

Das swfPanel-Objekt repräsentiert ein Window-SWF-Bedienfeld. Windows-SWF-Bedienfelder sind SWF-Dateien, die Anwendungen implementieren, die von der Flash-Authoringumgebung aus ausgeführt werden können; sie stehen über das Menü „Fenster“ > „Andere Bedienfelder“ zur Verfügung. Standardmäßig werden Window-SWF-Bedienfelder in einem Unterordner des Ordners „Configuration“ gespeichert (siehe „[Speichern von JSFL-Dateien](#)“ auf Seite 2). Unter Windows XP befindet sich der Ordner zum Beispiel hier: *Startlaufwerk*\Dokumente und Einstellungen\Benutzer\Lokale Einstellungen\Anwendungsdaten\Adobe\Flash CS4\Sprache\Configuration\WindowSWF. Ein Beispiel für ein Window-SWF-Bedienfeld ist verfügbar, siehe „[Beispiel-Bedienfeld „Bitmap nachzeichnen](#)““ auf Seite 14. Das Array registrierter Window-SWF-Bedienfelder ist in der `fl.swfPanels`-Eigenschaft gespeichert.

Übersicht über Methoden

Mit dem swfPanel-Objekt können Sie die folgende Methode verwenden:

Methode	Beschreibung
<code>swfPanel.call()</code>	Stellt zusammen mit den ActionScript-Methoden <code>ExternalInterface.addCallback()</code> und <code>MMExecute()</code> die Kommunikation zwischen dem SWF-Bedienfeld und der Authoring-Umgebung her.
„ <code>swfPanel.setFocus()</code> “ auf Seite 422	Setzt den Tastaturfokus auf das angegebene SWF-Bedienfeld.

Übersicht über Eigenschaften

Mit dem swfPanel-Objekt können Sie die folgenden Eigenschaften verwenden:

Eigenschaft	Beschreibung
<code>swfPanel.name</code>	Schreibgeschützt; ein String, der den Namen des angegebenen Windows-SWF-Bedienfeldes angibt.
<code>swfPanel.path</code>	Schreibgeschützt; ein String, der den Pfad der SWF-Datei darstellt, die im angegebenen Window SWF-Bedienfeld verwendet wird.

swfPanel.call()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
swfPanel.call(request)
```

Parameter

request An die Funktion zu übergebende Parameter (siehe „Beschreibung“ und „Beispiel“ unten).

Rückgabewerte

Entweder `null` oder ein String, der durch den Funktionsaufruf zurückgegeben wird. Das Ergebnis der Funktion könnte ein leerer String sein.

Beschreibung

Methode; stellt zusammen mit den ActionScript-Methoden `ExternalInterface.addCallback()` und `MMExecute()` die Kommunikation zwischen dem SWF-Bedienfeld und der Authoring-Umgebung her.

Beispiel

Im folgenden Beispiel wird die Verwendung von ActionScript- und JavaScript-Code zum Erstellen eines Window-SWF-Bedienfeldes und für die Kommunikation damit aus der Authoringumgebung veranschaulicht.

- 1 Erstellen Sie eine ActionScript 3.0-FLA-Datei, stellen Sie die Farbe auf Mittelgrau und die Größe auf 400 Pixel (Breite) mal 250 Pixel (Höhe) ein.
- 2 Platzieren Sie ein dynamisches Textfeld in der Mitte der Bühne, stellen Sie den Instanznamen auf `myTextField` ein und geben Sie das Wort „Status“ in das Textfeld ein.
- 3 Stellen Sie andere Eigenschaften des Textfeldes ein, zum Beispiel:
 - Zentriert
 - 355 Pixel breit und 46 Pixel hoch
 - Schriftart Times New Roman, 28 Punkt, rot
- 4 Fügen Sie den folgenden ActionScript-Code hinzu:

```
// Here's the callback function to be called from JSAPI
function callMeFromJavascript(arg:String):void
{
    try {
        var name:String = String(arg);
        myTextField.text = name;
    } catch (e:Error) {
    }
}

// Expose the callback function as "callMySWF"
ExternalInterface.addCallback("callMySWF", callMeFromJavascript);

// run the JSAPI to wire up the callback
MMExecute("fl.runScript( fl.configURI + \"WindowSWF/fileOp.jsfl\" );");

MMExecute("fl.trace(\"AS3 File Status Panel Initialized\");");
```

- 5 Speichern Sie die Datei als „fileStatus fla“ und veröffentlichen Sie die SWF-Datei mit den Standardveröffentlichungseinstellungen.
- 6 Schließen Sie Flash.
- 7 Kopieren Sie die Datei „fileStatus.swf“ in den Ordner „WindowSWF“, der sich im Ordner „Configuration“ befindet (siehe „[Speichern von JSFL-Dateien](#)“ auf Seite 2). Unter Windows XP befindet sich der Ordner zum Beispiel hier: *Startlaufwerk*\\Dokumente und Einstellungen\\Benutzer\\Lokale Einstellungen\\Anwendungsdaten\\Adobe\\Flash CS4*Sprache*\\Configuration\\WindowSWF.

8 Starten Sie Flash.

9 Erstellen Sie eine JSFL-Datei mit dem folgenden Code:

```
function callMyPanel(panelName, arg)
{
    if(fl.swfPanels.length > 0){
        for(x = 0; x < fl.swfPanels.length; x++){
            // look for a SWF panel of the specified name, then call the specified AS3
function
            // in this example, the panel is named "test" and the AS3 callback is "callMySWF"
            if(fl.swfPanels[x].name == panelName) // name busted?
            {
                fl.swfPanels[x].call("callMySWF",arg);
                break;
            }
        }
    }
    else
        fl.trace("no panels");
}

// define the various handlers for events
documentClosedHandler = function () { callMyPanel("fileStatus", "Document Closed");};
fl.addEventListener("documentClosed", documentClosedHandler );

var dater = "New Document";
documentNewHandler = function () { callMyPanel("fileStatus", dater );};
fl.addEventListener("documentNew", documentNewHandler );

documentOpenedHandler = function () { callMyPanel("fileStatus", "Document Opened");};
fl.addEventListener("documentOpened", documentOpenedHandler );
```

10 Speichern Sie die JSFL-Datei unter dem Namen „fileOp.jsfl“ in demselben Verzeichnis wie die SWF-Datei.

11 Wählen Sie „Fenster“ > „Andere Bedienfelder“ > „fileStatus“.

Wenn Sie FLA-Dateien erstellen, öffnen und schließen, zeigt das Bedienfeld „fileStatus“ eine Meldung an, die angibt, welche Aktion gerade ausgeführt wurde.

swfPanel.name

Verfügbarkeit

Flash CS4 Professional.

Verwendung

swfPanel.name

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der den Namen des angegebenen Window SWF-Bedienfelds darstellt.

Beispiel

Im folgenden Beispiel wird im Bedienfeld „Ausgabe“ der Name des ersten registrierten Window SWF-Bedienfelds angezeigt:

```
fl.trace(fl.swfPanels[0].name);
```

Siehe auch

[swfPanel.path](#), [fl.swfPanels](#)

swfPanel.path

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`swfPanel.path`

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der den Pfad der SWF-Datei darstellt, die im angegebenen Window SWF-Bedienfeld verwendet wird.

Beispiel

Im folgenden Beispiel wird im Bedienfeld „Ausgabe“ der Pfad der SWF-Datei angezeigt, die im ersten registrierten Window SWF-Bedienfeld verwendet wird:

```
fl.trace(fl.swfPanels[0].path);
```

Siehe auch

[swfPanel.name](#), [fl.swfPanels](#)

swfPanel.setFocus()

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

`swfPanel.setFocus()`

Beschreibung

Methode; setzt den Tastaturfokus auf das angegebene SWF-Bedienfeld.

Beispiel

Mit dem folgenden Code wird der Fokus auf das SWF-Bedienfeld namens „Project“ gesetzt:

Führen Sie die folgenden Schritte durch, bevor Sie diesen Befehl ausführen:

- 1 Lösen Sie das Bedienfeld „Projekt“, um es zu einem schwebenden Bedienfeld zu machen.

- 2 Öffnen Sie das Dialogfeld „Datei erstellen“ über das Bedienfeld „Projekt“ und klicken Sie dann auf die Bühne.
- 3 Drücken Sie mehrmals die Tabulatortaste, um sicherzustellen, dass der Fokus nicht auf dem Bedienfeld „Projekt“ liegt.
- 4 Führen Sie das folgende Skript vom Menü „Befehle“ aus (platzieren Sie eine JSFL-Datei mit dem unten gezeigten Code in das Verzeichnis „user/config/Commands“):
- 5 Drücken Sie die Tabulatortaste. Daraufhin sollte eine Einfügemarke in einem der Textfelder des Dialogfelds „Datei erstellen“ angezeigt werden.

```
flash.getSwfPanel ("Project") .setFocus () ;
```

Siehe auch

`swfPanel.name`, `fl.swfPanels`

Kapitel 40: SymbolInstance-Objekt

Vererbung [Element-Objekt](#) > [Instance-Objekt](#) > SymbolInstance-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das SymbolInstance-Objekt ist eine Unterklasse des Instance-Objekts und entspricht einem Symbol in einem Bild (siehe [Instance-Objekt](#)).

Übersicht über Eigenschaften

Abgesehen von den Eigenschaften des Instance-Objekts stehen für das SymbolInstance-Objekt die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
symbolInstance.accName	Ein String, der dem Feld „Name“ im Bedienfeld „Eingabehilfen“ entspricht.
symbolInstance.actionScript	Ein String, der die dem Symbol zugewiesenen Aktionen angibt.
symbolInstance.backgroundColor	Ein String, der die Hintergrundfarbe angibt, wenn „Undurchsichtig“ ausgewählt ist.
symbolInstance.bitmapRenderMode	Ein String, der den Anzeigetyp für die Symbolinstanz angibt.
symbolInstance.blendMode	Ein String, der den auf ein Movieclip-Symbol anzuwendenden Mischmodus angibt.
symbolInstance.buttonTracking	Ein String, der ausschließlich für Schaltflächensymbole gilt und dieselbe Eigenschaft wie das Popupmenü „Als Schaltfläche behandeln“ oder „Als Menüelement behandeln“ im Eigenschafteninspektor festlegt.
symbolInstance.cacheAsBitmap	Ein boolescher Wert, der angibt, ob Bitmap-Zwischenspeicherung zur Laufzeit aktiviert ist.
symbolInstance.colorAlphaAmount	Eine Ganzzahl, die einen Teil der Farbtransformation für die Instanz festlegt und die Einstellungen für „Erweiterter Effekt/Alpha“ angibt; entspricht der Auswahl der Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor und dem Verschieben der Steuerungen rechts im Dialogfeld.
symbolInstance.colorAlphaPercent	Eine Ganzzahl, die einen Teil der Farbtransformation der Instanz festlegt; die gleiche Wirkung erzielen Sie, wenn Sie im Eigenschafteninspektor für Instanzen die Einstellung „Farbe“ > „Erweitert“ verwenden (die Prozentwertsteuerungen links im Dialogfeld).
symbolInstance.colorBlueAmount	Eine Ganzzahl, die einen Teil der Farbtransformation für die Instanz festlegt; entspricht der Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor für Instanzen.
symbolInstance.colorBluePercent	Eine Ganzzahl, die einen Teil der Farbtransformation der Instanz festlegt; die gleiche Wirkung erzielen Sie, wenn Sie im Eigenschafteninspektor für Instanzen die Einstellung „Farbe“ > „Erweitert“ verwenden (die Prozentwertsteuerungen links im Dialogfeld).
symbolInstance.colorGreenAmount	Eine Ganzzahl, die einen Teil der Farbtransformation für die Instanz festlegt; entspricht der Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor für Instanzen. Es sind Werte von -255 bis 255 zulässig.

Eigenschaft	Beschreibung
<code>symbolInstance.colorGreenPercent</code>	Eine Ganzzahl, die einen Teil der Farbtransformation für die Instanz festlegt; entspricht der Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor für Instanzen (die Prozentwertsteuerungen links im Dialogfeld).
<code>symbolInstance.colorMode</code>	Ein String, der den Farbmodus angibt, der im Eigenschafteninspektor für Symbole im Popupmenü „Farbe“ festgelegt wurde.
<code>symbolInstance.colorRedAmount</code>	Eine Ganzzahl, die einen Teil der Farbtransformation für die Instanz festlegt; entspricht der Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor für Instanzen.
<code>symbolInstance.colorRedPercent</code>	Eine Ganzzahl, die einen Teil der Farbtransformation für die Instanz festlegt; entspricht der Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor für Instanzen (die Prozentwertsteuerungen links im Dialogfeld).
<code>symbolInstance.description</code>	Ein String, der dem Feld „Beschreibung“ im Bedienfeld „Eingabehilfen“ entspricht.
<code>symbolInstance.filters</code>	Ein Array von Filter-Objekten (siehe Filter-Objekt).
<code>symbolInstance.firstFrame</code>	Eine auf Null basierende Ganzzahl, die das erste in der Zeitleiste der Grafik anzuzeigende Bild angibt.
<code>symbolInstance.forceSimple</code>	Ein boolescher Wert, der die Eingabehilfe für die Child-Objekte des Objekts aktiviert oder deaktiviert; entspricht der umgekehrten Logik der Einstellung „Child-Objekte mit Eingabehilfen versehen“ im Bedienfeld „Eingabehilfen“.
<code>symbolInstance.loop</code>	Ein String, der für Grafiksymbbole gilt und dieselbe Eigenschaft wie das Popupmenü „Wiederholen“ im Eigenschafteninspektor festlegt.
<code>symbolInstance.shortcut</code>	Ein String mit dem Kurzbefehl, der dem Symbol zugewiesen wurde; entspricht dem Feld „Kurzbefehl“ im Bedienfeld „Eingabehilfen“.
<code>symbolInstance.silent</code>	Ein boolescher Wert, der die Eingabehilfe für das Objekt aktiviert oder deaktiviert; entspricht der umgekehrten Logik der Einstellung „Objekt mit Eingabehilfen versehen“ im Bedienfeld „Eingabehilfen“.
<code>symbolInstance.symbolType</code>	Ein String, der den Typ des Symbols angibt; entspricht dem Wert für „Verhalten“ in den Dialogfeldern „Neues Symbol erstellen“ und „In Symbol konvertieren“.
<code>symbolInstance.tabIndex</code>	Eine Ganzzahl, die dem Feld „Reihenfolgenposition“ im Bedienfeld „Eingabehilfen“ entspricht.
„ <code>symbolInstance.usesBackgroundColor</code> “ auf Seite 436	Ein boolescher Wert, der das Anzeigeformat angibt.
„ <code>symbolInstance.visible</code> “ auf Seite 437	Ein boolescher Wert, der angibt, ob die Instanz sichtbar ist oder nicht.

symbolInstance.accName

Verfügbarkeit
Flash MX 2004.

Verwendung
`symbolInstance.accName`

Beschreibung

Eigenschaft; ein String, der dem Feld „Name“ im Bedienfeld „Eingabehilfen“ entspricht. Bildschirmleseprogramme geben die Namen von Objekten akustisch wieder, um diese zu identifizieren. Diese Eigenschaft ist nicht für Grafiksymbole verfügbar.

Beispiel

Im folgenden Beispiel wird der Wert des im Bedienfeld „Eingabehilfen“ angezeigten Objektnamens in der Variablen `theName` gespeichert:

```
var theName = fl.getDocumentDOM().selection[0].accName;
```

Im folgenden Beispiel wird dem im Bedienfeld „Eingabehilfen“ angezeigten Objektnamen der Wert `Home Button` zugewiesen:

```
fl.getDocumentDOM().selection[0].accName = "Home Button";
```

symbolInstance.actionScript

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.actionScript`

Beschreibung

Eigenschaft; ein String, der die dem Symbol zugewiesenen Aktionen angibt. Gilt nur für Movieclip- und Schaltflächeninstanzen. Bei einer Grafiksymbollinstanz wird der Wert `undefined` zurückgegeben.

Beispiel

Im folgenden Beispiel wird die Aktion `onClipEvent` dem ersten Element zugewiesen, das sich im ersten Bild der ersten Ebene in der Zeitleiste befindet:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].actionScript  
= "onClipEvent(enterFrame) {trace('movie clip enterFrame');}";
```

symbolInstance.backgroundColor

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

`symbolInstance.backgroundColor`

Beschreibung

Eigenschaft; ein String, der die Hintergrundfarbe angibt, wenn der 24-Bit-Modus für die Instanz ausgewählt ist. Dies ist ein String im hexadezimalen `#rrggbb`-Format oder eine Ganzzahl, die den Wert enthält.

Beispiel

Im folgenden Beispiel wird der Symbolinstanz die Hintergrundfarbe Schwarz zugewiesen:

```
var bitmapInstance = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];  
bitmapInstance.backgroundColor = "#000000";
```

symbolInstance.bitmapRenderMode

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

symbolInstance.bitmapRenderMode

Beschreibung

Eigenschaft; ein String, der den Anzeigetyp für das Symbol festlegt.

Zulässige Werte sind:

- „none“
- „cache“ – legt fest, dass Flash Player das Symbol zur Laufzeit als Bitmap zwischenspeichert.
- „export“ – legt fest, dass das Symbol beim Kompilieren der SWF als Bitmap exportiert wird.

Die ältere „[symbolInstance.cacheAsBitmap](#)“ auf [Seite 428](#)-Eigenschaft ähnelt dieser Eigenschaft, bietet jedoch weniger Optionen da sie den Boolean-Typ hat. Die cacheAsBitmap-Eigenschaft wird in Zukunft möglicherweise als veraltet gelten. Deshalb wird empfohlen, zur neuen Eigenschaft zu wechseln. Die Optionen „true/false“ der booleschen cacheAsBitmap-Eigenschaft entsprechen den Werten „cache“ / „none“ der neuen Eigenschaft.

Beispiel

Im folgenden Beispiel wird bitmapRenderMode für das Symbol auf „export“ eingestellt:

```
var symbol = fl.getDocumentDOM().selection[0];  
fl.trace(symbol.bitmapRenderMode);  
symbol.bitmapRenderMode = "export";
```

symbolInstance.blendMode

Verfügbarkeit

Flash 8.

Verwendung

symbolInstance.blendMode

Beschreibung

Eigenschaft; ein String, der den auf ein Movieclip-Symbol anzuwendenden Mischmodus angibt. Die zulässigen Werte lauten "normal", "layer", "multiply", "screen", "overlay", "hardlight", "lighten", "darken", "difference", "add", "subtract", "invert", "alpha" und "erase".

Beispiel

Im folgenden Beispiel wird dem ersten Movieclip-Symbol im ersten Bild auf der ersten Ebene der Mischmodus add zugewiesen:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].blendMode = "add";
```

Siehe auch

[document.setBlendMode\(\)](#)

symbolInstance.buttonTracking

Verfügbarkeit

Flash MX 2004.

Verwendung

symbolInstance.buttonTracking

Beschreibung

Eigenschaft; ein String, der ausschließlich für Schaltflächensymbole gilt und dieselbe Eigenschaft wie das Popupmenü „Als Schaltfläche behandeln“ oder „Als Menüelement behandeln“ im Eigenschafteninspektor festlegt. Bei anderen Symboltypen wird diese Eigenschaft ignoriert. Die zulässigen Werte lauten "button" und "menu".

Beispiel

Im folgenden Beispiel wird das erste Symbol im ersten Bild der ersten Ebene in der Zeitleiste auf „Als Menüelement behandeln“ eingestellt, sofern es sich bei dem Symbol um eine Schaltfläche handelt:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].buttonTracking = "menu";
```

symbolInstance.cacheAsBitmap

Verfügbarkeit

Flash 8.

Verwendung

symbolInstance.cacheAsBitmap

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob Bitmap-Zwischenspeicherung zur Laufzeit aktiviert ist.

Hinweis: Ab Flash Professional CS5.5 sollten Benutzer anstelle dieser Eigenschaft die „[symbolInstance.bitmapRenderMode](#)“ auf [Seite 427](#)-Eigenschaft verwenden.

Beispiel

Im folgenden Beispiel wird die Bitmap-Zwischenspeicherung zur Laufzeit für das erste Element aktiviert, das sich im ersten Bild auf der ersten Ebene befindet:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].cacheAsBitmap = true;
```

symbolInstance.colorAlphaAmount

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.colorAlphaAmount`

Beschreibung

Eigenschaft; eine Ganzzahl, die einen Teil der Farbtransformation für die Instanz festlegt und die Einstellungen für „Erweiterter Effekt/Alpha“ angibt. Diese Eigenschaft entspricht der Auswahl der Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor und dem Verschieben der Steuerungen rechts im Dialogfeld. Dieser Wert erhöht oder verringert die Farbton- und Alphawerte um einen konstanten Betrag. Der Wert wird dem aktuellen Wert hinzugefügt. Diese Eigenschaft ist vor allem in Kombination mit [symbolInstance.colorAlphaPercent](#) überaus nützlich. Es sind Werte von -255 bis 255 zulässig.

Beispiel

Im folgenden Beispiel wird die Alphaeinstellung der ausgewählten Symbolinstanz um den Betrag 100 verringert:

```
fl.getDocumentDOM().selection[0].colorAlphaAmount = -100;
```

symbolInstance.colorAlphaPercent

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.colorAlphaPercent`

Beschreibung

Eigenschaft; eine Ganzzahl, die einen Teil der Farbtransformation für die Instanz festlegt. Die gleiche Wirkung erzielen Sie, wenn Sie die Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor für Instanzen wählen (die Prozentwertsteuerungen rechts im Dialogfeld). Dieser Wert ändert die Farbton- und Alphawerte auf einen bestimmten Prozentwert. Es sind Werte von -100 bis 100 zulässig. Siehe auch [symbolInstance.colorAlphaAmount](#).

Beispiel

Im folgenden Beispiel wird der Wert `colorAlphaPercent` der ausgewählten Symbolinstanz auf 80 eingestellt:

```
fl.getDocumentDOM().selection[0].colorAlphaPercent = 80;
```

symbolInstance.colorBlueAmount

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.colorBlueAmount`

Beschreibung

Eigenschaft; eine Ganzzahl, die einen Teil der Farbtransformation für die Instanz festlegt. Diese Eigenschaft entspricht der Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor für Instanzen. Es sind Werte von -255 bis 255 zulässig.

symbolInstance.colorBluePercent

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.colorBluePercent`

Beschreibung

Eigenschaft; eine Ganzzahl, die einen Teil der Farbtransformation für die Instanz festlegt. Die gleiche Wirkung erzielen Sie, wenn Sie die Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor für Instanzen wählen (die Prozentwertsteuerungen rechts im Dialogfeld). Dieser Wert weist den Blau-Werten einen bestimmten Prozentwert zu. Es sind Werte von -100 bis 100 zulässig.

Beispiel

Im folgenden Beispiel wird der Wert `colorBluePercent` der ausgewählten Symbolinstanz auf 80 eingestellt:

```
fl.getDocumentDOM().selection[0].colorBluePercent = 80;
```

symbolInstance.colorGreenAmount

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.colorGreenAmount`

Beschreibung

Eigenschaft; eine Ganzzahl, die einen Teil der Farbtransformation für die Instanz festlegt. Diese Eigenschaft entspricht der Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor für Instanzen. Es sind Werte von -255 bis 255 zulässig.

symbolInstance.colorGreenPercent

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.colorGreenPercent`

Beschreibung

Eigenschaft; legt einen Teil der Farbtransformation für die Instanz fest. Die gleiche Wirkung erzielen Sie, wenn Sie die Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor für Instanzen wählen (die Prozentwertsteuerungen rechts im Dialogfeld). Dieser Wert weist den Grün-Werten einen bestimmten Prozentwert zu. Es sind Werte von -100 bis 100 zulässig.

Beispiel

Im folgenden Beispiel wird der Wert `colorGreenPercent` der ausgewählten Symbolinstanz auf 70 eingestellt:

```
fl.getDocumentDOM().selection[0].colorGreenPercent = 70;
```

symbolInstance.colorMode

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.colorMode`

Beschreibung

Eigenschaft; ein String, der den im Eigenschafteninspektor für Symbole im Popupmenü „Farbe“ festgelegten Farbmodus angibt. Die zulässigen Werte lauten "none", "brightness", "tint", "alpha" und "advanced".

Beispiel

Im folgenden Beispiel wird die Eigenschaft `colorMode` des ersten Elements, das sich im ersten Bild in der ersten Ebene der Zeitleiste befindet, auf alpha eingestellt:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].colorMode = "alpha";
```

symbolInstance.colorRedAmount

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.colorRedAmount`

Beschreibung

Eigenschaft; eine Ganzzahl, die einen Teil der Farbtransformation für die Instanz festlegt. Diese Eigenschaft entspricht der Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor für Instanzen. Es sind Werte von -255 bis 255 zulässig.

Beispiel

Im folgenden Beispiel wird der Wert `colorRedAmount` der ausgewählten Symbolinstanz auf 255 eingestellt:

```
fl.getDocumentDOM().selection[0].colorRedAmount = 255;
```

symbolInstance.colorRedPercent

Verfügbarkeit

Flash MX 2004.

Verwendung

```
symbolInstance.colorRedPercent
```

Beschreibung

Eigenschaft; legt einen Teil der Farbtransformation für die Instanz fest. Die gleiche Wirkung erzielen Sie, wenn Sie die Einstellung „Farbe“ > „Erweitert“ im Eigenschafteninspektor für Instanzen wählen (die Prozentwertsteuerungen rechts im Dialogfeld). Dieser Wert weist den Rot-Werten einen bestimmten Prozentwert zu. Es sind Werte von -100 bis 100 zulässig.

Beispiel

Im folgenden Beispiel wird der Wert `colorRedPercent` der ausgewählten Symbolinstanz auf 10 eingestellt:

```
fl.getDocumentDOM().selection[0].colorRedPercent = 10;
```

symbolInstance.description

Verfügbarkeit

Flash MX 2004.

Verwendung

```
symbolInstance.description
```

Beschreibung

Eigenschaft; ein String, der dem Feld „Beschreibung“ im Bedienfeld „Eingabehilfen“ entspricht. Die Beschreibung wird von Bildschirmleseprogrammen akustisch wiedergegeben. Diese Eigenschaft ist nicht für Grafiksymbole verfügbar.

Beispiel

Im folgenden Beispiel wird der Wert der im Bedienfeld „Eingabehilfen“ angezeigten Objektbeschreibung in der Variablen `theDescription` gespeichert:

```
var theDescription = fl.getDocumentDOM().selection[0].description;
```

Im folgenden Beispiel wird der im Bedienfeld „Eingabehilfen“ angezeigten Objektbeschreibung der Wert `Click the home button to go to home` zugewiesen:

```
fl.getDocumentDOM().selection[0].description= "Click the home button to go to home";
```

symbolInstance.filters

Verfügbarkeit

Flash 8.

Verwendung

`symbolInstance.filters`

Beschreibung

Eigenschaft; ein Array von Filter-Objekten (siehe [Filter-Objekt](#)). Wenn Sie Filtereigenschaften ändern möchten, schreiben Sie diese Änderungen nicht direkt in das Array. Sie müssen vielmehr das Array abrufen, die einzelnen Eigenschaften festlegen und dann das Array den neuen Eigenschaften entsprechend einstellen.

Beispiel

Im folgenden Beispiel wird der Name der Filters an Indexposition 0 erfasst. Handelt es sich dabei um einen Glühen-Filter, wird dessen `blurX`-Eigenschaft auf 100 gesetzt und der neue Wert in das Array des Filters eingetragen.

```
var filterName =  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].filters[0].name;  
fl.trace(filterName);  
var filterArray = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].filters;  
if (filterName == 'glowFilter'){  
    filterArray[0].blurX = 100;  
}  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].filters = filterArray;
```

symbolInstance.firstFrame

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.firstFrame`

Beschreibung

Eigenschaft; eine auf Null basierende Ganzzahl, die das erste in der Zeitleiste der Grafik anzuzeigende Bild angibt. Diese Eigenschaft gilt nur für Grafiksymbole und legt dieselbe Eigenschaft wie das Feld „Erstes“ im Eigenschafteninspektor fest. Bei anderen Symboltypen ist diese Eigenschaft `undefined`.

Beispiel

Im folgenden Beispiel wird angegeben, dass Bild 10 das erste Bild sein soll, das in der Zeitleiste des angegebenen Elements angezeigt wird:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].firstFrame = 10;
```

symbolInstance.forceSimple

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.forceSimple`

Beschreibung

Eigenschaft; ein boolescher Wert, der die Eingabehilfe für die Child-Objekte des Objekts aktiviert und deaktiviert. Dies entspricht der umgekehrten Logik der Einstellung „Child-Objekte mit Eingabehilfen versehen“ im Bedienfeld „Eingabehilfen“. Wenn `forceSimple` den Wert `true` aufweist, entspricht dies einer Deaktivierung der Option „Child-Objekte mit Eingabehilfen versehen“. Wenn `forceSimple` den Wert `false` aufweist, entspricht dies einer Aktivierung der Option „Child-Objekte mit Eingabehilfen versehen“.

Diese Eigenschaft ist nur für MovieClip-Objekte verfügbar.

Beispiel

Im folgenden Beispiel wird geprüft, ob die Child-Objekte des Objekts mit Eingabehilfen versehen sind; ist dies der Fall, wird `false` zurückgegeben:

```
var areChildrenAccessible = fl.getDocumentDOM().selection[0].forceSimple;
```

Im folgenden Beispiel werden die Child-Objekte des Objekts mit Eingabehilfen versehen:

```
fl.getDocumentDOM().selection[0].forceSimple = false;
```

symbolInstance.loop

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.loop`

Beschreibung

Eigenschaft; ein String, der für Grafiksymbole gilt und dieselbe Eigenschaft wie das Pop-up-Menü „Wiederholen“ im Eigenschafteninspektor festlegt. Bei anderen Symboltypen ist diese Eigenschaft `undefined`. Die Grafikanimation wird mit den zulässigen Werten `"loop"`, `"play once"` und `"single frame"` entsprechend eingestellt.

Beispiel

Im folgenden Beispiel wird dem ersten Symbol, das sich im ersten Bild in der ersten Ebene der Zeitleiste befindet, der Wert `single frame` (Einzelbild) zugewiesen (d. h. ein bestimmtes Bild der Grafikzeitleiste wird angezeigt), sofern es sich bei dem Symbol um eine Grafik handelt:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].loop = 'single frame';
```


symbolInstance.shortcut

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.shortcut`

Beschreibung

Eigenschaft; ein String mit dem Kurzbefehl, der dem Symbol zugewiesen wurde. Diese Eigenschaft entspricht dem Feld „Kurzbefehl“ im Bedienfeld „Eingabehilfen“. Der Kurzbefehl wird von Bildschirmleseprogrammen akustisch wiedergegeben. Diese Eigenschaft ist nicht für Grafiksymbole verfügbar.

