

ADOBE® FLASH® PROFESSIONAL CS5 및 CS5.5 확장



마지막 업데이트 2011 년 5 월 12 일

법적 고지 사항

법적 고지 사항은 http://help.adobe.com/ko_KR/legalnotices/index.html을 참조하십시오.

목차

1장: 소개

JavaScript API 작업	1
JavaScript API의 새로운 기능	4
JavaScript API 객체	6
샘플 구현	12

2장: 최상위 함수 및 메서드

activate()	14
alert()	15
configureTool()	15
confirm()	16
deactivate()	17
keyDown()	17
keyUp()	18
mouseDoubleClick()	18
mouseDown()	19
mouseMove()	20
mouseUp()	20
notifySettingsChanged()	21
prompt()	22
setCursor()	22

3장: actionsPanel 객체

actionsPanel.getClassForObject()	24
actionsPanel.getScriptAssistMode()	25
actionsPanel.getSelectedText()	25
actionsPanel.getText()	26
actionsPanel.hasSelection()	27
actionsPanel.replaceSelectedText()	27
actionsPanel.setScriptAssistMode()	28
actionsPanel.setSelection()	29
actionsPanel.setText()	29

4장: BitmapInstance 객체

bitmapInstance.getBits()	31
bitmapInstance.hPixels	32
bitmapInstance.setBits()	32
bitmapInstance.vPixels	33

5장: BitmapItem 객체

bitmapItem.allowSmoothing	34
bitmapItem.compressionType	35
bitmapItem.exportToFile()	35
bitmapItem.fileLastModifiedDate	36
bitmapItem.originalCompressionType	36
bitmapItem.quality	37
bitmapItem.sourceFileExists	37
bitmapItem.sourceFilesCurrent	38
bitmapItem.sourceFilePath	38
bitmapItem.useDeblocking	39
bitmapItem.useImportedJPEGQuality	39

6장: CompiledClipInstance 객체

compiledClipInstance.accName	40
compiledClipInstance.actionScript	41
compiledClipInstance.description	41
compiledClipInstance.forceSimple	41
compiledClipInstance.shortcut	42
compiledClipInstance.silent	42
compiledClipInstance.tabIndex	43

7장: compilerErrors 객체

compilerErrors.clear()	44
compilerErrors.save()	45

8장: ComponentInstance 객체

componentInstance.parameters	46
------------------------------	----

9장: componentsPanel 객체

componentsPanel.addItemToDocument()	47
componentsPanel.reload()	48

10장: Contour 객체

contour.fill	49
contour.getHalfEdge()	50
contour.interior	51
contour.orientation	51

11장: Document 객체

document.accName	59
document.addDataToDocument()	60
document.addDataToSelection()	60
document.addFilter()	61

document.addItem()	62
document.addNewLine()	62
document.addNewOval()	63
document.addNewPrimitiveOval()	64
document.addNewPrimitiveRectangle()	64
document.addNewPublishProfile()	65
document.addNewRectangle()	66
document.addNewScene()	67
document.addNewText()	68
document.align()	68
document.allowScreens()	69
document.arrange()	69
document.as3AutoDeclare	70
document.as3Dialect	70
document.as3ExportFrame	71
document.as3StrictMode	71
document.as3WarningsMode	72
document.asVersion	72
document.autoLabel	73
document.backgroundColor	73
document.breakApart()	74
document.canEditSymbol()	74
document.canRevert()	75
document.canTestMovie()	75
document.canTestScene()	76
document.changeFilterOrder()	76
document.clipCopy()	77
document.clipCut()	78
document.clipPaste()	78
document.close()	79
document.convertLinesToFills()	79
document.convertToSymbol()	80
document.crop()	81
document.currentPublishProfile	81
document.currentTimeline	82
document.debugMovie()	82
document.deleteEnvelope()	83
document.deletePublishProfile()	83
document.deleteScene()	84
document.deleteSelection()	84
document.description	85
document.disableAllFilters()	85

document.disableFilter()	86
document.disableOtherFilters()	86
document.distribute()	87
document.distributeToLayers()	88
document.docClass	88
document.documentHasData()	89
document.duplicatePublishProfile()	89
document.duplicateScene()	90
document.duplicateSelection()	90
document.editScene()	91
document.enableAllFilters()	91
document.enableFilter()	92
document.enterEditMode()	92
document.exitEditMode()	93
document.exportPNG()	93
document.exportPublishProfile()	94
document.exportPublishProfileString()	95
document.exportSWF()	95
document.externalLibraryPath	96
document.forceSimple	96
document.frameRate	97
document.getAlignToDocument()	97
document.getBlendMode()	98
document.getCustomFill()	98
document.getCustomStroke()	99
document.getDataFromDocument()	100
document.getElementProperty()	101
document.getElementTextAttr()	101
document.getFilters()	102
document.getMetadata()	103
document.getMobileSettings()	103
document.getPlayerVersion()	104
document.getSelectionRect()	104
document.getTextString()	105
document.getTimeline()	106
document.getTransformationPoint()	107
document.group()	107
document.height	108
document.id	108
document.importFile()	109
document.importPublishProfile()	109
document.importPublishProfileString()	110

document.importSWF()	110
document.intersect()	111
document.library	111
document.libraryPath	112
document.livePreview	112
document.loadCuepointXML()	113
document.match()	113
document.mouseClick()	114
document.mouseDbClick()	114
document.moveSelectedBezierPointsBy()	115
document.moveSelectionBy()	116
document.name	116
document.optimizeCurves()	117
document.path	117
document.pathURI	118
document.publish()	118
document.publishProfiles	119
document.punch()	119
document.removeAllFilters()	120
document.removeDataFromDocument()	120
document.removeDataFromSelection()	121
document.removeFilter()	121
document.renamePublishProfile()	122
document.renameScene()	122
document.reorderScene()	123
document.resetOvalObject()	123
document.resetRectangleObject()	124
document.resetTransformation()	125
document.revert()	125
document.rotate3DSelection()	126
document.rotateSelection()	126
document.save()	127
document.saveAndCompact()	127
document.scaleSelection()	128
document.screenOutline	129
document.selectAll()	129
document.selection	130
document.selectNone()	131
document.setAlignToDocument()	132
document.setBlendMode()	132
document.setCustomFill()	133
document.setCustomStroke()	134

document.setElementProperty()	134
document.setElementTextAttr()	135
document.setFillColor()	136
document.setFilterProperty()	136
document.setFilters()	137
document.setInstanceAlpha()	138
document.setInstanceBrightness()	138
document.setInstanceTint()	139
document.setMetadata()	139
document.setMobileSettings()	141
document.setOvalObjectProperty()	141
document.setPlayerVersion()	142
document.setRectangleObjectProperty()	142
document.setSelectionBounds()	143
document.setSelectionRect()	144
document.setStageVanishingPoint()	144
document.setStageViewAngle()	145
document.setStroke()	145
document.setStrokeColor()	146
document.setStrokeSize()	147
document.setStrokeStyle()	147
document.setTextRectangle()	148
document.setTextSelection()	148
document.setTextString()	149
document.setTransformationPoint()	150
document.silent	150
document.skewSelection()	151
document.smoothSelection()	151
document.sourcePath	152
document.space()	152
document.straightenSelection()	153
document.swapElement()	154
document.swapStrokeAndFill()	154
document.testMovie()	155
document.testScene()	155
document.timelines	156
document.traceBitmap()	156
document.translate3DCenter()	157
document.translate3DSelection()	157
document.transformSelection()	158
document.unGroup()	159
document.union()	159

document.unlockAllElements()	160
document.viewMatrix	160
document.width	161
document.xmlPanel()	161
document.zoomFactor	162
12장: drawingLayer 객체	
drawingLayer.beginDraw()	163
drawingLayer.beginFrame()	164
drawingLayer.cubicCurveTo()	164
drawingLayer.curveTo()	165
drawingLayer.drawPath()	166
drawingLayer.endDraw()	166
drawingLayer.endFrame()	167
drawingLayer.lineTo()	167
drawingLayer.moveTo()	168
drawingLayer.newPath()	168
drawingLayer.setColor()	169
drawingLayer.setFill()	169
drawingLayer.setStroke()	169
13장: Edge 객체	
edge.cubicSegmentIndex	170
edge.getControl()	171
edge.getHalfEdge()	171
edge.id	172
edge.isLine	172
edge.setControl()	173
edge.splitEdge()	173
edge.stroke	174
14장: Element 객체	
element.depth	176
element.elementType	176
element.getPersistentData()	177
element.getTransformationPoint()	177
element.hasPersistentData()	178
element.height	179
element.layer	179
element.left	179
element.locked	180
element.matrix	180
element.name	181

element.removePersistentData()	181
element.rotation	181
element.scaleX	182
element.scaleY	182
element.selected	183
element.setPersistentData()	183
element.setTransformationPoint()	184
element.skewX	184
element.skewY	185
element.top	185
element.transformX	186
element.transformY	186
element.width	187
element.x	187
element.y	187
15장: Fill 객체	
fill.bitmapplsClipped	189
fill.bitmapPath	190
fill.color	190
fill.colorArray	191
fill.focalPoint	191
fill.linearRGB	192
fill.matrix	192
fill.overflow	193
fill.posArray	193
fill.style	194
16장: Filter 객체	
filter.angle	196
filter.blurX	196
filter.blurY	197
filter.brightness	197
filter.color	198
filter.contrast	198
filter.distance	199
filter.enabled	199
filter.hideObject	200
filter.highlightColor	200
filter.hue	201
filter.inner	201
filter.knockout	202
filter.name	202

filter.quality	203
filter.saturation	203
filter.shadowColor	204
filter.strength	205
filter.type	205
17장: Flash 객체(fl)	
fl.actionsPanel	210
fl.addEventListener()	210
fl.as3PackagePaths	211
fl.browseForFileURL()	211
fl.browseForFolderURL()	212
fl.clearPublishCache()	212
fl.clipCopyString()	213
fl.closeAll()	213
fl.closeAllPlayerDocuments()	214
fl.closeDocument()	215
fl.compilerErrors	215
fl.componentsPanel	216
fl.configDirectory	216
fl.configURI	216
fl.contactSensitiveSelection	217
fl.createDocument()	217
fl.createNewDocList	218
fl.createNewDocListType	218
fl.createNewTemplateList	219
fl.documents	219
fl.drawingLayer	219
fl.exportPublishProfileString()	220
fl.externalLibraryPath	220
fl.fileExists()	221
fl.findDocumentDOM()	221
fl.findDocumentIndex()	222
fl.findObjectInDocByName()	223
fl.findObjectInDocByType()	223
fl.flexSDKPath	225
fl.getAppMemoryInfo()	225
fl.getDocumentDOM()	226
fl.getSwfPanel()	227
fl.installedPlayers	227
fl.isFontInstalled()	228
fl.languageCode	229

fl.libraryPath	229
fl.mapPlayerURL()	229
fl.Math	230
fl.mruRecentFileList	230
fl.mruRecentFileListType	231
fl.objectDrawingMode	231
fl.openDocument()	232
fl.openScript()	232
fl.outputPanel	233
fl.packagePaths	233
fl.presetPanel	234
fl.publishCacheDiskSizeMax	234
fl.publishCacheEnabled	235
fl.publishCacheMemoryEntrySizeLimit	235
fl.publishCacheMemorySizeMax	236
fl.publishDocument()	236
fl.quit()	237
fl.reloadEffects()	237
fl.reloadTools()	238
fl.removeEventListener()	238
fl.resetAS3PackagePaths()	239
fl.resetPackagePaths()	239
fl.revertDocument()	240
fl.runScript()	240
fl.saveAll()	241
fl.saveDocument()	242
fl.saveDocumentAs()	243
fl.scriptURI	243
fl.selectElement()	244
fl.selectTool()	244
fl.setActiveWindow()	245
fl.showIdleMessage()	246
fl.sourcePath	246
fl.swfPanels	247
fl.toggleBreakpoint()	247
fl.tools	248
fl.trace()	248
fl.version	249
fl.xmlui	249

18장: FLfile 객체

FLfile.copy()	251
FLfile.createFolder()	252
FLfile.exists()	252
FLfile.getAttributes()	253
FLfile.getCreationDate()	254
FLfile.getCreationDateObj()	255
FLfile.getModificationDate()	255
FLfile.getModificationDateObj()	256
FLfile.getSize()	257
FLfile.listFolder()	257
FLfile.platformPathToURI()	258
FLfile.read()	259
FLfile.remove()	260
FLfile.setAttributes()	261
FLfile.uriToPlatformPath()	262
FLfile.write()	262

19장: folderItem 객체

20장: fontItem 객체

fontItem.bitmap	265
fontItem.bold	266
fontItem.embeddedCharacters	266
fontItem.embedRanges	266
fontItem.embedVariantGlyphs	267
fontItem.font	268
fontItem.isDefineFont4Symbol	269
fontItem.italic	269
fontItem.size	270