Beispiel

Im folgenden Beispiel wird der Wert des Kurzbefehls des Objekts in der Variablen `theShortcut` gespeichert:

```
var theShortcut = fl.getDocumentDOM().selection[0].shortcut;
```

Im folgenden Beispiel wird dem Objekt der Kurzbefehl "Ctrl+i" zugewiesen:

```
fl.getDocumentDOM().selection[0].shortcut = "Ctrl+i";
```

symbolInstance.silent

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.silent`

Beschreibung

Eigenschaft; ein boolescher Wert, der die Eingabehilfe für das Objekt aktiviert oder deaktiviert. Diese Eigenschaft entspricht der umgekehrten Logik der Einstellung „Objekt mit Eingabehilfen versehen“ im Bedienfeld „Eingabehilfen“. Wenn `silent` den Wert `true` aufweist, entspricht dies einer Deaktivierung der Option „Objekt mit Eingabehilfen versehen“. Wenn `silent` den Wert `false` aufweist, entspricht dies einer Aktivierung der Option „Objekt mit Eingabehilfen versehen“.

Diese Eigenschaft ist nicht für Grafikobjekte verfügbar.

Beispiel

Im folgenden Beispiel wird geprüft, ob das Objekt mit Eingabehilfen versehen ist; ist dies der Fall, wird der Wert `false` zurückgegeben:

```
var isSilent = fl.getDocumentDOM().selection[0].silent;
```

Im folgenden Beispiel wird das Objekt mit Eingabehilfen versehen:

```
fl.getDocumentDOM().selection[0].silent = false;
```

symbolInstance.symbolType

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.symbolType`

Beschreibung

Eigenschaft; ein String, der den Typ des Symbols angibt. Diese Eigenschaft entspricht dem Wert für „Verhalten“ in den Dialogfeldern „Neues Symbol erstellen“ und „In Symbol konvertieren“. Die zulässigen Werte lauten "button", "movie clip" und "graphic".

Beispiel

Im folgenden Beispiel wird angegeben, dass das erste Symbol, das sich im ersten Bild in der ersten Ebene der Zeitleiste befindet, sich wie ein Grafiksymbolsymbol verhalten soll:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].symbolType = "graphic";
```

symbolInstance.tabIndex

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolInstance.tabIndex`

Beschreibung

Eigenschaft; eine Ganzzahl, die dem Feld „Reihenfolgenposition“ im Bedienfeld „Eingabehilfen“ entspricht. Erstellt die Reihenfolge, in der ein Benutzer Objekte durch Drücken der Tabulatortaste ansteuern kann. Diese Eigenschaft ist nicht für Grafiksymbole verfügbar.

Beispiel

Im folgenden Beispiel wird der Eigenschaft `tabIndex` des Objekts „mySymbol“ der Wert 3 zugewiesen und dieser Wert im Bedienfeld „Ausgabe“ angezeigt:

```
var mySymbol = fl.getDocumentDOM().selection[0];  
mySymbol.tabIndex = 3;  
fl.trace(mySymbol.tabIndex);
```

symbolInstance.backgroundColor

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

`symbolInstance.usesBackgroundColor`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob für die Instanz der 24-Bit-Modus (true) oder der 32-Bit-Modus mit Alpha (false) verwendet wird. Beim Wert „true“ wird für die Instanz die Hintergrundfarbe verwendet, die über `backgroundColor` festgelegt ist.

Beispiel

Im folgenden Beispiel wird die `usesBackgroundColor`-Eigenschaft einer Instanz auf „true“ gesetzt:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].useTransparentBackground = true;
```

symbolInstance.visible

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

`symbolInstance.visible`

Beschreibung

Eigenschaft; ein boolescher Wert, der die `visible`-Eigenschaft eines Objekts aktiviert (true) oder deaktiviert (false).

Beispiel

Im folgenden Beispiel wird die Sichtbarkeit des ersten Elements im ersten Bild auf der ersten Ebene auf „false“ gesetzt:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].visible = false;
```

Kapitel 41: SymbolItem-Objekt

Vererbung [Item-Objekt](#) > SymbolItem-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das SoundItem-Objekt ist eine Unterklasse aus dem [Item-Objekt](#).

Übersicht über Methoden

Abgesehen von den Methoden des Item-Objekts stehen für das SymbolItem-Objekt die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>symbolItem.convertToCompiledClip()</code>	Konvertiert ein Symbolelement in der Bibliothek in einen kompilierten Movieclip.
<code>symbolItem.exportSWC()</code>	Exportiert das Symbolelement in eine SWC-Datei.
<code>symbolItem.exportSWF()</code>	Exportiert das Symbolelement in eine SWF-Datei.

Übersicht über Eigenschaften

Abgesehen von den Eigenschaften des Item-Objekts stehen für das SymbolItem-Objekt die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>symbolItem.scalingGrid</code>	Ein boolescher Wert, der angibt, ob die Skalierung im 9-teiligen Segmentraster für das Element aktiviert ist.
<code>symbolItem.scalingGridRect</code>	Ein Rechteckobjekt, das die Lage der vier Hilfslinien im 9-teiligen Segmentraster angibt.
<code>symbolItem.sourceAutoUpdate</code>	Ein boolescher Wert, der angibt, ob das Element aktualisiert wird, wenn die FLA-Datei veröffentlicht wird.
<code>symbolItem.sourceFilePath</code>	Ein String, der den Pfad der FLA-Quelldatei im Format file:/// URI angibt.
<code>symbolItem.sourceLibraryName</code>	Ein String, der den Namen des Elements in der Bibliothek der Quelldatei angibt.
<code>symbolItem.symbolType</code>	Ein String, der den Symboltyp angibt.
<code>symbolItem.timeline</code>	Schreibgeschützt; ein Timeline-Objekt .

symbolItem.convertToCompiledClip()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
symbolItem.convertToCompiledClip()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; konvertiert ein Symbolelement in der Bibliothek in einen kompilierten Movieclip.

Beispiel

Im folgenden Beispiel wird ein Element in der Bibliothek in einen kompilierten Movieclip konvertiert:

```
fl.getDocumentDOM().library.items[3].convertToCompiledClip();
```

symbolItem.exportSWC()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
symbolItem.exportSWC(outputURI)
```

Parameter

outputURI Ein String im Format file:/// URI, der die SWC-Datei angibt, in die das Symbol exportiert wird. *outputURI* muss auf eine lokale Datei verweisen. Wenn *outputURI* nicht vorhanden ist, wird von Flash kein Ordner erstellt.

Rückgabewerte

Keine.

Beschreibung

Methode; exportiert das Symbolelement in eine SWC-Datei.

Beispiel

Im folgenden Beispiel wird ein Bibliothekselement in die SWC-Datei `mySymbol.swc` im Ordner „tests“ exportiert:

```
fl.getDocumentDOM().library.selectItem("mySymbol");  
var currentSelection = fl.getDocumentDOM().library.getSelectedItems();  
currentSelection[0].exportSWC("file:///Macintosh HD/SWCDirectory/mySymbol.swc");
```

symbolItem.exportSWF()

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolItem.exportSWF(outputURI)`

Parameter

outputURI Ein String im Format `file:/// URI`, der die SWF-Datei angibt, in die das Symbol exportiert wird. *outputURI* muss auf eine lokale Datei verweisen. Wenn *outputURI* nicht vorhanden ist, wird von Flash kein Ordner erstellt.

Rückgabewerte

Keine.

Beschreibung

Methode; exportiert das Symbolelement in eine SWF-Datei.

Beispiel

Im folgenden Beispiel wird ein Bibliothekselement in die SWF-Datei `my.swf` im Ordner „tests“ exportiert:

```
fl.getDocumentDOM().library.items[0].exportSWF("file:///c:/tests/my.swf");
```

symbolItem.scalingGrid

Verfügbarkeit

Flash 8.

Verwendung

`symbolItem.scalingGrid`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob die Skalierung im 9-teiligen Segmentraster für das Element aktiviert ist.

Beispiel

Im folgenden Beispiel wird die Skalierung im 9-teiligen Segmentraster für ein Bibliothekselement aktiviert:

```
fl.getDocumentDOM().library.items[0].scalingGrid = true;
```

Siehe auch

[symbolItem.scalingGridRect](#)

symbolItem.scalingGridRect

Verfügbarkeit

Flash 8.

Verwendung

`symbolItem.scalingGridRect`

Beschreibung

Eigenschaft; ein Rechteckobjekt, das die Lage der vier Hilfslinien im 9-teiligen Segmentraster angibt. Informationen zum Format des Rechtecks finden Sie unter [document.addNewRectangle\(\)](#).

Beispiel

Im folgenden Beispiel wird die Position der Hilfslinien im 9-teiligen Segmentraster festgelegt:

```
fl.getDocumentDOM().library.items[0].scalingGridRect = {left:338, top:237, right:3859,  
bottom:713};
```

Siehe auch

[symbolItem.scalingGrid](#)

symbolItem.sourceAutoUpdate

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolItem.sourceAutoUpdate`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob das Element aktualisiert wird, wenn die FLA-Datei veröffentlicht wird. Der Standardwert lautet `false`. Wird für Symbole der gemeinsamen Bibliothek verwendet.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `sourceAutoUpdate` für ein Bibliothekselement eingestellt:

```
fl.getDocumentDOM().library.items[0].sourceAutoUpdate = true;
```

symbolItem.sourceFilePath

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolItem.sourceFilePath`

Beschreibung

Eigenschaft; ein String, der den Pfad der FLA-Quelldatei im Format `file:///` URI angibt. Bei dem Pfad muss es sich um einen absoluten Pfad und nicht um einen relativen Pfad handeln. Diese Eigenschaft wird für Symbole der gemeinsamen Bibliothek verwendet.

Beispiel

Im folgenden Beispiel wird der Wert der Eigenschaft `sourceFilePath` im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace(fl.getDocumentDOM().library.items[0].sourceFilePath);
```

symbolItem.sourceLibraryName

Verfügbarkeit

Flash MX 2004.

Verwendung

```
symbolItem.sourceLibraryName
```

Beschreibung

Eigenschaft; ein String, der den Namen des Elements in der Bibliothek der Quelldatei angibt. Sie wird für Symbole der gemeinsamen Bibliothek verwendet.

Beispiel

Im folgenden Beispiel wird der Wert der Eigenschaft `sourceLibraryName` im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace(fl.getDocumentDOM().library.items[0].sourceLibraryName);
```

symbolItem.symbolType

Verfügbarkeit

Flash MX 2004.

Verwendung

```
symbolItem.symbolType
```

Beschreibung

Eigenschaft; ein String, der den Typ des Symbols angibt. Die zulässigen Werte lauten "movie clip", "button" und "graphic".

Beispiel

Im folgenden Beispiel wird der aktuelle Wert der Eigenschaft `symbolType` angezeigt, in `button` geändert und dann erneut angezeigt:

```
alert(fl.getDocumentDOM().library.items[0].symbolType);  
fl.getDocumentDOM().library.items[0].symbolType = "button";  
alert(fl.getDocumentDOM().library.items[0].symbolType);
```

symbolItem.timeline

Verfügbarkeit

Flash MX 2004.

Verwendung

`symbolItem.timeline`

Beschreibung

Schreibgeschützte Eigenschaft; ein [Timeline-Objekt](#).

Beispiel

Im folgenden Beispiel wird die Anzahl der Ebenen abgerufen und angezeigt, die der in der Bibliothek ausgewählte Movieclip enthält:

```
var tl = fl.getDocumentDOM().library.getSelectedItems()[0].timeline;  
alert(tl.layerCount);
```

Kapitel 42: Text-Objekt

Vererbung [Element-Objekt](#) > Text-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Text-Objekt stellt ein einzelnes Textelement in einem Dokument dar. Alle Eigenschaften des Textes gelten für den gesamten Textblock.

Wie Sie die Eigenschaften einer Textreihe innerhalb des Textfelds einstellen, erfahren Sie in der Übersicht über die Eigenschaften für das [TextAttrs-Objekt](#). Wenn Sie die Eigenschaften einer Auswahl in einem Textfeld ändern möchten, können Sie `document.setElementTextAttr()` eingeben und einen Textbereich angeben bzw. die aktuelle Auswahl verwenden.

Um die generischen Eigenschaften des ausgewählten Textfelds festzulegen, verwenden Sie `document.setElementProperty()`. Im folgenden Beispiel wird der `x`-Wert des Registrierungspunkts des ausgewählten Textfelds auf 50 eingestellt:

```
fl.getDocumentDOM().setElementProperty("x", 50);
```

Übersicht über Methoden

Abgesehen von den Methoden des Element-Objekts stehen für das Text-Objekt die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>text.getTextAttr()</code>	Ruft das angegebene Attribut für den Text ab, der durch die optionalen Parameter <i>startIndex</i> und <i>endIndex</i> gekennzeichnet ist.
<code>text.getTextString()</code>	Ruft den angegebenen Textbereich ab.
<code>text.setTextAttr()</code>	Stellt das angegebene Attribut für den Text ein, der durch <i>startIndex</i> und <i>endIndex</i> gekennzeichnet ist.
<code>text.setTextString()</code>	Ersetzt den Textstring in diesem Text-Objekt.

Übersicht über Eigenschaften

Abgesehen von den Eigenschaften des Element-Objekts stehen für das Text-Objekt die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>text.accName</code>	Ein String, der dem Feld „Name“ im Bedienfeld „Eingabehilfen“ entspricht.
<code>text.antiAliasSharpness</code>	Ein Gleitkommawert, der das Anti-Aliasing für die Textschärfe festlegt.
<code>text.antiAliasThickness</code>	Ein Gleitkommawert, der das Anti-Aliasing für die Textbreite festlegt.
<code>text.autoExpand</code>	Ein boolescher Wert, der die Zunahme der Begrenzungsbreite von statischen Textfeldern bzw. der Begrenzungsbreite und -höhe von dynamischen oder Eingabetextfeldern steuert.
<code>text.border</code>	Ein boolescher Wert, der steuert, ob in Flash für dynamischen Text oder Eingabetext eine Umrandung angezeigt (<code>true</code>) oder nicht angezeigt (<code>false</code>) wird.

Eigenschaft	Beschreibung
<code>text.description</code>	Ein String, der dem Feld „Beschreibung“ im Bedienfeld „Eingabehilfen“ entspricht.
<code>text.embeddedCharacters</code>	Ein String, der die einzubettenden Zeichen angibt. Dies entspricht der Eingabe von Text im Dialogfeld „Zeicheneinbettung“.
<code>text.embedRanges</code>	Ein String, der sich aus Ganzzahlen mit Trennzeichen zusammensetzt. Diese Ganzzahlen entsprechen den Elementen, die im Dialogfeld „Zeicheneinbettung“ ausgewählt werden können.
<code>text.embedVariantGlyphs</code>	Ein boolescher Wert, mit dem angegeben wird, ob das Einbetten von Glyphenvarianten aktiviert werden soll.
<code>text.fontRenderingMode</code>	Ein String, der den Wiedergabemodus für den Text angibt.
<code>text.length</code>	Schreibgeschützt; eine Ganzzahl, die angibt, wie viele Zeichen das Text-Objekt enthält.
<code>text.lineType</code>	Ein String, der den Zeilentyp auf "single line", "multiline", "multiline no wrap" oder "password" einstellt.
<code>text.maxCharacters</code>	Eine Ganzzahl, die angibt, wie viele Zeichen der Benutzer maximal in dieses Text-Objekt eingeben kann.
<code>text.orientation</code>	Ein String, der die Ausrichtung des Textfelds angibt.
<code>text.renderAsHTML</code>	Ein boolescher Wert, der steuert, ob Flash den Text als HTML zeichnet und eingebettete HTML-Tags auswertet.
<code>text.scrollable</code>	Ein boolescher Wert, der steuert, ob ein Bildlauf durch den Text möglich (<code>true</code>) oder nicht möglich (<code>false</code>) ist.
<code>text.selectable</code>	Ein boolescher Wert, der steuert, ob der Text auswählbar (<code>true</code>) oder nicht auswählbar (<code>false</code>) ist. Eingabetext kann immer ausgewählt werden.
<code>text.selectionEnd</code>	Eine auf Null basierende Ganzzahl, die den Abstand am Ende einer Textunterauswahl festlegt.
<code>text.selectionStart</code>	Eine auf Null basierende Ganzzahl, die den Abstand am Anfang einer Textunterauswahl festlegt.
<code>text.shortcut</code>	Ein String, der dem Feld „Kurzbehl“ im Bedienfeld „Eingabehilfen“ entspricht.
<code>text.silent</code>	Ein boolescher Wert, der angibt, ob das Objekt mit Eingabehilfen versehen ist.
<code>text.tabIndex</code>	Eine Ganzzahl, die dem Feld „Reihenfolgenposition“ im Bedienfeld „Eingabehilfen“ entspricht.
<code>text.textRuns</code>	Schreibgeschützt; ein Array von TextRun-Objekten.
<code>text.textType</code>	Ein String, der den Typ des Textfelds angibt. Die zulässigen Werte lauten "static", "dynamic" und "input".
<code>text.useDeviceFonts</code>	Ein boolescher Wert. Bei dem Wert <code>true</code> wird der Text anhand von Geräteschriftarten reproduziert.
<code>text.variableName</code>	Ein String mit dem Inhalt des Text-Objekts.

text.accName

Verfügbarkeit
Flash MX 2004.

Verwendung
`text.accName`

Beschreibung

Eigenschaft; ein String, der dem Feld „Name“ im Bedienfeld „Eingabehilfen“ entspricht. Bildschirmleseprogramme geben die Namen von Objekten akustisch wieder, um diese zu identifizieren. Diese Eigenschaft kann bei dynamischem Text nicht verwendet werden.

Beispiel

Im folgenden Beispiel wird der Name des Objekts abgerufen:

```
var doc = fl.getDocumentDOM();  
var theName = doc.selection[0].accName;
```

Im folgenden Beispiel wird der Name des gegenwärtig ausgewählten Objekts festgelegt:

```
fl.getDocumentDOM().selection[0].accName = "Home Button";
```

text.antiAliasSharpness

Verfügbarkeit

Flash 8.

Verwendung

```
text.antiAliasSharpness
```

Beschreibung

Eigenschaft; ein Gleitkommawert, der das Anti-Aliasing für die Textschärfe festlegt. Diese Eigenschaft steuert, mit welcher Schärfe die Textkanten dargestellt werden. Höhere Werte definieren schärfere Textkanten. Ein Wert von 0 gibt eine normale Schärfe an. Diese Eigenschaft ist nur verfügbar, wenn `text.fontRenderingMode` auf `customThicknessSharpness` eingestellt ist.

Beispiel

Siehe [text.fontRenderingMode](#).

Siehe auch

[text.antiAliasThickness](#), [text.fontRenderingMode](#)

text.antiAliasThickness

Verfügbarkeit

Flash 8.

Verwendung

```
text.antiAliasThickness
```

Beschreibung

Eigenschaft; ein Gleitkommawert, der das Anti-Aliasing für die Textbreite festlegt. Diese Eigenschaft steuert, mit welcher Linienstärke Text dargestellt wird. Höhere Werte definieren dickeren Text. Ein Wert von 0 gibt eine normale Linienstärke für den Text an. Diese Eigenschaft ist nur verfügbar, wenn `text.fontRenderingMode` auf `customThicknessSharpness` eingestellt ist.

Beispiel

Siehe `text.fontRenderingMode`.

Siehe auch

`text.antiAliasSharpness`, `text.fontRenderingMode`

text.autoExpand

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.autoExpand`

Beschreibung

Eigenschaft; ein boolescher Wert. Bei statischen Textfeldern bewirkt der Wert `true` eine Zunahme der Begrenzungsbreite, damit der gesamte Text angezeigt werden kann. Bei dynamischen oder Eingabetextfeldern bewirkt der Wert `true` eine Zunahme der Begrenzungsbreite und -höhe, damit der gesamte Text angezeigt werden kann.

Beispiel

Im folgenden Beispiel wird der Eigenschaft `autoExpand` der Wert `true` zugewiesen:

```
fl.getDocumentDOM().selection[0].autoExpand = true;
```

text.border

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.border`

Beschreibung

Eigenschaft; ein boolescher Wert. Wenn der Wert `true` lautet, wird um den Text eine Umrandung angezeigt.

Beispiel

Im folgenden Beispiel wird der Eigenschaft `border` der Wert `true` zugewiesen:

```
fl.getDocumentDOM().selection[0].border = true;
```

text.description

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.description`

Beschreibung

Eigenschaft; ein String, der dem Feld „Beschreibung“ im Bedienfeld „Eingabehilfen“ entspricht. Die Beschreibung wird von Bildschirmleseprogrammen akustisch wiedergegeben.

Beispiel

Im folgenden Beispiel wird die Beschreibung des Objekts abgerufen:

```
var doc = fl.getDocumentDOM();  
var desc = doc.selection[0].description;
```

Im folgenden Beispiel wird die Beschreibung des Objekts festgelegt:

```
var doc = fl.getDocumentDOM();  
doc.selection[0].description= "Enter your name here";
```

text.embeddedCharacters

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.embeddedCharacters`

Beschreibung

Eigenschaft; ein String, der die einzubettenden Zeichen angibt. Dies entspricht der Eingabe von Text im Dialogfeld „Zeicheneinbettung“.

Diese Eigenschaft ist nur für dynamischen oder Eingabetext vorgesehen. Bei anderen Texttypen ruft sie eine Warnung hervor.

Hinweis: Ab Flash Professional CS5 wird die Schriftarteinbettung nicht mehr auf der Ebene des Textobjekts, sondern auf der Dokumentenebene gesteuert. Verwenden Sie die Eigenschaft „[fontItem.embeddedCharacters](#)“ auf Seite 287 anstelle der `text.embeddedCharacters`-Eigenschaft.

Beispiel

Im folgenden Beispiel wird davon ausgegangen, dass das erste oder einzige Element in der aktuellen Auswahl ein klassisches Textobjekt ist. Außerdem wird die `embeddedCharacters`-Eigenschaft auf `abc` gesetzt:

```
fl.getDocumentDOM().selection[0].embeddedCharacters = "abc";
```

text.embedRanges

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.embedRanges`

Beschreibung

Eigenschaft; ein String, der sich aus Ganzzahlen mit Trennzeichen zusammensetzt. Diese Ganzzahlen entsprechen den Elementen, die im Dialogfeld „Zeicheneinbettung“ ausgewählt werden können. Diese Eigenschaft ist ausschließlich für dynamischen oder Eingabetext vorgesehen. Bei statischem Text wird sie ignoriert.

Diese Eigenschaft entspricht der XML-Datei im Ordner „Configuration/Font Embedding“.

Hinweis: Ab Flash Professional CS5 wird die Schriftarteinbettung nicht mehr auf der Ebene des Textobjekts, sondern auf der Dokumentebene gesteuert. Verwenden Sie die Eigenschaft „[fontItem.embedRanges](#)“ auf Seite 288 anstelle der `text.embedRanges`-Eigenschaft.

Beispiel

Im folgenden Beispiel wird davon ausgegangen, dass das erste oder einzige Element in der aktuellen Auswahl ein klassisches Textobjekt ist. Außerdem wird die `embedRanges`-Eigenschaft auf `"1|3|7"` gesetzt:

```
var doc = fl.getDocumentDOM();  
doc.selection[0].embedRanges = "1|3|7";
```

Im folgenden Beispiel wird die Eigenschaft zurückgesetzt:

```
var doc = fl.getDocumentDOM();  
doc.selection[0].embedRanges = "";
```

text.embedVariantGlyphs

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`text.embedVariantGlyphs`

Beschreibung

Eigenschaft; ein boolescher Wert, mit dem angegeben wird, ob das Einbetten von Glyphenvarianten aktiviert werden soll (`true`) oder nicht (`false`). Diese Eigenschaft ist ausschließlich für dynamischen oder Eingabetext vorgesehen. Bei statischem Text wird sie ignoriert. Der Standardwert lautet `false`.

Hinweis: Ab Flash Professional CS5 wird die Schriftarteinbettung nicht mehr auf der Ebene des Textobjekts, sondern auf der Dokumentebene gesteuert. Verwenden Sie die Eigenschaft „[fontItem.embedVariantGlyphs](#)“ auf Seite 288 anstelle der `text.embedVariantGlyphs`-Eigenschaft. In Flash Professional CS5 hat die `text.embedVariantGlyphs`-Eigenschaft keine Auswirkung mehr, da Flash Glyphenvarianten für TLF-Text immer und für klassischen Text nie einbettet.

Beispiel

Im folgenden Beispiel wird das Einbetten von Glyphenvarianten im ausgewählten Text-Objekt aktiviert:

```
fl.getDocumentDOM().selection[0].embedVariantGlyphs = true;
```

Siehe auch

[fontItem.embedVariantGlyphs](#)

text.fontRenderingMode

Verfügbarkeit

Flash 8.

Verwendung

`text.fontRenderingMode`

Beschreibung

Eigenschaft; ein String, der den Wiedergabemodus für den Text angibt. Diese Eigenschaft wirkt sich darauf aus, wie der Text auf der Bühne und im Flash Player aussieht. Eine Liste der zulässigen Werte finden Sie in der folgenden Tabelle:

Eigenschaftswert	Textwiedergabe
device	Der Text wird mit Geräteschriftarten wiedergegeben.
bitmap	Text mit Alias wird, ähnlich einer Pixelschriftart, als Bitmap wiedergegeben.
standard	Der Text wird mithilfe der Standardglättungsmethode von Flash MX 2004 wiedergegeben. Dies ist die beste Einstellung für animierten, sehr umfangreichen oder geneigten Text.
advanced	Der Text wird mithilfe der in Flash 8 implementierten erweiterten Anti-Aliasing-Schriftwiedergabetechnologie wiedergegeben, die ein verbessertes Anti-Aliasing und damit, besonders bei kleiner Schrift, eine höhere Lesbarkeit ergibt.
customThicknessSharpness	Ermöglicht bei Nutzung der in Flash 8 implementierten erweiterten Anti-Aliasing-Schriftwiedergabetechnologie die Angabe von benutzerdefinierten Einstellungen für Schärfe und Dicke von Text.

Beispiel

Im folgenden Beispiel wird demonstriert, wie man mit dem Wert `customThicknessSharpness` die Textschärfe und -breite festlegen kann:

```
fl.getDocumentDOM().setProperty("fontRenderingMode", "customThicknessSharpness");  
fl.getDocumentDOM().setProperty("antiAliasSharpness", 400);  
fl.getDocumentDOM().setProperty("antiAliasThickness", -200);
```

Siehe auch

[text.antiAliasSharpness](#), [text.antiAliasThickness](#)

text.getTextAttr()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
text.getTextAttr(attrName [, startIndex [, endIndex]])
```

Parameter

attrName Ein String mit dem Namen der zurückzugebenden TextAttrs-Eigenschaft. Eine Liste der für *attrName* zulässigen Werte finden Sie in der Übersicht über die Eigenschaften für das [TextAttrs-Objekt](#).

startIndex Eine Ganzzahl, die die Indexposition des ersten Zeichens angibt. Dieser Parameter ist optional.

endIndex Eine Ganzzahl, die das Ende des Textbereichs angibt. Der Bereich beginnt bei *startIndex* und endet bei *endIndex*, schließt das Zeichen an letzter Position aber nicht ein. Dieser Parameter ist optional.

Rückgabewerte

Der Wert des im Parameter *attrName* angegebenen Attributs.

Beschreibung

Methode; ruft das im Parameter *attrName* angegebene Attribut für den Text ab, der durch die optionalen Parameter *startIndex* und *endIndex* gekennzeichnet ist. Wenn es für den angegebenen Bereich kein einheitliches Attribut gibt, gibt Flash den Wert `undefined` zurück. Wenn die optionalen Parameter *startIndex* und *endIndex* fehlen, verwendet die Methode den gesamten Textbereich. Wenn Sie nur *startIndex* angeben, beschränkt sich der Bereich auf ein einzelnes Zeichen an dieser Position. Wenn Sie sowohl *startIndex* als auch *endIndex* angeben, beginnt der Bereich bei *startIndex* und endet bei *endIndex*, ohne aber das Zeichen an letzter Position einzuschließen.

Beispiel

Im folgenden Beispiel wird die Schriftgröße des gegenwärtig ausgewählten Textfelds abgerufen und angezeigt:

```
var TheTextSize = fl.getDocumentDOM().selection[0].getTextAttr("size");  
fl.trace(TheTextSize);
```

Im folgenden Beispiel wird die Textfüllfarbe des ausgewählten Textfelds abgerufen:

```
var TheFill = fl.getDocumentDOM().selection[0].getTextAttr("fillColor");  
fl.trace(TheFill);
```

Im folgenden Beispiel wird die Größe des dritten Zeichens abgerufen:

```
var Char3 = fl.getDocumentDOM().selection[0].getTextAttr("size", 2);  
fl.trace(Char3);
```

Im folgenden Beispiel wird die Farbe des ausgewählten Textfelds vom dritten bis zum achten Zeichen abgerufen:

```
fl.getDocumentDOM().selection[0].getTextAttr("fillColor", 2, 8);
```

text.getTextString()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
text.getTextString([startIndex [, endIndex]])
```

Parameter

startIndex Eine Ganzzahl, die die Indexposition (auf Null basierend) des ersten Zeichens angibt. Dieser Parameter ist optional.

endIndex Eine Ganzzahl, die das Ende des Textbereichs angibt. Der Bereich beginnt bei *startIndex* und endet bei *endIndex*, schließt das Zeichen an letzter Position aber nicht ein. Dieser Parameter ist optional.

Rückgabewerte

Ein String mit dem Text in dem angegebenen Bereich.

Beschreibung

Methode; ruft den angegebenen Textbereich ab. Wenn Sie die optionalen Parameter *startIndex* und *endIndex* weglassen, wird der gesamte Textstring zurückgegeben. Wenn Sie nur *startIndex* angeben, gibt die Methode einen String zurück, der von dieser Indexposition bis zum Ende des Felds reicht. Wenn Sie sowohl *startIndex* als auch *endIndex* angeben, gibt die Methode einen String zurück, der bei *startIndex* beginnt und bei *endIndex* endet, das Zeichen an letzter Position aber nicht einschließt.

Beispiel

Im folgenden Beispiel wird im ausgewählten Textfeld ein Zeichenbereich abgerufen, der vom fünften Zeichen bis zum Ende des Textfelds reicht:

```
var myText = fl.getDocumentDOM().selection[0].getTextString(4);  
fl.trace(myText);
```

Im folgenden Beispiel wird im ausgewählten Textfeld ein Zeichenbereich abgerufen, der vom vierten bis zum neunten Zeichen reicht:

```
var myText = fl.getDocumentDOM().selection[0].getTextString(3, 9);  
fl.trace(myText);
```

text.length

Verfügbarkeit

Flash MX 2004.

Verwendung

```
text.length
```

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die angibt, wie viele Zeichen das Text-Objekt enthält.

Beispiel

Im folgenden Beispiel wird die Anzahl der Zeichen im ausgewählten Text zurückgegeben:

```
var textLength = fl.getDocumentDOM().selection[0].length;
```

text.lineType

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.lineType`

Beschreibung

Eigenschaft; ein String, der den Zeilentyp festlegt. Die zulässigen Werte lauten "single line", "multiline", "multiline no wrap" und "password".

Diese Eigenschaft ist nur für dynamischen oder Eingabetext vorgesehen. Bei statischem Text ruft sie eine Warnung hervor. Der Wert "password" funktioniert nur bei Eingabetext.

Beispiel

Im folgenden Beispiel wird der Eigenschaft `lineType` der Wert `multiline no wrap` zugewiesen:

```
fl.getDocumentDOM().selection[0].lineType = "multiline no wrap";
```

text.maxCharacters

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.maxCharacters`

Beschreibung

Eigenschaft; eine Ganzzahl, die angibt, wie viele Zeichen der Benutzer maximal in dieses Text-Objekt eingeben kann.

Diese Eigenschaft ist nur für Eingabetext vorgesehen. Bei anderen Texttypen ruft sie eine Warnung hervor.

Beispiel

Im folgenden Beispiel wird der Wert der Eigenschaft `maxCharacters` auf „30“ eingestellt:

```
fl.getDocumentDOM().selection[0].maxCharacters = 30;
```

text.orientation

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.orientation`

Beschreibung

Eigenschaft; ein String, der die Ausrichtung des Textfelds angibt. Die zulässigen Werte lauten "horizontal", "vertical left to right" und "vertical right to left".

Diese Eigenschaft ist nur für statischen Text vorgesehen. Bei anderen Texttypen ruft sie eine Warnung hervor.

Beispiel

Im folgenden Beispiel wird die orientation-Eigenschaft auf `vertical right to left` eingestellt:

```
fl.getDocumentDOM().selection[0].orientation = "vertical right to left";
```

text.renderAsHTML

Verfügbarkeit

Flash MX 2004.

Verwendung

```
text.renderAsHTML
```

Beschreibung

Eigenschaft; ein boolescher Wert. Bei dem Wert `true` zeichnet Flash den Text als HTML und wertet eingebettete HTML-Tags aus.

Diese Eigenschaft ist nur für dynamischen oder Eingabetext vorgesehen. Bei anderen Texttypen ruft sie eine Warnung hervor.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `renderAsHTML` auf `true` eingestellt:

```
fl.getDocumentDOM().selection[0].renderAsHTML = true;
```

text.scrollable

Verfügbarkeit

Flash MX 2004.

Verwendung

```
text.scrollable
```

Beschreibung

Eigenschaft; ein boolescher Wert. Wenn der Wert `true` lautet, ist ein Bildlauf durch den Text möglich.

Diese Eigenschaft ist nur für dynamischen oder Eingabetext vorgesehen. Bei statischem Text ruft sie eine Warnung hervor.

Beispiel

Im folgenden Beispiel wird der Eigenschaft `scrollable` der Wert `false` zugewiesen:

```
fl.getDocumentDOM().selection[0].scrollable = false;
```

text.selectable

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.selectable`

Beschreibung

Eigenschaft; ein boolescher Wert. Wenn der Wert `true` lautet, kann der Text ausgewählt werden.

Eingabetext kann immer ausgewählt werden. Wenn diese Eigenschaft den Wert `false` aufweist und bei Eingabetext verwendet wird, wird eine Warnung angezeigt.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `selectable` auf `true` eingestellt:

```
fl.getDocumentDOM().selection[0].selectable = true;
```

text.selectionEnd

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.selectionEnd`

Beschreibung

Eigenschaft; eine auf Null basierende Ganzzahl, die das Ende einer Textunterauswahl festlegt. Weitere Informationen hierzu finden Sie unter [text.selectionStart](#).

text.selectionStart

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.selectionStart`

Beschreibung

Eigenschaft; eine auf Null basierende Ganzzahl, die den Anfang einer Textunterauswahl festlegt. Wenn Sie diese Eigenschaft mit `text.selectionEnd` kombinieren, können Sie einen Zeichenbereich auswählen. Der ausgewählte Zeichenbereich reicht bis `text.selectionEnd`, schließt aber das Zeichen an letzter Position nicht ein. Siehe [text.selectionEnd](#).

- Wenn es eine Einfügemarke oder keine Auswahl gibt, sind `text.selectionEnd` und `text.selectionStart` identisch.
- Wenn der Wert für `text.selectionStart` größer als der Wert für `text.selectionEnd` ist, werden `text.selectionEnd` und `text.selectionStart` gleichgesetzt und es wird kein Text ausgewählt.

Beispiel

Im folgenden Beispiel wird festgelegt, dass die Textunterauswahl bei dem sechsten Zeichen beginnen soll:

```
fl.getDocumentDOM().selection[0].selectionStart = 5;
```

Im folgenden Beispiel werden in einem Textfeld, das den Text `My name is Barbara` enthält, die Zeichen `Barbara` ausgewählt und anschließend mit einer Fettformatierung und einer grünen Farbe versehen:

```
fl.getDocumentDOM().selection[0].selectionStart = 11;  
fl.getDocumentDOM().selection[0].selectionEnd = 18;  
var s = fl.getDocumentDOM().selection[0].selectionStart;  
var e = fl.getDocumentDOM().selection[0].selectionEnd;  
fl.getDocumentDOM().setElementTextAttr('bold', true, s, e);  
fl.getDocumentDOM().setElementTextAttr("fillColor", "#00ff00", s, e);
```

text.setTextAttr()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
text.setTextAttr(attrName, attrValue [, startIndex [, endIndex]])
```

Parameter

attrName Ein String mit dem Namen der zu ändernden TextAttrs-Objekteigenschaft.

attrValue Der Wert der TextAttrs-Objekteigenschaft.

Eine Liste der für `attrName` und `attrValue` zulässigen Werte finden Sie in der Übersicht über die Eigenschaften für das [TextAttrs-Objekt](#).

startIndex Eine Ganzzahl, die die Indexposition (auf Null basierend) des ersten Zeichens in dem Array angibt. Dieser Parameter ist optional.

endIndex Eine Ganzzahl, die den Index des Endpunkts in dem ausgewählten Textbereich angibt. Der Bereich beginnt bei `startIndex` und endet bei `endIndex`, schließt das Zeichen an letzter Position aber nicht ein. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; legt das von dem Parameter *attrName* angegebene Attribut für den Text, der durch *startIndex* und *endIndex* gekennzeichnet ist, auf den in *attrValue* angegebenen Wert fest. Mit dieser Methode können Sie Attribute von Textstellen ändern, die sich möglicherweise über mehrere TextRun-Elemente erstrecken (siehe [TextRun-Objekt](#)) oder bei denen es sich um Teile vorhandener TextRun-Elemente handelt. Diese Methode kann die Position und Anzahl der TextRun-Elemente innerhalb des `text.textRuns`-Arrays des Objekts ändern (siehe [text.textRuns](#)).

Wenn Sie die optionalen Parameter weglassen, verwendet die Methode den gesamten Zeichenbereich des Text-Objekts. Wenn Sie nur *startIndex* angeben, beschränkt sich der Bereich auf ein einzelnes Zeichen an dieser Position. Wenn Sie sowohl *startIndex* als auch *endIndex* angeben, beginnt der Bereich bei *startIndex* und endet bei *endIndex*, ohne aber das Zeichen an letzter Position einzuschließen.

Beispiel

Im folgenden Beispiel wird dem ausgewählten Textfeld Kursivdruck zugewiesen:

```
fl.getDocumentDOM().selection[0].setTextAttr("italic", true);
```

Im folgenden Beispiel wird dem dritten Zeichen die Größe 10 zugewiesen:

```
fl.getDocumentDOM().selection[0].setTextAttr("size", 10, 2);
```

Im folgenden Beispiel wird den Zeichen 3 bis 8 in dem ausgewählten Text eine rote Farbe zugewiesen:

```
fl.getDocumentDOM().selection[0].setTextAttr("fillColor", 0xff0000, 2, 8);
```

text.setTextString()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
text.setTextString(text [, startIndex [, endIndex]])
```

Parameter

text Ein String mit den Zeichen, die in dieses Text-Objekt eingefügt werden sollen.

startIndex Eine Ganzzahl, die die Indexposition (auf Null basierend) des Zeichens in dem String angibt, an der der Text eingefügt werden soll. Dieser Parameter ist optional.

endIndex Eine Ganzzahl, die die Indexposition des Endpunkts im ausgewählten Textstring angibt. Der neue Text ersetzt den vorhandenen Text, der von *startIndex* bis *endIndex* reicht, mit Ausnahme des Zeichens an letzter Position. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Eigenschaft; ersetzt den Textstring in diesem Text-Objekt. Wenn Sie diesen Parameter weglassen, wird das gesamte Text-Objekt ersetzt. Wenn Sie nur *startIndex* angeben, wird der angegebene String an der Position *startIndex* eingefügt. Wenn Sie sowohl *startIndex* als auch *endIndex* angeben, ersetzt der angegebene String das Textsegment, das von *startIndex* bis *endIndex* reicht, mit Ausnahme des Zeichens an letzter Position.

Beispiel

Im folgenden Beispiel wird dem ausgewählten Textfeld der String `this is a string` zugewiesen:

```
fl.getDocumentDOM().selection[0].setTextString("this is a string");
```

Im folgenden Beispiel wird der String `abc` im ausgewählten Textfeld an der Position des fünften Zeichens eingefügt:

```
fl.getDocumentDOM().selection[0].setTextString("01234567890");  
fl.getDocumentDOM().selection[0].setTextString("abc", 4);  
// text field is now "0123abc4567890"
```

Im folgenden Beispiel werden die Zeichen 3 bis 8 des ausgewählten Textstrings durch den String `abcdefghij` ersetzt. Die Zeichen, die bisher in dem durch *startIndex* und *endIndex* gekennzeichneten Bereich vorhanden waren, werden überschrieben. Alle Zeichen ab der Position *endIndex* folgen auf den eingefügten String.

```
fl.getDocumentDOM().selection[0].setTextString("01234567890");  
fl.getDocumentDOM().selection[0].setTextString("abcdefghij", 2, 8);  
// text field is now "01abcdefghij890"
```

text.shortcut

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.shortcut`

Beschreibung

Eigenschaft; ein String, der dem Feld „Kurzbehl“ im Bedienfeld „Eingabehilfen“ entspricht. Der Kurzbehl wird vom Bildschirmleseprogramm akustisch wiedergegeben. Diese Eigenschaft kann bei dynamischem Text nicht verwendet werden.

Beispiel

Im folgenden Beispiel wird der Tastaturbehl des ausgewählten Objekts abgerufen und dessen Wert angezeigt:

```
var theShortcut = fl.getDocumentDOM().selection[0].shortcut;  
fl.trace(theShortcut);
```

Im folgenden Beispiel wird der Tastaturbehl des ausgewählten Objekts festgelegt:

```
fl.getDocumentDOM().selection[0].shortcut = "Ctrl+i";
```

text.silent

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.silent`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob das Objekt mit Eingabehilfen versehen ist. Dies entspricht der umgekehrten Logik der Einstellung „Objekt mit Eingabehilfen versehen“ im Bedienfeld „Eingabehilfen“. Wenn der Wert für `silent` also `true` lautet, ist die Option „Objekt mit Eingabehilfen versehen“ deaktiviert. Bei dem Wert `false` hingegen ist „Objekt mit Eingabehilfen versehen“ aktiviert.

Beispiel

Im folgenden Beispiel wird festgestellt, ob das Objekt mit Eingabehilfen versehen ist (bei dem Wert `false` ist dies der Fall):

```
var isSilent = fl.getDocumentDOM().selection[0].silent;
```

Im folgenden Beispiel wird das Objekt mit Eingabehilfen versehen:

```
fl.getDocumentDOM().selection[0].silent = false;
```

text.tabIndex

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.tabIndex`

Beschreibung

Eigenschaft; eine Ganzzahl, die dem Feld „Reihenfolgenposition“ im Bedienfeld „Eingabehilfen“ entspricht. Anhand dieses Wertes können Sie die Reihenfolge festlegen, in der ein Benutzer die Objekte durch Drücken der Tabulatortaste ansteuern kann.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `tabIndex` des gegenwärtig ausgewählten Objekts abgerufen:

```
var theTabIndex = fl.getDocumentDOM().selection[0].tabIndex;
```

Im folgenden Beispiel wird die Eigenschaft `tabIndex` des gegenwärtig ausgewählten Objekts festgelegt:

```
fl.getDocumentDOM().selection[0].tabIndex = 1;
```

text.textRuns

Verfügbarkeit

Flash MX 2004.

Verwendung

`text.textRuns`

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von `TextRun`-Objekten (siehe [TextRun-Objekt](#)).

Beispiel

Im folgenden Beispiel wird der Wert der Eigenschaft `textRuns` in der Variablen `myTextRuns` gespeichert:

```
var myTextRuns = fl.getDocumentDOM().selection[0].textRuns;
```

text.textType

Verfügbarkeit

Flash MX 2004.

Verwendung

```
text.textType
```

Beschreibung

Eigenschaft; ein String, der den Typ des Textfelds angibt. Die zulässigen Werte lauten "static", "dynamic" und "input".

Beispiel

Im folgenden Beispiel wird der `textType`-Eigenschaft der Wert `input` zugewiesen:

```
fl.getDocumentDOM().selection[0].textType = "input";
```

text.useDeviceFonts

Verfügbarkeit

Flash MX 2004.

Verwendung

```
text.useDeviceFonts
```

Beschreibung

Eigenschaft; ein boolescher Wert. Bei dem Wert `true` wird der Text anhand von Geräteschriftarten reproduziert.

Beispiel

Im folgenden Beispiel wird Flash angewiesen, bei der Darstellung von Text Geräteschriftarten zu verwenden:

```
fl.getDocumentDOM().selection[0].useDeviceFonts = true;
```

text.variableName

Verfügbarkeit

Flash MX 2004.

Verwendung

```
text.variableName
```

Beschreibung

Eigenschaft; ein String, der den Namen der mit dem Text-Objekt verknüpften Variablen enthält. Diese Eigenschaft ist nur für dynamischen oder Eingabetext vorgesehen. Bei anderen Texttypen ruft sie eine Warnung hervor.

Diese Eigenschaft wird nur in ActionScript 1.0 und ActionScript 2.0 unterstützt.

Beispiel

Im folgenden Beispiel wird der Variablenname des ausgewählten Textfelds auf `firstName` eingestellt:

```
fl.getDocumentDOM().selection[0].variableName = "firstName";
```

Kapitel 43: TextAttrs-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das TextAttrs-Objekt enthält alle Texteigenschaften, die auf eine Unterauswahl angewendet werden können. Es ist eine Eigenschaft des TextRun-Objekts (`textRun.textAttrs`).

Übersicht über Eigenschaften

Für das TextAttrs-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>textAttrs.aliasText</code>	Ein boolescher Wert, der angibt, dass der Text in Flash anhand einer speziellen Methode gezeichnet werden soll, die die Lesbarkeit einer kleineren Schrift steigert.
<code>textAttrs.alignment</code>	Ein String, der die Absatzausrichtung angibt. Die zulässigen Werte lauten "left", "center", "right" und "justify".
<code>textAttrs.autoKern</code>	Ein boolescher Wert, der angibt, ob die in den Schriften gespeicherten Unterscheidungsinformationen für Buchstabenpaare bei der Textunterscheidung verwendet (<code>true</code>) oder ignoriert (<code>false</code>) werden.
<code>textAttrs.bold</code>	Ein boolescher Wert. Wenn der Wert <code>true</code> lautet, wird der Text in der fettgedruckten Version der Schrift angezeigt.
<code>textAttrs.characterPosition</code>	Ein String, der die Grundlinie des Textes festlegt.
<code>textAttrs.characterSpacing</code>	Veraltet für <code>textAttrs.letterSpacing</code> . Eine Ganzzahl, die den Abstand zwischen Zeichen festlegt.
<code>textAttrs.face</code>	Ein String, der den Namen der Schrift angibt, z. B. "Arial".
<code>textAttrs.fillColor</code>	Ein String, ein hexadezimaler Wert oder eine Ganzzahl zur Darstellung der Füllfarbe.
<code>textAttrs.indent</code>	Eine Ganzzahl, die den Absatzeinzug angibt.
<code>textAttrs.italic</code>	Ein boolescher Wert. Wenn der Wert <code>true</code> lautet, wird der Text in der kursiven Version der Schrift angezeigt.
<code>textAttrs.leftMargin</code>	Eine Ganzzahl, die den linken Rand des Absatzes festlegt.
<code>textAttrs.letterSpacing</code>	Eine Ganzzahl, die den Abstand zwischen Zeichen angibt.
<code>textAttrs.lineSpacing</code>	Eine Ganzzahl, die den Zeilenabstand des Absatzes festlegt.
<code>textAttrs.rightMargin</code>	Eine Ganzzahl, die den rechten Rand des Absatzes festlegt.
<code>textAttrs.rotation</code>	Ein boolescher Wert. Wenn der Wert <code>true</code> lautet, werden die Textzeichen um 90° gedreht. Der Standardwert lautet <code>false</code> .
<code>textAttrs.size</code>	Eine Ganzzahl, die die Schriftgröße angibt.
<code>textAttrs.target</code>	Ein String, der die <code>target</code> -Eigenschaft des Textfelds darstellt.
<code>textAttrs.url</code>	Ein String, der die <code>URL</code> -Eigenschaft des Textfelds darstellt.

textAttrs.aliasText

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.aliasText`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, dass der Text in Flash anhand einer speziellen Methode gezeichnet werden soll, die die Lesbarkeit kleiner Textzeichen steigert.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `aliasText` für den gesamten Text in dem gegenwärtig ausgewählten Textfeld auf `true` eingestellt:

```
fl.getDocumentDOM().setElementTextAttr('aliasText', true);
```

textAttrs.alignment

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.alignment`

Beschreibung

Eigenschaft; ein String, der die Absatzausrichtung angibt. Die zulässigen Werte lauten "left", "center", "right" und "justify".

Beispiel

Im folgenden Beispiel werden die Absätze, die Zeichen zwischen Indexposition 0 und (ausschließlich) 3 enthalten, auf "justify" eingestellt. Dies kann sich auf Zeichen außerhalb des angegebenen Bereichs auswirken, sofern sich diese Zeichen in demselben Absatz befinden.

```
fl.getDocumentDOM().setTextSelection(0, 3);  
fl.getDocumentDOM().setElementTextAttr("alignment", "justify");
```

textAttrs.autoKern

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.autoKern`

Beschreibung

Eigenschaft; ein boolescher Wert, der angibt, ob die in den Schriften gespeicherten Unterschneidungsinformationen für Buchstabenpaare zur Textunterschneidung verwendet (`true`) oder ignoriert (`false`) werden.

Beispiel

Im folgenden Beispiel werden die Zeichen von Indexposition 2 bis (ausschließlich) 6 ausgewählt und anschließend wird der `autoKern`-Eigenschaft der Wert `true` zugewiesen:

```
fl.getDocumentDOM().setTextSelection(3, 6);  
fl.getDocumentDOM().setElementTextAttr('autoKern', true);
```

textAttrs.bold

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.bold`

Beschreibung

Eigenschaft; ein boolescher Wert. Wenn der Wert `true` lautet, wird der Text in der fettgedruckten Version der Schrift angezeigt.

Beispiel

Im folgenden Beispiel wird das erste Zeichen des angegebenen Textobjekts ausgewählt und der `bold`-Eigenschaft der Wert `true` zugewiesen:

```
fl.getDocumentDOM().setTextSelection(0, 1);  
fl.getDocumentDOM().setElementTextAttr('bold', true);
```

textAttrs.characterPosition

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.characterPosition`

Beschreibung

Eigenschaft; ein String, der die Grundlinie des Textes festlegt. Die zulässigen Werte lauten `"normal"`, `"subscript"` und `"superscript"`. Diese Eigenschaft gilt nur für statischen Text.

Beispiel

Im folgenden Beispiel werden die Zeichen von Indexposition 2 bis (ausschließlich) Indexposition 6 des angegebenen Textfelds ausgewählt und dann wird der Eigenschaft `characterPosition` der Wert `subscript` zugewiesen:

```
fl.getDocumentDOM().setTextSelection(2, 6);  
fl.getDocumentDOM().setElementTextAttr("characterPosition", "subscript");
```

textAttrs.characterSpacing

Verfügbarkeit

Flash MX 2004. In Flash 8 veraltet für [textAttrs.letterSpacing](#).

Verwendung

`textAttrs.characterSpacing`

Beschreibung

Eigenschaft; eine Ganzzahl, die den Abstand zwischen Zeichen angibt. Die zulässigen Werte liegen zwischen -60 und 60.

Diese Eigenschaft gilt nur für statischen Text. Bei anderen Texttypen wird eine Warnung angezeigt.

Beispiel

Im folgenden Beispiel wird der Zeichenabstand des ausgewählten Textfelds auf 10 eingestellt:

```
fl.getDocumentDOM().setElementTextAttr("characterSpacing", 10);
```

textAttrs.face

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.face`

Beschreibung

Eigenschaft; ein String, der den Namen der Schrift angibt, z. B. "Arial".

Beispiel

Im folgenden Beispiel wird dem ausgewählten Textfeld von dem Zeichen an Indexposition 2 bis (ausschließlich) zu dem Zeichen an Indexposition 8 die Schrift `Arial` zugewiesen:

```
fl.getDocumentDOM().selection[0].setTextAttr("face", "Arial", 2, 8);
```

textAttrs.fillColor

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.fillColor`

Beschreibung

Eigenschaft; die Farbe der Füllung. Sie wird in einem der folgenden Formate angegeben:

- als String im Format "#RRGGBB" oder "#RRGGBBAA"
- als Hexadezimalzahl im Format "0xRRGGBB"
- als Ganzzahl, die die dezimale Entsprechung einer hexadezimalen Zahl darstellt

Beispiel

Im folgenden Beispiel wird dem ausgewählten Textfeld von dem Zeichen an Indexposition 2 bis (ausschließlich) zu dem Zeichen an Indexposition 8 eine rote Farbe zugewiesen:

```
fl.getDocumentDOM().selection[0].setTextAttr("fillColor", 0xff0000, 2, 8);
```

textAttrs.indent

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.indent`

Beschreibung

Eigenschaft; eine Ganzzahl, die den Absatzeinzug angibt. Die zulässigen Werte liegen zwischen -720 und 720.

Beispiel

Im folgenden Beispiel wird dem ausgewählten Textfeld von dem Zeichen an Indexposition 2 bis (ausschließlich) zu dem Zeichen an Indexposition 8 eine Einrückung mit dem Wert 100 zugewiesen. Dies kann sich auf Zeichen außerhalb des angegebenen Bereichs auswirken, sofern sich diese Zeichen in demselben Absatz befinden.

```
fl.getDocumentDOM().selection[0].setTextAttr("indent", 100, 2, 8);
```

textAttrs.italic

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.italic`

Beschreibung

Eigenschaft; ein boolescher Wert. Wenn der Wert `true` lautet, wird der Text in der kursiven Version der Schrift angezeigt.

Beispiel

Im folgenden Beispiel wird dem ausgewählten Textfeld Kursivdruck zugewiesen:


```
fl.getDocumentDOM().selection[0].setTextAttr("italic", true);
```

textAttrs.leftMargin

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.leftMargin`

Beschreibung

Eigenschaft; eine Ganzzahl, die den linken Rand des Absatzes festlegt. Die zulässigen Werte liegen zwischen 0 und 720.

Beispiel

Im folgenden Beispiel wird dem ausgewählten Textfeld von dem Zeichen an Indexposition 2 bis (ausschließlich) zu dem Zeichen an Indexposition 8 die Eigenschaft `leftMargin` mit dem Wert 100 zugewiesen. Dies kann sich auf Zeichen außerhalb des angegebenen Bereichs auswirken, sofern sich diese Zeichen in demselben Absatz befinden.

```
fl.getDocumentDOM().selection[0].setTextAttr("leftMargin", 100, 2, 8);
```

textAttrs.letterSpacing

Verfügbarkeit

Flash 8.

Verwendung

`textAttrs.letterSpacing`

Beschreibung

Eigenschaft; eine Ganzzahl, die den Abstand zwischen Zeichen angibt. Die zulässigen Werte liegen zwischen -60 und 60.

Diese Eigenschaft gilt nur für statischen Text. Bei anderen Texttypen wird eine Warnung angezeigt.

Beispiel

Im folgenden Beispiel wird dem ausgewählten Textfeld von dem Zeichen an Indexposition 0 bis (ausschließlich) zu dem Zeichen an Indexposition 10 ein Zeichenabstand mit dem Wert 60 zugewiesen:

```
fl.getDocumentDOM().setTextSelection(0, 10);  
fl.getDocumentDOM().setElementTextAttr("letterSpacing", 60);
```

textAttrs.lineSpacing

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.lineSpacing`

Beschreibung

Eigenschaft; eine Ganzzahl, die den Zeilenabstand des Absatzes festlegt. Die zulässigen Werte liegen zwischen -360 und 720.

Beispiel

Im folgenden Beispiel wird der Eigenschaft `lineSpacing` des ausgewählten Textfelds der Wert 100 zugewiesen:

```
fl.getDocumentDOM().selection[0].setTextAttr("lineSpacing", 100);
```

textAttrs.rightMargin

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.rightMargin`

Beschreibung

Eigenschaft; eine Ganzzahl, die den rechten Rand des Absatzes festlegt. Die zulässigen Werte liegen zwischen 0 und 720.

Beispiel

Im folgenden Beispiel wird dem ausgewählten Textfeld von dem Zeichen an Indexposition 2 bis (ausschließlich) zu dem Zeichen an Indexposition 8 die Eigenschaft `rightMargin` mit dem Wert 100 zugewiesen. Dies kann sich auf Zeichen außerhalb des angegebenen Bereichs auswirken, sofern sich diese Zeichen in demselben Absatz befinden.

```
fl.getDocumentDOM().selection[0].setTextAttr("rightMargin", 100, 2, 8);
```

textAttrs.rotation

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.rotation`

Beschreibung

Eigenschaft; ein boolescher Wert. Wenn der Wert `true` lautet, werden die Textzeichen um 90° gedreht. Der Standardwert lautet `false`. Diese Eigenschaft gilt nur für statischen Text mit einer vertikalen Ausrichtung. Bei anderen Texttypen wird eine Warnung angezeigt.

Beispiel

Im folgenden Beispiel wird die Drehung des ausgewählten Textfelds auf `true` eingestellt:

```
fl.getDocumentDOM().setElementTextAttr("rotation", true);
```

textAttrs.size

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.size`

Beschreibung

Eigenschaft; eine Ganzzahl, die die Schriftgröße angibt.

Beispiel

Im folgenden Beispiel wird die Größe des Zeichens an Indexposition 2 abgerufen und das Ergebnis im Bedienfeld „Ausgabe“ angezeigt:

```
fl.outputPanel.trace(fl.getDocumentDOM().selection[0].getTextAttr("size", 2));
```

textAttrs.target

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.target`

Beschreibung

Eigenschaft; ein String, der die Eigenschaft `target` des Textfelds darstellt. Diese Eigenschaft funktioniert nur bei statischem Text.

Beispiel

Im folgenden Beispiel wird die Eigenschaft `target` des Textfelds aus dem ersten Bild in der obersten Ebene der aktuellen Szene abgerufen und im Bedienfeld „Ausgabe“ angezeigt:

```
fl.outputPanel.trace(fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].getTextAttr("target"));
```

textAttrs.url

Verfügbarkeit

Flash MX 2004.

Verwendung

`textAttrs.url`

Beschreibung

Eigenschaft; ein String, der die Eigenschaft `URL` des Textfelds darstellt. Diese Eigenschaft funktioniert nur bei statischem Text.

Beispiel

Im folgenden Beispiel wird dem ausgewählten Textfeld die folgende URL zugewiesen: `http://www.adobe.com`:

```
fl.getDocumentDOM().setElementTextAttr("url", "http://www.adobe.com");
```

Kapitel 44: TextRun-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das TextRun-Objekt stellt eine Zeichenreihe dar, deren Attribute mit allen Eigenschaften für das [TextAttrs-Objekt](#) übereinstimmen. Dieses Objekt ist eine Eigenschaft des Text-Objekts ([text.textRuns](#)).

Übersicht über Eigenschaften

Abgesehen von den Eigenschaften des Text-Objekts stehen für das TextRun-Objekt die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
textRun.characters	Ein String, der den im TextRun-Objekt enthaltenen Text darstellt.
textRun.textAttrs	Das TextAttrs-Objekt mit den Attributen der Zeichenreihe.

textRun.textAttrs

Verfügbarkeit

Flash MX 2004.

Verwendung

```
textRun.textAttrs
```

Beschreibung

Eigenschaft; das [TextAttrs-Objekt](#) mit den Attributen der Zeichenreihe.

Beispiel

Im folgenden Beispiel werden die Eigenschaften der ersten Zeichenreihe des ausgewählten Textfelds im Bedienfeld „Ausgabe“ angezeigt:

```
var curTextAttrs = fl.getDocumentDOM().selection[0].textRuns[0].textAttrs;
for (var prop in curTextAttrs) {
    fl.trace(prop + " = " + curTextAttrs[prop]);
}
```

textRun.characters

Verfügbarkeit

Flash MX 2004.

Verwendung

`textRun.characters`

Beschreibung

Eigenschaft; der in dem TextRun-Objekt enthaltene Text.

Beispiel

Im folgenden Beispiel werden die Zeichen, aus denen sich die erste Zeichenreihe des ausgewählten Textfelds zusammensetzt, im Bedienfeld „Ausgabe“ angezeigt:

```
fl.trace(fl.getDocumentDOM().selection[0].textRuns[0].characters);
```

Kapitel 45: Timeline-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Timeline-Objekt entspricht der Flash-Zeitleiste. Sie können über `fl.getDocumentDOM().getTimeline()` auf die Flash-Zeitleiste des aktuellen Dokuments zugreifen. Diese Methode gibt die Zeitleiste der aktuellen Szene oder des zurzeit bearbeiteten Symbols zurück.

Wenn Sie mit Szenen arbeiten, hat die Zeitleiste jeder Szene einen Indexwert. Sie können mit `fl.getDocumentDOM().timelines[i]` auf die Zeitleiste einer Szene des aktuellen Dokuments zugreifen. (In diesem Beispiel ist `i` der Index des Zeitleistenwerts.)

Wenn Sie die Methoden und Eigenschaften des Timeline-Objekts bei Bildern einsetzen, sollten Sie daran denken, dass dieser Bildwert einen auf Null basierenden Index darstellt und nicht die tatsächliche Bildnummer in der Bildsequenz der Zeitleiste ist. Dies bedeutet, dass das erste Bild den Bildindex 0 hat.