21장: Frame 객체

frame.convertMotionObjectTo2D()	272
frame.convertMotionObjectTo3D()	273
frame.actionScript	273
frame.duration	274
frame.elements	274
frame.getCustomEase()	275
frame.getMotionObjectXML()	275
frame.hasCustomEase	276
frame.hasMotionPath()	276
frame.is3DMotionObject()	277
frame.isMotionObject()	277

frame.labelType	278
frame.motionTweenOrientToPath	278
frame.motionTweenRotate	279
frame.motionTweenRotateTimes	279
frame.motionTweenScale	279
frame.motionTweenSnap	280
frame.motionTweenSync	280
frame.name	280
frame.selectMotionPath()	281
frame.setCustomEase()	281
frame.setMotionObjectDuration()	282
frame.setMotionObjectXML()	283
frame.shapeTweenBlend	283
frame.soundEffect	284
frame.soundLibraryItem	284
frame.soundLoop	284
frame.soundLoopMode	285
frame.soundName	285
frame.soundSync	286
frame.startFrame	286
frame.tweenEasing	286
frame.tweenInstanceName	287
frame.tweenType	287
frame.useSingleEaseCurve	288
22장: HalfEdge 객체	
halfEdge.getEdge()	289
halfEdge.getNext()	290
halfEdge.getOppositeHalfEdge()	290
halfEdge.getPrev()	291
halfEdge.getVertex()	291
halfEdge.id	292
halfEdge.index	292
23장: Instance 객체	
instance.instanceType	294
instance.libraryItem	294
24장: Item 객체	
item.addData()	296
item.getData()	297
item.hasData()	298
item.itemType	298

item.linkageBaseClass	298
item.linkageClassName	299
item.linkageExportForAS	300
item.linkageExportForRS	300
item.linkageExportInFirstFrame	300
item.linkageIdentifier	301
item.linkageImportForRS	301
item.linkageURL	302
item.name	302
item.removeData()	303
25장: Layer 객체	
layer.color	304
layer.frameCount	305
layer.frames	305
layer.height	306
layer.layerType	306
layer.locked	306
layer.name	307
layer.outline	307
layer.parentLayer	307
layer.visible	308
26장: library 객체	
library.addItemToDocument()	310
library.addNewItem()	310
library.deleteItem()	311
library.duplicateItem()	311
library.editItem()	312
library.expandFolder()	313
library.findItemIndex()	313
library.getItemProperty()	314
library.getItemType()	314
library.getSelectedItems()	315
library.importEmbeddedSWF()	315
library.itemExists()	316
library.items	316
library.moveToFolder()	317
library.newFolder()	317
library.renameItem()	318
library.selectAll()	318
library.selectItem()	319
library.selectNone()	320

library.setItemProperty()	320
library.updateItem()	321
27장: Math 객체	
Math.concatMatrix()	322
Math.invertMatrix()	323
Math.pointDistance()	323
28장: Matrix 객체	
matrix.a	324
matrix.b	325
matrix.c	325
matrix.d	326
matrix.tx	326
matrix.ty	326
29장: outputPanel 객체	
outputPanel.clear()	328
outputPanel.save()	329
outputPanel.trace()	329
30장: Oval 객체	
OvalObject.closePath	331
OvalObject.endAngle	332
OvalObject.innerRadius	332
OvalObject.startAngle	332
31장: Parameter 객체	
parameter.category	334
parameter.insertItem()	335
parameter.listIndex	335
parameter.name	336
parameter.removeItem()	336
parameter.value	337
parameter.valueType	337
parameter.verbose	338
32장: Path 객체	
path.addCubicCurve()	339
path.addCurve()	340
path.addPoint()	341
path.clear()	341
path.close()	342
path.makeShape()	342

path.newContour()	343
path.nPts	343
33장: presetItem 객체	
presetItem.isDefault	345
presetItem.isFolder	346
presetItem.level	346
presetItem.name	347
presetItem.open	347
presetItem.path	347
34장: presetPanel 객체	
presetPanel.addNewItem()	349
presetPanel.applyPreset()	350
presetPanel.deleteFolder()	351
presetPanel.deleteItem()	351
presetPanel.expandFolder()	352
presetPanel.exportItem()	353
presetPanel.findItemIndex()	353
presetPanel.getSelectedItems()	354
presetPanel.importItem()	355
presetPanel.items	355
presetPanel.moveToFolder()	356
presetPanel.newFolder()	357
presetPanel.renameItem()	357
presetPanel.selectItem()	358
35장: Rectangle 객체	
RectangleObject.bottomLeftRadius	359
RectangleObject.bottomRightRadius	360
RectangleObject.lockFlag	360
RectangleObject.topLeftRadius	360
RectangleObject.topRightRadius	361
36장: Shape 객체	
shape.beginEdit()	363
shape.contours	363
shape.deleteEdge()	363
shape.edges	364
shape.endEdit()	364
shape.getCubicSegmentPoints()	365
shape.isDrawingObject	365
shape.isGroup	366
shape.isOvalObject	366

shape.isRectangleObject	367
shape.members	367
shape.numCubicSegments	368
shape.vertices	368
37장: SoundItem 객체	
soundItem.bitRate	370
soundItem.bits	370
soundItem.compressionType	371
soundItem.convertStereoToMono	371
soundItem.exportToFile()	372
soundItem.fileLastModifiedDate	372
soundItem.originalCompressionType	373
soundItem.quality	373
soundItem.sampleRate	374
soundItem.sourceFileExists	374
soundItem.sourceFilesCurrent	374
soundItem.sourceFilePath	375
soundItem.useImportedMP3Quality	375
38장: Stroke 객체	
stroke.breakAtCorners	378
stroke.capType	378
stroke.color	379
stroke.curve	379
stroke.dash1	380
stroke.dash2	380
stroke.density	380
stroke.dotSize	381
stroke.dotSpace	381
stroke.hatchThickness	382
stroke.jiggle	382
stroke.joinType	383
stroke.length	383
stroke.miterLimit	384
stroke.pattern	384
stroke.rotate	384
stroke.scaleType	385
stroke.shapeFill	385
stroke.space	386
stroke.strokeHinting	386
stroke.style	387
stroke.thickness	387

stroke.variation	388
stroke.waveHeight	388
stroke.waveLength	388
39장: swfPanel 객체	
swfPanel.call()	390
swfPanel.name	392
swfPanel.path	393
swfPanel.setFocus()	393
40장: SymbolInstance 객체	
symbolInstance.accName	395
symbolInstance.actionScript	395
symbolInstance.backgroundColor	396
symbolInstance.bitmapRenderMode	396
symbolInstance.blendMode	397
symbolInstance.buttonTracking	397
symbolInstance.cacheAsBitmap	398
symbolInstance.colorAlphaAmount	398
symbolInstance.colorAlphaPercent	398
symbolInstance.colorBlueAmount	399
symbolInstance.colorBluePercent	399
symbolInstance.colorGreenAmount	399
symbolInstance.colorGreenPercent	400
symbolInstance.colorMode	400
symbolInstance.colorRedAmount	401
symbolInstance.colorRedPercent	401
symbolInstance.description	401
symbolInstance.filters	402
symbolInstance.firstFrame	402
symbolInstance.forceSimple	403
symbolInstance.loop	403
symbolInstance.shortcut	403
symbolInstance.silent	404
symbolInstance.symbolType	404
symbolInstance.tabIndex	405
symbolInstance.usesBackgroundColor	405
symbolInstance.visible	406
41장: SymbolItem 객체	
symbolItem.convertToCompiledClip()	407
symbolItem.exportSWC()	408
symbolItem.exportSWF()	408

symbolItem.scalingGrid	409
symbolItem.scalingGridRect	409
symbolItem.sourceAutoUpdate	410
symbolItem.sourceFilePath	410
symbolItem.sourceLibraryName	410
symbolItem.symbolType	411
symbolItem.timeline	411
42장: Text 객체	
text.accName	413
text.antiAliasSharpness	414
text.antiAliasThickness	414
text.autoExpand	414
text.border	415
text.description	415
text.embeddedCharacters	416
text.embedRanges	416
text.embedVariantGlyphs	417
text.fontRenderingMode	417
text.getTextAttr()	418
text.getTextString()	419
text.length	419
text.lineType	420
text.maxCharacters	420
text.orientation	420
text.renderAsHTML	421
text.scrollable	421
text.selectable	422
text.selectionEnd	422
text.selectionStart	422
text.setTextAttr()	423
text.setTextString()	424
text.shortcut	424
text.silent	425
text.tabIndex	425
text.textRuns	426
text.textType	426
text.useDeviceFonts	426
text.variableName	427
43장: TextAttrs 객체	
textAttrs.aliasText	429
textAttrs.alignment	429

textAttrs.autoKern	429
textAttrs.bold	430
textAttrs.characterPosition	430
textAttrs.characterSpacing	430
textAttrs.face	431
textAttrs.fillColor	431
textAttrs.indent	432
textAttrs.italic	432
textAttrs.leftMargin	432
textAttrs.letterSpacing	433
textAttrs.lineSpacing	433
textAttrs.rightMargin	433
textAttrs.rotation	434
textAttrs.size	434
textAttrs.target	435
textAttrs.url	435
44장: TextRun 객체	
textRun.textAttrs	436
textRun.characters	436
45장: Timeline 객체	
timeline.addMotionGuide()	440
timeline.addNewLayer()	440
timeline.clearFrames()	441
timeline.clearKeyframes()	442
timeline.convertToBlankKeyframes()	442
timeline.convertToKeyframes()	443
timeline.copyFrames()	444
timeline.copyLayers()	444
timeline.copyMotion()	445
timeline.copyMotionAsAS3()	446
timeline.createMotionObject()	446
timeline.createMotionTween()	447
timeline.currentFrame	447
timeline.currentLayer	448
timeline.cutFrames()	448
timeline.cutLayers()	449
timeline.deleteLayer()	450
timeline.duplicateLayers()	450
timeline.expandFolder()	451
timeline.findLayerIndex()	452
timeline.frameCount	452

timeline.getFrameProperty()	453
timeline.getGuidelines()	453
timeline.getLayerProperty()	454
timeline.getSelectedFrames()	454
timeline.getSelectedLayers()	455
timeline.insertBlankKeyframe()	455
timeline.insertFrames()	456
timeline.insertKeyframe()	457
timeline.layerCount	458
timeline.layers	458
timeline.libraryItem	459
timeline.name	459
timeline.pasteFrames()	460
timeline.pasteLayers()	460
timeline.pasteMotion()	461
timeline.removeFrames()	461
timeline.removeMotionObject()	462
timeline.reorderLayer()	463
timeline.reverseFrames()	463
timeline.selectAllFrames()	464
timeline setFrameProperty()	465
timeline.setGuidelines()	465
timeline.setLayerProperty()	466
timeline.setSelectedFrames()	467
timeline.setSelectedLayers()	467
timeline.showLayerMasking()	468
timeline.startPlayback()	469
timeline.stopPlayback()	469
46장: ToolObj 객체	
toolObj.depth	470
toolObj.enablePIControl()	471
toolObj.iconID	472
toolObj.position	472
toolObj.setIcon()	473
toolObj.setMenuString()	473
toolObj.setOptionsFile()	474
toolObj.setPI()	475
toolObj.setToolName()	475
toolObj.setToolTip()	476
toolObj.showPIControl()	476
toolObj.showTransformHandles()	477

47장: Tools 객체

tools.activeTool	479
tools.altIsDown	480
tools.constrainPoint()	480
tools.ctrlIsDown	481
tools.getKeyDown()	481
tools.mouselsDown	482
tools.penDownLoc	482
tools.penLoc	482
tools.setCreatingBbox()	483
tools.setCursor()	483
tools.shiftIsDown	484
tools.snapPoint()	484
tools.toolObjs	485

48장: Vertex 객체

vertex.getHalfEdge()	486
vertex.setLocation()	487
vertex.x	487
vertex.y	488

49장: VideoItem 객체

videoitem.exportToFLV()	489
videoitem.fileLastModifiedDate	490
videoitem.sourceFileExists	490
videoitem.sourceFilesCurrent	491
videoitem.sourceFilePath	491
videoitem.videoType	492

50장: XMLUI 객체

xmlui.accept()	493
xmlui.cancel()	494
xmlui.get()	494
xmlui.getControlItemElement()	495
xmlui.setEnabled()	495
xmlui.getVisible()	496
xmlui.set()	497
xmlui.setControlItemElement()	497
xmlui.setControlItemElements()	498
xmlui.setEnabled()	499
xmlui.setVisible()	499

51장: C 레벨 확장성

확장성	501
C 함수 통합	501
데이터 유형	506
C 레벨 API	507

1장: 소개

Adobe® ActionScript®에 익숙한 Adobe® Flash® Professional CS5 또는 CS5.5 사용자는 Adobe® Flash® Player에서 런타임에 실행되는 스크립트를 만들 수 있습니다. 이 문서에서 설명하는 Flash JavaScript API(Application Programming Interface)는 제작 환경에서 실행되는 스크립트를 만들 수 있는 보완적인 프로그래밍 도구입니다.

이 문서에서는 JavaScript API에서 제공하는 객체, 메서드 및 속성에 대해 설명합니다. 여기에서는 독자가 제작 환경에서 작업할 때 문서에 정의된 명령을 사용할 줄 아는 것으로 가정합니다. 특정 명령의 기능을 확인하려면 **Flash** 사용 등과 같은 Flash 도움말의 다른 문서를 참조하십시오.

또한 이 문서에서는 사용자가 함수, 매개 변수, 데이터 유형 등의 기본 프로그래밍 개념과 JavaScript 또는 ActionScript 구문에 익숙한 것으로 가정합니다.

JavaScript API 작업

Flash JavaScript API를 사용하면 Flash 제작 환경에서(즉, Flash 프로그램이 열려 있는 동안) 여러 가지 액션을 수행하는 스크립트를 작성할 수 있습니다. 이 기능은 Flash Player 환경에서(즉, SWF 파일이 재생되는 동안) 액션을 수행하는 스크립트를 작성하는 데 사용할 수 있는 ActionScript 언어와 다릅니다. 이 기능은 웹 브라우저에 표시되는 페이지에서 사용할 수 있는 JavaScript 명령과도 다릅니다.

JavaScript API를 사용하면 제작 프로세스의 능력을 높여 주는 Flash 응용 프로그램 스크립트를 작성할 수 있습니다. 예를 들어 스크립트를 작성하여 반복적인 작업을 자동화하거나 [도구] 패널에 사용자 정의 도구를 추가할 수 있습니다.

Flash JavaScript API는 Netscape JavaScript API를 기반으로 설계된 * Dreamweaver® 및 Adobe® Fireworks® JavaScript API와 비슷한 기능을 제공하기 위한 것입니다. Flash JavaScript API는 JavaScript 객체를 통해 Flash 문서에 액세스할 수 있게 하는 DOM(Document Object Model)을 기반으로 합니다. Flash JavaScript API에는 Flash DOM뿐 아니라 Netscape JavaScript API의 모든 요소가 포함되어 있습니다. 이 문서에서는 이러한 추가 객체 및 이러한 객체의 메서드와 속성에 대해 설명합니다. Flash 스크립트에서는 기본 JavaScript 언어의 모든 요소를 사용할 수 있지만 Flash 문서의 컨텍스트에서 의미가 있는 요소만 효과가 있습니다.

또한 JavaScript API에는 JavaScript와 사용자 정의 C 코드의 조합을 통해 확장을 구현할 수 있는 메서드가 포함되어 있습니다. 자세한 내용은 501페이지의 “**C 레벨 확장성**”을 참조하십시오.

Flash의 JavaScript 인터프리터는 www.mozilla.org/js/spidermonkey/ 웹 사이트에서 다운로드할 수 있는 Mozilla SpiderMonkey 엔진 버전 1.6입니다. SpiderMonkey는 Mozilla.org에서 개발한 두 개의 JavaScript 언어 참조 구현 중 하나로, Mozilla 브라우저에 포함된 것과 동일한 엔진입니다.

SpiderMonkey는 ECMAScript(ECMA-262) Edition 3 언어 사양에 정의된 대로 기본 JavaScript 언어를 구현하며 이 사양과 완벽하게 호환됩니다. ECMA-262 사양에 포함되지 않은 브라우저 전용 호스트 객체만 유일하게 지원되지 않습니다. 마찬가지로 많은 JavaScript 참조 안내서에서는 기본 JavaScript와 클라이언트측(브라우저 관련) JavaScript가 구분되어 있습니다. 기본 JavaScript만 Flash JavaScript 인터프리터에 적용됩니다.