Übersicht über Methoden

Für das Timeline-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>timeline.addMotionGuide()</code>	Fügt über der aktuellen Ebene eine Pfadebene hinzu und verknüpft diese beiden Ebenen miteinander.
<code>timeline.addNewLayer()</code>	Fügt dem Dokument eine neue Ebene hinzu und wählt diese Ebene aus.
<code>timeline.clearFrames()</code>	Löscht auf der aktuellen Ebene den gesamten Inhalt eines Bilds oder Bildbereichs.
<code>timeline.clearKeyframes()</code>	Konvertiert auf der aktuellen Ebene ein Schlüsselbild in ein normales Bild und löscht seinen Inhalt.
<code>timeline.convertToBlankKeyframes()</code>	Konvertiert auf der aktuellen Ebene Bilder in leere Schlüsselbilder.
<code>timeline.convertToKeyframes()</code>	Konvertiert auf der aktuellen Ebene einen Bildbereich in Schlüsselbilder (oder konvertiert die Auswahl, wenn keine Bilder angegeben wurden).
<code>timeline.copyFrames()</code>	Kopiert einen Bildbereich aus der aktuellen Ebene in die Zwischenablage.
<code>timeline.copyLayers()</code>	Kopiert einen Bereich mit Zeitleistenebenen in die Zwischenablage.
<code>timeline.copyMotion()</code>	Kopiert Bewegung auf ausgewählte Bilder, entweder von einem Bewegungs-Tween oder von einer Einzelbildanimation, sodass die Bewegung auf andere Bilder angewendet werden kann.
<code>timeline.copyMotionAsAS3()</code>	Kopiert Bewegung auf ausgewählten Bildern – entweder von einem Bewegungs-Tween oder von einer Einzelbildanimation – als ActionScript 3.0-Code in die Zwischenablage.
<code>timeline.createMotionObject()</code>	Erstellt ein neues Bewegungsobjekt an einem angegebenen Start- und Endbild.

Methode	Beschreibung
<code>timeline.createMotionTween()</code>	Stellt die Eigenschaft <code>frame.tweenType</code> für alle auf der aktuellen Ebene ausgewählten Schlüsselbilder auf <code>motion</code> ein und konvertiert ggf. den Inhalt sämtlicher Bilder in eine einzelne Symbolinstanz.
<code>timeline.cutFrames()</code>	Schneidet auf der aktuellen Ebene einen Bildbereich aus der Zeitleiste aus und speichert ihn in der Zwischenablage.
<code>timeline.cutLayers()</code>	Schneidet einen Bereich mit Zeitleistenebenen aus und speichert ihn in der Zwischenablage.
<code>timeline.deleteLayer()</code>	Löscht eine Ebene.
<code>timeline.duplicateLayers()</code>	Dupliziert die ausgewählten oder angegebenen Ebenen.
<code>timeline.expandFolder()</code>	Blendet die Inhalte der angegebenen Ordner ein oder aus.
<code>timeline.findLayerIndex()</code>	Sucht für die Ebenen mit den angegebenen Namen ein Indexarray.
<code>timeline.getFrameProperty()</code>	Ruft den Wert der angegebenen Eigenschaft für die ausgewählten Bilder ab.
<code>timeline.getGuidelines()</code>	Gibt einen XML-String zurück, der die gegenwärtigen Positionen der horizontalen und vertikalen Hilfslinien einer Zeitleiste darstellen („Ansicht“ > „Hilfslinien“ > „Hilfslinien anzeigen“).
<code>timeline.getLayerProperty()</code>	Ruft den Wert der angegebenen Eigenschaft für die ausgewählten Ebenen ab.
<code>timeline.getSelectedFrames()</code>	Ruft ein Array der gegenwärtig ausgewählten Bilder ab.
<code>timeline.getSelectedLayers()</code>	Ruft die auf Null basierenden Indexwerte der ausgewählten Ebenen ab.
<code>timeline.insertBlankKeyframe()</code>	Fügt an der angegebenen Bildindexposition ein leeres Schlüsselbild ein. Wurde keine Indexposition angegeben, wird das leere Schlüsselbild anhand des Abspielkopfes bzw. anhand der Auswahl eingefügt.
<code>timeline.insertFrames()</code>	Fügt die angegebene Anzahl von Bildern an der angegebenen Bildnummer.
<code>timeline.insertKeyframe()</code>	Fügt an der angegebenen Bildposition ein Schlüsselbild ein.
<code>timeline.pasteFrames()</code>	Fügt den Bildbereich aus der Zwischenablage in die angegebenen Bilder ein.
<code>timeline.pasteLayers()</code>	Fügt kopierte Ebenen oberhalb des angegebenen Ebenenindex in die Zeitleiste ein.
<code>timeline.pasteMotion()</code>	Fügt den Bereich der durch <code>timeline.copyMotion()</code> abgerufenen Bewegungsbilder in die Zeitleiste ein.
<code>timeline.removeFrames()</code>	Löscht das Bild.
<code>timeline.removeMotionObject()</code>	Entfernt das mit <code>timeline.createMotionObject()</code> erstellte Bewegungsobjekt und konvertiert die Bilder in statische Bilder.
<code>timeline.reorderLayer()</code>	Verschiebt die erste angegebene Ebene über oder unter die zweite angegebene Ebene.
<code>timeline.reverseFrames()</code>	Kehrt einen Bildbereich um.
<code>timeline.selectAllFrames()</code>	Wählt alle Bilder in der aktuellen Zeitleiste aus.
<code>timeline setFrameProperty()</code>	Legt die Eigenschaft des Frame-Objekts für die ausgewählten Bilder fest.

Methode	Beschreibung
<code>timeline.setGuidelines()</code>	Ersetzt die Hilfslinien für die Zeitleiste durch die angegebenen Informationen.
<code>timeline.setLayerProperty()</code>	Weist der angegebenen Eigenschaft auf allen ausgewählten Ebenen einen bestimmten Wert zu.
<code>timeline.setSelectedFrames()</code>	Wählt einen Bildbereich in der aktuellen Ebene aus oder stellt die ausgewählten Bilder auf das an diese Methode übergebene Auswahl-Array ein.
<code>timeline.setSelectedLayers()</code>	Legt die auszuwählende Ebene fest. Legt außerdem fest, dass die angegebene Ebene die aktuelle Ebene sein soll.
<code>timeline.showLayerMasking()</code>	Sperrt die Maskenebenen und die maskierten Ebenen und zeigt so die Ebenenmaskierung während des Authorings an.
<code>timeline.startPlayback()</code>	Startet die automatische Wiedergabe der Zeitleiste, wenn sie derzeit nicht abgespielt wird.
<code>timeline.stopPlayback()</code>	Stoppt die automatische Wiedergabe der Zeitleiste, wenn sie derzeit abgespielt wird.

Übersicht über Eigenschaften

Für das Timeline-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>timeline.currentFrame</code>	Ein auf Null basierender Index für das Bild an der aktuellen Abspielkopfposition.
<code>timeline.currentLayer</code>	Ein auf Null basierender Index für die gegenwärtig aktive Ebene.
<code>timeline.frameCount</code>	Schreibgeschützt; eine Ganzzahl, die angibt, wie viele Bilder die längste Ebene dieser Zeitleiste enthält.
<code>timeline.layerCount</code>	Schreibgeschützt; eine Ganzzahl, die die Anzahl der Ebenen in der angegebenen Zeitleiste angibt.
<code>timeline.layers</code>	Schreibgeschützt; ein Array von Layer-Objekten.
<code>timeline.libraryItem</code>	Schreibgeschützte Eigenschaft; gibt an, ob die Zeitleiste zu einer Szene gehört.
<code>timeline.name</code>	Ein String, der den Namen der aktuellen Zeitleiste angibt.

timeline.addMotionGuide()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.addMotionGuide()
```

Parameter

Keine.

Rückgabewerte

Eine Ganzzahl, die die Indexposition (auf Null basierend) der neu hinzugefügten Führungsebene angibt. Wenn die aktuelle Ebene nicht dem Typ "Normal" angehört, gibt Flash -1 zurück.

Beschreibung

Methode; fügt über der aktuellen Ebene eine Pfadebene hinzu und verknüpft diese beiden Ebenen miteinander. Dabei wird die aktuelle Ebene in eine Ebene des Typs "Guided" konvertiert.

Diese Methode ist ausschließlich für Ebenen des Typs "Normal" vorgesehen. Sie hat keine Auswirkung auf eine Ebene des Typs "Folder", "Mask", "Masked", "Guide" oder "Guided".

Beispiel

Im folgenden Beispiel wird über der aktuellen Ebene eine Pfadebene hinzugefügt und die aktuelle Ebene in Guided konvertiert:

```
fl.getDocumentDOM().getTimeline().addMotionGuide();
```

timeline.addNewLayer()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.addNewLayer([name] [, layerType [, bAddAbove]])
```

Parameter

name Ein String, der den Namen der neuen Ebene angibt. Wenn dieser Parameter fehlt, wird der neuen Ebene ein Standardname zugewiesen („Ebene n“ wobei *n* der Gesamtanzahl der Ebenen entspricht). Dieser Parameter ist optional.

layerType Ein String, der den Typ der hinzuzufügenden Ebene angibt. Wenn dieser Parameter fehlt, wird eine Ebene des Typs „Normal“ erstellt. Dieser Parameter ist optional. Die zulässigen Werte lauten "normal", "guide", "guided", "mask", "masked" und "folder".

bAddAbove Ein boolescher Wert; bei dem Wert `true` (Standardwert), wird die neue Ebene über der aktuellen Ebene eingefügt und bei `false` darunter. Dieser Parameter ist optional.

Rückgabewerte

Eine Ganzzahl, die den auf Null basierenden Index der neu hinzugefügten Ebene angibt.

Beschreibung

Methode; fügt dem Dokument eine neue Ebene hinzu und wählt diese Ebene aus.

Beispiel

Im folgenden Beispiel wird der Zeitleiste eine neue Ebene mit einem von Flash generierten Standardnamen hinzugefügt:

```
fl.getDocumentDOM().getTimeline().addNewLayer();
```

Im folgenden Beispiel wird eine neue Ordnerebene über der aktuellen Ebene hinzugefügt und der neuen Ebene der Name `Folder1` zugewiesen:

```
fl.getDocumentDOM().getTimeline().addNewLayer("Folder1", "folder", true);
```

timeline.clearFrames()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.clearFrames([startFrameIndex [, endFrameIndex]])
```

Parameter

startFrameIndex Ein auf Null basierender Index, der den Anfang des zu löschenden Bildbereichs angibt. Wenn Sie *startFrameIndex* auslassen, verwendet die Methode die aktuelle Auswahl. Dieser Parameter ist optional.

endFrameIndex Ein auf Null basierender Index, der das Ende des zu löschenden Bildbereichs angibt. Der Bereich endet bei *endFrameIndex*, schließt das Bild an dieser Indexposition aber nicht ein. Wenn Sie nur *startFrameIndex* angeben, erhält *endFrameIndex* automatisch den gleichen Wert wie *startFrameIndex*. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; löscht auf der aktuellen Ebene den gesamten Inhalt eines Bilds oder Bildbereichs.

Beispiel

Im folgenden Beispiel werden alle Bilder von Bild 6 bis (ausschließlich) Bild 11 gelöscht (denken Sie daran, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().clearFrames(5, 10);
```

Im folgenden Beispiel wird Bild 15 gelöscht:

```
fl.getDocumentDOM().getTimeline().clearFrames(14);
```

timeline.clearKeyframes()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.clearKeyframes([startFrameIndex [, endFrameIndex]])
```

Parameter

startFrameIndex Ein auf Null basierender Index, der den Anfang des zu löschenden Bildbereichs angibt. Wenn Sie *startFrameIndex* auslassen, verwendet die Methode die aktuelle Auswahl. Dieser Parameter ist optional.

endFrameIndex Ein auf Null basierender Index, der das Ende des zu löschenden Bildbereichs angibt. Der Bereich endet bei *endFrameIndex*, schließt das Bild an dieser Indexposition aber nicht ein. Wenn Sie nur *startFrameIndex* angeben, erhält *endFrameIndex* automatisch den gleichen Wert wie *startFrameIndex*. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; konvertiert auf der aktuellen Ebene ein Schlüsselbild in ein normales Bild und löscht seinen Inhalt.

Beispiel

Im folgenden Beispiel werden alle Schlüsselbilder von Bild 5 bis (ausschließlich) Bild 10 gelöscht (denken Sie daran, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().clearKeyframes(4, 9);
```

Im folgenden Beispiel wird das Schlüsselbild in Bild 15 gelöscht und in ein normales Bild konvertiert:

```
fl.getDocumentDOM().getTimeline().clearKeyframes(14);
```

timeline.convertToBlankKeyframes()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.convertToBlankKeyframes([startFrameIndex [, endFrameIndex]])
```

Parameter

startFrameIndex Ein auf Null basierender Index, der das erste der in Schlüsselbilder zu konvertierenden Bilder angibt. Wenn Sie *startFrameIndex* auslassen, konvertiert die Methode die aktuell ausgewählten Bilder. Dieser Parameter ist optional.

endFrameIndex Ein auf Null basierender Index, der angibt, bis zu welchem Bild die Konvertierung in Schlüsselbilder durchgeführt werden soll. Der Bereich der zu konvertierenden Bilder reicht bis *endFrameIndex*, schließt das Bild an dieser Position aber nicht ein. Wenn Sie nur *startFrameIndex* angeben, erhält *endFrameIndex* automatisch den gleichen Wert wie *startFrameIndex*. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; konvertiert auf der aktuellen Ebene Bilder in leere Schlüsselbilder.

Beispiel

Im folgenden Beispiel werden alle Bilder von Bild 2 bis (ausschließlich) Bild 10 in leere Schlüsselbilder konvertiert (denken Sie daran, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().convertToBlankKeyframes(1, 9);
```

Im folgenden Beispiel wird Bild 5 in ein leeres Schlüsselbild konvertiert:

```
fl.getDocumentDOM().getTimeline().convertToBlankKeyframes(4);
```

timeline.convertToKeyframes()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.convertToKeyframes([startFrameIndex [, endFrameIndex]])
```

Parameter

startFrameIndex Ein auf Null basierender Index, der das erste der in Schlüsselbilder zu konvertierenden Bilder angibt. Wenn Sie *startFrameIndex* auslassen, konvertiert die Methode die aktuell ausgewählten Bilder. Dieser Parameter ist optional.

endFrameIndex Ein auf Null basierender Index, der angibt, bis zu welchem Bild die Konvertierung in Schlüsselbilder durchgeführt werden soll. Der Bereich der zu konvertierenden Bilder reicht bis *endFrameIndex*, schließt das Bild an dieser Position aber nicht ein. Wenn Sie nur *startFrameIndex* angeben, erhält *endFrameIndex* automatisch den gleichen Wert wie *startFrameIndex*. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; konvertiert auf der aktuellen Ebene einen Bildbereich in Schlüsselbilder (oder konvertiert die Auswahl, wenn keine Bilder angegeben wurden).

Beispiel

Im folgenden Beispiel werden die ausgewählten Bilder in Schlüsselbilder konvertiert:

```
fl.getDocumentDOM().getTimeline().convertToKeyframes();
```

Im folgenden Beispiel werden alle Bilder von Bild 2 bis (ausschließlich) Bild 10 in Schlüsselbilder konvertiert (denken Sie daran, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().convertToKeyframes(1, 9);
```

Im folgenden Beispiel wird Bild 5 in ein Schlüsselbild konvertiert:

```
fl.getDocumentDOM().getTimeline().convertToKeyframes(4);
```

timeline.copyFrames()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.copyFrames([startFrameIndex [, endFrameIndex]])
```

Parameter

startFrameIndex Ein auf Null basierender Index, der den Anfang des zu kopierenden Bildbereichs angibt. Wenn Sie *startFrameIndex* auslassen, verwendet die Methode die aktuelle Auswahl. Dieser Parameter ist optional.

endFrameIndex Ein auf Null basierender Index, der angibt, bis zu welchem Bild der Kopiervorgang durchgeführt werden soll. Der Bereich der zu kopierenden Bilder reicht bis *endFrameIndex*, schließt das Bild an dieser Position aber nicht ein. Wenn Sie nur *startFrameIndex* angeben, erhält *endFrameIndex* automatisch den gleichen Wert wie *startFrameIndex*. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; kopiert einen Bildbereich aus der aktuellen Ebene in die Zwischenablage.

Beispiel

Im folgenden Beispiel werden die ausgewählten Bilder in die Zwischenablage kopiert:

```
fl.getDocumentDOM().getTimeline().copyFrames();
```

Im folgenden Beispiel werden alle Bilder von Bild 2 bis (ausschließlich) Bild 10 in die Zwischenablage kopiert (denken Sie daran, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().copyFrames(1, 9);
```

Im folgenden Beispiel wird Bild 5 in die Zwischenablage kopiert:

```
fl.getDocumentDOM().getTimeline().copyFrames(4);
```

timeline.copyLayers()

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

```
timeline.copyLayers([startLayerIndex [, endLayerIndex]])
```

Parameter

startLayerIndex Optional. Ein auf Null basierender Index, der den Anfang des zu kopierenden Ebenenbereichs angibt. Wenn Sie *startLayerIndex* auslassen, verwendet die Methode die aktuelle Auswahl.

endLayerIndex Optional. Ein auf Null basierender Index, der angibt, bis zu welcher Ebene der Kopiervorgang durchgeführt werden soll. Der Bereich der zu kopierenden Ebenen reicht bis einschließlich *endLayerIndex*. Wenn Sie nur *startLayerIndex* angeben, wird *endLayerIndex* standardmäßig auf den Wert *startLayerIndex* eingestellt.

Rückgabewerte

Keine.

Beschreibung

Methode; kopiert die derzeit in der Zeitleiste ausgewählten Ebenen oder die Ebenen im angegebenen Bereich. Über optionale Argumente können Sie angeben, welche Ebene oder welcher Ebenenbereich kopiert werden soll.

Beispiel

Im folgenden Beispiel werden die Ebenen von Index 2 bis Index 7 in der Zeitleiste kopiert:

```
fl.getDocumentDOM().getTimeline().copyLayers(2, 7);
```

Siehe auch

[timeline.cutLayers\(\)](#), [timeline.pasteLayers\(\)](#), [timeline.duplicateLayers\(\)](#)

timeline.copyMotion()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
timeline.copyMotion()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; kopiert Bewegung auf ausgewählte Bilder, entweder von einem Bewegungs-Tween oder von einer Einzelbildanimation. Anschließend können Sie die Bewegung mithilfe von [timeline.pasteMotion\(\)](#) auf andere Bilder übertragen.

Um Bewegung als Text(code) zu kopieren, der in ein Skript eingefügt werden kann, informieren Sie sich unter [timeline.copyMotionAsAS3\(\)](#).

Beispiel

Im folgenden Beispiel wird die Bewegung vom ausgewählten Bild bzw. von den ausgewählten Bildern kopiert:

```
fl.getDocumentDOM().getTimeline().copyMotion();
```

Siehe auch

[timeline.copyMotionAsAS3\(\)](#), [timeline.pasteMotion\(\)](#)

timeline.copyMotionAsAS3()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
timeline.copyMotionAsAS3()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; kopiert Bewegung auf ausgewählten Bildern – entweder von einem Bewegungs-Tween oder von einer Einzelbildanimation – als ActionScript 3.0-Code in die Zwischenablage. Anschließend können Sie den Code in ein Skript einfügen.

Um Bewegung in ein Format zu kopieren, das Sie auf andere Bilder übertragen können, informieren Sie sich unter [timeline.copyMotion\(\)](#).

Beispiel

Im folgenden Beispiel wird die Bewegung vom ausgewählten Bild bzw. von den ausgewählten Bildern als ActionScript 3.0-Code in die Zwischenablage kopiert:

```
fl.getDocumentDOM().getTimeline().copyMotionAsAS3();
```

Siehe auch

[timeline.copyMotion\(\)](#)

timeline.createMotionObject()

Verfügbarkeit

Flash Professional CS5.

Verwendung

```
timeline.createMotionObject([startFrame [,endFrame]])
```

Parameter

startFrame Gibt das erste Bild an, an dem Bewegungsobjekte erstellt werden sollen. Wenn Sie *startFrame* weglassen, verwendet die Methode die aktuelle Auswahl. Ist keine Auswahl vorhanden, werden alle Bilder am aktuellen Abspielkopf auf allen Ebenen entfernt. Dieser Parameter ist optional.

endFrame Gibt das Bild an, an dem die Erstellung von Bewegungsobjekten beendet werden soll. Der Bildbereich reicht bis zu *endFrame*, schließt dieses Bild aber nicht ein. Wenn Sie nur *startFrame* angeben, ist *endFrame* standardmäßig auf den Wert von *startFrame* eingestellt. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; erstellt ein neues Bewegungsobjekt. Die Parameter sind optional. Wenn die Parameter angegeben werden, wird die Auswahl der Zeitleiste auf die angegebenen Bilder eingestellt, bevor das Bewegungsobjekt erstellt wird.

Beispiel

Mit dem folgenden Beispiel wird ein Bewegungsobjekt an der aktuellen Position des Abspielkopfs auf der obersten Ebene erstellt:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().createMotionObject();
```

Mit dem folgenden Beispiel wird auf der obersten Ebene in der aktuellen Szene ein Bewegungsobjekt erstellt, das bei Bild 5 beginnt und bis zu Bild 15 reicht (dieses jedoch nicht mehr einschließt):

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().createMotionObject(5, 15);
```

timeline.createMotionTween()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.createMotionTween([startFrameIndex [, endFrameIndex]])
```

Parameter

startFrameIndex Ein auf Null basierender Index, der angibt, bei welchem Bild ein Bewegungs-Tween beginnen soll. Wenn Sie *startFrameIndex* auslassen, verwendet die Methode die aktuelle Auswahl. Dieser Parameter ist optional.

endFrameIndex Ein auf Null basierender Index, der angibt, bis zu welchem Bild der Bewegungs-Tween durchgeführt werden soll. Der Bildbereich reicht bis *endFrameIndex*, schließt das Bild an dieser Position aber nicht ein. Wenn Sie nur *startFrameIndex* angeben, erhält *endFrameIndex* automatisch den gleichen Wert wie *startFrameIndex*. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; stellt die Eigenschaft `frame.tweenType` für alle auf der aktuellen Ebene ausgewählten Schlüsselbilder auf `motion` ein und konvertiert ggf. den Inhalt sämtlicher Bilder in eine einzelne Symbolinstanz. Diese Eigenschaft entspricht dem Menübefehl „Bewegungs-Tween erstellen“ im Flash-Authoring-Tool.

Beispiel

Im folgenden Beispiel wird die Form in einem Bildbereich, der von Bild 1 bis (ausschließlich) Bild 10 reicht, in eine Grafiksymbollinstanz konvertiert und der Eigenschaft `frame.tweenType` der Wert `motion` zugewiesen (denken Sie daran, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().createMotionTween(0, 9);
```

timeline.currentFrame

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.currentFrame
```

Beschreibung

Eigenschaft; ein auf Null basierender Index für das Bild an der aktuellen Abspielkopfposition.

Beispiel

Im folgenden Beispiel wird der Abspielkopf der aktuellen Zeitleiste an Bild 10 gesetzt (beachten Sie hierbei, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().currentFrame = 9;
```

Im folgenden Beispiel wird der Wert der aktuellen Abspielkopfposition in der Variablen `curFrame` gespeichert:

```
var curFrame = fl.getDocumentDOM().getTimeline().currentFrame;
```

timeline.currentLayer

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.currentLayer
```

Beschreibung

Eigenschaft; ein auf Null basierender Index für die gegenwärtig aktive Ebene. Der Wert 0 bezieht sich auf die oberste Ebene, der Wert 1 auf die direkt darunter liegende Ebene usw.

Beispiel

Im folgenden Beispiel wird die oberste Ebene aktiviert:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
```

Im folgenden Beispiel wird der Index der gegenwärtig aktiven Ebene in der Variablen `curLayer` gespeichert:

```
var curLayer = fl.getDocumentDOM().getTimeline().currentLayer;
```

timeline.cutFrames()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.cutFrames([startFrameIndex [, endFrameIndex]])
```

Parameter

startFrameIndex Ein auf Null basierender Index, der den Anfang des auszuschneidenden Bildbereichs angibt. Wenn Sie *startFrameIndex* auslassen, verwendet die Methode die aktuelle Auswahl. Dieser Parameter ist optional.

endFrameIndex Ein auf Null basierender Index, der das Bild angibt, bei dem der Ausschneidevorgang endet. Der Bildbereich reicht bis *endFrameIndex*, schließt das Bild an dieser Position aber nicht ein. Wenn Sie nur *startFrameIndex* angeben, erhält *endFrameIndex* automatisch den gleichen Wert wie *startFrameIndex*. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; schneidet auf der aktuellen Ebene einen Bildbereich aus der Zeitleiste aus und speichert ihn in der Zwischenablage.

Beispiel

Im folgenden Beispiel werden die ausgewählten Bilder aus der Zeitleiste ausgeschnitten und in der Zwischenablage gespeichert:

```
fl.getDocumentDOM().getTimeline().cutFrames();
```

Im folgenden Beispiel werden alle Bilder von Bild 2 bis (ausschließlich) Bild 10 ausgeschnitten und in der Zwischenablage gespeichert (denken Sie daran, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().cutFrames(1, 9);
```

Im folgenden Beispiel wird Bild 5 aus der Zeitleiste ausgeschnitten und in der Zwischenablage gespeichert:

```
fl.getDocumentDOM().getTimeline().cutFrames(4);
```

timeline.cutLayers()

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

```
timeline.cutLayers([startLayerIndex [, endLayerIndex]])
```

Parameter

startLayerIndex Optional. Ein auf Null basierender Index, der den Anfang des zu auszuschneidenden Ebenenbereichs angibt. Wenn Sie *startLayerIndex* auslassen, verwendet die Methode die aktuelle Auswahl.

endLayerIndex Optional. Ein auf Null basierender Index, der angibt, bis zu welcher Ebene der Ausschneidevorgang durchgeführt werden soll. Der Bereich der auszuschneidenden Ebenen reicht bis einschließlich *endLayerIndex*. Wenn Sie nur *startLayerIndex* angeben, wird *endLayerIndex* standardmäßig auf den Wert *startLayerIndex* eingestellt.

Rückgabewerte

Keine.

Beschreibung

Methode; schneidet die derzeit in der Zeitleiste ausgewählten Ebenen oder die Ebenen im angegebenen Bereich aus. Über optionale Argumente können Sie angeben, welche Ebene oder welcher Ebenenbereich ausgeschnitten werden soll.

Beispiel

Im folgenden Beispiel werden die Ebenen von Index 2 bis Index 7 in der Zeitleiste ausgeschnitten:

```
fl.getDocumentDOM().getTimeline().cutLayers(2, 7);
```

Siehe auch

[timeline.copyLayers\(\)](#), [timeline.pasteLayers\(\)](#), [timeline.duplicateLayers\(\)](#)

timeline.deleteLayer()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.deleteLayer([index])
```

Parameter

index Ein auf Null basierender Index, der die zu löschende Ebene angibt. Wenn die Zeitleiste nur eine Ebene enthält, bleibt diese Methode wirkungslos. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; löscht eine Ebene. Wenn es sich bei der Ebene um einen Ordner handelt, werden alle Ebenen gelöscht, die sich in diesem Ordner befinden. Wenn Sie keinen Ebenenindex angeben, werden alle ausgewählten Ebenen von Flash gelöscht.

Beispiel

Im folgenden Beispiel wird die zweitoberste Ebene gelöscht:

```
fl.getDocumentDOM().getTimeline().deleteLayer(1);
```

Im folgenden Beispiel werden die ausgewählten Ebenen gelöscht:

```
fl.getDocumentDOM().getTimeline().deleteLayer();
```

timeline duplicateLayers()

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

```
timeline.duplicateLayers([startLayerIndex [, endLayerIndex]])
```

Parameter

startLayerIndex Optional. Ein auf Null basierender Index, der den Anfang des zu kopierenden Ebenenbereichs angibt. Dieser Wert gibt auch die Ebene an, über der die Ebenen in der Zwischenablage eingefügt werden. Wenn Sie **startLayerIndex** auslassen, verwendet die Methode die aktuelle Ebenenauswahl.

endLayerIndex Optional. Ein auf Null basierender Index, der angibt, bis zu welcher Ebene der Kopiervorgang durchgeführt werden soll. Der Bereich der zu kopierenden Ebenen reicht bis einschließlich **endLayerIndex**. Wenn Sie nur **startLayerIndex** angeben, wird **endLayerIndex** standardmäßig auf den Wert **startLayerIndex** eingestellt.

Rückgabewerte

Keine.

Beschreibung

Methode; dupliziert die derzeit in der Zeitleiste ausgewählten Ebenen oder die Ebenen im angegebenen Bereich. Über optionale Argumente können Sie angeben, welche Ebene oder welcher Ebenenbereich dupliziert werden soll.

Beispiel

Im folgenden Beispiel wird die derzeit in der Zeitleiste ausgewählte Ebene dupliziert:

```
fl.getDocumentDOM().getTimeline().duplicateLayers();
```

Im folgenden Beispiel werden die Ebenen von Index 2 bis Index 7 oberhalb der Ebene an Index 2 dupliziert:

```
fl.getDocumentDOM().getTimeline().duplicateLayers(2,7);
```

Siehe auch

[timeline.copyLayers\(\)](#), [timeline.cutLayers\(\)](#), [timeline.pasteLayers\(\)](#)

timeline.expandFolder()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.expandFolder(bExpand [, bRecurseNestedParents [, index]])
```

Parameter

bExpand Ein boolescher Wert; bei dem Wert **true** werden die Ordnerinhalte eingeblendet und bei **false** werden sie ausgeblendet.

bRecurseNestedParents Ein boolescher Wert; bei dem Wert `true` werden alle Ebenen innerhalb des angegebenen Ordners je nach dem Wert des Parameters *bExpand* entweder ein- oder ausgeblendet. Dieser Parameter ist optional.

index Ein auf Null basierender Index des Ordners, dessen Inhalte ein- oder ausgeblendet werden sollen. Wenn der Befehl für alle Ebenen gelten soll, verwenden Sie den Wert `-1` (Sie müssen außerdem *bRecurseNestedParents* auf `true` einstellen). Diese Eigenschaft entspricht den Menübefehlen „Alle Ordnerinhalte einblenden“ und „Alle Ordnerinhalte ausblenden“ im Flash-Authoring-Tool. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; blendet die Inhalte der angegebenen Ordner ein oder aus. Wenn Sie keine Ebene angeben, wirkt die Methode sich auf die aktuelle Ebene aus.