JSFL 파일 생성

Adobe Flash Professional 또는 주로 이용하는 다른 텍스트 편집기를 사용하여 Flash JavaScript(JSFL) 파일을 작성하고 편집할 수 있습니다. Flash를 사용할 경우 기본적으로 이러한 파일의 확장명은 .jsfl입니다. 스크립트를 작성하려면 [파일] > [새로 만들기] > [Flash JavaScript 파일]을 선택합니다.

[작업 내역] 패널에서 명령을 선택하여 JSFL 파일을 만들 수도 있습니다. 이때 [기록 내역] 패널에서 [저장] 버튼을 클릭하거나 패널 메뉴에서 [명령으로 저장]을 선택합니다. 명령(JSFL) 파일은 Commands 폴더에 저장됩니다(2페이지의 “**JSFL 파일 저장**” 참조). 저장된 파일을 다른 스크립트 파일처럼 열고 편집할 수 있습니다.

[작업 내역] 패널에서는 다른 유용한 옵션도 제공하고 있습니다. 선택된 명령을 클립보드에 복사할 수 있으며, Flash에서 작업하는 동안 생성되는 JavaScript 명령을 볼 수 있습니다.

클립보드에 [작업 내역] 패널의 명령을 복사하려면

- 1 [작업 내역] 패널에서 명령을 하나 이상 선택합니다.
- 2 다음 중 하나를 수행합니다.
 - [복사] 버튼을 클릭합니다.
 - 패널 메뉴에서 [단계 복사]를 선택합니다.

[작업 내역] 패널에서 JavaScript 명령을 보려면

- 패널 메뉴에서 [보기] > [패널의 JavaScript]를 선택합니다.

JSFL 파일 저장

Configuration 폴더 안에 있는 여러 폴더 중 하나에 JSFL 스크립트를 저장하여 Flash 제작 환경 안에서 JSFL 스크립트를 사용할 수 있습니다. Configuration 폴더의 기본 위치는 다음과 같습니다.

- Windows® 7™:
부트 드라이브\Users\사용자 이름\AppData\Local\Adobe\Flex CS5 또는 CS5.5\언어\Configuration\
- Windows® Vista™:
부트 드라이브\Users\사용자 이름\Local Settings\Application Data\Adobe\Flex CS5 또는 CS5.5\언어\Configuration\
- Windows XP:
부트 드라이브\Documents and Settings\사용자 이름\Local Settings\Application Data\Adobe\Flex CS5 또는 CS5.5\언어\Configuration\
- Mac OS® X:
Macintosh HD/Users/사용자 이름/Library/Application Support/Adobe/Flex CS5 또는 CS5.5/언어/Configuration/

Configuration 폴더의 위치를 따로 지정하려면 다음 예제와 같이 `fl.configDirectory` 또는 `fl.configURI`를 사용합니다.

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

Configuration 폴더 내의 Behaviors(비헤이비어에 대한 사용자 인터페이스를 지원하는 경우), Commands([명령] 메뉴에 표시되는 스크립트의 경우), JavaScript(사용자 인터페이스 컨트롤을 채우기 위해 스크립트 도우미에 의해 사용되는 스크립트의 경우), Tools([도구] 패널의 확장 가능한 도구의 경우) 및 WindowSWF(Windows 메뉴에 표시되는 패널의 경우) 폴더에는 제작 환경에 액세스할 수 있는 스크립트를 넣을 수 있습니다. 이 문서에서는 명령 및 도구에 사용되는 스크립트를 중점적으로 다룹니다.

Commands 폴더에서 스크립트를 편집하는 경우에는 Flash에서 그 새로운 스크립트를 즉시 사용할 수 있습니다. 확장 가능한 도구에 대한 스크립트를 편집하는 경우 Flash를 닫고 다시 시작하거나 `fl.reloadTools()` 명령을 사용합니다. 그러나 스크립트를 사용하여 [도구] 패널에 확장 가능한 도구를 추가한 후 그 스크립트를 편집했다면 [도구] 패널에서 그 도구를 일단 제거했다가 다시 추가하거나, Flash를 닫은 후 다시 시작해야 수정된 도구를 사용할 수 있습니다.

제작 환경에서 액세스할 수 있도록 명령 및 도구 파일을 저장할 수 있는 위치는 두 곳이 있습니다.

- [명령] 메뉴에 항목으로 표시되는 스크립트의 경우 다음 위치에 있는 Commands 폴더에 JSFL 파일을 저장합니다.

운영 체제	위치
Windows 7	부트 드라이브\Users\사용자 이름\AppData\Local\Adobe\Flash CS5 또는 CS5.5\언어 \Configuration\Commands
Windows Vista	부트 드라이브\Users\사용자 이름\Local Settings\Application Data\Adobe\Flash CS5\언어 \Configuration\Commands
Windows XP	부트 드라이브\Documents and Settings\사용자\Local Settings\Application Data\Adobe\Flash CS5\언어 \Configuration\Commands
Mac OS X	Macintosh HD/Users/사용자 이름/Library/Application Support/Adobe/Flash CS5/언어 /Configuration/Commands

- [도구] 패널에 확장 가능한 도구로 표시되는 스크립트의 경우 다음 위치에 있는 Tools 폴더에 JSFL 파일을 저장합니다.

운영 체제	위치
Windows 7	부트 드라이브\Users\사용자 이름\AppData\Local\Adobe\Flash CS5 또는 CS5.5\언어\Configuration\Tools
Windows Vista	부트 드라이브\Users\사용자 이름\Local Settings\Application Data\Adobe\Flash CS5\언어 \Configuration\Tools
Windows XP	부트 드라이브\Documents and Settings\사용자\Local Settings\Application Data\Adobe\Flash CS5\언어 \Configuration\Tools
Mac OS X	Macintosh HD/Users/사용자 이름/Library/Application Support/Adobe/Flash CS5/언어 /Configuration/Tools

JSFL 파일에 XML 파일과 같은 다른 파일이 수반될 경우, 이러한 파일을 JSFL 파일과 동일한 디렉토리에 저장합니다.

스크립트 실행

스크립트 실행 방법에는 여러 가지가 있습니다. 이 단원에서는 가장 일반적인 방법에 대해 설명합니다.

현재 보고 있거나 편집하고 있는 스크립트를 실행하려면

- 마우스 오른쪽 버튼을 클릭(Macintosh에서는 Command 키를 누른 상태에서 클릭)하고 [스크립트 실행]을 선택합니다.
- [스크립트] 윈도우 톨바에서 [스크립트 실행] 아이콘을 클릭합니다.

이 옵션을 사용하면 스크립트를 저장하기 전에 실행할 수 있습니다. 또한 FLA 파일이 열려 있지 않은 경우에도 스크립트를 실행할 수 있습니다.

Commands 폴더에 있는 스크립트를 실행하려면 다음 중 하나를 수행합니다.

- 제작 환경에서 [명령] > 스크립트 이름을 선택합니다.
- 스크립트에 할당해 놓은 키보드 단축키를 사용합니다. 키보드 단축키를 할당하려면 [편집] > [키보드 단축키]를 클릭하고 [명령] 팝업 메뉴에서 [드로잉 메뉴 명령]을 선택합니다. 메뉴 트리의 명령 노드를 확장하면 사용할 수 있는 스크립트의 목록이 표시됩니다.

Commands 폴더에 없는 명령 스크립트를 실행하려면 다음 중 하나를 수행합니다.

- 제작 환경에서 [명령] > [명령 실행]을 선택한 후 실행할 스크립트를 선택합니다.
- 스크립트 내에서 fl.runScript() 명령을 사용합니다.
- 파일 시스템에서 스크립트 파일을 두 번 클릭합니다.

JSFL 파일에서 구현된 도구를 [도구] 패널에 추가하려면

- 1 Tools 폴더에 도구에 필요한 JSFL 파일과 기타 연관된 파일을 복사합니다(2페이지의 “[JSFL 파일 저장](#)” 참조).
- 2 [편집] > [도구 패널 사용자 정의](Windows) 또는 [Flash] > [도구 패널 사용자 정의](Macintosh)를 선택합니다.
- 3 사용 가능한 도구 목록에 도구를 추가합니다.
- 4 [확인]을 클릭합니다.

ActionScript 3.0 언어 및 구성 요소 참조 설명서에 설명되어 있는 `MMExecute()` 함수를 사용하여 ActionScript 파일에 개별 JavaScript API 명령을 추가할 수 있습니다. 그러나 `MMExecute()` 함수는 구성 요소 속성 관리자나 제작 환경 내의 SWF 패널과 같은 사용자 정의 사용자 인터페이스 요소의 컨텍스트에서 사용될 때만 효과가 있습니다. JavaScript API 명령은 ActionScript에서 호출되더라도 제작 환경 밖이나 Flash Player에서는 효과가 없습니다.

ActionScript 스크립트에서 명령을 실행하려면

- 다음 구문을 사용합니다(한 문자열에 여러 가지 명령 연결 가능).

```
MMExecute(Javascript command string);
```

명령줄에서 스크립트를 실행할 수도 있습니다.

Windows의 명령줄에서 스크립트를 실행하려면

- 다음 구문을 사용합니다(필요에 따라 경로 정보 추가).

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

Macintosh의 "Terminal" 응용 프로그램에서 스크립트를 실행하려면

- 다음 구문을 사용합니다(필요에 따라 경로 정보 추가).

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

osascript 명령은 파일에서 AppleScript를 실행할 수도 있습니다. 예를 들어 myScript라는 파일에 다음 텍스트를 포함할 수 있습니다.

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

그런 다음 스크립트를 실행하려면 이 명령을 사용합니다.

```
osascript myScript
```

JavaScript API의 새로운 기능

Flash CS5 및 CS5.5에는 추가된 객체, 메서드 및 속성도 있고 제거된 객체, 메서드 및 속성도 있습니다. 이러한 변경 사항이 아래에 요약되어 있습니다.

이전에 JavaScript API를 사용한 적이 없으면 이 단원을 건너뛰고 6페이지의 “[JavaScript API 객체](#)”로 바로 이동할 수도 있습니다.

새로운 메서드 및 속성

기존 객체에 대해 다음 메서드 및 속성이 Flash Pro CS5에서 새로 추가되었습니다.

- [Document](#) 객체
 - `document.debugMovie()`
 - `document.loadCuepointXML()`

- **Flash 객체 (fl)**
 - fl.languageCode
 - fl.toggleBreakpoint
- **Frame 객체**
 - frame.convertMotionObjectTo2D()
 - frame.convertMotionObjectTo3D()
 - frame.getMotionObjectXML()
 - frame.hasMotionPath()
 - frame.isMotionObject()
 - frame.is3DMotionObject()
 - frame.selectMotionPath()
 - frame.setMotionObjectDuration()
 - frame.setMotionObjectXML()
 - frame.tweenInstanceName
- **Timeline 객체**
 - timeline.createMotionObject()
 - timeline.libraryItem
 - timeline.removeMotionObject()
 - timeline.startPlayback
 - timeline.stopPlayback

기존 객체에 대해 다음 메서드 및 속성이 Flash Pro CS5.5에서 새로 추가되었습니다.

- **SymbolInstance 객체**
 - symbolInstance.bitmapRenderMode
 - symbolInstance.backgroundColor
 - symbolInstance.usesBackgroundColor
 - symbolInstance.visible
- **Timeline 객체**
 - timeline.copyLayers()
 - timeline.cutLayers()
 - timeline.duplicateLayers()
 - timeline.pasteLayers()
- **Flash 객체 (fl)**
 - fl.getSwfPanel()
 - fl.installedPlayers()
 - fl.publishCacheEnabled
 - fl.publishCacheDiskSizeMax
 - fl.publishCacheMemorySizeMax

- fl.publishCacheMemoryEntrySizeLimit
- fl.clearPublishCache()
- **swfPanel 객체**
 - swfPanel.setFocus()

기타 변경 내용

다음 메서드 및 속성이 Flash CS5에서 업데이트되었습니다.

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

다음 객체 및 메서드는 Flash CS5에서 사용할 수 없습니다.

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

JavaScript API 객체

이 단원에서는 Flash JavaScript API에서 사용할 수 있는 객체와 이러한 객체를 사용하여 작업하는 방법을 요약하여 보여줍니다. JavaScript API를 사용하여 작업할 때는 모든 표준 JavaScript 명령도 사용할 수 있습니다.

다음 표에서는 JavaScript API의 각 객체에 대해 간략히 설명합니다. 객체는 알파벳 순서로 나열되어 있습니다.

객체	설명
actionsPanel 객체	actionsPanel 객체는 현재 표시된 액션 패널을 나타냅니다.
BitmapInstance 객체	BitmapInstance 객체는 Instance 객체의 하위 클래스이며 프레임 내의 비트맵을 나타냅니다.
BitmapItem 객체	BitmapItem 객체는 문서 라이브러리 내의 비트맵을 나타내며, BitmapItem 객체는 Item 객체의 하위 클래스입니다.
CompiledClipInstance 객체	CompiledClipInstance 객체는 Instance 객체의 하위 클래스입니다.
compilerErrors 객체	[컴파일러 오류] 패널을 나타내는 compilerErrors 객체는 Flash 객체(fl.compilerErrors)의 속성입니다.

객체	설명
ComponentInstance 객체	ComponentInstance 객체는 SymbolInstance 객체의 하위 클래스이며 프레임 내의 구성 요소를 나타냅니다.
componentsPanel 객체	[구성 요소] 패널을 나타내는 componentsPanel 객체는 Flash 객체(fl.componentsPanel)의 속성입니다.
Contour 객체	Contour 객체는 모양의 경계에 있는 양 반쪽 가장자리의 닫힌 경로를 나타냅니다.
Document 객체	Document 객체는 스테이지를 나타냅니다.
drawingLayer 객체	drawingLayer 객체는 flash 객체의 자식으로 JavaScript에서 액세스할 수 있습니다.
Edge 객체	Edge 객체는 스테이지에 있는 모양의 가장자리를 나타냅니다.
Element 객체	스테이지에 표시되는 모든 항목의 유형은 Element입니다.
Fill 객체	Fill 객체에는 [도구] 패널 또는 선택한 모양의 채움 색상 설정에 대한 모든 속성이 포함됩니다.
Filter 객체	Filter 객체에는 모든 필터에 대한 속성이 모두 들어 있습니다.
Flash 객체(fl)	flash 객체는 Flash 응용 프로그램을 나타냅니다.
FLfile 객체	FLfile 객체를 사용하면 로컬 파일 시스템에 있는 파일과 폴더를 액세스, 수정 및 제거할 수 있는 Flash 확장을 작성할 수 있습니다.
folderItem 객체	folderItem 객체는 Item 객체의 하위 클래스입니다.
fontItem 객체	fontItem 객체는 Item 객체의 하위 클래스입니다.
Frame 객체	Frame 객체는 레이어 내의 프레임을 나타냅니다.
HalfEdge 객체	Shape 객체 가장자리의 지정된 반쪽입니다.
Instance 객체	Instance 객체는 Element 객체의 하위 클래스입니다.
Item 객체	Item 객체는 기본 추상 클래스입니다.
Layer 객체	Layer 객체는 타임라인 내의 레이어를 나타냅니다.
library 객체	library 객체는 [라이브러리] 패널을 나타냅니다.
Math 객체	Math 객체는 flash 객체(fl.Math)의 읽기 전용 속성으로 사용할 수 있습니다.
Matrix 객체	Matrix 객체는 변형 행렬을 나타냅니다.
outputPanel 객체	구문 오류와 같은 문제 해결 정보를 표시하는 [출력] 패널을 나타내는 outputPanel 객체는 Flash 객체(fl.outputPanel)의 속성입니다.
Oval 객체	Oval 객체는 [타원형 도구]를 사용하여 그려진 모양입니다. 항목이 Oval 객체인지 확인하려면 shape.isOvalObject를 사용합니다.
Parameter 객체	Parameter 객체 유형은 screen.parameters 배열(Flash 제작 도구의 스크린 속성 관리자에 해당) 또는 componentInstance.parameters 배열(제작 도구의 구성 요소 속성 관리자에 해당)에서 액세스합니다.
Path 객체	Path 객체는 주로 확장 가능한 도구를 만들 때 사용하는 일련의 선분(직선, 곡선 또는 둘 다)을 정의합니다.
presetItem 객체	presetItem 객체는 [모션 프리셋] 패널 항목(프리셋 또는 폴더)을 나타냅니다.
presetPanel 객체	[모션 프리셋] 패널([원도우] > [모션 프리셋])을 나타내는 presetPanel 객체는 flash 객체(fl.presetPanel)의 속성입니다.
Rectangle 객체	Rectangle 객체는 [사각형 도구]를 사용하여 그려진 모양입니다. 항목이 Rectangle 객체인지 확인하려면 shape.isRectangleObject를 사용합니다.

객체	설명
Screen 객체	Screen 객체는 슬라이드 또는 양식 문서 내의 단일 스크린을 나타냅니다.
ScreenOutline 객체	ScreenOutline 객체는 슬라이드 또는 양식 문서 내의 스크린 그룹을 나타냅니다.
Shape 객체	Shape 객체는 Element 객체의 하위 클래스입니다. Shape 객체는 스테이지에서 기하 도형을 만들거나 조작할 때 드로잉 API보다 더 정확한 컨트롤을 제공합니다.
SoundItem 객체	SoundItem 객체는 Item 객체의 하위 클래스입니다. 사운드를 만드는 데 사용되는 라이브러리 항목을 나타냅니다.
Stroke 객체	Stroke 객체에는 사용자 정의 설정을 포함하여 획에 대한 모든 설정이 포함됩니다.
swfPanel 객체	swfPanel 객체는 [원도우 SWF] 패널을 나타냅니다. [원도우 SWF] 패널은 Flash 제작 환경에서 실행할 수 있는 응용 프로그램을 구현하는 SWF 파일입니다. swfPanel 객체 배열은 flash 객체(fl.swfPanels)의 속성입니다.
SymbolInstance 객체	SymbolInstance 객체는 Instance 객체의 하위 클래스이며 프레임 내의 심볼을 나타냅니다.
SymbolItem 객체	SymbolItem 객체는 Item 객체의 하위 클래스입니다.
Text 객체	Text 객체는 문서 내의 단일 텍스트 항목을 나타냅니다.
TextAttrs 객체	TextAttrs 객체에는 세부 선택에 적용할 수 있는 텍스트의 모든 속성이 포함됩니다. 이 객체는 Text 객체의 하위 클래스입니다.
TextRun 객체	TextRun 객체는 TextAttrs 객체의 모든 속성과 일치하는 특성을 가진 문자의 흐름을 나타냅니다.
Timeline 객체	Timeline 객체는 fl.getDocumentDOM().getTimeline()을 통해 액세스할 수 있는 현재 문서의 Flash 타임라인을 나타냅니다.
ToolObj 객체	ToolObj 객체는 [도구] 패널 내의 개별 도구를 나타냅니다.
Tools 객체	Tools 객체는 Flash 객체(fl.tools)에서 액세스할 수 있습니다.
Vertex 객체	Vertex 객체는 좌표 데이터가 포함된 모양 데이터 구조의 일부입니다.
VideoItem 객체	VideoItem 객체는 Item 객체의 하위 클래스입니다.
XMLUI 객체	XMLUI 객체에서는 XMLUI 대화 상자의 속성을 가져와서 설정하고, 수락하거나 취소함으로써 해당 대화 상자에서 벗어나도록 할 수 있습니다.

Flash DOM

Flash JavaScript API용 Flash DOM(Document Object Model)은 최상위 함수 집합(14페이지의 “[최상위 함수 및 메서드](#)” 참조)과 두 가지 최상위 객체인 FLfile 객체 및 Flash 객체(fl)로 구성됩니다. 모든 객체는 Flash 제작 환경이 열려 있을 때 항상 존재하므로 스크립트에서 반드시 사용할 수 있습니다. 자세한 내용은 [FLfile 객체](#) 및 [Flash 객체\(fl\)](#)를 참조하십시오.

Flash 객체를 참조할 경우 flash 또는 fl을 사용할 수 있습니다. 예를 들어 열려 있는 모든 FLA 파일을 닫으려면 다음 명령문 중 하나를 사용할 수 있습니다.

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

Flash 객체에는 다음과 같은 자식 객체가 포함됩니다.

객체	액세스 방법
actionsPanel 객체	fl.actionsPanel을 사용하여 actionsPanel 객체에 액세스합니다. 이 객체는 Flash 제작 환경의 [액션] 패널에 해당합니다.
compilerErrors 객체	fl.compilerErrors를 사용하여 compilerErrors 객체에 액세스합니다. 이 객체는 Flash 제작 환경의 [컴파일러 오류] 패널에 해당합니다.
componentsPanel 객체	fl.componentsPanel을 사용하여 componentsPanel 객체에 액세스합니다. 이 객체는 Flash 제작 환경의 [구성 요소] 패널에 해당합니다.
Document 객체	fl.documents를 사용하여 열려 있는 모든 문서의 배열을 검색합니다. 특정 문서에 액세스하려면 fl.documents[index]를 사용하고 현재 문서(포커스가 있는 문서)에 액세스하려면 fl.getDocumentDOM()을 사용합니다.
drawingLayer 객체	fl.drawingLayer를 사용하여 drawingLayer 객체에 액세스합니다.
Math 객체	fl.Math를 사용하여 Math 객체에 액세스합니다.
outputPanel 객체	fl.outputPanel을 사용하여 outputPanel 객체에 액세스합니다. 이 객체는 Flash 제작 환경의 [출력] 패널에 해당합니다.
presetPanel 객체	fl.presetPanel을 사용하여 presetPanel 객체에 액세스합니다. 이 객체는 [모션 프리셋] 패널([윈도우] > [모션 프리셋])에 해당합니다.
swfPanel 객체	fl.swfPanels를 사용하여 swfPanel 객체의 배열에 액세스합니다. 이러한 객체는 윈도우 SWF 패널에 해당합니다.
Tools 객체	fl.tools를 사용하여 Tools 객체의 배열에 액세스합니다.
XMLUI 객체	fl.xmlui를 사용하여 XMLUI(XML 사용자 인터페이스) 객체에 액세스합니다. XMLUI 객체에서는 XMLUI 대화 상자의 속성을 가져와서 설정할 수 있습니다.

Document 객체

최상위 Flash 객체의 중요한 속성 중 하나는 [fl.documents](#) 속성입니다. 이 속성에는 제작 환경에 현재 열려 있는 FLA 파일 중 하나를 각각 나타내는 Document 객체의 배열이 포함됩니다. 각 Document 객체의 속성은 FLA 파일이 포함할 수 있는 대부분의 요소를 나타냅니다. 따라서 DOM의 대부분은 Document 객체의 자식 객체와 속성으로 구성됩니다. 자세한 내용은 [Document 객체](#)를 참조하십시오.

예를 들어 열려 있는 첫 번째 문서를 참조하려면 `flash.documents[0]` 또는 `fl.documents[0]` 문을 사용합니다. 첫 번째 문서는 제작 환경의 현재 세션에서 제일 먼저 연 Flash 문서입니다. 열려 있는 첫 번째 문서를 닫으면 열려 있는 다른 문서의 인덱스가 감소합니다.

특정 문서의 인덱스를 찾으려면 `flash.findDocumentIndex(nameOfDocument)` 또는 `fl.findDocumentIndex(nameOfDocument)`를 사용합니다. [fl.findDocumentIndex\(\)](#)를 참조하십시오.

현재 포커스가 있는 문서에 액세스하려면 `flash.getDocumentDOM()` 또는 `fl.getDocumentDOM()` 문을 사용합니다. [fl.getDocumentDOM\(\)](#)을 참조하십시오. 두 번째 문은 이 문서에 나오는 대부분의 예제에 사용되는 구문입니다.

`fl.documents` 배열에서 특정 문서를 찾으려면 배열을 반복하고 각 문서의 `document.name` 속성을 테스트합니다. [fl.documents](#) 및 `document.name`을 참조하십시오.

앞의 표에 나와 있지 않은 DOM에 있는 모든 객체(8페이지의 “Flash DOM” 참조)는 Document 객체에서 액세스합니다. 예를 들어 문서의 라이브러리에 액세스하려면 `Library` 객체를 가져오는 `document.library` 속성을 사용합니다.

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

라이브러리 항목의 배열에 액세스하려면 `library.items` 속성을 사용합니다. 여기서 배열의 각 요소는 `Item` 객체입니다.

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

특정 라이브러리 항목에 액세스하려면 `library.items` 배열의 멤버를 지정합니다.

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

즉, **Library** 객체는 **Document** 객체의 자식이고 **Item** 객체는 **Library** 객체의 자식입니다. 자세한 내용은 [document.library](#), [library](#) 객체, [library.items](#) 및 [Item](#) 객체를 참조하십시오.

액션 대상 지정

별도로 지정하지 않으면 메시드는 현재 포커스나 선택한 항목에 영향을 줍니다. 예를 들어 다음 스크립트에서는 특정 객체가 지정되어 있지 않기 때문에 현재 선택한 항목의 크기를 두 배로 늘립니다.

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

액션 대상을 특별히 **Flash** 문서의 현재 선택한 항목으로 지정해야 하는 경우도 있습니다. 이렇게 하려면 `document.selection` 속성에 포함된 배열을 사용합니다([document.selection](#) 참조). 다음 예제에서처럼 배열의 첫 번째 요소는 현재 선택한 항목을 나타냅니다.

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

다음 스크립트에서는 현재 선택한 항목 대신 요소 배열에 저장된 스테이지의 첫 번째 요소 크기를 두 배로 늘립니다.

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

또한 다음 예제에서처럼 스테이지의 모든 요소를 반복하고 지정된 양만큼 폭과 높이를 늘리는 등의 작업을 수행할 수도 있습니다.

```
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;  
}
```

DOM 구조의 요약

다음 목록에는 DOM 구조가 간략하게 표시됩니다. 각 줄의 시작 부분에 있는 숫자는 객체의 레벨을 나타냅니다. 예를 들어, 앞에 "03"이 오는 객체는 차상위 "02" 객체의 자식이고 "02" 객체는 그 다음 차상위 객체인 "01" 객체의 자식입니다.

객체를 사용하기 위해서는 부모 객체의 속성을 지정해야 하는 경우도 있습니다. 예를 들어 `document.timelines` 속성에는 **Timeline** 객체의 배열이 포함됩니다. 이러한 속성은 다음 개요에 설명되어 있습니다.

일부 객체는 다른 객체의 자식이 아닌 다른 객체의 하위 클래스입니다. 다른 객체의 하위 클래스인 객체에는 부모 객체(수퍼 클래스)의 메서드와 속성뿐 아니라 자신의 메서드 및/또는 속성이 있습니다. 하위 클래스는 계층 구조 내에서 해당 수퍼 클래스와 같은 레벨을 공유합니다. 예를 들어 **Item** 객체는 **BitmapItem** 객체의 수퍼 클래스입니다. 이러한 관계는 다음 개요에 나와 있습니다.

- 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

샘플 구현

Adobe Flash Professional CS5 및 CS5.5에는 여러 가지 샘플 JSFL 구현이 포함되어 있습니다. JavaScript API에 익숙해지도록 이러한 파일을 검토하고 설치할 수 있습니다. 샘플은 www.adobe.com/go/learn_fl_samples_kr에 있는 Samples.zip 파일의 Samples/ExtendingFlash라는 폴더에 있습니다.

모양 명령 샘플

Shape.jsfl이라는 샘플 JavaScript API 스크립트는 ExtendingFlash/Shape 폴더에 있습니다(위의 "샘플 구현" 참조). 이 스크립트에서는 [출력] 패널에 있는 모양의 윤곽선에 대한 정보를 표시합니다.

Shape 스크립트를 설치하고 실행하려면

- 1 Shape.jsfl 파일을 Configuration/Commands 폴더에 복사합니다(2페이지의 "JSFL 파일 저장" 참조).
- 2 Flash 문서(FLA 파일)에서 Shape 객체를 선택합니다.
- 3 [명령] > [Shape]를 선택하여 스크립트를 실행합니다.

필터 가져오기 및 설정 명령 샘플

filtersGetSet.jsfl이라는 샘플 JavaScript API 스크립트는 ExtendingFlash/filtersGetSet 폴더에 있습니다(위의 "샘플 구현" 참조). 이 스크립트에서는 선택한 객체에 필터를 추가하고 [출력] 패널에 추가되는 필터에 대한 정보를 표시합니다.

filtersGetSet 스크립트를 설치하고 실행하려면

- 1 filtersGetSet.jsfl 파일을 Configuration/Commands 폴더에 복사합니다(2페이지의 "JSFL 파일 저장" 참조).
- 2 Flash 문서(FLA 파일)에서 텍스트, 무비 클립 또는 버튼 객체를 선택합니다.
- 3 [명령] > [filtersGetSet]을 선택하여 스크립트를 실행합니다.

다각형 별 도구 샘플

PolyStar.jsfl이라는 샘플 JavaScript API 스크립트는 ExtendingFlash/PolyStar 폴더에 있습니다(위의 "샘플 구현" 참조).