Beispiel

Für die nachfolgenden Beispiele gilt die folgende Ordnerstruktur:

```
Folder 1 ***  
--layer 7  
--Folder 2 ****  
----Layer 5
```

Im folgenden Beispiel werden nur die Inhalte des Ordners 1 eingeblendet:

```
fl.getDocumentDOM().getTimeline().currentLayer = 1;  
fl.getDocumentDOM().getTimeline().expandFolder(true);
```

Im folgenden Beispiel werden nur die Inhalte des Ordners 1 eingeblendet (dies setzt voraus, dass die Inhalte des Ordners 2 ausgeblendet waren, als die Inhalte des Ordners 1 zuletzt ausgeblendet wurden; andernfalls werden die Inhalte des Ordners 2 eingeblendet):

```
fl.getDocumentDOM().getTimeline().expandFolder(true, false, 0);
```

Im folgenden Beispiel werden alle Ordner in der aktuellen Zeitleiste ausgeblendet:

```
fl.getDocumentDOM().getTimeline().expandFolder(false, true, -1);
```

timeline.findLayerIndex()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.findLayerIndex(name)
```

Parameter

name Ein String, der den Namen der zu suchenden Ebene angibt.

Rückgabewerte

Ein Array von Indexwerten für die angegebene Ebene. Wird die angegebene Ebene nicht gefunden, gibt Flash `undefined` zurück.

Beschreibung

Methode; sucht für die Ebenen mit den angegebenen Namen ein Indexarray. Der Ebenenindex ist flach, d. h. alle Ordner gehören dem Hauptindex an.

Beispiel

Im folgenden Beispiel werden die Indexwerte aller Ebenen mit dem Namen `Layer 7` im Bedienfeld „Ausgabe“ angezeigt:

```
var layerIndex = fl.getDocumentDOM().getTimeline().findLayerIndex("Layer 7");  
fl.trace(layerIndex);
```

Im folgenden Beispiel wird demonstriert, wie man die von dieser Methode zurückgegebenen Werte wieder an `timeline.setSelectedLayers()` übergibt:

```
var layerIndex = fl.getDocumentDOM().getTimeline().findLayerIndex("Layer 1");  
fl.getDocumentDOM().getTimeline().setSelectedLayers(layerIndex[0], true);
```

timeline.frameCount

Verfügbarkeit

Flash MX 2004.

Verwendung

`timeline.frameCount`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die angibt, wie viele Bilder die längste Ebene dieser Zeitleiste enthält.

Beispiel

Im folgenden Beispiel wird die Anzahl der Bilder in der längsten Ebene des aktuellen Dokuments in der Variablen `countNum` gespeichert:

```
var countNum = fl.getDocumentDOM().getTimeline().frameCount;
```

timeline.getFrameProperty()

Verfügbarkeit

Flash MX 2004.

Verwendung

`timeline.getFrameProperty(property [, startframeIndex [, endFrameIndex]])`

Parameter

property Ein String, der den Namen der Eigenschaft angibt, für die der Wert abgerufen werden soll. Eine Liste sämtlicher Eigenschaften finden Sie in der Übersicht über die Eigenschaften für das [Frame-Objekt](#).

startFrameIndex Ein auf Null basierender Index, der die Nummer des ersten Bilds angibt, für das der Wert abgerufen werden soll. Wenn Sie *startFrameIndex* auslassen, verwendet die Methode die aktuelle Auswahl. Dieser Parameter ist optional.

endFrameIndex Ein auf Null basierender Index, der das Ende des auszuwählenden Bildbereichs angibt. Der Bereich endet bei *endFrameIndex*, schließt das Bild an dieser Indexposition aber nicht ein. Wenn Sie nur *startFrameIndex* angeben, erhält *endFrameIndex* automatisch den gleichen Wert wie *startFrameIndex*. Dieser Parameter ist optional.

Rückgabewerte

Ein Wert für die angegebene Eigenschaft bzw. *undefined*, wenn die ausgewählten Bilder nicht den gleichen Eigenschaftswert aufweisen.

Beschreibung

Methode; ruft den Wert der angegebenen Eigenschaft für die ausgewählten Bilder ab.

Beispiel

Im folgenden Beispiel wird der Name des ersten Bilds abgerufen, das sich auf der obersten Ebene des aktuellen Dokuments befindet, und im Bedienfeld „Ausgabe“ angezeigt:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().setSelectedFrames(0, 0, true);  
var frameName = fl.getDocumentDOM().getTimeline().getFrameProperty("name");  
fl.trace(frameName);
```

timeline.getGuidelines()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
timeline.getGuidelines()
```

Parameter

Keine.

Rückgabewerte

Ein XML-String.

Beschreibung

Methode; gibt einen XML-String zurück, der die gegenwärtigen Positionen der horizontalen und vertikalen Hilfslinien einer Zeitleiste darstellen („Ansicht“ > „Hilfslinien“ > „Hilfslinien anzeigen“). Mit `timeline.setGuidelines()` können diese Hilfslinien auf eine Zeitleiste angewendet werden.

Beispiel

Sofern Sie über einige Hilfslinien auf der ersten Zeitleiste verfügen, werden sie im folgenden Beispiel im Bedienfeld „Ausgabe“ als XML-String angezeigt:

```
var currentTimeline = fl.getDocumentDOM().timelines[0];  
fl.trace(currentTimeline.getGuidelines());
```


timeline.getLayerProperty()

Verfügbarkeit

Flash MX 2004.

Verwendung

`timeline.getLayerProperty(property)`

Parameter

property Ein String, der den Namen der Eigenschaft angibt, deren Wert abgerufen werden soll. Eine Liste der Eigenschaften finden Sie in der Übersicht über die Eigenschaften für das [Frame-Objekt](#).

Rückgabewerte

Der Wert der angegebenen Eigenschaft. Flash ermittelt den Typ einer Ebene anhand der Ebeneneigenschaften. Wenn die Eigenschaftswerte der angegebenen Ebenen nicht übereinstimmen, gibt Flash den Wert `undefined` zurück.

Beschreibung

Methode; ruft den Wert der angegebenen Eigenschaft für die ausgewählten Ebenen ab.

Beispiel

Im folgenden Beispiel wird der Name der obersten Ebene in dem aktiven Dokument abgerufen und im Bedienfeld „Ausgabe“ angezeigt:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
var layerName = fl.getDocumentDOM().getTimeline().getLayerProperty("name");  
fl.trace(layerName);
```

timeline.getSelectedFrames()

Verfügbarkeit

Flash MX 2004.

Parameter

Keine.

Rückgabewerte

Ein Array mit $3n$ Ganzzahlen, wobei n für die Anzahl der ausgewählten Bereiche steht. Die erste Ganzzahl in einer Gruppe ist der Ebenenindex, die zweite Ganzzahl ist das Bild am Anfang der Auswahl und die dritte Ganzzahl ist das Bild, bei dem der Auswahlbereich endet. Das zuletzt bezeichnete Bild gehört nicht der Auswahl an.

Beschreibung

Methode; ruft ein Array der gegenwärtig ausgewählten Bilder ab.

Beispiel

Im folgenden Beispiel wird davon ausgegangen, dass die oberste Ebene die aktuelle Ebene ist. Im Bedienfeld „Ausgabe“ wird dann `0, 5, 10, 0, 20, 25` angezeigt:

```
var timeline = fl.getDocumentDOM().getTimeline();
timeline.setSelectedFrames(5,10);
timeline.setSelectedFrames(20,25,false);
var theSelectedFrames = timeline.getSelectedFrames();
fl.trace(theSelectedFrames);
```

Siehe auch

[timeline.setSelectedFrames\(\)](#)

timeline.getSelectedLayers()

Verfügbarkeit

Flash MX 2004.

Parameter

Keine.

Rückgabewerte

Ein Array der auf Null basierenden Indexwerte der ausgewählten Ebenen.

Beschreibung

Methode; ruft die auf Null basierenden Indexwerte der ausgewählten Ebenen ab.

Beispiel

Im folgenden Beispiel wird im Bedienfeld „Ausgabe“ 1,0 angezeigt:

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(0);
fl.getDocumentDOM().getTimeline().setSelectedLayers(1, false);
var layerArray = fl.getDocumentDOM().getTimeline().getSelectedLayers();
fl.trace(layerArray);
```

Siehe auch

[timeline.setSelectedLayers\(\)](#)

timeline.insertBlankKeyframe()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.insertBlankKeyframe([frameNumIndex])
```

Parameter

frameNumIndex Ein auf Null basierender Index, der angibt, an welcher Bildposition das Schlüsselbild eingefügt werden soll. Wenn Sie *frameNumIndex* auslassen, verwendet die Methode die aktuelle Bildnummer am Abspielkopf. Dieser Parameter ist optional.

Wenn das angegebene oder ausgewählte Bild ein normales Bild ist, wird das Schlüsselbild an dieser Bildposition eingefügt. Wenn beispielsweise ein Bildbereich 10 Bilder enthält, die von 1 bis 10 nummeriert sind, und Sie Bild 5 auswählen, wird Bild 5 bei dieser Methode durch ein leeres Schlüsselbild ersetzt und der Bildbereich enthält weiterhin 10 Bilder. Wenn Bild 5 ausgewählt wurde, ein Schlüsselbild ist und direkt neben einem normalen Bild liegt, fügt diese Methode an der Position von Bild 6 ein leeres Schlüsselbild ein. Handelt es sich sowohl bei Bild 5 als auch bei dem daneben liegenden Bild um Schlüsselbilder, so wird kein Schlüsselbild eingefügt, sondern der Abspielkopf wird zu Bild 6 verschoben.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt an der angegebenen Bildindexposition ein leeres Schlüsselbild ein. Wurde keine Indexposition angegeben, wird das leere Schlüsselbild anhand des Abspielkopfes bzw. anhand der Auswahl eingefügt. Siehe auch `timeline.insertKeyframe()`.

Beispiel

Im folgenden Beispiel wird an der Position von Bild 20 ein leeres Schlüsselbild eingefügt (denken Sie daran, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().insertBlankKeyframe(19);
```

Im folgenden Beispiel wird an der Position des ausgewählten Bilds ein leeres Schlüsselbild eingefügt (bzw. an der Position des Abspielkopfes, wenn kein Bild ausgewählt wurde):

```
fl.getDocumentDOM().getTimeline().insertBlankKeyframe();
```

timeline.insertFrames()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.insertFrames([numFrames [, bAllLayers [, frameNumIndex]])
```

Parameter

numFrames Eine Ganzzahl, die die Anzahl der einzufügenden Bilder angibt. Wenn Sie diesen Parameter auslassen, werden die Bilder in der aktuellen Ebene an der aktuellen Position eingefügt. Dieser Parameter ist optional.

bAllLayers Ein boolescher Wert; bei dem Wert `true` (Standard), fügt die Methode in sämtliche Ebenen die im Parameter `numFrames` angegebene Anzahl von Bildern ein. Bei `false` werden die Bilder nur in der aktuellen Ebene eingefügt. Dieser Parameter ist optional.

frameNumIndex Ein auf Null basierender Index, der die Bildposition angibt, an der ein neues Bild eingefügt werden soll. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt die angegebene Anzahl von Bildern an der angegebenen Indexposition ein.

Wenn keine Parameter festgelegt wurden, geschieht Folgendes:

- Ist mindestens ein Bild ausgewählt, fügt die Methode die ausgewählte Anzahl von Bildern in der aktuellen Ebene an der Stelle ein, an der sich das erste ausgewählte Bild befindet. Wenn also die Bilder 6 bis 10 ausgewählt sind (insgesamt fünf Bilder), werden auf der Ebene, die die ausgewählten Bilder enthält, an der Position von Bild 6 fünf Bilder hinzugefügt.
- Sind keine Bilder ausgewählt, fügt die Methode auf allen Ebenen an der aktuellen Bildposition ein Bild hinzu.

Wenn Parameter festgelegt wurden, geschieht Folgendes:

- Wurde nur *numFrames* angegeben, wird die angegebene Anzahl von Bildern in der aktuellen Ebene an der aktuellen Bildposition eingefügt.
- Wurde *numFrames* angegeben und hat *bAllLayers* den Wert *true*, wird die angegebene Anzahl von Bildern auf allen Ebenen an der aktuellen Bildposition hinzugefügt.
- Wurden alle drei Parameter festgelegt, wird die angegebene Anzahl von Bildern an der angegebenen Indexposition (*frameIndex*) eingefügt; der für *bAllLayers* übergebene Wert bestimmt, ob die Bilder nur auf der aktuellen Ebene oder auf allen Ebenen hinzugefügt werden.

Ist das angegebene oder ausgewählte Bild ein normales Bild, wird das Bild an dieser Bildposition eingefügt. Wenn beispielsweise ein Bildbereich 10 Bilder enthält, die von 1 bis 11 nummeriert sind, und Sie Bild 5 auswählen (bzw. für *frameIndex* den Wert 4 übergeben), wird an der Position von Bild 5 ein Bild hinzugefügt und der Bildbereich auf 11 Bilder erweitert. Wenn Bild 5 ausgewählt und ein Schlüsselbild ist, fügt diese Methode an der Position von Bild 6 ein leeres Bild ein, unabhängig davon, ob das daneben liegende Bild ebenfalls ein Schlüsselbild ist.

Beispiel

Im folgenden Beispiel wird ein Bild (bzw. je nach der Auswahl auch mehrere Bilder) auf der aktuellen Ebene an der Position der aktuellen Auswahl eingefügt:

```
fl.getDocumentDOM().getTimeline().insertFrames();
```

Im folgenden Beispiel werden auf allen Ebenen an der aktuellen Bildposition jeweils fünf Bilder eingefügt:

```
fl.getDocumentDOM().getTimeline().insertFrames(5);
```

Hinweis: Nehmen wir einmal an, Ihnen liegen mehrere Ebenen mit Bildern vor. Wenn Sie dann ein Bild in einer Ebene auswählen und den vorherigen Befehl verwenden, werden die Bilder nur in der ausgewählten Ebene eingefügt. Gibt es mehrere Ebenen, in denen keine Bilder ausgewählt sind, werden die Bilder in allen Ebenen eingefügt.

Im folgenden Beispiel werden nur in der aktuellen Ebene drei Bilder eingefügt:

```
fl.getDocumentDOM().getTimeline().insertFrames(3, false);
```

Im folgenden Beispiel werden in allen Ebenen an der Position des ersten Bilds vier Bilder eingefügt:

```
fl.getDocumentDOM().getTimeline().insertFrames(4, true, 0);
```

timeline.insertKeyframe()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.insertKeyframe([frameNumIndex])
```

Parameter

frameNumIndex Ein auf Null basierender Index, der die Bildindexposition angibt, an der das Schlüsselbild in der aktuellen Ebene eingefügt werden soll. Wenn *frameNumIndex* fehlt, verwendet die Methode die Bildnummer des aktuellen Abspielkopfes oder des ausgewählten Bilds. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt an der angegebenen Bildposition ein Schlüsselbild ein. Wenn Sie diesen Parameter auslassen, fügt die Methode das Schlüsselbild anhand der Position des Abspielkopfes oder der Auswahl ein.

Diese Methode funktioniert genau wie `timeline.insertBlankKeyframe()`, allerdings enthält das eingefügte Schlüsselbild den Inhalt des konvertierten Bilds (es ist also nicht leer).

Beispiel

Im folgenden Beispiel wird an der Position des Abspielkopfes bzw. der Auswahl ein Schlüsselbild eingefügt:

```
fl.getDocumentDOM().getTimeline().insertKeyframe();
```

Im folgenden Beispiel wird auf der zweiten Ebene an der Position von Bild 10 ein Schlüsselbild eingefügt (denken Sie daran, dass Index-, Bild- und Ebenennummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().currentLayer = 1;  
fl.getDocumentDOM().getTimeline().insertKeyframe(9);
```

timeline.layerCount

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.layerCount
```

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die die Anzahl der Ebenen in der angegebenen Zeitleiste angibt.

Beispiel

Im folgenden Beispiel wird die Anzahl der in der aktuellen Szene vorhandenen Ebenen in der Variablen `NumLayer` gespeichert:

```
var NumLayer = fl.getDocumentDOM().getTimeline().layerCount;
```

timeline.layers

Verfügbarkeit

Flash MX 2004.

Verwendung

`timeline.layers`

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von Layer-Objekten.

Beispiel

Im folgenden Beispiel wird das Array der im aktuellen Dokument vorhandenen Layer-Objekte in der Variablen `currentLayers` gespeichert:

```
var currentLayers = fl.getDocumentDOM().getTimeline().layers;
```

timeline.libraryItem

Verfügbarkeit

Flash Professional CS5.

Verwendung

`timeline.libraryItem`

Beschreibung

Schreibgeschützte Eigenschaft; wenn die `libraryItem`-Eigenschaft der Zeitleiste auf Null eingestellt ist, gehört die Zeitleiste zu einer Szene. Bei einem anderen Wert als Null kann sie wie ein `LibraryItem`-Objekt gehandhabt werden.

Beispiel

Mit dem folgenden Beispiel wird der Name von `libraryItem` ausgegeben, wenn der Wert von `libraryItem` nicht Null lautet, und der Name der Szene, wenn `libraryItem` den Wert Null hat:

```
var item = fl.getDocumentDOM().getTimeline().libraryItem;
if (item)
    fl.trace("libraryItem name: " + item.name);
else
    fl.trace("scene name: " + fl.getDocumentDOM().getTimeline().name);
```

timeline.name

Verfügbarkeit

Flash MX 2004.

Verwendung

`timeline.name`

Beschreibung

Eigenschaft; ein String, der den Namen der aktuellen Zeitleiste angibt. Dies ist der Name der Szene, des Bildschirms (Folie oder Formular) oder des Symbols, an der bzw. dem Sie zurzeit arbeiten.

Beispiel

Im folgenden Beispiel wird der Name der ersten Szene abgerufen:

```
var sceneName = fl.getDocumentDOM().timelines[0].name;
```

Im folgenden Beispiel wird der ersten Szene der Name `FirstScene` zugewiesen:

```
fl.getDocumentDOM().timelines[0].name = "FirstScene";
```

timeline.pasteFrames()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.pasteFrames([startFrameIndex [, endFrameIndex]])
```

Parameter

startFrameIndex Ein auf Null basierender Index, der angibt, an welcher Bildposition mit dem Einfügen eines Bildbereichs begonnen werden soll. Wenn Sie *startFrameIndex* auslassen, verwendet die Methode die aktuelle Auswahl. Dieser Parameter ist optional.

endFrameIndex Ein auf Null basierender Index, der die Bildposition angibt, an der der Einfügevorgang endet. Der Bildbereich endet bei *endFrameIndex*, schließt aber das Bild an dieser Position nicht ein. Wenn Sie nur *startFrameIndex* angeben, erhält *endFrameIndex* automatisch den gleichen Wert wie *startFrameIndex*. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt den Bildbereich aus der Zwischenablage in die angegebenen Bilder ein.

Beispiel

Im folgenden Beispiel werden die Bilder aus der Zwischenablage an der Position des gegenwärtig ausgewählten Bilds bzw. des Abspielkopfes eingefügt:

```
fl.getDocumentDOM().getTimeline().pasteFrames();
```

Im folgenden Beispiel werden die Bilder aus der Zwischenablage in den von Bild 2 bis (ausschließlich) Bild 10 reichenden Bereich eingefügt (denken Sie daran, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().pasteFrames(1, 9);
```

Im folgenden Beispiel werden die Bilder aus der Zwischenablage an der Position von Bild 5 eingefügt:

```
fl.getDocumentDOM().getTimeline().pasteFrames(4);
```

timeline.pasteLayers()

Verfügbarkeit

Flash CS5.5 Professional.

Verwendung

```
timeline.pasteLayers ([layerIndex])
```

Parameter

layerIndex Optional. Ein auf Null basierender Index, der die Ebene angibt, über der die Ebenen in der Zwischenablage eingefügt werden. Wenn Sie „layerIndex“ auslassen, verwendet die Methode die aktuelle Auswahl.

Rückgabewerte

Ganzzahl für den niedrigsten Ebenenindex der eingefügten Ebenen.

Beschreibung

Methode; fügt Ebenen, die zuvor ausgeschnitten oder kopiert wurden, oberhalb der derzeit ausgewählten Ebene oder oberhalb des angegebenen Ebenenindex ein. Wenn es sich bei der angegebenen Ebene um eine Ordnerebene handelt, werden die Ebenen in den Ordner eingefügt. Gibt den niedrigsten Ebenenindex der eingefügten Ebenen zurück. Dieser Vorgang wirkt sich nicht auf die Systemzwischenablage aus.

Beispiel

Im folgenden Beispiel werden die Ebenen aus der Ebenenzwischenablage oberhalb der derzeit ausgewählten Ebene in der Zeitleiste eingefügt:

```
fl.getDocumentDOM().getTimeline().pasteLayers();
```

Im folgenden Beispiel werden die Ebenen aus der Zwischenablage oberhalb der Ebene an Index 2 eingefügt:

```
fl.getDocumentDOM().getTimeline().pasteLayers(2);
```

Siehe auch

[timeline.cutLayers\(\)](#), [timeline.copyLayers\(\)](#), [timeline.duplicateLayers\(\)](#)

timeline.pasteMotion()

Verfügbarkeit

Flash CS3 Professional.

Verwendung

```
timeline.pasteMotion()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; fügt den Bereich der durch `timeline.copyMotion()` abgerufenen Bewegungsbilder in die Zeitleiste ein. Vorhandene Bilder werden, falls nötig, verschoben (nach rechts), um Platz für die eingefügten Bilder zu schaffen.

Beispiel

Im folgenden Beispiel wird die Bewegung aus der Zwischenablage in das zurzeit ausgewählte Bild oder an der Position des Abspielkopfs eingefügt, wobei das dort befindliche Bild nach rechts verschoben wird:

```
fl.getDocumentDOM().getTimeline().pasteMotion();
```

Siehe auch

[timeline.copyMotion\(\)](#)

timeline.removeFrames()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.removeFrames([startFrameIndex [,endFrameIndex]])
```

Parameter

startFrameIndex Ein auf Null basierender Index, der angibt, an welcher Bildposition mit dem Entfernen von Bildern begonnen werden soll. Wenn Sie *startFrameIndex* auslassen, verwendet die Methode die aktuelle Auswahl. Falls es keine Auswahl gibt, werden alle Bilder an der aktuellen Abspielkopfposition auf allen Ebenen entfernt. Dieser Parameter ist optional.

endFrameIndex Ein auf Null basierender Index, der die Bildposition angibt, an der das Entfernen von Bildern endet. Der Bildbereich reicht bis *endFrameIndex*, schließt aber das Bild an dieser Position nicht ein. Wenn Sie nur *startFrameIndex* angeben, erhält *endFrameIndex* automatisch den gleichen Wert wie *startFrameIndex*. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; löscht das Bild.

Beispiel

Im folgenden Beispiel werden alle Bilder von Bild 5 bis (ausschließlich) Bild 10 in der aktuellen Szene aus der obersten Ebene gelöscht (denken Sie daran, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().removeFrames(4, 9);
```

Im folgenden Beispiel wird Bild 8 der aktuellen Szene aus der obersten Ebene gelöscht:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().removeFrames(7);
```

timeline.removeMotionObject()

Verfügbarkeit

Flash Professional CS5.

Verwendung

```
timeline.removeMotionObject([startFrame [,endFrame]])
```

Parameter

startFrame Gibt das erste Bild an, an dem mit dem Entfernen von Bewegungsobjekten begonnen werden soll. Wenn Sie *startFrame* weglassen, verwendet die Methode die aktuelle Auswahl. Ist keine Auswahl vorhanden, werden alle Bilder am aktuellen Abspielkopf auf allen Ebenen entfernt. Dieser Parameter ist optional.

endFrame Gibt das Bild an, an dem das Entfernen von Bewegungsobjekten beendet werden soll. Der Bildbereich reicht bis zu *endFrame*, schließt dieses Bild aber nicht ein. Wenn Sie nur *startFrame* angeben, ist *endFrame* standardmäßig auf den Wert von *startFrame* eingestellt. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; entfernt das Bewegungsobjekt und konvertiert die Bilder wieder in statische Bilder. Die Parameter sind optional. Wenn die Parameter angegeben werden, wird die Auswahl der Zeitleiste auf die angegebenen Bilder eingestellt, bevor das Bewegungsobjekt entfernt wird.

Beispiel

Das folgende Beispiel löscht alle Bewegungsobjekte und konvertiert die Bilder an der aktuellen Position des Abspielkopfs auf der obersten Ebene wieder in statische Bilder:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().removeMotionObject();
```

Das folgende Beispiel löscht Bewegungsobjekte auf der obersten Ebene der aktuellen Szene ab Bild 5 bis Bild 15 (aber nicht einschließlich Bild 15).

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().removeMotionObject(5, 15);
```

Siehe auch

„[timeline.createMotionObject\(\)](#)“ auf Seite 482

timeline.reorderLayer()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.reorderLayer(layerToMove, layerToPutItBy [, bAddBefore])
```

Parameter

layerToMove Ein auf Null basierender Index, der die zu verschiebende Ebene angibt.

layerToPutItBy Ein auf Null basierender Index, der die Ebene angibt, neben die die Ebene verschoben werden soll. Wenn Sie beispielsweise für *layerToMove* den Wert 1 angeben und für *layerToPutItBy* den Wert 0, wird die zweite Ebene neben der ersten Ebene platziert.

addBefore Gibt an, ob die Ebene über oder unter *layerToPutItBy* verschoben werden soll. Bei dem Wert `false` wird die Ebene unter *layerToPutItBy* verschoben. Der Standardwert lautet `true`. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; verschiebt die erste angegebene Ebene über oder unter die zweite angegebene Ebene.

Beispiel

Im folgenden Beispiel wird die Ebene an Indexposition 2 an die oberste Position verschoben (also über die Ebene an Indexposition 0):

```
fl.getDocumentDOM().getTimeline().reorderLayer(2, 0);
```

Im folgenden Beispiel wird die Ebene an Indexposition 3 unter die Ebene an Indexposition 5 verschoben:

```
fl.getDocumentDOM().getTimeline().reorderLayer(3, 5, false);
```

timeline.reverseFrames()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.reverseFrames([startFrameIndex [, endFrameIndex]])
```

Parameter

startFrameIndex Ein auf Null basierender Index, der die Bildposition angibt, an der die Bildumkehrung beginnen soll. Wenn Sie *startFrameIndex* auslassen, verwendet die Methode die aktuelle Auswahl. Dieser Parameter ist optional.

endFrameIndex Ein auf Null basierender Index, der die Bildposition angibt, an der die Bildumkehrung beginnen soll. Der Bildbereich reicht bis *endFrameIndex*, schließt aber das Bild an dieser Position nicht ein. Wenn Sie nur *startFrameIndex* angeben, erhält *endFrameIndex* automatisch den gleichen Wert wie *startFrameIndex*. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; kehrt einen Bildbereich um.

Beispiel

Im folgenden Beispiel werden die Positionen der ausgewählten Bilder umgekehrt:

```
fl.getDocumentDOM().getTimeline().reverseFrames();
```

Im folgenden Beispiel werden die Bilder von Bild 10 bis (ausschließlich) Bild 15 umgekehrt (denken Sie daran, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().reverseFrames(9, 14);
```

timeline.selectAllFrames()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.selectAllFrames()
```

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; wählt alle Bilder in der aktuellen Zeitleiste aus.

Beispiel

Im folgenden Beispiel werden alle Bilder in der aktuellen Zeitleiste ausgewählt:

```
fl.getDocumentDOM().getTimeline().selectAllFrames();
```

timeline setFrameProperty()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.setFrameProperty(property, value [, startFrameIndex [, endFrameIndex]])
```

Parameter

property Ein String, der den Namen der zu ändernden Eigenschaft angibt. Eine vollständige Liste der Eigenschaften und Werte finden Sie in der Übersicht über die Eigenschaften für das [Frame-Objekt](#).

Mit dieser Methode können Sie für schreibgeschützte Eigenschaften wie z. B. `frame.duration` und `frame.elements` keine Werte festlegen.

value Gibt den Wert an, der der Eigenschaft zugewiesen werden soll. Informationen zu den möglichen Werten und Typen finden Sie in der Übersicht über die Eigenschaften für das [Frame-Objekt](#).

startFrameIndex Ein auf Null basierender Index, der die Bildposition angibt, an der mit der Änderung begonnen werden soll. Wenn Sie *startFrameIndex* auslassen, verwendet die Methode die aktuelle Auswahl. Dieser Parameter ist optional.

endFrameIndex Ein auf Null basierender Index, der die Bildposition angibt, an der die Änderung beendet werden soll. Der Bildbereich reicht bis *endFrameIndex*, schließt das Bild an dieser Position aber nicht ein. Wenn Sie *startFrameIndex* angeben, aber *endFrameIndex* auslassen, erhält *endFrameIndex* automatisch den gleichen Wert wie *startFrameIndex*. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; legt die Eigenschaft des Frame-Objekts für die ausgewählten Bilder fest.

Beispiel

Im folgenden Beispiel wird dem ersten Bild auf der obersten Ebene des aktuellen Dokuments der `ActionScript`-Befehl `stop()` zugewiesen:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().setSelectedFrames(0,0,true);  
fl.getDocumentDOM().getTimeline().setFrameProperty("actionScript", "stop();");
```

Im folgenden Beispiel wird in den Bildbereich von Bild 2 bis (ausschließlich) Bild 5 auf der aktuellen Ebene ein `Bewegungs-Tween` eingefügt (denken Sie daran, dass `Index`- und `Bildnummernwerte` unterschiedlich sind):

```
var doc = fl.getDocumentDOM();  
doc.getTimeline().setFrameProperty("tweenType", "motion", 1,4);
```

timeline.setGuidelines()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
timeline.setGuidelines(xmlString)
```

Parameter

xmlString Ein XML-String, der Informationen zu den anzuwendenden Hilfslinien enthält.

Rückgabewerte

Der boolescher Wert `true`, wenn die Hilfslinien erfolgreich angewendet wurden; andernfalls `false`.

Beschreibung

Methode; ersetzt die Hilfslinien einer Zeitleiste („Ansicht“ > „Hilfslinien“ > „Hilfslinien anzeigen“) durch die in *xmlString* enthaltenen Informationen. Mit `timeline.getGuidelines()` kann ein XML-String abgerufen werden, der an diese Methode übergeben werden kann.

Um die neu festgelegten Hilfslinien anzuzeigen, müssen Sie sie möglicherweise erst ausblenden und dann anzeigen.