PolyStar.jsfl에서는 Flash [도구] 패널에 있는 다각형 별 도구를 복제합니다. 이 스크립트에서는 JavaScript API를 사용하여 다각형 별 도구를 구성하는 방법을 보여 주며 여기에는 코드가 수행하는 작업을 설명하는 자세한 주석이 포함되어 있습니다. 이 파일을 읽으면 JavaScript API의 사용 방법을 보다 잘 이해할 수 있습니다. 사용자 고유의 도구를 구성하는 방법에 대한 자세한 내용을 배우려면 Tools 디렉토리에 있는 PolyStar.xml 파일도 읽어보는 것이 좋습니다.

비트맵 추적 패널 샘플

TraceBitmap.fla 및 TraceBitmap.swf라는 파일의 집합은 ExtendingFlash/TraceBitmapPanel 폴더에 있습니다(위의 "샘플 구현" 참조). 이러한 파일에서는 Flash의 기능을 제어하기 위해 패널을 디자인하고 만드는 방법을 보여 줍니다. 또한 MMExecute() 함수를 사용하여 ActionScript 스크립트에서 JavaScript 명령을 호출하는 방법을 보여 줍니다.

TraceBitmap 샘플을 실행하려면

- 1 Flash가 실행 중이면 Flash를 종료합니다.
- 2 Configuration 폴더의 하위 디렉토리인 WindowSWF 폴더에 TraceBitmap.swf 파일을 복사합니다(2페이지의 "JSFL 파일 저장" 참조). 예를 들어 Windows XP에서는 이 폴더가 부트 드라이브\Documents and Settings\사용자\Local Settings\Application Data\Adobe\Flex CS5\언어\Configuration\WindowSWF입니다.
- 3 Flash를 시작합니다.

- 4 Flash 문서(FLA 파일)를 열거나 새로 만들고 비트맵 또는 JPEG 이미지를 파일로 가져옵니다.
TraceBitmapPanel 폴더에 있는 flower.jpg 파일을 사용하거나, 따로 원하는 이미지를 선택해서 사용할 수 있습니다.
- 5 가져온 이미지를 선택한 상태에서 [윈도우] > [기타 패널] > [TraceBitmap]을 선택합니다.
- 6 [전송]을 클릭합니다.
이미지가 일단의 모양 그룹으로 변환됩니다.

DLL 샘플

DLL 샘플 구현은 ExtendingFlash/dllSampleComputeSum 폴더에 있습니다(위의 "샘플 구현" 참조). DLL 구성에 대한 자세한 내용은 501페이지의 "[C 레벨 확장성](#)"을 참조하십시오.

2장: 최상위 함수 및 메서드

단원 정보

이 단원에서는 Adobe Flash JavaScript API(Application Programming Interface)에서 사용할 수 있는 최상위 함수 및 메서드에 대해 설명합니다. JavaScript API 파일 저장 위치에 대한 자세한 내용은 2페이지의 “[JSFL 파일 저장](#)”을 참조하십시오.

Global 메서드

다음 메서드는 모든 JavaScript API 스크립트에서 호출할 수 있습니다.

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

확장 가능한 도구

다음 함수는 확장 가능한 도구를 만드는 스크립트에서 사용할 수 있습니다.

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

activate()

지원 버전

Flash MX 2004

사용법

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

매개 변수

없음

반환값

없음

설명

함수, 확장 가능한 도구가 활성화될 때(즉, [도구] 패널에서 선택될 때) 호출됩니다. 이 함수를 사용하여 도구에 필요한 모든 초기화 작업을 수행합니다.

예제

다음 예제에서는 [도구] 패널에서 확장 가능한 도구를 선택했을 때 `tools.activeTool` 값을 설정합니다.

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

참고 사항

[tools.activeTool](#)

alert()

지원 버전

Flash MX 2004

사용법

```
alert ( alertText )
```

매개 변수

`alertText` [경고] 대화 상자에 표시할 메시지를 지정하는 문자열

반환값

없음

설명

메서드, 모달 [경고] 대화 상자에 [확인] 버튼과 함께 문자열을 표시합니다.

예제

다음 예제에서는 [경고] 대화 상자에 "Process Complete" 메시지를 표시합니다.

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

참고 사항

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

configureTool()

지원 버전

Flash MX 2004

사용법

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

매개 변수

없음

반환값

없음

설명

함수, Flash를 열고 [도구] 패널에 확장 가능한 도구를 로드할 때 호출됩니다. 이 함수를 사용하여 Flash에 필요한 도구 정보를 설정할 수 있습니다.

예제

다음 예제에서는 이 함수의 두 가지 가능한 구현을 보여 줍니다.

```
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()

지원 버전

Flash 8

사용법

```
confirm ( strAlert )
```

매개 변수

strAlert [경고] 대화 상자에 표시할 메시지를 지정하는 문자열

반환값

부울 값, 사용자가 [확인]을 클릭하면 true가 반환되고, [취소]를 클릭하면 false가 반환됩니다.

설명

메서드, 모달 [경고] 대화 상자에 [확인] 및 [취소] 버튼과 함께 문자열을 표시합니다.

참고: 열려 있는 문서(FLA 파일)가 없으면 메서드가 실패하면서 오류가 발생합니다.

예제

다음 예제에서는 [경고] 대화 상자에 "Sort data?" 메시지를 표시합니다.

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


참고 사항

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

deactivate()

지원 버전

Flash MX 2004

사용법

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

매개 변수

없음

반환값

없음

설명

함수, 확장 가능한 도구가 비활성화될 때(즉, 활성 도구가 이 도구에서 다른 도구로 변경될 때) 호출됩니다. 이 함수를 사용하여 도구에 필요한 정리를 수행할 수 있습니다.

예제

다음 예제에서는 도구가 비활성화될 때 [출력] 패널에 메시지를 표시합니다.

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

keyDown()

지원 버전

Flash MX 2004

사용법

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

매개 변수

없음

반환값

없음

설명

함수, 확장 가능한 도구가 활성화되어 있을 때 키를 누르면 호출됩니다. 어떤 키를 눌렀는지 확인하려면 스크립트에서 [tools.getKeyDown\(\)](#)을 호출해야 합니다.

예제

다음 예제에서는 확장 가능한 도구가 활성화되어 있는 상태에서 키를 누른 경우 어떤 키를 눌렀는지에 대한 정보를 표시합니다.

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

참고 사항

[keyUp\(\)](#), [tools.getKeyDown\(\)](#)

keyUp()

지원 버전

Flash MX 2004

사용법

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

매개 변수

없음

반환값

없음

설명

함수, 확장 가능한 도구가 활성화되어 있을 때 키를 놓으면 호출됩니다.

예제

다음 예제에서는 확장 가능한 도구가 활성화되어 있는 상태에서 키를 놓은 경우 [출력] 패널에 메시지를 표시합니다.

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

참고 사항

[keyDown\(\)](#)

mouseDoubleClick()

지원 버전

Flash MX 2004

사용법

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

매개 변수

없음

반환값

없음

설명

함수, 확장 가능한 도구가 활성화되어 있는 상태에서 스테이지에서 마우스 버튼을 두 번 클릭할 때 호출됩니다.

예제

다음 예제에서는 확장 가능한 도구가 활성화되어 있는 상태에서 마우스 버튼을 두 번 클릭한 경우 [출력] 패널에 메시지를 표시합니다.

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

mouseDown()

지원 버전

Flash MX 2004

사용법

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

매개 변수

pt 마우스 버튼을 누를 때 마우스의 위치를 지정하는 점으로서, 마우스 버튼을 누르면 함수에 전달됩니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

함수, 포인터가 스테이지 위에 있는 동안 확장 가능한 도구가 활성화되어 있는 상태에서 마우스 버튼을 누를 때 호출됩니다.

예제

다음 예제에서는 확장 가능한 도구가 활성화되어 있을 때 이 함수의 사용 방법을 보여 줍니다. 첫 번째 예제에서는 [출력] 패널에 마우스 버튼을 눌렀다는 메시지를 표시합니다. 두 번째 예제에서는 마우스 버튼을 눌렀을 때의 마우스 위치에 대한 **x** 및 **y** 좌표를 표시합니다.

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

mouseMove()

지원 버전

Flash MX 2004

사용법

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

매개 변수

pt 마우스의 현재 위치를 지정하는 점으로서, 마우스를 이동할 때마다 마우스 위치를 추적하는 함수에 전달됩니다. 스테이지가 편집 또는 제자리에서 편집 모드에 있을 경우, 점 좌표는 편집 중인 객체를 기준으로 합니다. 그렇지 않을 경우, 점 좌표는 스테이지를 기준으로 합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

함수, 확장 가능한 도구가 활성화되어 있는 상태에서 스테이지에서 지정된 점 위로 마우스를 이동할 때마다 호출됩니다. 마우스 버튼은 누른 상태이거나 누르지 않은 상태일 수 있습니다.

예제

다음 예제에서는 이 함수의 사용 방법을 보여 줍니다. 첫 번째 예제에서는 [출력] 패널에 마우스를 이동 중이라는 메시지를 표시합니다. 두 번째 예제에서는 마우스를 이동할 때 마우스 위치에 대한 **x** 및 **y** 좌표를 표시합니다.

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

mouseUp()

지원 버전

Flash MX 2004

사용법

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

매개 변수

없음

반환값

없음

설명

함수, 확장 가능한 도구가 활성화되어 있는 상태에서 스테이지에서 누른 마우스 버튼을 놓을 때마다 호출됩니다.

예제

다음 예제에서는 확장 가능한 도구가 활성화되어 있는 상태에서 마우스 버튼을 놓은 경우 [출력] 패널에 메시지를 표시합니다.

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

notifySettingsChanged()

지원 버전

Flash MX 2004

사용법

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

매개 변수

없음

반환값

없음

설명

함수, 확장 가능한 도구가 활성화되어 있는 상태에서 속성 관리자에서 해당 도구의 옵션을 변경할 때 호출됩니다. `tools.activeTool` 속성을 사용하여 옵션의 현재 값을 조회할 수 있습니다([tools.activeTool](#) 참조).

예제

다음 예제에서는 확장 가능한 도구가 활성화되어 있는 상태에서 속성 관리자에서 해당 도구의 옵션을 변경하는 경우 [출력] 패널에 메시지를 표시합니다.

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

prompt()

지원 버전

Flash MX 2004

사용법

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

매개 변수

promptMsg [확인] 대화 상자에 표시할 문자열(Mac OSX에서는 256자로 제한됨)

text 텍스트 필드의 기본값으로 표시할 문자열(선택 사항)

반환값

사용자가 [확인]을 클릭하면 사용자가 입력한 문자열이 반환되고, [취소]를 클릭하면 `null`이 반환됩니다.

설명

메서드, 모달 [경고] 대화 상자에 [확인] 및 [취소] 버튼과 함께 확인 및 선택적 텍스트를 표시합니다.

예제

다음 예제에서는 사용자 이름을 입력하라는 메시지를 표시합니다. 사용자가 이름을 입력하고 [확인]을 클릭하면 [출력] 패널에 이 이름이 표시됩니다.

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

참고 사항

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

setCursor()

지원 버전

Flash MX 2004

사용법

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

매개 변수

없음

반환값

없음

설명

함수, 확장 가능한 도구가 활성화되어 있는 상태에서 스크립트에서 사용자 정의 포인터를 설정할 수 있도록 마우스를 이동할 때 호출됩니다. 사용할 포인터를 지정하려면 스크립트에서 `tools.setCursor()`를 호출해야 합니다. 각 포인터에 해당하는 정수 값을 보여 주는 목록을 보려면 [tools.setCursor\(\)](#)를 참조하십시오.

예제

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

3장: actionsPanel 객체

지원 버전

Flash CS3 Professional

설명

현재 표시된 [액션] 패널을 나타내는 actionsPanel 객체는 Flash 객체의 속성입니다([fl.actionsPanel](#) 참조).

메서드 요약

actionsPanel 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
actionsPanel.getClassForObject()	지정된 변수의 클래스를 반환합니다.
actionsPanel.getScriptAssistMode()	스크립트 도우미 모드를 활성화할지 여부를 지정합니다.
actionsPanel.getSelectedText()	[액션] 패널에 현재 선택되어 있는 텍스트를 반환합니다.
actionsPanel.getText()	[액션] 패널의 텍스트를 반환합니다.
actionsPanel.hasSelection()	[액션] 패널에 현재 선택된 텍스트가 있는지 여부를 지정합니다.
actionsPanel.replaceSelectedText()	현재 선택된 텍스트를 지정된 텍스트로 바꿉니다.
actionsPanel.setScriptAssistMode()	스크립트 도우미 모드를 활성화하거나 비활성화합니다.
actionsPanel.setSelection()	[액션] 패널에서 지정된 문자 세트를 선택합니다.
actionsPanel.setText()	[액션] 패널의 텍스트를 지운 다음 지정된 텍스트를 추가합니다.

actionsPanel.getClassForObject()

지원 버전

Flash CS3 Professional

사용법

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

매개 변수

ASvariableName ActionScript 변수의 이름을 나타내는 문자열

반환값

ASvariableName이 멤버로 속해 있는 클래스를 나타내는 문자열

설명

메서드, 지정된 변수의 클래스를 반환하며, 이 변수는 현재 표시된 [액션] 패널에 정의되어 있어야 합니다. 또한 [액션] 패널의 커서나 선택된 텍스트가 변수 정의 뒤에 있어야 합니다.

예제

다음 예제에서는 [액션] 패널에서 `var myVar:ActivityEvent;` 문 뒤에 커서가 있는 경우 `myVar` 변수에 할당된 클래스를 표시합니다.

```
// 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()

지원 버전

Flash CS3 Professional

사용법

```
actionsPanel.getScriptAssistMode()
```

매개 변수

없음

반환값

스크립트 도우미 모드가 활성화되어 있는지(`true`), 아니면 비활성화되어 있는지(`false`)를 지정하는 부울 값

설명

메서드, 스크립트 도우미 모드를 활성화할지 여부를 지정합니다.

예제

다음 예제에서는 스크립트 도우미 모드가 활성화되어 있지 않은 경우 메시지를 표시합니다.

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

참고 사항

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

actionsPanel.getSelectedText()

지원 버전

Flash CS3 Professional

사용법

```
actionsPanel.getSelectedText()
```

매개 변수

없음

반환값

[액션] 패널에 현재 선택되어 있는 텍스트가 들어 있는 문자열

설명

메서드, [액션] 패널에 현재 선택되어 있는 텍스트를 반환합니다.

예제

다음 예제에서는 [액션] 패널에 현재 선택되어 있는 텍스트를 표시합니다.

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

참고 사항

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

actionsPanel.getText()

지원 버전

Flash CS3 Professional

사용법

```
actionsPanel.getText()
```

매개 변수

없음

반환값

[액션] 패널의 모든 텍스트가 들어 있는 문자열

설명

메서드, [액션] 패널의 텍스트를 반환합니다.

예제

다음 예제에서는 [액션] 패널에 있는 텍스트를 표시합니다.

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

참고 사항

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

actionsPanel.hasSelection()

지원 버전

Flash CS3 Professional

사용법

```
actionsPanel.hasSelection()
```

매개 변수

없음

반환값

[액션] 패널에 선택된 텍스트가 있는지(true), 아니면 없는지(false)를 지정하는 부울 값

설명

메서드, [액션] 패널에 현재 선택된 텍스트가 있는지 여부를 지정합니다.

예제

다음 예제에서는 [액션] 패널에 현재 선택되어 있는 텍스트를 표시합니다. 선택된 텍스트가 없으면 [액션] 패널의 모든 텍스트가 표시됩니다.

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

참고 사항

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

actionsPanel.replaceSelectedText()

지원 버전

Flash CS3 Professional

사용법

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

매개 변수

replacementText [액션] 패널의 선택된 텍스트를 바꿀 텍스트를 나타내는 문자열

반환값

[액션] 패널이 있으면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값

설명

메서드, 현재 선택된 텍스트를 **replacementText**에 지정된 텍스트로 바꿉니다. **replacementText**에 들어 있는 문자 수가 선택된 텍스트보다 많으면 선택된 텍스트 이후의 모든 문자는 **replacementText** 뒤에 오기 때문에 덮어쓰이지 않습니다.

예제

다음 예제에서는 [액션] 패널의 현재 선택된 텍스트를 바꿉니다.

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

참고 사항

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

actionsPanel.setScriptAssistMode()

지원 버전

Flash CS3 Professional

사용법

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

매개 변수

bScriptAssist 스크립트 도우미 모드를 활성화할지, 아니면 비활성화할지를 지정하는 부울 값

반환값

스크립트 도우미 모드가 성공적으로 활성화되었는지, 아니면 비활성화되었는지를 지정하는 부울 값

설명

메서드, 스크립트 도우미 모드를 활성화하거나 비활성화합니다.

예제

다음 예제에서는 스크립트 도우미 모드의 상태를 전환합니다.

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

참고 사항

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

actionsPanel.setSelection()

지원 버전

Flash CS3 Professional

사용법

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

매개 변수

startIndex 선택할 첫 번째 문자를 지정하는 0부터 시작하는 정수

numberOfChars 선택할 문자의 수를 지정하는 정수

반환값

요청된 문자를 선택할 수 있는지(**true**), 아니면 선택할 수 없는지(**false**)를 지정하는 부울 값

설명

메서드, [액션] 패널에서 지정된 문자 세트를 선택합니다.

예제

다음 예제에서는 [액션] 패널에 있는 문자 “2006”을 지정된 텍스트로 바꿉니다.

```
// 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");
```

참고 사항

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

actionsPanel.setText()

지원 버전

Flash CS3 Professional

사용법

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

매개 변수

replacementText [액션] 패널에 배치할 텍스트를 나타내는 문자열

반환값

지정된 텍스트가 [액션] 패널에 배치되었으면 **true**를 반환하고 그렇지 않으면 **false**를 반환하는 부울 값

설명

메서드, [액션] 패널의 모든 텍스트를 지운 다음 **replacementText**에 지정된 텍스트를 추가합니다.

예제

다음 예제에서는 [액션] 패널에 있는 모든 텍스트를 지정된 텍스트로 바꿉니다.

```
f1.actionsPanel.setText("// Deleted this code - no longer needed");
```

참고 사항

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

4장: BitmapInstance 객체

상속 [Element 객체](#) > [Instance 객체](#) > BitmapInstance 객체

지원 버전
Flash MX 2004

설명
BitmapInstance 객체는 Instance 객체의 하위 클래스이며 프레임 내의 비트맵을 나타냅니다([Instance 객체](#) 참조).

메서드 요약
[Instance 객체](#) 메서드 외에도 BitmapInstance 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
bitmapInstance.getBits()	비트맵에서 비트를 가져와서 조작한 다음 Flash에 반환하는 방식으로 비트맵 효과를 만들 수 있습니다.
bitmapInstance.setBits()	기존 비트맵 요소의 비트를 설정합니다.

속성 요약
[Instance 객체](#) 속성 외에도 BitmapInstance 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
bitmapInstance.hPixels	읽기 전용, 비트맵의 폭을 픽셀 단위로 나타내는 정수
bitmapInstance.vPixels	읽기 전용, 비트맵의 높이를 픽셀 단위로 나타내는 정수

bitmapInstance.getBits()

지원 버전
Flash MX 2004

사용법
`bitmapInstance.getBits()`

매개 변수
없음

반환값
width, height, depth, bits 및 cTab(비트맵에 색상표가 포함된 경우) 속성을 포함하는 객체. bits 요소는 바이트의 배열이고 cTab 요소는 "#RRGGBB" 형식의 색상 값 배열입니다. 배열의 길이는 색상표의 길이입니다.

바이트 배열은 DLL 또는 공유 라이브러리에 의해 참조될 때만 사용할 수 있으며 대개 확장 가능한 도구나 효과를 만들 때만 사용합니다. Flash JavaScript와 함께 사용할 DLL을 만드는 방법에 대한 자세한 내용은 501페이지의 “[C 레벨 확장성](#)”을 참조하십시오.

설명

메서드, 비트맵에서 비트를 가져와서 조작한 다음 Flash로 반환하는 방식으로 비트맵 효과를 만들 수 있습니다.

예제

다음 코드에서는 현재 선택된 객체에 대한 참조를 만들고 객체가 비트맵인지 테스트한 다음 비트맵의 높이, 폭 및 비트 심도를 추적합니다.

```
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);
}
```

참고 사항

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

bitmapInstance.hPixels

지원 버전

Flash MX 2004

사용법

`bitmapInstance.hPixels`

설명

읽기 전용 속성, 비트맵의 폭을 가로 픽셀 수로 나타내는 정수입니다.

예제

다음 코드에서는 비트맵의 폭을 픽셀 단위로 가져옵니다.

```
// 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;
}
```

참고 사항

[bitmapInstance.vPixels](#)

bitmapInstance.setBits()

지원 버전

Flash MX 2004

사용법

`bitmapInstance.setBits(bitmap)`

매개 변수

bitmap height, width, depth, bits 및 cTab 속성을 포함하는 객체. height, width 및 depth 속성은 정수이고 bits 속성은 바이트 배열입니다. cTab 속성은 비트 심도가 8 이하인 비트맵에만 필요하며 "#RRGGBB" 형식의 색상 값을 나타내는 문자열입니다.

참고: 바이트 배열은 외부 라이브러리에 의해 참조될 때만 사용할 수 있으며 대개 확장 가능한 도구나 효과를 만들 때만 사용됩니다.

반환값

없음

설명

메서드, 기존 비트맵 요소의 비트를 설정합니다. 이 메서드를 사용하면 비트맵에서 비트를 가져와서 조작한 다음 비트맵을 Flash에 반환하는 방식으로 비트맵 효과를 만들 수 있습니다.

예제

다음 코드는 현재 선택한 항목이 비트맵인지 테스트한 다음 비트맵의 높이를 150픽셀로 설정합니다.

```
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);
}
```

참고 사항

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

bitmapInstance.vPixels

지원 버전

Flash MX 2004

사용법

bitmapInstance.vPixels

설명

읽기 전용 속성, 비트맵의 높이를 세로 픽셀 수로 나타내는 정수입니다.

예제

다음 코드에서는 비트맵의 높이를 픽셀 단위로 가져옵니다.

```
// 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;
}
```

참고 사항

[bitmapInstance.hPixels](#)

5장: BitmapItem 객체

상속 [Item 객체](#) > [BitmapItem 객체](#)

지원 버전
Flash MX 2004

설명

BitmapItem 객체는 문서 라이브러리 내의 비트맵을 나타내며, BitmapItem 객체는 Item 객체의 하위 클래스입니다([Item 객체 참조](#)).

속성 요약

[Item 객체](#) 속성 외에도 BitmapItem 객체에는 다음과 같은 속성이 있습니다.

속성	설명
bitmapItem.allowSmoothing	비트맵을 매끄럽게 할 수 있는지 여부를 지정하는 부울 값
bitmapItem.compressionType	비트맵에 적용된 이미지 압축의 유형을 나타내는 문자열
bitmapItem.fileLastModifiedDate	1970년 1월 1일부터 원본 파일 수정 날짜까지 경과된 시간(초)을 나타냅니다.
bitmapItem.originalCompressionType	항목을 jpeg 파일로 가져왔는지 여부를 지정합니다.
bitmapItem.sourceFileExists	라이브러리로 가져온 파일이 파일을 가져온 위치에 계속 존재하는지 여부를 지정합니다.
bitmapItem.sourceFilesCurrent	라이브러리 항목의 파일 수정 날짜가 가져온 파일의 디스크 수정 날짜와 동일할지 여부를 지정합니다.
bitmapItem.sourceFilePath	라이브러리로 가져온 파일의 경로와 이름입니다.
bitmapItem.useDeblocking	디블로킹이 활성화되어 있는지 여부를 지정합니다.
bitmapItem.useImportedJPEGQuality	가져온 기본 JPEG 품질을 사용할지 여부를 지정하는 부울 값

메서드 요약

[Item 객체](#) 속성 외에도 BitmapItem 객체에는 다음과 같은 메서드가 있습니다.

메서드	설명
bitmapItem.exportToFile()	지정된 항목을 PNG 또는 JPG 파일로 내보냅니다.

bitmapItem.allowSmoothing

지원 버전
Flash MX 2004

사용법

`bitmapItem.allowSmoothing`

설명

속성, 비트맵을 매끄럽게 할 수 있는지(true), 아니면 없는지(false)를 지정하는 부울 값입니다.

예제

다음 코드에서는 현재 문서의 라이브러리에 있는 첫 번째 항목의 allowSmoothing 속성을 true로 설정합니다.

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

bitmapItem.compressionType

지원 버전

Flash MX 2004

사용법

```
bitmapItem.compressionType
```

설명

속성, 비트맵에 적용된 이미지 압축의 유형을 나타내는 문자열입니다. 사용할 수 있는 값은 "photo" 또는 "lossless"입니다. bitmapItem.useImportedJPEGQuality 값이 false인 경우 "photo"는 0부터 100까지의 품질을 사용하는 JPEG를 나타내며, bitmapItem.useImportedJPEGQuality 값이 true인 경우 "photo"는 기본 문서 품질 값을 사용하는 JPEG를 나타냅니다. "lossless" 값은 GIF 또는 PNG 형식을 나타냅니다(bitmapItem.useImportedJPEGQuality 참조).

예제

다음 코드에서는 현재 문서의 라이브러리에 있는 첫 번째 항목의 compressionType 속성을 "photo"로 설정합니다.

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

bitmapItem.exportToFile()

지원 버전

Flash CS4 Professional

사용법

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

매개 변수

fileURI 내보낸 파일의 경로와 이름을 지정하는 문자열(file:/// URI로 표시됨)

반환값

파일을 성공적으로 내보냈으면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값

설명

메서드, 지정된 항목을 PNG 또는 JPG 파일로 내보냅니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 비트맵 항목으로 가정하고 JPG 파일로 내보냅니다.

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

bitmapItem.fileLastModifiedDate

지원 버전

Flash CS4 Professional

사용법

```
bitmapItem.fileLastModifiedDate
```

설명

읽기 전용 속성, 1970년 1월 1일부터 원본 파일을 라이브러리로 가져왔을 때의 파일 수정 날짜까지 경과된 시간(초 단위)을 나타내는 16진수를 포함하는 문자열입니다. 파일이 없으면 이 값은 "00000000"입니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 비트맵 항목으로 가정하고 위에서 설명한 16진수 숫자를 표시합니다.

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

참고 사항

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

bitmapItem.originalCompressionType

지원 버전

Flash CS4 Professional

사용법

```
bitmapItem.originalCompressionType
```

설명

읽기 전용 속성, 지정된 항목을 jpeg 파일로 가져왔는지 여부를 지정하는 문자열입니다. 이 속성에 가능한 값은 "photo"(jpeg 파일의 경우) 및 "lossless"(GIF 및 PNG와 같은 압축되지 않은 파일 유형)입니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 비트맵 항목으로 가정하고 파일을 jpeg 파일로 라이브러리에 가져온 경우 "photo"를 표시하고 그렇지 않은 경우 "lossless"를 표시합니다.

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

참고 사항

[bitmapItem.compressionType](#)

bitmapItem.quality

지원 버전

Flash MX 2004

사용법

`bitmapItem.quality`

설명

속성, 비트맵의 품질을 지정하는 정수입니다. 기본 문서 품질을 사용하려면 -1을 지정하고 그렇지 않으면 0부터 100까지의 정수를 지정합니다. 이 속성은 JPEG 압축에만 사용할 수 있습니다.

예제

다음 코드에서는 현재 문서의 라이브러리에 있는 첫 번째 항목의 `quality` 속성을 65로 설정합니다.

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

bitmapItem.sourceFileExists

지원 버전

Flash CS4 Professional

사용법

`bitmapItem.sourceFileExists`

설명

읽기 전용 속성, 라이브러리로 가져온 파일이 아직 가져온 위치에 있으면 `true`를 반환하고 그렇지 않으면 `false`를 반환하는 부울 값입니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 비트맵 항목으로 가정하고 라이브러리로 가져온 파일이 있으면 `"true"`를 표시합니다.

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

참고 사항

[bitmapItem.sourceFilesCurrent](#),

[bitmapItem.sourceFilePath](#)

bitmapItem.sourceFileIsCurrent

지원 버전

Flash CS4 Professional

사용법

```
bitmapItem.sourceFileIsCurrent
```

설명

읽기 전용 속성, 라이브러리 항목의 파일 수정 날짜가 가져온 파일을 디스크에서 수정한 날짜와 같으면 **true**를 반환하고 그렇지 않으면 **false**를 반환하는 부울 값입니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 비트맵 항목으로 가정하고 가져온 파일이 가져온 이후 디스크에서 수정되지 않았으면 "true"를 표시합니다.

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

참고 사항

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

bitmapItem.sourceFilePath

지원 버전

Flash CS4 Professional

사용법

```
bitmapItem.sourceFilePath
```

설명

읽기 전용 속성, 라이브러리로 가져온 파일의 경로와 이름을 나타내는 문자열(**file:///** URI로 표시됨)입니다.