Beispiel

Im folgenden Beispiel werden die Hilfslinien einer FLA-Datei auf eine andere FLA-Datei angewendet:

```
var doc0 = fl.documents[0];  
var guides0 = doc0.timelines[0].getGuidelines();  
var doc1 = fl.documents[1];  
doc1.timelines[0].setGuidelines(guides0);
```

timeline.setLayerProperty()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.setLayerProperty(property, value [, layersToChange])
```

Parameter

property Ein String, der die festzulegende Eigenschaft angibt. Eine Liste der Eigenschaften finden Sie unter „[Layer-Objekt](#)“ auf Seite 326.

value Der Wert, der der Eigenschaft zugewiesen werden soll. Verwenden Sie dabei den Wertetyp, den Sie beim Festlegen der Eigenschaft für das Layer-Objekt verwenden würden.

layersToChange Ein String, der angibt, welche Ebenen modifiziert werden sollen. Die zulässigen Werte lauten "selected", "all" und "others". Wenn Sie diesen Parameter auslassen, wird der Standardwert "selected" verwendet. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; weist der angegebenen Eigenschaft auf allen ausgewählten Ebenen einen bestimmten Wert zu.

Beispiel

Im folgenden Beispiel werden die ausgewählten Ebenen unsichtbar gemacht:

```
fl.getDocumentDOM().getTimeline().setLayerProperty("visible", false);
```

Im folgenden Beispiel wird den ausgewählten Ebenen der Name `selLayer` zugewiesen:

```
fl.getDocumentDOM().getTimeline().setLayerProperty("name", "selLayer");
```

timeline.setSelectedFrames()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.setSelectedFrames(startFrameIndex, endFrameIndex [, bReplaceCurrentSelection])  
timeline.setSelectedFrames(selectionList [, bReplaceCurrentSelection])
```

Parameter

startFrameIndex Ein auf Null basierender Index, der das erste der einzustellenden Bilder angibt.

endFrameIndex Ein auf Null basierender Index, der das Ende der Auswahl angibt. *endFrameIndex* ist das Bild, das unmittelbar auf das letzte dem Auswahlbereich angehörende Bild folgt.

bReplaceCurrentSelection Ein boolescher Wert; bei dem Wert `true` wird die gegenwärtige Bildauswahl aufgehoben, bevor die angegebenen Bilder ausgewählt werden. Der Standardwert ist `true`.

selectionList Ein Array mit drei Ganzzahlen, die von `timeline.getSelectedFrames()` zurückgegeben wurden.

Rückgabewerte

Keine.

Beschreibung

Methode; wählt einen Bildbereich in der aktuellen Ebene aus oder stellt die ausgewählten Bilder auf das an diese Methode übergebene Auswahl-Array ein.

Beispiel

Die folgenden Beispiele zeigen zwei Möglichkeiten, um die oberste Ebene, Bild 1 bis (ausschließlich) Bild 10 auszuwählen und anschließend der aktuellen Auswahl die Bilder 12 bis (ausschließlich) 15 hinzuzufügen, die sich auf derselben Ebene befinden (denken Sie daran, dass Index- und Bildnummernwerte unterschiedlich sind):

```
fl.getDocumentDOM().getTimeline().setSelectedFrames(0, 9);  
fl.getDocumentDOM().getTimeline().setSelectedFrames(11, 14, false);  
fl.getDocumentDOM().getTimeline().setSelectedFrames([0, 0, 9]);  
fl.getDocumentDOM().getTimeline().setSelectedFrames([0, 11, 14], false);
```

Im folgenden Beispiel wird zuerst das Array der ausgewählten Bilder in der Variablen `savedSelectionList` gespeichert. Anhand dieses Arrays werden diese Bilder zu einem späteren Zeitpunkt erneut ausgewählt, nachdem die Auswahl durch einen Befehl oder eine Benutzerinteraktion verändert wurde:

```
var savedSelectionList = fl.getDocumentDOM().getTimeline().getSelectedFrames();  
// Do something that changes the selection.  
fl.getDocumentDOM().getTimeline().setSelectedFrames(savedSelectionList);
```

Siehe auch

[timeline.getSelectedFrames\(\)](#)

timeline.setSelectedLayers()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.setSelectedLayers(index [, bReplaceCurrentSelection])
```

Parameter

index Ein auf Null basierender Index, der die auszuwählende Ebene angibt.

bReplaceCurrentSelection Ein boolescher Wert; bei dem Wert `true` wird die aktuelle Auswahl ersetzt und bei `false` wird sie erweitert. Der Standardwert ist `true`. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; legt die auszuwählende Ebene fest. Legt außerdem fest, dass die angegebene Ebene die aktuelle Ebene sein soll. Bei der Auswahl einer Ebene werden gleichzeitig alle Bilder in dieser Ebene ausgewählt.

Beispiel

Im folgenden Beispiel wird die oberste Ebene ausgewählt:

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(0);
```

Im folgenden Beispiel wird der Auswahl die nächste Ebene hinzugefügt:

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(1, false);
```

Siehe auch

[timeline.getSelectedLayers\(\)](#)

timeline.showLayerMasking()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
timeline.showLayerMasking([layer])
```

Parameter

layer Ein auf Null basierender Index einer Maskenebene bzw. einer maskierten Ebene, mit dessen Hilfe die Maskierung während des Authorings angezeigt werden kann. Dieser Parameter ist optional.

Rückgabewerte

Keine.

Beschreibung

Methode; sperrt die Maskenebenen und die maskierten Ebenen und zeigt so die Ebenenmaskierung während des Authorings an. Wenn keine Ebene angegeben wurde, wird die aktuelle Ebene verwendet. Wenn Sie diese Methode bei einer Ebene einsetzen, die nicht dem Typ „Mask“ bzw. „Masked“ angehört, wird im Bedienfeld „Ausgabe“ ein Fehler angezeigt.

Beispiel

Im folgenden Beispiel wird angegeben, dass die Ebenenmaskierung der ersten Ebene während des Authorings angezeigt werden soll:

```
fl.getDocumentDOM().getTimeline().showLayerMasking(0);
```

timeline.startPlayback()

Verfügbarkeit

Flash Professional CS5.

Verwendung

```
timeline.startPlayback()
```

Rückgabewerte

Keine.

Beschreibung

Methode; startet die automatische Wiedergabe der Zeitleiste, wenn sie derzeit abgespielt wird. Diese Methode kann mit SWF-Bedienfeldern verwendet werden, um die Wiedergabe der Zeitleiste in der Authoring-Umgebung zu steuern.

Beispiel

Mit dem folgenden Beispiel wird die Wiedergabe der Zeitleiste gestartet.

```
fl.getDocumentDOM().getTimeline().startPlayback();
```

timeline.stopPlayback()

Verfügbarkeit

Flash Professional CS5.

Verwendung

```
timeline.stopPlayback()
```

Rückgabewerte

Keine.

Beschreibung

Methode; stoppt die automatische Wiedergabe der Zeitleiste, wenn sie derzeit abgespielt wird. Diese Methode kann mit SWF-Bedienfeldern verwendet werden, um die Wiedergabe der Zeitleiste in der Authoring-Umgebung zu steuern.

Beispiel

Mit dem folgenden Beispiel wird die Wiedergabe der Zeitleiste gestoppt.

```
fl.getDocumentDOM().getTimeline().stopPlayback();
```

Kapitel 46: ToolObj-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Ein ToolObj-Objekt stellt ein einzelnes Werkzeug im Bedienfeld „Werkzeuge“ dar. Wenn Sie auf ein ToolObj-Objekt zugreifen möchten, verwenden Sie die Eigenschaften von **Tools-Objekt**: entweder das Array `tools.toolObjs` oder `tools.activeTool`.

Übersicht über Methoden

Für das ToolObj-Objekt stehen die folgenden Methoden zur Auswahl:

Hinweis: Die folgenden Methoden werden nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Methode	Beschreibung
<code>toolObj.enablePIControl()</code>	Aktiviert oder deaktiviert die angegebene Steuerung in einem Eigenschafteninspektor. Wird nur beim Erstellen erweiterbarer Werkzeuge verwendet.
<code>toolObj.setIcon()</code>	Gibt eine PNG-Datei an, die im Flash-Bedienfeld „Werkzeuge“ als Werkzeugsymbol verwendet werden soll.
<code>toolObj.setMenuString()</code>	Gibt den String an, der im Popupmenü als Name des Werkzeugs angezeigt werden soll.
<code>toolObj.setOptionsFile()</code>	Verknüpft eine XML-Datei mit dem Werkzeug.
<code>toolObj.setPI()</code>	Legt fest, dass ein bestimmter Eigenschafteninspektor verwendet werden soll, wenn das Werkzeug aktiviert ist.
<code>toolObj.setToolName()</code>	Weist dem Werkzeug einen Namen zu, der bei der Konfiguration des Bedienfelds „Werkzeuge“ verwendet wird.
<code>toolObj.setToolTip()</code>	Gibt die QuickInfo an, die angezeigt werden soll, wenn sich die Maus direkt über dem Werkzeugsymbol befindet.
<code>toolObj.showPIControl()</code>	Blendet eine Steuerung im Eigenschafteninspektor ein oder aus.
<code>toolObj.showTransformHandles()</code>	Wird in der Methode <code>configureTool()</code> der JavaScript-Datei eines erweiterbaren Werkzeugs aufgerufen und gibt an, dass die Griffe für die freie Transformation angezeigt werden sollen, wenn das Werkzeug aktiv ist.

Übersicht über Eigenschaften

Für das ToolObj-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>toolObj.depth</code>	Eine Ganzzahl, die die Tiefe des Werkzeugs im Popupmenü im Bedienfeld „Werkzeuge“ angibt.
<code>toolObj.iconID</code>	Eine Ganzzahl, die Ressourcenbezeichner des Werkzeugs angibt.
<code>toolObj.position</code>	Schreibgeschützt; eine Ganzzahl, die angibt, wo sich das Werkzeug im Bedienfeld „Werkzeuge“ befindet.

toolObj.depth

Verfügbarkeit

Flash MX 2004.

Verwendung

`toolObj.depth`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die die Tiefe des Werkzeugs im Popupmenü im Bedienfeld „Werkzeuge“ angibt. Diese Eigenschaft wird nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Im folgenden Beispiel wird angegeben, dass das Werkzeug eine Tiefe von 1 haben, also im Bedienfeld „Werkzeuge“ eine Ebene unterhalb eines vorhandenen Werkzeugs liegen soll:

```
fl.tools.activeTool.depth = 1;
```

toolObj.enablePIControl()

Verfügbarkeit

Flash MX 2004.

Verwendung

`toolObj.enablePIControl(control, bEnable)`

Parameter

control Ein String, der den Namen der Steuerung angibt, die aktiviert oder deaktiviert werden soll. Die zulässigen Werte hängen von dem Eigenschafteninspektor ab, der bei der Aktivierung dieses Werkzeugs aufgerufen wird (siehe [toolObj.setPI\(\)](#)).

Ein Eigenschafteninspektor für Formen hat die folgenden Steuerungen:

stroke	fill
--------	------

Ein Eigenschafteninspektor für Text hat die folgenden Steuerungen:

type	font	pointsize
color	bold	italic
direction	alignLeft	alignCenter
alignRight	alignJustify	spacing
position	autoKern	small
rotation	format	lineType

selectable	html	border
deviceFonts	varEdit	options
link	maxChars	target

Ein Eigenschafteninspektor für Filme hat die folgenden Steuerungen:

size	publish	background
framerate	player	profile

bEnable Ein boolescher Wert, der bestimmt, ob die Steuerung aktiviert (`true`) oder deaktiviert (`false`) werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; aktiviert oder deaktiviert die angegebene Steuerung in einem Eigenschafteninspektor. Wird nur beim Erstellen erweiterbarer Werkzeuge verwendet.

Beispiel

Mit dem folgenden Befehl in der JavaScript-Datei eines erweiterbaren Werkzeugs wird Flash angewiesen, für dieses Werkzeug im Eigenschafteninspektor keine Strichoptionen anzuzeigen:

```
theTool.enablePControl("stroke", false);
```

toolObj.iconID

Verfügbarkeit

Flash MX 2004.

Verwendung

`toolObj.iconID`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl mit dem Wert -1. Diese Eigenschaft wird nur beim Erstellen erweiterbarer Werkzeuge eingesetzt. Ein `iconID`-Wert von -1 bedeutet, dass Flash nicht versucht, ein Symbol für das Werkzeug zu suchen. Stattdessen wird das im Bedienfeld „Werkzeuge“ angezeigte Symbol des Werkzeugs mithilfe des Skripts angegeben. Weitere Hinweise finden Sie unter [toolObj.setIcon\(\)](#).

Beispiel

Im folgenden Beispiel wird der Variablen `toolIconID` der Wert -1 (der Symbolbezeichner des aktuellen Werkzeugs) zugewiesen:

```
var toolIconID = fl.tools.activeTool.iconID
```

toolObj.position

Verfügbarkeit

Flash MX 2004.

Verwendung

`toolObj.position`

Beschreibung

Schreibgeschützte Eigenschaft; eine Ganzzahl, die die Position des Werkzeugs im Bedienfeld „Werkzeuge“ angibt. Diese Eigenschaft wird nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Mit den folgenden Befehlen in der Methode `mouseDown()` der JavaScript-Datei eines Werkzeugs wird die Position dieses Werkzeugs innerhalb des Bedienfelds „Werkzeuge“ im Bedienfeld „Ausgabe“ als Ganzzahl angezeigt:

```
myToolPos = fl.tools.activeTool.position;  
fl.trace(myToolPos);
```

toolObj.setIcon()

Verfügbarkeit

Flash MX 2004.

Verwendung

`toolObj.setIcon(file)`

Parameter

file Ein String, der den Namen der als Symbol zu verwendenden PNG-Datei angibt. Die PNG-Datei muss in demselben Ordner wie die JSFL-Datei abgelegt werden.

Rückgabewerte

Keine.

Beschreibung

Methode; bezeichnet eine PNG-Datei, die im Bedienfeld „Werkzeuge“ als Werkzeugsymbol verwendet werden soll. Diese Methode wird nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Im folgenden Beispiel wird festgelegt, dass das Bild in der Datei „PolyStar.png“ als Symbol für das Werkzeug `PolyStar` verwendet werden soll. Dieser Code stammt aus der Beispieldatei „PolyStar.jsfl“ (siehe „[Beispiel-Polysternwerkzeug](#)“ auf Seite 14):

```
theTool = fl.tools.activeTool;  
theTool.setIcon("PolyStar.png");
```

toolObj.setMenuString()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
toolObj.setMenuString(menuStr)
```

Parameter

menuStr Ein String mit dem Namen, der in dem Popupmenü als Name des Werkzeugs angezeigt werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; gibt den String an, der im Popupmenü als Name des Werkzeugs angezeigt werden soll. Diese Methode wird nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Im folgenden Beispiel wird festgelegt, dass im Popupmenü des Werkzeugs `theTool` der Name „PolyStar Tool“ angezeigt werden soll. Dieser Code stammt aus der Beispieldatei „PolyStar.jsfl“ (siehe „[Beispiel-Polysternwerkzeug](#)“ auf Seite 14):

```
theTool = fl.tools.activeTool;  
theTool.setMenuString("PolyStar Tool");
```

toolObj.setOptionsFile()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
toolObj.setOptionsFile(xmlFile)
```

Parameter

xmlFile Ein String, der den Namen der XML-Datei angibt, die die Beschreibung der Werkzeugoptionen enthält. Die XML-Datei muss in demselben Ordner abgelegt werden wie die JSFL-Datei.

Rückgabewerte

Keine.

Beschreibung

Methode; verknüpft eine XML-Datei mit dem Werkzeug. Die Datei gibt an, welche Optionen in einem modalen Bedienfeld angezeigt werden sollen, das über eine „Optionen“-Schaltfläche im Eigenschafteninspektor aufgerufen wird. Normalerweise wird diese Methode im Rahmen der Funktion `configureTool()` in der JSFL-Datei verwendet. Siehe [configureTool\(\)](#).

Die Datei „PolyStar.xml“ nennt z. B. für das Polygonwerkzeug drei Optionen:

```
<properties>
  <property name="Style"
    variable="style"
    list="polygon,star"
    defaultValue="0"
    type="Strings" />

  <property name="Number of Sides"
    variable="nsides"
    min="3"
    max="32"
    defaultValue="5"
    type="Number" />

  <property name="Star point size"
    variable="pointParam"
    min="0"
    max="1"
    defaultValue=".5"
    type="Double" />

</properties>
```

Beispiel

Im folgenden Beispiel wird die Datei „PolyStar.xml“ mit dem gegenwärtig aktiven Werkzeug verknüpft. Dieser Code stammt aus der Beispieldatei „PolyStar.jsfl“ (siehe „[Beispiel-Polysternwerkzeug](#)“ auf Seite 14):

```
theTool = fl.tools.activeTool;
theTool.setOptionsFile("PolyStar.xml");
```

toolObj.setPI()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
toolObj.setPI(pi)
```

Parameter

pi Ein String, der angibt, welcher Eigenschafteninspektor für dieses Werkzeug aufgerufen werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; legt fest, welcher Eigenschafteninspektor mit dem aktivierten Werkzeug verwendet werden soll. Diese Methode wird nur beim Erstellen erweiterbarer Werkzeuge eingesetzt. Die zulässigen Werte lauten "shape" (Standard), "text" und "movie".

Beispiel

Im folgenden Beispiel wird festgelegt, dass der Eigenschafteninspektor für Formen verwendet werden soll, wenn das Werkzeug aktiviert wird. Dieser Code stammt aus der Beispieldatei „PolyStar.jsfl“ (siehe „[Beispiel-Polysternwerkzeug](#)“ auf Seite 14):

```
theTool = fl.tools.activeTool;  
theTool.setPI("shape");
```

toolObj.setToolName()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
toolObj.setToolName(name)
```

Parameter

name Ein String, der den Namen des Werkzeugs angibt.

Rückgabewerte

Keine.

Beschreibung

Methode; weist dem Werkzeug einen Namen zu, der bei der Konfiguration des Bedienfelds „Werkzeuge“ verwendet wird. Diese Methode wird nur beim Erstellen erweiterbarer Werkzeuge eingesetzt. Der Name wird ausschließlich von der XML-Layoutdatei verwendet, die von Flash gelesen wird, um das Bedienfeld „Werkzeuge“ aufzubauen. Der Name wird nicht in der Flash-Benutzeroberfläche angezeigt.

Beispiel

Im folgenden Beispiel wird dem Werkzeug `theTool` der Name `polystar` zugewiesen. Dieser Code stammt aus der Beispieldatei „PolyStar.jsfl“ (siehe „[Beispiel-Polysternwerkzeug](#)“ auf Seite 14):

```
theTool = fl.tools.activeTool;  
theTool.setToolName("polystar");
```

toolObj.setToolTip()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
toolObj.setToolTip(toolTip)
```

Parameter

toolTip Ein String mit der QuickInfo für das Werkzeug.

Rückgabewerte

Keine.

Beschreibung

Methode; gibt die QuickInfo an, die angezeigt werden soll, wenn die Maus sich direkt über dem Werkzeugsymbol befindet. Diese Methode wird nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Im folgenden Beispiel wird festgelegt, dass die QuickInfo für das Werkzeug `PolyStar Tool` lauten soll. Dieser Code stammt aus der Beispieldatei „PolyStar.jsfl“ (siehe „[Beispiel-Polysternwerkzeug](#)“ auf Seite 14):

```
theTool = fl.tools.activeTool;  
theTool.setToolTip("PolyStar Tool");
```

toolObj.showPIControl()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
toolObj.showPIControl(control, bShow)
```

Parameter

control Ein String mit dem Namen der Steuerung, die ein- oder ausgeblendet werden soll. Diese Methode wird nur beim Erstellen erweiterbarer Werkzeuge eingesetzt. Die zulässigen Werte hängen von dem Eigenschafteninspektor ab, der von diesem Werkzeug aufgerufen wird (siehe [toolObj.setPI\(\)](#)).

Ein Eigenschafteninspektor für Formen hat die folgenden Steuerungen:

stroke	fill
--------	------

Ein Eigenschafteninspektor für Text hat die folgenden Steuerungen:

type	font	pointsize
color	bold	italic
direction	alignLeft	alignCenter
alignRight	alignJustify	spacing
position	autoKern	small
rotation	format	lineType
selectable	html	border
deviceFonts	varEdit	options
link	maxChars	target

Der Eigenschafteninspektor für Filme hat die folgenden Steuerungen:

size	publish	background
framerate	player	profile

bShow Ein boolescher Wert, der bestimmt, ob die angegebene Steuerung eingeblendet (`true`) oder ausgeblendet (`false`) werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; blendet eine Steuerung im Eigenschafteninspektor ein oder aus. Diese Methode wird nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Mit dem folgenden Befehl in der JavaScript-Datei eines erweiterbaren Werkzeugs wird Flash angewiesen, für dieses Werkzeug im Eigenschafteninspektor keine Füllungsoptionen anzuzeigen:

```
fl.tools.activeTool.showPControl("fill", false);
```

toolObj.showTransformHandles()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
toolObj.showTransformHandles(bShow)
```

Parameter

bShow Ein boolescher Wert, der bestimmt, ob die Griffe für die freie Transformation für das aktuelle Werkzeug eingeblendet (`true`) oder ausgeblendet (`false`) werden sollen.

Rückgabewerte

Keine.

Beschreibung

Methode; wird in der Methode `configureTool()` der JavaScript-Datei eines erweiterbaren Werkzeugs aufgerufen und gibt an, dass die freien Transformationsgriffe angezeigt werden sollen, wenn das Werkzeug aktiv ist. Diese Methode wird nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Beispiel

Siehe `configureTool()`.

Kapitel 47: Tools-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Sie können vom `flash`-Objekt (`fl.tools`) aus auf das Tools-Objekt zugreifen. Die Eigenschaft `tools.toolObjs` enthält ein Array von `ToolObj`-Objekten und die Eigenschaft `tools.activeTool` gibt das `ToolObj`-Objekt für das gegenwärtig aktive Werkzeug zurück. (Siehe auch [ToolObj-Objekt](#) und die Liste der erweiterbaren Werkzeuge unter „[Top-Level-Funktionen und -Methoden](#)“ auf Seite 15.)

Hinweis: Die folgenden Methoden und Eigenschaften werden nur beim Erstellen erweiterbarer Werkzeuge eingesetzt.

Übersicht über Methoden

Für das Tools-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>tools.constrainPoint()</code>	Erstellt anhand zweier Punkte einen neuen angepassten oder <i>eingeschränkten</i> Punkt.
<code>tools.getKeyDown()</code>	Gibt die zuletzt gedrückte Taste zurück.
„ <code>tools.setCreatingBbox()</code> “ auf Seite 521	PLATZHALTER
<code>tools.setCursor()</code>	Weist dem Zeiger ein bestimmtes Erscheinungsbild zu.
<code>tools.snapPoint()</code>	Erstellt anhand eines angegebenen Punkts einen neuen Punkt, der an dem nächsten geometrischen Objekt ausgerichtet ist.

Übersicht über Eigenschaften

Für das Tools-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>tools.activeTool</code>	Schreibgeschützt; gibt das ToolObj-Objekt für das gegenwärtig aktive Werkzeug zurück.
<code>tools.altIsDown</code>	Schreibgeschützt; ein boolescher Wert, der angibt, ob die Alt-Taste gedrückt ist.
<code>tools.ctrlIsDown</code>	Schreibgeschützt; ein boolescher Wert, der angibt, ob die Strg-Taste gedrückt ist.
<code>tools.mouseIsDown</code>	Schreibgeschützt; ein boolescher Wert, der angibt, ob die linke Maustaste gegenwärtig gedrückt ist.
<code>tools.penDownLoc</code>	Schreibgeschützt; ein Punkt, der die Position des letzten Mouse-down-Ereignisses auf der Bühne angibt.
<code>tools.penLoc</code>	Schreibgeschützt; ein Punkt, der die aktuelle Position der Maus angibt.
<code>tools.shiftIsDown</code>	Schreibgeschützt; ein boolescher Wert, der angibt, ob die Umschalttaste gedrückt ist.
<code>tools.toolObjs</code>	Schreibgeschützt; ein Array von <code>ToolObj</code> -Objekten.

tools.activeTool

Verfügbarkeit

Flash MX 2004.

Verwendung

`tools.activeTool`

Beschreibung

Schreibgeschützte Eigenschaft; gibt das [ToolObj-Objekt](#) für das gegenwärtig aktive Werkzeug zurück.

Beispiel

Im folgenden Beispiel wird ein Objekt, das das gegenwärtig aktive Werkzeug darstellt, in der Variablen `theTool` gespeichert:

```
var theTool = fl.tools.activeTool;
```

tools.altIsDown

Verfügbarkeit

Flash MX 2004.

Verwendung

`tools.altIsDown`

Beschreibung

Schreibgeschützte Eigenschaft; ein boolescher Wert, der angibt, ob die Alt-Taste gedrückt ist. Der Wert lautet `true`, wenn die Alt-Taste gedrückt ist, ansonsten `false`.

Beispiel

Im folgenden Beispiel wird festgestellt, ob die Alt-Taste gegenwärtig gedrückt ist:

```
var isAltDown = fl.tools.altIsDown;
```

tools.constrainPoint()

Verfügbarkeit

Flash MX 2004.

Verwendung

`tools.constrainPoint(pt1, pt2)`

Parameter

pt1, **pt2** Bestimmen den Punkt, an dem der erste Mausklick erfolgt, und den Punkt, zu dem der Zeiger gezogen wird.

Rückgabewerte

Ein neuer angepasster oder eingeschränkter Punkt.

Beschreibung

Methode; erstellt anhand zweier Punkte einen neuen angepassten oder eingeschränkten Punkt. Wenn bei der Ausführung des Befehls die Umschalttaste gedrückt ist, wird der zurückgegebene Punkt so eingeschränkt, dass er entweder einer 45°-Linie folgt (hilfreich beispielsweise bei einer Linie mit einer Pfeilspitze) oder dass das Seitenverhältnis eines Objekts erhalten bleibt (beispielsweise um mit dem Rechteckwerkzeug ein perfektes Quadrat zu zeichnen).

Beispiel

Im folgenden Beispiel wird ein Punkt mit eingeschränkten Positionsmöglichkeiten zurückgegeben:

```
pt2 = fl.tools.constrainPoint(pt1, tempPt);
```

tools.ctlsDown

Verfügbarkeit

Flash MX 2004.

Verwendung

```
tools.ctlsDown
```

Beschreibung

Schreibgeschützte Eigenschaft; ein boolescher Wert. Bei dem Wert `true` ist die Strg-Taste gedrückt und bei `false` ist sie nicht gedrückt.

Beispiel

Im folgenden Beispiel wird festgestellt, ob die Strg-Taste gedrückt ist:

```
var isCtrlDown = fl.tools.ctrlIsDown;
```

tools.getKeyDown()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
tools.getKeyDown()
```

Parameter

Keine.

Rückgabewerte

Der ganzzahlige Wert der Taste.

Beschreibung

Methode; gibt die zuletzt gedrückte Taste zurück.

Beispiel

Im folgenden Beispiel wird der ganzzahlige Wert der zuletzt gedrückten Taste angezeigt:

```
var theKey = fl.tools.getKeyDown();  
fl.trace(theKey);
```

tools.mouseIsDown

Verfügbarkeit

Flash MX 2004.

Verwendung

`tools.mouseIsDown`

Beschreibung

Schreibgeschützte Eigenschaft; ein boolescher Wert. Bei dem Wert `true` ist die linke Maustaste gegenwärtig gedrückt und bei `false` ist sie nicht gedrückt.

Beispiel

Im folgenden Beispiel wird festgestellt, ob die linke Maustaste gedrückt ist:

```
var isMouseDown = fl.tools.mouseIsDown;
```

tools.penDownLoc

Verfügbarkeit

Flash MX 2004.

Verwendung

`tools.penDownLoc`

Beschreibung

Schreibgeschützte Eigenschaft; ein Punkt, der die Position des letzten Mouse-down-Ereignisses auf der Bühne angibt. Die Eigenschaft `tools.penDownLoc` hat die beiden Eigenschaften `x` und `y`, die der x, y -Position des Mauszeigers entsprechen.

Beispiel

Im folgenden Beispiel wird die Position des letzten Mouse-down-Ereignisses auf der Bühne ermittelt. Anschließend werden die x - und y -Werte im Bedienfeld „Ausgabe“ angezeigt:

```
var pt1 = fl.tools.penDownLoc;  
fl.trace("x,y location of last mouseDown event was " + pt1.x + ", " + pt1.y)
```

Siehe auch

[tools.penLoc](#)

tools.penLoc

Verfügbarkeit

Flash MX 2004.

Verwendung

`tools.penLoc`

Beschreibung

Schreibgeschützte Eigenschaft; ein Punkt, der die aktuelle Position des Mauszeigers angibt. Die Eigenschaft `tools.penLoc` hat die beiden Eigenschaften *x* und *y*, die der *x,y*-Position des Mauszeigers entsprechen.

Beispiel

Im folgenden Beispiel wird die aktuelle Position der Maus ermittelt:

```
var tempPt = fl.tools.penLoc;
```

Siehe auch

[tools.penDownLoc](#)

tools.setCreatingBbox()

Verfügbarkeit

Flash 11.

Verwendung

`tools.setCreatingBbox()`

Parameter

Platzhalter Eine Ganzzahl, die das Erscheinungsbild des Mauszeigers wie folgt definiert:

Rückgabewerte

Keine.

Beschreibung

Methode; weist dem Mauszeiger ein bestimmtes Erscheinungsbild zu.

Beispiel

Im folgenden Beispiel wird dem Zeiger die Form eines schwarzen Pfeils zugewiesen:

```
fl.tools.setCursor(1);
```

tools.setCursor()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
tools.setCursor(cursor)
```

Parameter

cursor Eine Ganzzahl, die das Erscheinungsbild des Mauszeigers wie folgt definiert:

- 0 = Plus-Cursor (+)
- 1 = Schwarzer Pfeil
- 2 = Weißer Pfeil
- 3 = Pfeil mit vier Spitzen
- 4 = Horizontaler Pfeil mit zwei Spitzen
- 5 = Vertikaler Pfeil mit zwei Spitzen
- 6 = X
- 7 = Handcursor

Rückgabewerte

Keine.

Beschreibung

Methode; weist dem Mauszeiger ein bestimmtes Erscheinungsbild zu.

Beispiel

Im folgenden Beispiel wird dem Zeiger die Form eines schwarzen Pfeils zugewiesen:

```
fl.tools.setCursor(1);
```

tools.shiftIsDown

Verfügbarkeit

Flash MX 2004.

Verwendung

```
tools.shiftIsDown
```

Beschreibung

Schreibgeschützte Eigenschaft; ein boolescher Wert. Bei dem Wert `true` ist die Umschalttaste gedrückt und bei `false` ist sie nicht gedrückt.

Beispiel

Im folgenden Beispiel wird festgestellt, ob die Umschalttaste gedrückt ist:

```
var isShiftDown = fl.tools.shiftIsDown;
```

tools.snapPoint()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
tools.snapPoint(pt)
```

Parameter

pt Gibt die Lage des Punkts an, für den Sie einen Ausrichtungspunkt zurückgeben möchten.