예제

다음 예제에서는 유형이 "bitmap"인 라이브러리에 있는 모든 항목의 이름과 소스 파일 경로를 표시합니다.

```
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);  
  }  
}
```

참고 사항

[bitmapItem.sourceFileExists](#)

bitmapItem.useDeblocking

지원 버전

Flash CS4 Professional

사용법

```
bitmapItem.useDeblocking
```

설명

속성, 디블로킹이 활성화되었는지(true), 아니면 비활성화되었는지(false)를 지정하는 부울 값입니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 비트맵 항목으로 가정하고 항목에 대해 디블로킹을 활성화합니다.

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

bitmapItem.useImportedJPEGQuality

지원 버전

Flash MX 2004

사용법

```
bitmapItem.useImportedJPEGQuality
```

설명

속성, 가져온 기본 JPEG 품질을 사용할지(true), 아니면 사용하지 않을지(false)를 지정하는 부울 값입니다. 이 속성은 JPEG 압축에만 사용할 수 있습니다.

예제

다음 코드에서는 현재 문서의 라이브러리에 있는 첫 번째 항목의 useImportedJPEGQuality 속성을 true로 설정합니다.

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

6장: CompiledClipInstance 객체

상속 [Element 객체](#) > [Instance 객체](#) > CompiledClipInstance 객체

지원 버전
Flash MX 2004

설명

CompiledClipInstance 객체는 Instance 객체의 하위 클래스입니다. 본질적으로 컴파일된 클립 라이브러리 항목으로 변환된 무비 클립의 인스턴스입니다([Instance 객체](#) 참조).

속성 요약

[Instance 객체](#)의 속성 외에도 CompiledClipInstance 객체에는 다음과 같은 속성이 있습니다.

속성	설명
compiledClipInstance.accName	[액세스 가능성] 패널의 [이름] 필드에 해당하는 문자열
compiledClipInstance.actionScript	이 인스턴스의 ActionScript 를 나타내며 symbolInstance.actionScript 에 해당하는 문자열
compiledClipInstance.description	[액세스 가능성] 패널의 [설명] 필드에 해당하는 문자열
compiledClipInstance.forceSimple	객체의 자식에 대한 액세스 가능 여부를 지정하는 부울 값
compiledClipInstance.shortcut	[액세스 가능성] 패널의 [단축키] 필드에 해당하는 문자열
compiledClipInstance.silent	객체에 대한 액세스 가능 여부를 지정하는 부울 값으로, [액세스 가능성] 패널에 있는 [객체를 액세스 가능하게 만들기] 설정의 역논리에 해당합니다.
compiledClipInstance.tabIndex	[액세스 가능성] 패널의 [탭 인덱스] 필드에 해당하는 정수

compiledClipInstance.accName

지원 버전
Flash MX 2004

사용법

```
compiledClipInstance.accName
```

설명

속성, [액세스 가능성] 패널의 [이름] 필드에 해당하는 문자열입니다. 화면 판독기에서는 이름을 읽어 객체를 식별합니다.

예제

다음 예제에서는 선택된 첫 번째 객체의 액세스 가능성 이름을 가져오고 설정합니다.

```
// 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

지원 버전
Flash MX 2004

사용법

`compiledClipInstance.actionScript`

설명

속성, 이 인스턴스의 `ActionScript`를 나타내며 `symbolInstance.actionScript`에 해당하는 문자열입니다.

예제

다음 코드에서는 지정된 요소에 `ActionScript`를 할당합니다.

```
// 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

지원 버전
Flash MX 2004

사용법

`compiledClipInstance.description`

설명

속성, [액세스 가능성] 패널의 [설명] 필드에 해당하는 문자열입니다. 이 설명은 화면 판독기에서 읽습니다.

예제

다음 예제에서는 `description` 속성을 가져오고 설정하는 방법을 보여 줍니다.

```
// 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

지원 버전
Flash MX 2004

사용법

`compiledClipInstance.forceSimple`

설명

속성, 객체의 자식에 대한 액세스 가능 여부를 지정하는 부울 값으로, [액세스 가능성] 패널에 있는 [자식 객체 액세스 가능] 설정의 역논리에 해당합니다. `forceSimple`이 `true`이면 [자식 객체 액세스 가능] 옵션이 선택되어 있지 않은 것과 같고, `forceSimple`이 `false`이면 [자식 객체 액세스 가능] 옵션이 선택되어 있는 것과 같습니다.

예제

다음 예제에서는 `forceSimple` 속성을 가져오고 설정하는 방법을 보여 줍니다.

```
// 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

지원 버전

Flash MX 2004

사용법

`compiledClipInstance.shortcut`

설명

속성, [액세스 가능성] 패널의 [단축키] 필드에 해당하는 문자열입니다. 이 단축키는 화면 판독기에서 읽습니다. 이 속성은 동적 텍스트 필드에 사용할 수 없습니다.

예제

다음 예제에서는 `shortcut` 속성을 가져오고 설정하는 방법을 보여 줍니다.

```
// 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

지원 버전

Flash MX 2004

사용법

`compiledClipInstance.silent`

설명

속성, 객체에 대한 액세스 가능 여부를 지정하는 부울 값으로, [액세스 가능성] 패널에 있는 [객체를 액세스 가능하게 만들기] 설정의 역논리에 해당합니다. 즉, `silent`가 `true`이면 [객체를 액세스 가능하게 만들기]가 선택 취소되고 `silent`가 `false`이면 [객체를 액세스 가능하게 만들기]가 선택됩니다.

예제

다음 예제에서는 `silent` 속성을 가져오고 설정하는 방법을 보여 줍니다.

```
// 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

지원 버전

Flash MX 2004

사용법

compiledClipInstance.tabIndex

설명

속성, [액세스 가능성] 패널의 [탭 인덱스] 필드에 해당하는 정수입니다. 사용자가 Tab 키를 누를 때 객체가 액세스되는 순서를 지정하는 탭 순서를 만듭니다.

예제

다음 예제에서는 tabIndex 속성을 가져오고 설정하는 방법을 보여 줍니다.

```
// 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;
```

7장: compilerErrors 객체

지원 버전

Flash CS3 Professional

설명

[컴파일러 오류] 패널을 나타내는 `compilerErrors` 객체는 Flash 객체 (`fl`)의 속성이며 `fl.compilerErrors`를 통해 액세스할 수 있습니다([Flash 객체 \(fl\)](#) 참조).

메서드 요약

`compilerErrors` 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
<code>compilerErrors.clear()</code>	[컴파일러 오류] 패널의 내용을 지웁니다.
<code>compilerErrors.save()</code>	[컴파일러 오류] 패널의 내용을 로컬 텍스트 파일에 저장합니다.

`compilerErrors.clear()`

지원 버전

Flash CS3 Professional

사용법

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

매개 변수

없음

반환값

없음

설명

메서드, [컴파일러 오류] 패널의 내용을 지웁니다.

예제

다음 예제에서는 [컴파일러 오류] 패널의 내용을 지웁니다.

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

참고 사항

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

compilerErrors.save()

지원 버전

Flash CS3 Professional

사용법

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

매개 변수

fileURI 저장된 파일의 이름을 지정하는 문자열(file:/// URI로 표시됨). **fileURI**가 이미 존재하고, **bAppendToFile**의 값을 true로 지정하지 않은 경우 경고 없이 **fileURI**를 덮어씁니다.

bAppendToFile [컴파일러 오류] 패널의 내용을 **fileURI**에 추가해야 하는지(true) 또는 추가하지 않아야 하는지(false)를 지정하는 선택적 부울 값. 기본값은 false입니다.

bUseSystemEncoding 시스템 인코딩을 사용하여 [컴파일러 오류] 패널 텍스트를 저장할지 여부를 지정하는 선택적 부울 값. 이 값이 false(기본값)이면 텍스트 시작 부분에 BOM(Byte Order Mark) 문자가 있는 UTF-8 인코딩을 사용하여 [컴파일러 오류] 패널 텍스트를 저장합니다. 기본값은 false입니다.

반환값

없음

설명

메서드, [컴파일러 오류] 패널의 내용을 로컬 텍스트 파일에 저장합니다.

예제

다음 예제에서는 [컴파일러 오류] 패널의 내용을 C:\tests 폴더에 있는 errors.log 파일에 저장합니다.

```
f1.compilerErrors.save("file:///c:/tests/errors.log");
```

참고 사항

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

8장: ComponentInstance 객체

상속 [Element 객체](#) > [Instance 객체](#) > [SymbolInstance 객체](#) > ComponentInstance 객체

지원 버전
Flash MX 2004

설명
ComponentInstance 객체는 SymbolInstance 객체의 하위 클래스이며 프레임 내의 구성 요소를 나타냅니다 ([SymbolInstance 객체](#) 참조).

속성 요약
[SymbolInstance 객체](#)의 속성 외에도 ComponentInstance 객체에는 다음과 같은 속성이 있습니다.

속성	설명
componentInstance.parameters	읽기 전용, 구성 요소 속성 관리자에서 액세스할 수 있는 ActionScript 2.0 속성의 배열

componentInstance.parameters

지원 버전
Flash MX 2004

사용법
`componentInstance.parameters`

설명
읽기 전용 속성, 구성 요소 속성 관리자에서 액세스할 수 있는 ActionScript 2.0 속성의 배열입니다. [Parameter 객체](#)를 참조하십시오.

예제
다음 예제에서는 parameters 속성을 가져오고 설정하는 방법을 보여 줍니다.

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

참고 사항
[Parameter 객체](#)

9장: componentsPanel 객체

지원 버전

Flash MX 2004

설명

[구성 요소] 패널을 나타내는 componentsPanel 객체는 Flash 객체(fl)의 속성이며 fl.componentsPanel을 통해 액세스할 수 있습니다(Flash 객체(fl) 참조).

메서드 요약

componentsPanel 객체에서는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
componentsPanel.addItemToDocument()	지정된 위치의 문서에 특정 구성 요소를 추가합니다.
componentsPanel.reload()	[구성 요소] 패널의 구성 요소 목록을 새로 고칩니다.

componentsPanel.addItemToDocument()

지원 버전

Flash MX 2004

사용법

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

매개 변수

position 구성 요소를 추가할 위치를 지정하는 점(예: {x:0,y:100}). 구성 요소의 등록 포인트(원점)가 아닌 구성 요소의 중심점을 기준으로 **position**을 지정합니다.

categoryName 구성 요소 범주의 이름을 지정하는 문자열(예: "Data"). [구성 요소] 패널에는 유효한 범주 이름이 나열되어 있습니다.

componentName 지정된 범주에 있는 구성 요소의 이름을 지정하는 문자열(예: "WebServiceConnector"). [구성 요소] 패널에는 유효한 구성 요소 이름이 나열되어 있습니다.

반환값

없음

설명

지정된 위치의 문서에 특정 구성 요소를 추가합니다.

예제

다음 예제에서는 이 메서드를 사용하는 몇 가지 방법을 보여 줍니다.

```
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()

지원 버전

Flash 8

사용법

```
componentsPanel.reload()
```

매개 변수

없음

반환값

[구성 요소] 패널 목록이 새로 고쳐지면 `true`를 반환하고 그렇지 않으면 `false`를 반환하는 부울 값

설명

메서드, [구성 요소] 패널의 구성 요소 목록을 새로 고칩니다.

예제

다음 예제에서는 [구성 요소] 패널을 새로 고칩니다.

```
f1.componentsPanel.reload();
```


10장: Contour 객체

지원 버전

Flash MX 2004

설명

Contour 객체는 모양의 경계에 있는 양 반쪽 가장자리의 닫힌 경로를 나타냅니다.

메서드 요약

Contour 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
<code>contour.getHalfEdge()</code>	선택 영역의 윤곽선에서 HalfEdge 객체를 반환합니다.

속성 요약

Contour 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>contour.fill</code>	Fill 객체
<code>contour.interior</code>	읽기 전용, 윤곽선이 영역을 둘러싸면 true 이고 그렇지 않으면 false 입니다.
<code>contour.orientation</code>	읽기 전용, 윤곽선의 방향을 나타내는 정수

contour.fill

지원 버전

Flash CS4 Professional

사용법

`contour.fill`

설명

속성, **Fill** 객체입니다.

예제

다음 예제에서는 채우기가 선택된 윤곽선이 있다고 가정하고 [출력] 패널에 윤곽선의 채움 색상을 표시합니다.

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

contour.getHalfEdge()

지원 버전
Flash MX 2004

사용법
contour.getHalfEdge()

매개 변수
없음

반환값
[HalfEdge 객체](#)

설명
메서드, 선택 영역의 윤곽선에서 [HalfEdge 객체](#)를 반환합니다.

예제
이 예제에서는 선택한 모양의 모든 윤곽선을 순회하고 꼭지점의 좌표를 [출력] 패널에 표시합니다.

```
// 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

지원 버전
Flash MX 2004

사용법
contour.interior

설명
읽기 전용 속성, 윤곽선이 영역을 둘러싸면 true이고 그렇지 않으면 false입니다.

예제
이 예제에서는 선택한 모양의 모든 윤곽선을 순회하고 각 윤곽선의 interior 속성 값을 [출력] 패널에 표시합니다.

```
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

지원 버전
Flash MX 2004

사용법
contour.orientation

설명
읽기 전용 속성, 윤곽선의 방향을 나타내는 정수입니다. 방향이 시계 반대 방향이면 정수 값은 -1이고, 시계 방향이면 1이며, 윤곽선에 포함된 영역이 없으면 0입니다.

예제
다음 예제에서는 선택한 모양의 모든 윤곽선을 순회하고 각 윤곽선의 orientation 속성 값을 [출력] 패널에 표시합니다.