Rückgabewerte

Ein neuer Punkt, der ggf. am nächstgelegenen geometrischen Objekt ausgerichtet wird.

Beschreibung

Methode; erstellt anhand eines angegebenen Punkts einen neuen Punkt, der an dem nächsten geometrischen Objekt ausgerichtet ist. Wenn die Ausrichtung in der Flash-Benutzeroberfläche im Menü „Ansicht“ deaktiviert ist, wird der ursprüngliche Punkt zurückgegeben.

Beispiel

Im folgenden Beispiel wird ein neuer Punkt zurückgegeben, der ggf. am nächsten geometrischen Objekt ausgerichtet ist:

```
var theSnapPoint = fl.tools.snapPoint(pt1);
```

tools.toolObjs

Verfügbarkeit

Flash MX 2004.

Verwendung

```
tools.toolObjs
```

Beschreibung

Schreibgeschützte Eigenschaft; ein Array von ToolObj-Objekten (siehe [ToolObj-Objekt](#)).

Kapitel 48: Vertex-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Das Vertex-Objekt ist der Teil der Formdatenstruktur, der die Koordinatendaten enthält.

Übersicht über Methoden

Für das Vertex-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>vertex.getHalfEdge()</code>	Ruft ein HalfEdge-Objekt ab, das diesen Scheitelpunkt verwendet.
<code>vertex.setLocation()</code>	Legt fest, an welcher Stelle der Scheitelpunkt platziert werden soll.

Übersicht über Eigenschaften

Für das Vertex-Objekt stehen die folgenden Eigenschaften zur Auswahl:

Eigenschaft	Beschreibung
<code>vertex.x</code>	Schreibgeschützt; die x-Koordinate des Scheitelpunkts in Pixel.
<code>vertex.y</code>	Schreibgeschützt; die y-Koordinate des Scheitelpunkts in Pixel.

vertex.getHalfEdge()

Verfügbarkeit

Flash MX 2004.

Verwendung

`vertex.getHalfEdge()`

Parameter

Keine.

Rückgabewerte

Ein [HalfEdge-Objekt](#).

Beschreibung

Methode; ruft ein [HalfEdge-Objekt](#) ab, das diesen Scheitelpunkt verwendet.

Beispiel

Im folgenden Beispiel wird veranschaulicht, wie andere Halbkanten am gleichen Scheitelpunkt ausgerichtet werden können:

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge(0);  
var theVertex = hEdge.getVertex();  
var someHEdge = theVertex.getHalfEdge(); // Not necessarily the same half edge  
var theSameVertex = someHEdge.getVertex();  
fl.trace('the same vertex: ' + theSameVertex);
```

vertex.setLocation()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
vertex.setLocation(x, y)
```

Parameter

x Ein Gleitkommawert, der die *x*-Koordinate der Stelle angibt (in Pixel), an der der Scheitelpunkt platziert werden soll.

y Ein Gleitkommawert, der die *y*-Koordinate der Stelle angibt (in Pixel), an der der Scheitelpunkt platziert werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; legt fest, an welcher Stelle der Scheitelpunkt platziert werden soll. Sie müssen `shape.beginEdit()` aufrufen, bevor Sie diese Methode verwenden.

Beispiel

Im folgenden Beispiel wird festgelegt, dass sich der Scheitelpunkt am Nullpunkt befindet:

```
var shape = fl.getDocumentDOM().selection[0];  
shape.beginEdit();  
var hEdge = shape.edges[0].getHalfEdge(0);  
var vertex = hEdge.getVertex();  
var someHEdge = vertex.getHalfEdge();  
var vertex = someHEdge.getVertex();  
// Move the vertex to the origin.  
vertex.setLocation(0.0, 0.0);  
shape.endEdit();
```

vertex.x

Verfügbarkeit

Flash MX 2004.

Verwendung

`vertex.x`

Beschreibung

Schreibgeschützte Eigenschaft; die *x*-Koordinate des Scheitelpunkts in Pixel.

Beispiel

Im folgenden Beispiel werden die *x*- und *y*-Koordinaten des Scheitelpunkts im Bedienfeld „Ausgabe“ angezeigt:

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge(0);  
var vertex = hEdge.getVertex();  
  
fl.trace('x location of vertex is: ' + vertex.x);  
fl.trace('y location of vertex is: ' + vertex.y);
```

vertex.y

Verfügbarkeit

Flash MX 2004.

Verwendung

`vertex.y`

Beschreibung

Schreibgeschützte Eigenschaft; die *y*-Koordinate des Scheitelpunkts in Pixel.

Beispiel

Siehe [vertex.x](#).

Kapitel 49: VideoItem-Objekt

Vererbung [Item-Objekt](#) > VideoItem-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Beim VideoItem-Objekt handelt es sich um eine Unterklasse aus dem [Item-Objekt](#).

Übersicht über Methoden

Neben den Methoden des Item-Objekts steht für das VideoItem-Objekt die folgende Methode zur Verfügung:

Eigenschaft	Beschreibung
<code>videoItem.exportToFLV()</code>	Exportiert das angegebene Element in eine FLV-Datei.

Übersicht über Eigenschaften

Zusätzlich zu den Eigenschaften des Item-Objekts stehen Ihnen in Verbindung mit dem VideoItem-Objekt die folgenden Eigenschaften zur Verfügung:

Eigenschaft	Beschreibung
<code>videoItem.fileLastModifiedDate</code>	Schreibgeschützt; ein String mit einer hexadezimalen Zahl, die die Anzahl der Sekunden angibt, die seit dem 1. Januar 1970 bis zum Änderungsdatum der ursprünglichen Datei zu dem Zeitpunkt vergangen sind, an dem die Datei in die Bibliothek importiert wurde.
<code>videoItem.sourceFileExists</code>	Schreibgeschützt; ein boolescher Wert der angibt, ob die in die Bibliothek importierte Datei noch an dem Speicherort existiert, von dem aus sie importiert wurde.
<code>videoItem.sourceFileIsCurrent</code>	Schreibgeschützt; ein boolescher Wert, der angibt, ob das Dateiänderungsdatum des Bibliothekselements mit dem Änderungsdatum der importierten Datei (auf der Festplatte) identisch ist.
<code>videoItem.sourceFilePath</code>	Schreibgeschützt; ein String, der den Pfad zum Videoelement angibt.
<code>videoItem.videoType</code>	Schreibgeschützt; ein String, der angibt, welchen Videotyp das Element repräsentiert.

videoItem.exportToFLV()

Verfügbarkeit

Flash CS4 Professional.

Verwendung

```
videoItem.exportToFLV(fileURI)
```

Parameter

fileURI Ein String im Format file:/// URI, der den Namen der exportierten Datei angibt.

Rückgabewerte

Ein boolescher Wert; `true`, wenn die Datei erfolgreich exportiert wurde; andernfalls `false`.

Beschreibung

Methode; exportiert das angegebene Element in eine FLV-Datei.

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Video-Element handelt, wird es im folgenden Beispiel als FLV-Datei exportiert:

```
var videoFileURL = "file:///C:/out.flv";  
var libItem = fl.getDocumentDOM().library.items[0];  
libItem.exportToFLV(videoFileURL);
```

videoItem.fileLastModifiedDate

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`videoItem.fileLastModifiedDate`

Beschreibung

Schreibgeschützte Eigenschaft; ein String mit einer hexadezimalen Zahl, die die Anzahl der Sekunden angibt, die seit dem 1. Januar 1970 bis zum Änderungsdatum der ursprünglichen Datei zu dem Zeitpunkt vergangen sind, an dem die Datei in die Bibliothek importiert wurde. Sollte die Datei nicht mehr existieren, lautet dieser Wert "00000000".

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Video-Element handelt, wird im folgenden Beispiel wie oben beschrieben eine hexadezimale Zahl angezeigt:

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("Mod date when imported = " + libItem.fileLastModifiedDate);
```

Siehe auch

[videoItem.sourceFileExists](#), [videoItem.sourceFileIsCurrent](#), [videoItem.sourceFilePath](#), [FLfile.getModificationDate\(\)](#)

videoItem.sourceFileExists

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`videoItem.sourceFileExists`

Beschreibung

Schreibgeschützte Eigenschaft; der boolesche Wert `true`, wenn die in die Bibliothek importierte Datei noch an dem Speicherort existiert, von dem aus sie importiert wurde; andernfalls `false`.

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Video-Element handelt, wird im folgenden Beispiel "true" angezeigt, wenn die in die Bibliothek importierte Datei noch existiert.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("sourceFileExists = "+ libItem.sourceFileExists);
```

Siehe auch

[videoItem.sourceFileIsCurrent](#), [videoItem.sourceFilePath](#)

videoItem.sourceFileIsCurrent

Verfügbarkeit

Flash CS4 Professional.

Verwendung

`videoItem.sourceFileIsCurrent`

Beschreibung

Schreibgeschützte Eigenschaft; der boolesche Wert `true`, wenn das Dateiänderungsdatum des Bibliothekselements mit dem Änderungsdatum der importierten Datei auf der Festplatte identisch ist; andernfalls `false`.

Beispiel

Sofern es sich bei dem ersten Element in der Bibliothek um ein Video-Element handelt, wird im folgenden Beispiel "true" angezeigt, wenn die importierte Datei auf der Festplatte seit dem Import nicht geändert wurde.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("fileIsCurrent = "+ libItem.sourceFileIsCurrent);
```

Siehe auch

[videoItem.fileLastModifiedDate](#), [videoItem.sourceFilePath](#)

videoItem.sourceFilePath

Verfügbarkeit

Flash 8.

Verwendung

`videoItem.sourceFilePath`

Beschreibung

Schreibgeschützte Eigenschaft; ein String im Format file:/// URI, der den Pfad zum Videoelement angibt.

Beispiel

Im folgenden Beispiel werden der Name und der Quelldateipfad aller in der Bibliothek enthaltenen Elemente vom Typ video angezeigt:

```
for (idx in fl.getDocumentDOM().library.items) {  
    if (fl.getDocumentDOM().library.items[idx].itemType == "video") {  
        var myItem = fl.getDocumentDOM().library.items[idx];  
        fl.trace(myItem.name + " source is " + myItem.sourceFilePath);  
    }  
}
```

Siehe auch

[videoItem.sourceFileExists](#)

videoItem.videoType

Verfügbarkeit

Flash 8.

Verwendung

videoItem.videoType

Beschreibung

Schreibgeschützte Eigenschaft; ein String, der angibt, welchen Videotyp das Element repräsentiert. Mögliche Werte sind "embedded video", "linked video" und video.

Beispiel

Im folgenden Beispiel werden der Name und der Typ aller in der Bibliothek enthaltenen Elemente vom Typ video angezeigt:

```
for (idx in fl.getDocumentDOM().library.items) {  
    if (fl.getDocumentDOM().library.items[idx].itemType == "video") {  
        var myItem = fl.getDocumentDOM().library.items[idx];  
        fl.trace(myItem.name + " is " + myItem.videoType);  
    }  
}
```


Kapitel 50: XMLUI-Objekt

Verfügbarkeit

Flash MX 2004.

Beschreibung

Flash 8 unterstützt benutzerdefinierte Dialogfelder, die in einer Untermenge der XML User Interface Language (XMLUI) geschrieben sind. XMLUI-Dialogfelder können von mehreren Flash-Funktionen genutzt werden (z. B. von Befehlen und Verhalten) und ermöglichen so die Bereitstellung einer Benutzeroberfläche für Funktionen, die Sie mithilfe der Erweiterungsfunktionalität entwickeln. Mit dem XMLUI-Objekt können Sie die Eigenschaften eines XMLUI-Dialogfelds abrufen und festlegen, die Einstellungen eines XMLUI-Dialogfelds akzeptieren sowie ein XMLUI-Dialogfeld schließen, ohne die Einstellungen zu akzeptieren. Die XMLUI-Methoden können in Rückrufen (Callbacks) eingesetzt werden, z. B. bei `oncommand`-Prozeduren in Schaltflächen.

Sie können eine `dialog.xml`-Datei schreiben und von der JavaScript-API aus mit der Methode `document.xmlPanel()` aufrufen. Um ein Objekt abzurufen, das das gegenwärtige XMLUI-Dialogfeld darstellt, verwenden Sie `fl.xmlui`.

Übersicht über Methoden

Für das XMLUI-Objekt stehen die folgenden Methoden zur Auswahl:

Methode	Beschreibung
<code>xmlui.accept()</code>	Schließt das aktuelle XMLUI-Dialogfeld mit einem Akzeptieren-Status.
<code>xmlui.cancel()</code>	Schließt das aktuelle XMLUI-Dialogfeld mit einem Abbrechen-Status.
<code>xmlui.get()</code>	Ruft den Wert der angegebenen Eigenschaft des aktuellen XMLUI-Dialogfelds ab.
<code>xmlui.getControlItemElement()</code>	Gibt das aktuelle Steuerelement für die angegebene Steuerung zurück.
<code>xmlui.setEnabled()</code>	Gibt einen booleschen Wert zurück, der angibt, ob die Steuerung aktiviert oder deaktiviert (abgeblendet) ist.
<code>xmlui.getVisible()</code>	Gibt einen booleschen Wert zurück, der angibt, ob die Steuerung ein- oder ausgeblendet ist.
<code>xmlui.set()</code>	Ändert den Wert der angegebenen Eigenschaft des aktuellen XMLUI-Dialogfelds.
<code>xmlui.setControlItemElement()</code>	Legt die Bezeichnung und den Wert des aktuellen Elements fest.
<code>xmlui.setControlItemElements()</code>	Legt die Bezeichnung/Wert-Paare des aktuellen Elements fest.
<code>xmlui.setEnabled()</code>	Aktiviert oder deaktiviert eine Steuerung.
<code>xmlui.setVisible()</code>	Blendet eine Steuerung ein oder aus.

xmlui.accept()

Verfügbarkeit

Flash MX 2004.

Verwendung

`xmlui.accept()`

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; schließt das aktuelle XMLUI-Dialogfeld mit einem Akzeptieren-Status, was mit dem Klicken auf die Schaltfläche „OK“ vergleichbar ist.

Siehe auch

`fl.xmlui`, `document.xmlPanel()`, `xmlui.cancel()`

xmlui.cancel()

Verfügbarkeit

Flash MX 2004.

Verwendung

`xmlui.cancel()`

Parameter

Keine.

Rückgabewerte

Keine.

Beschreibung

Methode; schließt das aktuelle XMLUI-Dialogfeld mit einem Abbrechen-Status, was mit dem Klicken auf die Schaltfläche „Abbrechen“ vergleichbar ist.

Siehe auch

`fl.xmlui`, `document.xmlPanel()`, `xmlui.accept()`

xmlui.get()

Verfügbarkeit

Flash MX 2004.

Verwendung

`xmlui.get(controlPropertyName)`

Parameter

controlPropertyName Ein String, der den Namen der XMLUI-Eigenschaft angibt, deren Wert abgerufen werden soll.

Rückgabewerte

Ein String, der den Wert der angegebenen Eigenschaft darstellt. In den Fällen, in denen der boolesche Wert `true` oder `false` zu erwarten wäre, wird der String `"true"` oder `"false"` zurückgegeben.

Beschreibung

Methode; ruft den Wert der angegebenen Eigenschaft des aktuellen XMLUI-Dialogfelds ab.

Beispiel

Im folgenden Beispiel wird der Wert einer Eigenschaft mit dem Namen `URL` zurückgegeben:

```
fl.xmlui.get("URL");
```

Siehe auch

`fl.xmlui`, `document.xmlPanel()`, `xmlui.getControlItemElement()`, `xmlui.set()`

xmlui.getControlItemElement()

Verfügbarkeit

Flash 8.

Verwendung

```
xmlui.getControlItemElement(controlPropertyName)
```

Parameter

controlPropertyName Ein String, der die Eigenschaft angibt, deren Steuerelement Sie abrufen möchten.

Rückgabewerte

Ein Objekt, das das aktuelle Steuerelement der Steuerung darstellt, die in *controlPropertyName* angegeben wurde.

Beschreibung

Methode; gibt die Bezeichnung und den Wert der in einer Listen- oder Kombinationsfeldsteuerung ausgewählten Zeile für die Steuerung zurück, die in *controlPropertyName* angegeben wurde.

Beispiel

Im folgenden Beispiel werden die Bezeichnung und der Wert der gegenwärtig ausgewählten Zeile für die Steuerung `myListBox` zurückgegeben:

```
var elem = new Object();  
elem = fl.xmlui.getControlItemElement("myListBox");  
fl.trace("label = " + elem.label + " value = " + elem.value);
```

Siehe auch

```
fl.xmlui, document.xmlPanel(), xmlui.get(), xmlui.setControlItemElement(),  
xmlui.setControlItemElements()
```

xmlui.setEnabled()

Verfügbarkeit

Flash 8.

Verwendung

```
xmlui.setEnabled(controlID)
```

Parameter

controlID Ein String, der das ID-Attribut der Steuerung angibt, deren Status abgerufen werden soll.

Rückgabewerte

Der boolesche Wert `true`, wenn die Steuerung aktiviert ist, ansonsten `false`.

Beschreibung

Methode; gibt einen booleschen Wert zurück, der angibt, ob die Steuerung aktiviert oder deaktiviert (abgeblendet) ist.

Beispiel

Im folgenden Beispiel wird ein Wert zurückgegeben, der angibt, ob die Steuerung mit dem ID-Attribut `myListBox` aktiviert ist:

```
var isEnabled = fl.xmlui.setEnabled("myListBox");  
fl.trace(isEnabled);
```

Siehe auch

```
fl.xmlui, document.xmlPanel(), xmlui.setEnabled()
```

xmlui.getVisible()

Verfügbarkeit

Flash 8.

Verwendung

```
xmlui.getVisible(controlID)
```

Parameter

controlID Ein String, der das ID-Attribut der Steuerung angibt, deren Sichtbarkeit abgerufen werden soll.

Rückgabewerte

Der boolesche Wert `true`, falls die Steuerung eingeblendet ist, bzw. `false`, falls sie ausgeblendet (versteckt) ist.

Beschreibung

Methode; gibt einen booleschen Wert zurück, der angibt, ob die Steuerung ein- oder ausgeblendet ist.

Beispiel

Im folgenden Beispiel wird ein Wert zurückgegeben, der angibt, ob die Steuerung mit dem ID-Attribut `myListBox` eingeblendet ist:

```
var isVisible = fl.xmlui.getVisible("myListBox");  
fl.trace(isVisible);
```

Siehe auch

[xmlui.setVisible\(\)](#)

xmlui.set()

Verfügbarkeit

Flash MX 2004.

Verwendung

```
xmlui.set(controlPropertyName, value)
```

Parameter

controlPropertyName Ein String, der den Namen der zu ändernden XMLUI-Eigenschaft angibt.

value Ein String mit dem Wert, der der XMLUI-Eigenschaft zugewiesen werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; ändert den Wert der angegebenen Eigenschaft des aktuellen XMLUI-Dialogfelds.

Beispiel

Im folgenden Beispiel wird der Wert der Eigenschaft „URL“ auf `www.adobe.com` eingestellt:

```
fl.xmlui.set("URL", "www.adobe.com");
```

Siehe auch

[fl.xmlui.document.xmlPanel\(\)](#), [xmlui.get\(\)](#), [xmlui.setControlItemElement\(\)](#),
[xmlui.setControlItemElements\(\)](#)

xmlui.setControlItemElement()

Verfügbarkeit

Flash 8.

Verwendung

```
xmlui.setControlItemElement(controlPropertyName, elementItem)
```

Parameter

controlPropertyName Ein String, der das festzulegende Steuerelement angibt.

elementItem Ein JavaScript-Objekt mit einer Stringeigenschaft mit dem Namen `label` und einer optionalen Stringeigenschaft mit dem Namen `value`. Wenn Sie die Eigenschaft `value` auslassen, wird sie erstellt und sie erhält automatisch den gleichen Wert wie `label`.

Rückgabewerte

Keine.

Beschreibung

Methode; gibt die Bezeichnung und den Wert der gegenwärtig ausgewählten Zeile in einer Listen- oder Kombinationsfeldsteuerung zurück, die in *controlPropertyName* angegeben wurde.

Beispiel

Im folgenden Beispiel werden die Bezeichnung und der Wert für das aktuelle Element der Steuerungseigenschaft mit dem Namen `PhoneNumber` eingestellt:

```
var elem = new Object();  
elem.label = "Fax";  
elem.value = "707-555-5555";  
fl.xmlui.setControlItemElement("PhoneNumber", elem);
```

Siehe auch

```
fl.xmlui, document.xmlPanel(), xmlui.getControlItemElement(), xmlui.set(),  
xmlui.setControlItemElements()
```

xmlui.setControlItemElements()

Verfügbarkeit

Flash 8.

Verwendung

```
xmlui.setControlItemElements(controlID, elementItemArray)
```

Parameter

controlID Ein String, der das ID-Attribut der Steuerung angibt, deren Status festgelegt werden soll.

elementItemArray Ein Array von JavaScript-Objekten. Jedes dieser Objekte hat eine Stringeigenschaft mit dem Namen `label` und eine optionale Stringeigenschaft mit dem Namen `value`. Wenn Sie die Eigenschaft `value` auslassen, wird sie erstellt und sie erhält automatisch den gleichen Wert wie `label`.

Rückgabewerte

Keine.

Beschreibung

Methode; löscht die Werte der durch *controlID* bezeichneten Listen- oder Kombinationsfeldsteuerung und ersetzt die Liste bzw. die Menüelemente durch die *label,value*-Paare, die in *elementItemArray* angegeben wurden.

Beispiel

Im folgenden Beispiel werden die Bezeichnung und der Wert der Elemente in der Steuerung mit dem ID-Attribut *myControlID* auf die angegebenen *label,value*-Paare eingestellt:

```
var nameArray = new Array("January", "February", "March");
var monthArray = new Array();
for (i=0;i<nameArray.length;i++){
    elem = new Object();
    elem.label = nameArray[i];
    elem.value = i;
    monthArray[i] = elem;
}
fl.xmlui.setControlItemElements("myControlID", monthArray);
```

Siehe auch

[xmlui.getControlItemElement\(\)](#), [xmlui.set\(\)](#), [xmlui.setControlItemElement\(\)](#)

xmlui.setEnabled()

Verfügbarkeit

Flash 8.

Verwendung

```
xmlui.setEnabled(controlID, enable)
```

Parameter

controlID Ein String, der das ID-Attribut der Steuerung angibt, die aktiviert oder deaktiviert werden soll.

enable Der boolesche Wert *true*, wenn die Steuerung aktiviert, bzw. *false*, wenn sie deaktiviert (abgeblendet) werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; aktiviert oder deaktiviert eine Steuerung.

Beispiel

Im folgenden Beispiel wird die Steuerung mit dem ID-Attribut *myControl* abgeblendet:

```
fl.xmlui.setEnabled("myControl", false);
```

Siehe auch

[xmlui.setEnabled\(\)](#)

xmlui.setVisible()

Verfügbarkeit

Flash 8.

Verwendung

```
xmlui.setVisible(controlID, visible)
```

Parameter

controlID Ein String, der das ID-Attribut der Steuerung angibt, die ein- bzw. ausgeblendet werden soll.

visible Der boolesche Wert `true`, wenn die Steuerung eingeblendet, bzw. `false`, wenn sie ausgeblendet werden soll.

Rückgabewerte

Keine.

Beschreibung

Methode; blendet eine Steuerung ein oder aus.

Beispiel

Im folgenden Beispiel wird die Steuerung mit dem ID-Attribut `myControl` ausgeblendet:

```
fl.xmlui.setVisible("myControl", false);
```

Siehe auch

[xmlui.getVisible\(\)](#)

Kapitel 51: C-Level-Erweiterbarkeit

Dank der in diesem Kapitel beschriebenen C-Level-Erweiterbarkeit lassen sich Erweiterungsdateien für Adobe Flash CS4 Professional mit einer Kombination aus JavaScript und benutzerdefiniertem C-Code implementieren. In dieser Version von Flash wurden keine Änderungen an der C-Level-Erweiterbarkeit vorgenommen.

Erweiterbarkeit

Um die Erweiterbarkeit zu implementieren, werden Funktionen mit C definiert und in einer Dynamic Linked Library (DDL) oder einer Shared Library zusammengefasst. Anschließend wird die Bibliothek im entsprechenden Verzeichnis gespeichert. Die Funktionen können dann von JavaScript aus mit dem Adobe Flash JavaScript API aufgerufen werden.

Beispielsweise könnten Sie zur Leistungssteigerung eine Funktion definieren, die aufwändige Berechnungen effizienter als JavaScript ausführt, oder um damit fortschrittlichere Tools oder Effekte zu erstellen.

Bei dieser Erweiterbarkeit handelt es sich um einen Teilsatz der Adobe Dreamweaver CS3 API. Wenn Sie mit dieser API vertraut sind, werden Sie wahrscheinlich die Funktionen in der API der C-Level-Erweiterbarkeit erkennen. Diese API unterscheidet sich jedoch wie folgt von der Dreamweaver API:

- Diese API enthält nicht alle der Befehle in der Dreamweaver API.
- Alle Deklarationen vom Typ `wchar_t` und `char` der Dreamweaver API werden in dieser API als `unsigned short`-Deklarationen implementiert, um Unicode bei der Übergabe von Strings zu unterstützen.
- Die `JSVal JS_BytesToValue()`-Funktion dieser API ist nicht Teil der Dreamweaver API.
- Die DLL- oder Shared Library-Dateien werden an einem anderen Speicherort gespeichert (siehe „[Integrieren von C-Funktionen](#)“ auf Seite 539).

Integrieren von C-Funktionen

Dank der C-Level-Erweiterbarkeit lassen sich Erweiterungsdateien für Flash mit einer Kombination aus JavaScript und C-Code implementieren. In den folgenden Schritten werden das Verfahren zur Implementierung dieser Funktion zusammengefasst:

- 1 Definieren Sie Funktionen mit der Sprache C oder C++.
- 2 Fassen Sie sie in einer DLL-Datei (Windows) oder einer Shared Library (Macintosh) zusammen.
- 3 Speichern Sie die DLL-Datei oder Bibliothek am entsprechenden Speicherort:

- Windows 7:

Startlaufwerk\Benutzer\Benutzername\Anwendungsdaten\Adobe\Flash CS5 oder CS5.5\Sprache\Configuration\External Libraries

- Windows Vista:

Startlaufwerk\Benutzer\Benutzername\Lokale Einstellungen\Anwendungsdaten\Adobe\Flash CS5 oder CS5.5\Sprache\Configuration\External Libraries

- Windows XP:

*Startlaufwerk\Dokumente und Einstellungen\Benutzername\Lokale
Einstellungen\Anwendungsdaten\Adobe\Flash CS5 oder CS5.5\Sprache\Configuration\External Libraries*

- Mac OS X:

*Macintosh HD/Users/Benutzername/Library/Application Support/Adobe/Flash CS5 oder
CS5.5/Sprache/Configuration/External Libraries*

4 Erstellen Sie eine JSFL-Datei, mit der die Funktionen aufgerufen werden.

5 Führen Sie die JSFL-Datei über das Menü „Befehle“ in der Flash-Authoring-Umgebung aus.

Weitere Informationen finden Sie unter „[DLL-Beispielimplementierung](#)“ auf Seite 543.

C-Level-Erweiterbarkeit und der JavaScript-Interpreter

Der C-Code in der DLL oder Shared Library interagiert dreimal mit der Flash JavaScript API:

- beim Start, wenn die Bibliotheksfunktionen registriert werden
- beim Aufrufen der C-Funktion, um die Argumente, die von JavaScript an C übergeben werden, zu entpacken
- bevor die C-Funktion den Rückgabewert verpackt

Zu diesem Zweck definiert der Interpreter mehrere Datentypen und stellt eine API bereit. Die Definitionen für die in diesem Abschnitt aufgeführten Datentypen und Funktionen befinden sich in der Datei `mm_jsapi.h`. Damit die Bibliothek ordnungsgemäß arbeitet, müssen Sie die Datei `mm_jsapi.h` am Anfang der Bibliothek mit der folgenden Zeile aufnehmen:

```
#include "mm_jsapi.h"
```

Zusammen mit der Datei `mm_jsapi.h` wird die Datei `mm_jsapi_environment.h` aufgenommen, welche die `MM_Environment`-Struktur definiert.

Extrahieren Sie die Datei `mm_jsapi.h` aus der ZIP- oder SIT-Beispieldatei (siehe „[DLL-Beispielimplementierung](#)“ auf Seite 543) oder kopieren Sie den folgenden Code in eine `mm_jsapi.h` genannte Datei:

```
#ifndef _MM_JSAPI_H_
#define _MM_JSAPI_H_

/*****
 * Public data types
 *****/

typedef struct JSContext JSContext;
typedef struct JSObject JSObject;
typedef long jsval;
#ifndef JSBool
typedef long JSBool;
#endif

typedef JSBool (*JSNative)(JSContext *cx, JSObject *obj, unsigned int argc,
jsval *argv, jsval *rval);

/* Possible values for JSBool */
#define JS_TRUE 1
#define JS_FALSE 0
```

```

/*****
 * Public functions
 *****/

/* JSBool JS_DefineFunction(unsigned short *name, JSNative call, unsigned int nargs) */
#define JS_DefineFunction(n, c, a) \
(mmEnv.defineFunction ? (*(mmEnv.defineFunction))(mmEnv.libObj, n, c, a) \
: JS_FALSE)

/* unsigned short *JS_ValueToString(JSContext *cx, jsval v, unsigned int *pLength) */
#define JS_ValueToString(c, v, l) \
(mmEnv.valueToString? (*(mmEnv.valueToString))(c, v, l) : (char *)0)

/* unsigned char *JS_ValueToBytes(JSContext *cx, jsval v, unsigned int *pLength) */
#define JS_ValueToBytes(c, v, l) \
(mmEnv.valueToBytes? (*(mmEnv.valueToBytes))(c, v, l) : (unsigned char *)0)

/* JSBool JS_ValueToInteger(JSContext *cx, jsval v, long *lp); */
#define JS_ValueToInteger(c, v, l) \
(mmEnv.valueToInteger ? (*(mmEnv.valueToInteger))(c, v, l) : JS_FALSE)

/* JSBool JS_ValueToDouble(JSContext *cx, jsval v, double *dp); */
#define JS_ValueToDouble(c, v, d) \
(mmEnv.valueToDouble? (*(mmEnv.valueToDouble))(c, v, d) : JS_FALSE)

/* JSBool JS_ValueToBoolean(JSContext *cx, jsval v, JSBool *bp); */
#define JS_ValueToBoolean(c, v, b) \
(mmEnv.valueToBoolean ? (*(mmEnv.valueToBoolean))(c, v, b) : JS_FALSE)

/* JSBool JS_ValueToObject(JSContext *cx, jsval v, JSObject **op); */
#define JS_ValueToObject(c, v, o) \
(mmEnv.valueToObject? (*(mmEnv.valueToObject))(c, v, o) : JS_FALSE)

/* JSBool JS_StringToValue(JSContext *cx, unsigned short *bytes, uint sz, jsval *vp); */
#define JS_StringToValue(c, b, s, v) \
(mmEnv.stringToValue? (*(mmEnv.stringToValue))(c, b, s, v) : JS_FALSE)

/* JSBool JS_BytesToValue(JSContext *cx, unsigned char *bytes, uint sz, jsval *vp); */
#define JS_BytesToValue(c, b, s, v) \
(mmEnv.bytesToValue? (*(mmEnv.bytesToValue))(c, b, s, v) : JS_FALSE)

/* JSBool JS_DoubleToValue(JSContext *cx, double dv, jsval *vp); */
#define JS_DoubleToValue(c, d, v) \
(mmEnv.doubleToValue? (*(mmEnv.doubleToValue))(c, d, v) : JS_FALSE)

/* jsval JS_IntegerToValue(long lv); */
#define JS_IntegerToValue(lv) (((jsval)(lv) << 1) | 0x1)

/* jsval JS_BooleanToValue(JSBool bv); */
#define JS_BooleanToValue(bv) (((jsval)(bv) << 3) | 0x6)

/* jsval JS_ObjectToValue(JSObject *obj); */
#define JS_ObjectToValue(ov) ((jsval)(ov))

/* unsigned short *JS_ObjectType(JSObject *obj); */
#define JS_ObjectType(o) \
(mmEnv.objectType ? (*(mmEnv.objectType))(o) : (char *)0)

```

```
/* JSObject *JS_NewArrayObject(JSContext *cx, unsigned int length, jsval *v) */
#define JS_NewArrayObject(c, l, v) \
(mmEnv.newArrayObject ? (*(mmEnv.newArrayObject))(c, l, v) : (JSObject *)0)

/* long JS_GetArrayLength(JSContext *cx, JSObject *obj) */
#define JS_GetArrayLength(c, o) \
(mmEnv.getArrayLength ? (*(mmEnv.getArrayLength))(c, o) : -1)

/* JSBool JS_GetElement(JSContext *cx, JSObject *obj, jsint idx, jsval *vp) */
#define JS_GetElement(c, o, i, v) \
(mmEnv.getElement ? (*(mmEnv.getElement))(c, o, i, v) : JS_FALSE)

/* JSBool JS_SetElement(JSContext *cx, JSObject *obj, jsint idx, jsval *vp) */
#define JS_SetElement(c, o, i, v) \
(mmEnv.setElement ? (*(mmEnv.setElement))(c, o, i, v) : JS_FALSE)

/* JSBool JS_ExecuteScript(JSContext *cx, JSObject *obj, unsigned short *script,
    * unsigned int sz, jsval *rval) */
#define JS_ExecuteScript(c, o, s, z, r) \
(mmEnv.executeScript? (*(mmEnv.executeScript))(c, o, s, z, (LPCTSTR) __FILE__, \
    __LINE__, r) : JS_FALSE)

/* JSBool JS_ReportError(JSContext *cx, unsigned short *error, unsigned int sz) */
#define JS_ReportError(c, e, s) \
(mmEnv.reportError? (*(mmEnv.reportError))(c, e, s) : JS_FALSE)

/*****
    * Private data types, macros, and globals
    *****/

typedef struct {
JSObject *libObj;
JSBool (*defineFunction)(JSObject *libObj, unsigned short *name, JSNative call,
    unsigned int nargs);
unsigned short (*valueToString)(JSContext *cx, jsval v, unsigned int *pLength);
unsigned char *(*valueToBytes)(JSContext *cx, jsval v, unsigned int *pLength);
JSBool (*valueToInteger)(JSContext *cx, jsval v, long *lp);
JSBool (*valueToDouble)(JSContext *cx, jsval v, double *dp);
JSBool (*valueToBoolean)(JSContext *cx, jsval v, JSBool *bp);
JSBool (*valueToObject)(JSContext *cx, jsval v, JSObject **op);
JSBool (*stringToValue)(JSContext *cx, unsigned short *b, unsigned int sz, jsval *vp);
JSBool (*bytesToValue)(JSContext *cx, unsigned char *b, unsigned int sz, jsval *vp);
JSBool (*doubleToValue)(JSContext *cx, double dv, jsval *vp);
unsigned short (*objectType)(JSObject *obj);
JSObject *(*newArrayObject)(JSContext *cx, unsigned int length, jsval *vp);
long (*getArrayLength)(JSContext *cx, JSObject *obj);
JSBool (*getElement)(JSContext *cx, JSObject *obj, unsigned int idx,
    jsval *vp);
JSBool (*setElement)(JSContext *cx, JSObject *obj, unsigned int idx,
    jsval *vp);
JSBool (*executeScript)(JSContext *cx, JSObject *obj, unsigned short *script,
    unsigned int sz, unsigned short *file, unsigned int lineNum, jsval *rval);
JSBool (*reportError)(JSContext *cx, unsigned short *error, unsigned int sz);
} MM_Environment;
```

```
extern MM_Environment mmEnv;

// Declare the external entry point and linkage
#ifdef _WIN32
# ifndef _MAC
// Windows
__declspec( dllexport ) void MM_InitWrapper( MM_Environment *env, unsigned int envSize );
# endif
#else
extern void MM_InitWrapper( MM_Environment *env, unsigned int envSize );
#endif

#define MM_STATE\
/* Definitions of global variables */ \
MM_Environment mmEnv; \
\
void\
MM_InitWrapper(MM_Environment *env, unsigned int envSize) \
{ \
extern void MM_Init();\
\
char **envPtr = (char **)env; \
char **mmPtr = (char **)(&mmEnv);\
char **envEnd = (char **)((char *)envPtr + envSize);\
char **mmEnd = (char **)((char *)mmPtr+ sizeof(MM_Environment)); \
\
/* Copy fields from env to mmEnv, one pointer at a time */\
while (mmPtr < mmEnd && envPtr < envEnd)\
*mmPtr++ = *envPtr++; \
\
/* If env doesn't define all of mmEnv's fields, set extras to NULL */ \
while (mmPtr < mmEnd) \
*mmPtr++ = (char *)0; \
\
/* Call user's MM_Init function */\
MM_Init();\
} \

#endif /* _MM_JSAPI_H_ */
```

DLL-Beispielimplementierung

In diesem Abschnitt wird das Erstellen einer einfachen DLL-Implementierung erläutert. Wenn Sie sehen möchten, wie das Verfahren funktioniert, ohne die DLL selbst zu erstellen, können Sie die in der Datei Samples.zip bereitgestellten DLL-Beispieldateien installieren; die Dateien befinden sich im Ordner ExtendingFlash/dllSampleComputeSum. (Weitere Informationen zum Herunterladen der Datei Samples.zip finden Sie unter „[Implementierungsbeispiele](#)“ auf Seite 13.) Extrahieren Sie die Beispieldateien aus der Datei dllSampleComputeSum.dmg oder dllSampleComputeSum.zip und fahren Sie dann wie folgt fort:

- Speichern Sie die Datei Sample.jsfl im Verzeichnis Configuration/Commands (siehe „[Speichern von JSFL-Dateien](#)“ auf Seite 2).
- Speichern Sie die Datei Sample.dll im Verzeichnis Configuration/External Libraries (siehe „[Integrieren von C-Funktionen](#)“ auf Seite 539).

- Wählen Sie in der Flash-Authoring-Umgebung die Option „Befehle“ > „Beispiel“. Durch die „trace“-Anweisung in der JSFL-Datei werden die Ergebnisse der in Sample.dll definierten Funktion an das Bedienfeld „Ausgabe“ gesendet.

Im Rest dieses Abschnitts wird die Entwicklung des Beispiels erörtert. In diesem Fall enthält die DLL nur eine Funktion, die zwei Zahlen hinzufügt. Im folgenden Beispiel sehen Sie den C-Code:

```
// Source code in C
// Save the DLL or shared library with the name "Sample".
#include <windows.h>
#include <stdlib.h>

#include "mm_jsapi.h"

// A sample function
// Every implementation of a JavaScript function must have this signature.
JSBool computeSum(JSContext *cx, JSObject *obj, unsigned int argc, jsval *argv, jsval *rval)
{
    long a, b, sum;

    // Make sure the right number of arguments were passed in.
    if (argc != 2)
        return JS_FALSE;

    // Convert the two arguments from jsvals to longs.
    if (JS_ValueToInteger(cx, argv[0], &a) == JS_FALSE ||
        JS_ValueToInteger(cx, argv[1], &b) == JS_FALSE)
        return JS_FALSE;

    /* Perform the actual work. */
    sum = a + b;

    /* Package the return value as a jsval. */
    *rval = JS_IntegerToValue(sum);

    /* Indicate success. */
    return JS_TRUE;
}
```

Nachdem Sie den Code geschrieben haben, erstellen Sie die DLL-Datei oder Shared Library und speichern sie im entsprechenden Verzeichnis Configuration/External Libraries (siehe „[Integrieren von C-Funktionen](#)“ auf Seite 539). Erstellen Sie dann mithilfe des folgenden Codes eine JSFL-Datei und speichern Sie sie im Verzeichnis Configuration/Commands (siehe „[Speichern von JSFL-Dateien](#)“ auf Seite 2).

```
// JSFL file to run C function defined above.
var a = 5;
var b = 10;
var sum = Sample.computeSum(a, b);
fl.trace("The sum of " + a + " and " + b + " is " + sum );
```

Wählen Sie in der Flash-Authoring-Umgebung die Option „Befehle“ > „Beispiel“, um die in der DLL definierte Funktion auszuführen.

Datentypen

Der JavaScript-Interpreter definiert die in diesem Abschnitt beschriebenen Datentypen.

typedef struct JSContext JSContext

An die C-Level-Funktion wird ein Zeiger für diesen unbekannten Datentyp übergeben. Einige API-Funktionen akzeptieren diesen Zeiger als Argument.

typedef struct JSObject JSObject

An die C-Level-Funktion wird ein Zeiger für diesen unbekannten Datentyp übergeben. Dieser Datentyp repräsentiert ein Objekt (Array-Objekt oder anderer Objekttyp).

typedef struct jsval jsval

Eine unbekannte Datenstruktur kann einen Ganzzahlenwert oder einen Zeiger für eine Gleitkommazahl, einen String oder ein Objekt enthalten. Einige API-Funktionen können zum Einlesen der Werte der Funktionsargumente verwendet werden (indem der Inhalt von `jsval` gelesen wird), während andere API-Funktionen zum Schreiben des Rückgabewerts dienen (durch Schreiben von `jsval`).

typedef enum { JS_FALSE = 0, JS_TRUE = 1 } JSBool

Ein einfacher Datentyp, der einen booleschen Wert enthält.

Die C-Level-API

Die C-Level-Erweiterbarkeits-API besteht aus der Funktionssignatur `JSBool (*JSNative)` und folgenden Funktionen:

- `JSBool JS_DefineFunction()`
- `unsigned short *JS_ValueToString()`
- `JSBool JS_ValueToInteger()`
- `JSBool JS_ValueToDouble()`
- `JSBool JS_ValueToBoolean()`
- `JSBool JS_ValueToObject()`
- `JSBool JS_StringToValue()`
- `JSBool JS_DoubleToValue()`
- `JSVal JS_BooleanToValue()`
- `JSVal JS_BytesToValue()`
- `JSVal JS_IntegerToValue()`
- `JSVal JS_ObjectToValue()`
- `unsigned short *JS_ObjectType()`
- `JSObject *JS_NewArrayObject()`

- `long JS_GetArrayLength()`
- `JSBool JS_GetElement()`
- `JSBool JS_SetElement()`
- `JSBool JS_ExecuteScript()`

typedef JSBool (*JSNative)(JSContext *cx, JSObject *obj, unsigned int argc, jsval *argv, jsval *rval)

Beschreibung

Methode; beschreibt C-Level-Implementierungen von JavaScript-Funktionen in folgendem Kontext:

- *cx* ist ein Zeiger auf eine unbekannte `JSContext`-Struktur, der an einige Funktionen der JavaScript-API übergeben werden muss. Diese Variable enthält den Ausführungskontext des Interpreters.
- *obj* ist ein Zeiger auf das Objekt, in dessen Kontext das Skript ausgeführt wird. Während der Skriptaufführung entspricht das Schlüsselwort `this` dem Objekt.
- Die Ganzzahl *argc* ist die Anzahl der Argumente, die an die Funktion übergeben werden.
- Der Zeiger *argv* verweist auf ein Array von `jsval`-Strukturen. Das Array hat eine Länge von *argc* Elementen.
- *rval* ist ein Zeiger auf eine einzelne `jsvals`-Struktur. Der Rückgabewert der Funktion sollte nach **rval* geschrieben werden.

Wenn erfolgreich, gibt die Funktion `JS_TRUE` zurück, ansonsten `JS_FALSE`. Wenn die Funktion den Wert `JS_FALSE` zurückgibt, wird die Ausführung des aktuellen Skripts gestoppt und eine Fehlermeldung angezeigt.

JSBool JS_DefineFunction()

Verwendung

```
JSBool JS_DefineFunction(unsigned short *name, JSNative call, unsigned int nargs)
```

Beschreibung

Methode; registriert eine C-Level-Funktion mit dem JavaScript-Interpreter von Flash. Nachdem `JS_DefineFunction()` die C-Level-Funktion registriert hat, die Sie im *call*-Argument angeben, können Sie sie in einem JavaScript-Skript aufrufen, indem Sie mit dem von Ihnen im *name*-Argument angegebenen Namen darauf verweisen. Bei dem Argument *name* ist die Groß-/Kleinschreibung zu beachten.

Meist wird diese Funktion über `MM_Init()` aufgerufen, das Flash beim Starten aktiviert.

Argumente

```
unsigned short *name, JSNative call, unsigned int nargs
```

- *name* ist der Name der Funktion, die in JavaScript verwendet werden soll.
- *call* ist ein Zeiger auf eine C-Level-Funktion. Diese Funktion muss einen Wert des Typs `JSBool` zurückgeben, der angibt, ob ein Vorgang erfolgreich war oder fehlgeschlagen ist.
- *nargs* ist die von der Funktion erwartete Anzahl der Argumente.

Rückgabewerte

Ein boolescher Wert; `JS_TRUE` weist darauf hin, dass der Vorgang erfolgreich war; `JS_FALSE` weist darauf hin, dass der Vorgang fehlgeschlagen ist.

unsigned short *JS_ValueToString()

Verwendung

```
unsigned short *JS_ValueToString(JSContext *cx, jsval v, unsigned int *pLength)
```

Beschreibung

Methode; extrahiert ein Funktionsargument aus einer `jsval`-Struktur, konvertiert es in einen String (sofern möglich) und gibt den konvertierten Wert zurück.

Hinweis: Sie sollten den zurückgegebenen Pufferzeiger nicht verändern, da sonst die Datenstrukturen des JavaScript-Interpreters beschädigt werden. Wenn Sie den String verändern möchten, müssen Sie die Zeichen in einen anderen Puffer kopieren und einen neuen JavaScript-String erstellen.

Argumente

```
JSContext *cx, jsval v, unsigned int *pLength
```

- `cx` ist der undurchsichtige `JSContext`-Zeiger, der an die JavaScript-Funktion übergeben wird.
- `v` ist die `jsval`-Struktur, aus der der String extrahiert werden soll.
- `pLength` ist ein Zeiger auf eine vorzeichenlose Ganzzahl. Die Funktion setzt `*pLength` mit der Länge des Strings in Byte gleich.

Rückgabewerte

Ein Zeiger auf einen mit null abgeschlossenen String bei einem erfolgreichen Vorgang bzw. auf `null` bei einem nicht erfolgreichen Vorgang. Die Aufrufroutine darf diesen String nach der Bearbeitung nicht freigeben.

JSBool JS_ValueToInteger()

Verwendung

```
JSBool JS_ValueToInteger(JSContext *cx, jsval v, long *lp);
```

Beschreibung

Methode; extrahiert ein Funktionsargument aus einer `jsval`-Struktur, konvertiert es in eine Ganzzahl (sofern möglich) und gibt den konvertierten Wert zurück.

Argumente

```
JSContext *cx, jsval v, long *lp
```

- `cx` ist der undurchsichtige `JSContext`-Zeiger, der an die JavaScript-Funktion übergeben wird.
- `v` ist die `jsval`-Struktur, aus der die Ganzzahl extrahiert werden soll.
- `lp` ist ein Zeiger auf einen 4 Byte langen Ganzzahlwert. Diese Funktion speichert den konvertierten Wert in `*lp`.

Rückgabewerte

Ein boolescher Wert; `JS_TRUE` weist darauf hin, dass der Vorgang erfolgreich war; `JS_FALSE` weist darauf hin, dass der Vorgang fehlgeschlagen ist.

JSBool JS_ValueToDouble()

Verwendung

```
JSBool JS_ValueToDouble(JSContext *cx, jsval v, double *dp);
```

Beschreibung

Methode; extrahiert ein Funktionsargument aus einer `jsval`-Struktur, konvertiert es in einen Double-Wert (sofern möglich) und gibt den konvertierten Wert zurück.

Argumente

`JSContext *cx`, `jsval v`, `double *dp`

- `cx` ist der undurchsichtige `JSContext`-Zeiger, der an die JavaScript-Funktion übergeben wird.
- `v` ist die `jsval`-Struktur, aus der das Double extrahiert werden soll.
- `dp` ist ein Zeiger auf ein 8 Byte langes Double. Diese Funktion speichert den konvertierten Wert in `*dp`.

Rückgabewerte

Ein boolescher Wert; `JS_TRUE` weist darauf hin, dass der Vorgang erfolgreich war; `JS_FALSE` weist darauf hin, dass der Vorgang fehlgeschlagen ist.

JSBool JS_ValueToBoolean()

Verwendung

```
JSBool JS_ValueToBoolean(JSContext *cx, jsval v, JSBool *bp);
```

Beschreibung

Methode; extrahiert ein Funktionsargument aus einer `jsval`-Struktur, konvertiert es in einen booleschen Wert (sofern möglich) und gibt den konvertierten Wert zurück.

Argumente

`JSContext *cx`, `jsval v`, `JSBool *bp`

- `cx` ist der undurchsichtige `JSContext`-Zeiger, der an die JavaScript-Funktion übergeben wird.
- `v` ist die `jsval`-Struktur, aus der der boolesche Wert extrahiert werden soll.
- `bp` ist ein Zeiger auf einen booleschen Wert des Typs `JSBool`. Diese Funktion speichert den konvertierten Wert in `*bp`.

Rückgabewerte

Ein boolescher Wert; `JS_TRUE` weist darauf hin, dass der Vorgang erfolgreich war; `JS_FALSE` weist darauf hin, dass der Vorgang fehlgeschlagen ist.

JSBool JS_ValueToObject()

Verwendung

```
JSBool JS_ValueToObject(JSContext *cx, jsval v, JSObject **op);
```

Beschreibung

Methode; extrahiert ein Funktionsargument aus einer `jsval`-Struktur, konvertiert es in ein Objekt (sofern möglich) und gibt den konvertierten Wert zurück. Sollte es sich bei dem Objekt um ein Array handeln, verwenden Sie `JS_GetArrayLength()` und `JS_GetElement()`, um den Inhalt zu lesen.

Argumente

```
JSContext *cx, jsval v, JSObject **op
```

- `cx` ist der undurchsichtige `JSContext`-Zeiger, der an die JavaScript-Funktion übergeben wird.
- `v` ist die `jsval`-Struktur, aus der das Objekt extrahiert werden soll.
- `op` ist ein Zeiger auf einen `JSObject`-Zeiger. Diese Funktion speichert den konvertierten Wert in `*op`.

Rückgabewerte

Ein boolescher Wert; `JS_TRUE` weist darauf hin, dass der Vorgang erfolgreich war; `JS_FALSE` weist darauf hin, dass der Vorgang fehlgeschlagen ist.

JSBool JS_StringToValue()

Verwendung

```
JSBool JS_StringToValue(JSContext *cx, unsigned short *bytes, uint sz, jsval *vp);
```

Beschreibung

Methode; speichert einen Stringrückgabewert in einer `jsval`-Struktur. Weist ein neues JavaScript-String-Objekt zu.

Argumente

```
JSContext *cx, unsigned short *bytes, size_t sz, jsval *vp
```

- `cx` ist der undurchsichtige `JSContext`-Zeiger, der an die JavaScript-Funktion übergeben wird.
- `bytes` ist der in der `jsval`-Struktur zu speichernde String. Die Stringdaten werden kopiert, sodass der aufrufende Code den String freigeben sollte, sobald er nicht mehr benötigt wird. Wenn die Größe des Strings nicht angegeben wird (siehe Argument `sz`), muss der Wert mit null abgeschlossen werden.
- `sz` ist die Größe des Strings (in Byte). Wenn `sz` gleich 0 ist, wird die Länge des mit null abgeschlossenen Strings automatisch berechnet.
- `vp` ist ein Zeiger auf die `jsval`-Struktur, in die der Inhalt des Strings kopiert werden soll.

Rückgabewerte

Ein boolescher Wert; `JS_TRUE` weist darauf hin, dass der Vorgang erfolgreich war; `JS_FALSE` weist darauf hin, dass der Vorgang fehlgeschlagen ist.

JSBool JS_DoubleToValue()

Verwendung

```
JSBool JS_DoubleToValue(JSContext *cx, double dv, jsval *vp);
```

Beschreibung

Methode; speichert einen Gleitkommazahl-Rückgabewert in einer jsval-Struktur.

Argumente

JSContext *cx, double dv, jsval *vp

- cx ist der undurchsichtige JSContext-Zeiger, der an die JavaScript-Funktion übergeben wird.
- dv ist eine 8 Byte lange Gleitkommazahl.
- vp ist ein Zeiger auf die jsval-Struktur, in die der Inhalt des Double kopiert werden soll.

Rückgabewerte

Ein boolescher Wert; JS_TRUE weist darauf hin, dass der Vorgang erfolgreich war; JS_FALSE weist darauf hin, dass der Vorgang fehlgeschlagen ist.

JSVal JS_BooleanToValue()

Verwendung

```
jsval JS_BooleanToValue(JSBool bv);
```

Beschreibung

Methode; speichert einen booleschen Rückgabewert in einer jsval-Struktur.

Argumente

JSBool bv

- bv ist ein boolescher Wert; JS_TRUE weist darauf hin, dass der Vorgang erfolgreich war; JS_FALSE weist darauf hin, dass der Vorgang fehlgeschlagen ist.

Rückgabewerte

Eine JSVal-Struktur mit dem booleschen Wert, der als Argument an die Funktion übergeben wird.

JSVal JS_BytesToValue()

Verwendung

```
JSBool JS_BytesToValue(JSContext *cx, unsigned short *bytes, uint sz, jsval *vp);
```

Beschreibung

Methode; konvertiert Byte in einen JavaScript-Wert.

Argumente

`JSTContext *cx, unsignedshort *bytes, uintsz, jsval *vp`

- *cx* ist der JavaScript-Kontext.
- *bytes* ist der Byte-String, der in ein JavaScript-Objekt konvertiert werden soll.
- *sz* ist die Anzahl der zu konvertierenden Byte.
- *vp* ist der JavaScript-Wert.

Rückgabewerte

Ein boolescher Wert; `JS_TRUE` weist darauf hin, dass der Vorgang erfolgreich war; `JS_FALSE` weist darauf hin, dass der Vorgang fehlgeschlagen ist.

JSVal JS_IntegerToValue()

Verwendung

```
jsval JS_IntegerToValue(long lv);
```

Beschreibung

Methode; konvertiert einen langen Ganzzahlenwert in eine `JSVal`-Struktur.

Argumente

lv

lv ist der lange Ganzzahlwert, der in eine `jsval`-Struktur konvertiert werden soll.

Rückgabewerte

Eine `JSVal`-Struktur mit der Ganzzahl, die als Argument an die Funktion übergeben wurde.

JSVal JS_ObjectToValue()

Verwendung

```
jsval JS_ObjectToValue(JSObject *obj);
```

Beschreibung

Methode; speichert einen Objekt-Rückgabewert in einer `JSVal`-Struktur. Mit `JS_NewArrayObject()` können Sie ein Array-Objekt erstellen und mit `JS_SetElement()` können Sie den Inhalt des Objekts definieren.

Argumente

`JSObject *obj`

obj ist ein Zeiger auf das `JSObject`-Objekt, das in eine `JSVal`-Struktur konvertiert werden soll.

Rückgabewerte

Eine `JSVal`-Struktur mit dem Objekt, das als Argument an die Funktion übergeben wurde.

unsigned short *JS_ObjectType()

Verwendung

```
unsigned short *JS_ObjectType(JSObject *obj);
```

Beschreibung

Methode; gibt den Klassennamen des Objekts zurück, wenn eine Objektreferenz vorhanden ist. Für ein DOM-Objekt beispielsweise gibt die Funktion den Wert "Document" zurück. Ist das Objekt ein Knoten im Dokument, gibt die Funktion den Wert "Element" zurück. Für ein Array-Objekt gibt die Funktion den Wert "Array" zurück.

Hinweis: Sie sollten den zurückgegebenen Pufferzeiger nicht verändern, da sonst die Datenstrukturen des JavaScript-Interpreters beschädigt werden.

Argumente

JSObject *obj

In der Regel wird dieses Argument mithilfe der Funktion JS_ValueToObject() übergeben und konvertiert.

Rückgabewerte

Ein Zeiger auf einen mit Null abgeschlossenen String. Dieser String sollte vom aufrufenden Code nach Abschluss nicht freigegeben werden.

JSObject *JS_NewArrayObject()

Verwendung

```
JSObject *JS_NewArrayObject(JSContext *cx, unsigned int length [, jsval *v])
```

Beschreibung

Methode; erstellt ein neues Objekt, das ein Array von JSVals enthält.

Argumente

JSContext *cx, unsigned int length, jsval *v

- cx ist der undurchsichtige JSContext-Zeiger, der an die JavaScript-Funktion übergeben wird.
- length ist die Anzahl der Elemente, die das Array enthalten kann.
- v ist ein optionaler Zeiger auf den im Array zu speichernden Wert jsvals. Wenn der Rückgabewert nicht null lautet, ist v ein Array mit length-Elementen. Wenn der Rückgabewert null lautet, ist der anfängliche Inhalt des Array-Objekts nicht definiert und kann mittels JS_SetElement() gesetzt werden.

Rückgabewerte

Ein Zeiger auf ein neues Array-Objekt oder der Wert null, falls ein Fehler auftritt.

long JS_GetArrayLength()

Verwendung

```
long JS_GetArrayLength(JSContext *cx, JSObject *obj)
```

Beschreibung

Methode; fragt die Anzahl der Array-Elemente ab, wenn ein Zeiger auf ein Array-Objekt weist.

Argumente

`JSCContext *cx, JSObject *obj`

- *cx* ist der undurchsichtige `JSCContext`-Zeiger, der an die JavaScript-Funktion übergeben wird.
- *obj* ist ein Zeiger auf ein Array-Objekt.

Rückgabewerte

Die Anzahl der Array-Elemente. Beim Auftreten eines Fehlers wird `-1` zurückgegeben.

JSBool JS_GetElement()

Verwendung

`JSBool JS_GetElement(JSCContext *cx, JSObject *obj, jsint idx, jsval *vp)`

Beschreibung

Methode; liest ein einzelnes Element aus einem Array-Objekt.

Argumente

`JSCContext *cx, JSObject *obj, jsint idx, jsval *vp`

- *cx* ist der undurchsichtige `JSCContext`-Zeiger, der an die JavaScript-Funktion übergeben wird.
- *obj* ist ein Zeiger auf ein Array-Objekt.
- Das Argument *idx* ist ein Ganzzahlenindex im Array. Das erste Element hat den Indexwert `0` und das letzte Element hat den Indexwert `(length - 1)`.
- *vp* ist ein Zeiger auf `jsval`, in das der Inhalt der `jsval`-Struktur des Arrays kopiert werden soll.

Rückgabewerte

Ein boolescher Wert; `JS_TRUE` weist darauf hin, dass der Vorgang erfolgreich war; `JS_FALSE` weist darauf hin, dass der Vorgang fehlgeschlagen ist.

JSBool JS_SetElement()

Verwendung

`JSBool JS_SetElement(JSCContext *cx, JSObject *obj, jsint idx, jsval *vp)`

Beschreibung

Methode; schreibt ein einzelnes Element eines Array-Objekts.

Argumente

`JSCContext *cx, JSObject *obj, jsint idx, jsval *vp`

- *cx* ist der undurchsichtige `JSCContext`-Zeiger, der an die JavaScript-Funktion übergeben wird.
- *obj* ist ein Zeiger auf ein Array-Objekt.

- Das Argument *idx* ist ein Ganzzahlenindex im Array. Das erste Element hat den Indexwert 0 und das letzte Element hat den Indexwert (*length* - 1).
- *vp* ist ein Zeiger auf eine *j sval*-Struktur, deren Inhalt in den Wert *j sval* des Arrays kopiert werden soll.

Rückgabewerte

Ein boolescher Wert; *JS_TRUE* weist darauf hin, dass der Vorgang erfolgreich war; *JS_FALSE* weist darauf hin, dass der Vorgang fehlgeschlagen ist.

JSBool JS_ExecuteScript()

Verwendung

```
JS_ExecuteScript (JSContext *cx, JSObject *obj, unsigned short *script, unsigned int sz, jsval *rval)
```

Beschreibung

Methode; kompiliert einen JavaScript-String und führt ihn aus. Falls das Skript einen Rückgabewert generiert, wird dieser in **rval* zurückgegeben.

Argumente

```
JSContext *cx, JSObject *obj, unsigned short *script, unsigned intsz, jsval *rval
```

- *cx* ist der undurchsichtige *JSContext*-Zeiger, der an die JavaScript-Funktion übergeben wird.
- *obj* ist ein Zeiger auf das Objekt, in dessen Kontext das Skript ausgeführt wird. Während der Skriptausführung entspricht das Schlüsselwort *this* dem Objekt. In der Regel handelt es sich dabei um den Zeiger *JSObject*, der auf die JavaScript Funktion weist.
- *script* ist ein String, der JavaScript-Code enthält. Wenn die Größe des Strings nicht angegeben wird (siehe Argument *sz*), muss der Wert mit Null abgeschlossen werden.
- *sz* ist die Größe des Strings (in Byte). Wenn *sz* gleich 0 ist, wird die Länge des mit null abgeschlossenen Strings automatisch berechnet.
- *rval* ist ein Zeiger auf eine einzelne *j sval*-Struktur. Der Rückgabewert der Funktion wird in **rval* gespeichert.

Rückgabewerte

Ein boolescher Wert; *JS_TRUE* weist darauf hin, dass der Vorgang erfolgreich war; *JS_FALSE* weist darauf hin, dass der Vorgang fehlgeschlagen ist.