```
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();
```

11장: Document 객체

지원 버전

Flash MX 2004

설명

Document 객체는 스테이지를 나타냅니다. 즉, FLA 파일만 문서로 간주됩니다. 현재 문서에 대한 Document 객체를 변환하려면 `fl.getDocumentDOM()`을 사용합니다.

메서드 요약

Document 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
<code>document.addDataToDocument()</code>	지정된 데이터를 문서와 함께 저장합니다.
<code>document.addDataToSelection()</code>	지정된 데이터를 선택한 객체와 함께 저장합니다.
<code>document.addFilter()</code>	선택한 객체에 필터를 적용합니다.
<code>document.addItem()</code>	열려 있는 문서 또는 라이브러리의 항목을 지정된 Document 객체에 추가합니다.
<code>document.addNewLine()</code>	두 점 사이에 새 경로를 추가합니다.
<code>document.addNewOval()</code>	지정된 경계 사각형에 새 Oval 객체를 추가합니다.
<code>document.addNewPrimitiveOval()</code>	지정된 경계에 맞게 새 프리미티브 타원을 추가합니다.
<code>document.addNewPrimitiveRectangle()</code>	지정된 경계에 맞게 새 프리미티브 사각형을 추가합니다.
<code>document.addNewPublishProfile()</code>	새 제작 프로파일을 추가하고 이 프로파일을 현재 프로파일로 만듭니다.
<code>document.addNewRectangle()</code>	지정된 경계에 맞게 새 사각형 또는 둥근 사각형을 추가합니다.
<code>document.addNewScene()</code>	새 장면(Timeline 객체)을 현재 선택한 장면의 다음 장면으로 추가하고 이 장면을 현재 선택한 장면으로 만듭니다.
<code>document.addNewText()</code>	새로운 빈 텍스트 필드를 삽입합니다.
<code>document.align()</code>	선택 항목을 정렬합니다.
<code>document.allowScreens()</code>	129페이지의 "document.screenOutline" 속성을 사용하기 전에 사용합니다.
<code>document.arrange()</code>	스테이지에서 선택 항목을 정렬합니다.
<code>document.breakApart()</code>	현재 선택 영역에서 분리 작업을 수행합니다.
<code>document.canEditSymbol()</code>	심볼 편집 메뉴와 기능이 활성화되어 있는지 여부를 나타냅니다.
<code>document.canRevert()</code>	<code>document.revert()</code> 또는 <code>fl.revertDocument()</code> 메서드를 성공적으로 사용할 수 있는지 여부를 나타냅니다.
<code>document.canTestMovie()</code>	<code>document.testMovie()</code> 메서드를 성공적으로 사용할 수 있는지 여부를 나타냅니다.

메서드	설명
<code>document.canTestScene()</code>	<code>document.testScene()</code> 메서드를 성공적으로 사용할 수 있는지 여부를 나타냅니다.
<code>document.changeFilterOrder()</code>	필터 목록에서 필터의 인덱스를 변경합니다.
<code>document.clipCopy()</code>	문서의 현재 선택 영역을 클립보드에 복사합니다.
<code>document.clipCut()</code>	문서의 현재 선택 영역을 잘라내어 클립보드에 복사합니다.
<code>document.clipPaste()</code>	클립보드 내용을 문서에 붙여넣습니다.
<code>document.close()</code>	지정된 문서를 닫습니다.
<code>document.convertLinesToFills()</code>	선택한 객체에서 선을 채우기로 변환합니다.
<code>document.convertToSymbol()</code>	선택한 스테이지 항목을 새 심볼로 변환합니다.
<code>document.crop()</code>	맨 위에 선택된 Drawing 객체를 사용하여 그 아래에 선택되어 있는 모든 Drawing 객체를 자릅니다.
<code>document.debugMovie()</code>	해당 문서를 사용하여 디버그 세션을 시작합니다.
<code>document.deleteEnvelope()</code>	선택한 객체에서 엔벨로프(하나 이상의 객체를 포함하는 경계 상자)를 삭제합니다.
<code>document.deletePublishProfile()</code>	프로파일이 둘 이상 있는 경우 현재 활성화된 프로파일을 삭제합니다.
<code>document.deleteScene()</code>	현재 장면(Timeline 객체)을 삭제하고, 삭제된 장면이 마지막 장면이 아닐 경우 그 다음 장면을 현재 Timeline 객체로 설정합니다.
<code>document.deleteSelection()</code>	스테이지에서 현재 선택 항목을 삭제합니다.
<code>document.disableAllFilters()</code>	선택한 객체에서 모든 필터를 비활성화합니다.
<code>document.disableFilter()</code>	[필터] 목록에서 지정된 필터를 비활성화합니다.
<code>document.disableOtherFilters()</code>	[필터] 목록에서 지정된 위치의 필터를 제외한 모든 필터를 비활성화합니다.
<code>document.distribute()</code>	선택 항목을 배포합니다.
<code>document.distributeToLayers()</code>	현재 선택 항목에서 레이어에 배포 작업을 수행합니다. [레이어에 배포]를 선택하는 것과 같습니다.
<code>document.documentHasData()</code>	문서에 특정 이름을 가진 영구 데이터가 있는지 확인합니다.
<code>document.duplicatePublishProfile()</code>	현재 활성화된 프로파일을 복제하고 이 복제된 프로파일에 포커스를 둡니다.
<code>document.duplicateScene()</code>	현재 선택한 장면의 복사본을 만들고 고유한 이름을 지정한 다음 이 장면을 현재 장면으로 만듭니다.
<code>document.duplicateSelection()</code>	스테이지에서 선택 항목을 복제합니다.
<code>document.editScene()</code>	지정된 장면을 편집하기 위해 현재 선택한 장면으로 만듭니다.
<code>document.enableAllFilters()</code>	선택한 객체에 대한 필터 목록에서 모든 필터를 활성화합니다.
<code>document.enableFilter()</code>	선택한 객체에 대해 특정 필터를 활성화합니다.
<code>document.enterEditMode()</code>	제작 도구를 매개 변수에 의해 지정된 편집 모드로 전환합니다.
<code>document.exitEditMode()</code>	심볼 편집 모드를 종료하고 편집 모드보다 한 레벨 위로 포커스를 되돌립니다.

메서드	설명
<code>document.exportPNG()</code>	문서를 하나 이상의 PNG 파일로 내보냅니다.
<code>document.exportPublishProfile()</code>	현재 활성화된 프로파일을 XML 파일로 내보냅니다.
<code>document.exportPublishProfileString()</code>	지정된 프로파일을 XML 형식으로 지정하는 문자열을 반환합니다.
<code>document.exportSWF()</code>	Flash SWF 포맷으로 문서를 내보냅니다.
<code>document.getAlignToDocument()</code>	[정렬] 패널의 [스테이지로] 버튼 값을 가져오는 것과 같습니다.
<code>document.getBlendMode()</code>	선택한 객체에 대해 혼합 모드를 지정하는 문자열을 반환합니다.
<code>document.getCustomFill()</code>	선택한 모양의 Fill 객체를 가져오거나, 지정된 경우 [도구] 패널 및 속성 관리자의 Fill 객체를 가져옵니다.
<code>document.getCustomStroke()</code>	선택한 모양의 Stroke 객체를 반환하거나, 지정된 경우 [도구] 패널 및 속성 관리자의 Stroke 객체를 반환합니다.
<code>document.getDataFromDocument()</code>	지정된 데이터의 값을 가져옵니다.
<code>document.getElementProperty()</code>	현재 선택 항목의 지정된 Element 속성을 가져옵니다.
<code>document.getElementTextAttr()</code>	선택한 텍스트 객체의 지정된 TextAttrs 속성을 가져옵니다.
<code>document.getFilters()</code>	현재 선택되어 있는 객체에 적용된 필터의 목록이 포함된 배열을 반환합니다.
<code>document.getMetadata()</code>	문서와 관련된 XML 메타데이터가 포함된 문자열을 반환합니다.
<code>document.getMobileSettings()</code>	<code>document.setMobileSettings()</code> 에 전달된 문자열을 반환합니다.
<code>document.getPlayerVersion()</code>	지정한 문서의 대상 플레이어 버전을 나타내는 문자열을 반환합니다.
<code>document.getSelectionRect()</code>	현재 선택 항목의 경계 사각형을 가져옵니다.
<code>document.getTextString()</code>	현재 선택한 텍스트를 가져옵니다.
<code>document.getTimeline()</code>	문서에서 현재 Timeline 객체를 가져옵니다.
<code>document.getTransformationPoint()</code>	현재 선택 항목의 변형점 위치를 가져옵니다.
<code>document.group()</code>	현재 선택 항목을 그룹으로 변환합니다.
<code>document.importFile()</code>	파일을 문서로 가져옵니다.
<code>document.importPublishProfile()</code>	파일에서 프로파일을 가져옵니다.
<code>document.importPublishProfileString()</code>	제작 프로파일을 나타내는 XML 문자열을 가져오고 현재 프로파일로 설정합니다.
<code>document.importSWF()</code>	SWF 파일을 문서로 가져옵니다.
<code>document.intersect()</code>	선택한 모든 Drawing 객체에서 교차 Drawing 객체를 만듭니다.
<code>document.loadCuepointXML()</code>	큐 포인트 XML 파일을 로드합니다.
<code>document.match()</code>	선택한 객체의 크기를 동일하게 만듭니다.
<code>document.mouseClick()</code>	[선택 도구]에서 마우스 클릭을 수행합니다.
<code>document.mouseDbClick()</code>	[선택 도구]에서 마우스 두 번 클릭을 수행합니다.
<code>document.moveSelectedBezierPointsBy()</code>	선택 항목에 베지어 점이 최소 하나 이상 선택되어 있는 경로가 적어도 하나 이상 포함되어 있으면 이 메서드는 선택한 모든 경로에서 선택한 모든 베지어 점을 지정된 양만큼 이동합니다.

메서드	설명
<code>document.moveSelectionBy()</code>	선택한 객체를 지정된 거리만큼 이동합니다.
<code>document.optimizeCurves()</code>	최적의 매끄러움을 얻기 위해 여러 번 수행(지정된 경우)을 통해 현재 선택 항목의 매끄러움을 최적화합니다. [수정] > [모양] > [최적화]를 선택하는 것과 같습니다.
<code>document.publish()</code>	활성 제작 설정([파일] > [제작 설정])에 따라 문서를 제작합니다. [파일] > [제작]을 선택하는 것과 같습니다.
<code>document.punch()</code>	맨 위에 선택된 Drawing 객체를 사용하여 그 아래에 선택되어 있는 모든 Drawing 객체를 펀칭합니다.
<code>document.removeAllFilters()</code>	선택한 객체에서 모든 필터를 제거합니다.
<code>document.removeDataFromDocument()</code>	문서에 첨부된 특정 이름을 가진 영구 데이터를 제거합니다.
<code>document.removeDataFromSelection()</code>	선택 영역에 첨부된 특정 이름을 가진 영구 데이터를 제거합니다.
<code>document.removeFilter()</code>	선택한 객체의 [필터] 목록에서 지정된 필터를 제거합니다.
<code>document.renamePublishProfile()</code>	현재 프로파일의 이름을 변경합니다.
<code>document.renameScene()</code>	[장면] 패널에서 현재 선택한 장면의 이름을 변경합니다.
<code>document.reorderScene()</code>	특정 장면을 지정된 다른 장면 앞으로 이동합니다.
<code>document.resetOvalObject()</code>	속성 관리자의 모든 값을 기본 Oval 객체 설정으로 설정합니다.
<code>document.resetRectangleObject()</code>	속성 관리자의 모든 값을 기본 Rectangle 객체 설정으로 설정합니다.
<code>document.resetTransformation()</code>	변형 행렬을 재설정합니다. [수정] > [변형] > [변형 제거]를 선택하는 것과 같습니다.
<code>document.revert()</code>	지정된 문서를 이전에 저장한 버전으로 되돌립니다. [파일] > [복귀]를 선택하는 것과 같습니다.
<code>document.rotate3DSelection()</code>	선택 항목에 3D 회전을 적용합니다.
<code>document.rotateSelection()</code>	선택 항목을 지정된 각도만큼 회전합니다.
<code>document.save()</code>	문서를 기본 위치에 저장합니다. [파일] > [저장]을 선택하는 것과 같습니다.
<code>document.saveAndCompact()</code>	파일을 저장하고 압축합니다. [파일] > [저장 및 압축]을 선택하는 것과 같습니다.
<code>document.scaleSelection()</code>	지정된 양만큼 선택 항목의 크기를 조절합니다. [자유 변형 도구]를 사용하여 객체의 크기를 조절하는 것과 같습니다.
<code>document.selectAll()</code>	스태이지에 있는 모든 항목을 선택합니다. Ctrl+A(Windows) 또는 Command+A(Macintosh) 를 누르거나 [편집] > [전체 선택]을 선택하는 것과 같습니다.
<code>document.selectNone()</code>	선택한 항목을 선택 취소합니다.
<code>document.setAlignToDocument()</code>	문서에서 사용할 수 있는 <code>document.align()</code> , <code>document.distribute()</code> , <code>document.match()</code> 및 <code>document.space()</code> 에 대한 환경 설정을 설정합니다. [정렬] 패널에서 [스태이지로] 버튼을 활성화하는 것과 같습니다.
<code>document.setBlendMode()</code>	선택한 객체의 혼합 모드를 설정합니다.

메서드	설명
<code>document.setCustomFill()</code>	[도구] 패널, 속성 관리자 및 선택한 모양에 대한 채우기 설정을 설정합니다.
<code>document.setCustomStroke()</code>	[도구] 패널, 속성 관리자 및 선택한 모양에 대한 획 설정을 설정합니다.
<code>document.setElementProperty()</code>	선택한 문서 객체에 지정된 Element 속성을 설정합니다.
<code>document.setElementTextAttr()</code>	선택한 텍스트 항목의 지정된 TextAttrs 속성을 특정 값으로 설정합니다.
<code>document.setFillColor()</code>	선택 항목의 채움 색상을 지정된 색상으로 변경합니다.
<code>document.setFilterProperty()</code>	현재 선택한 객체에 대해 지정된 필터 속성을 설정합니다.
<code>document.setFilters()</code>	선택한 객체에 필터를 적용합니다.
<code>document.setInstanceAlpha()</code>	인스턴스의 불투명도를 설정합니다.
<code>document.setInstanceBrightness()</code>	인스턴스의 밝기를 설정합니다.
<code>document.setInstanceTint()</code>	인스턴스의 농도를 설정합니다.
<code>document.setMetadata()</code>	기존 메타데이터를 덮어쓰는 방식으로 지정된 문서에 대한 XML 메타데이터를 설정합니다.
<code>document.setMobileSettings()</code>	모바일 FLA 파일에 있는 XML 설정 문자열의 값을 설정합니다.
<code>document.setOvalObjectProperty()</code>	프리미티브 Oval 객체의 지정된 속성에 대한 값을 지정합니다.
<code>document.setPlayerVersion()</code>	지정된 문서에서 대상으로 지정한 Flash Player의 버전을 설정합니다.
<code>document.setRectangleObjectProperty()</code>	프리미티브 Rectangle 객체의 지정된 속성에 대한 값을 지정합니다.
<code>document.setSelectionBounds()</code>	한 번의 작업으로 선택 항목을 이동하고 크기를 조절합니다.
<code>document.setSelectionRect()</code>	지정된 좌표를 사용하여 스테이지를 기준으로 하는 사각형 선택 윤곽을 그립니다.
<code>document.setStageVanishingPoint()</code>	3D 객체를 보기 위한 소실점을 지정합니다.
<code>document.setStageViewAngle()</code>	3D 객체를 보기 위한 원근감 각도를 지정합니다.
<code>document.setStroke()</code>	선택한 획의 색상, 폭 및 스타일을 설정합니다.
<code>document.setStrokeColor()</code>	선택 항목의 획 색상을 지정된 색상으로 변경합니다.
<code>document.setStrokeSize()</code>	선택 항목의 획 크기를 지정된 크기로 변경합니다.
<code>document.setStrokeStyle()</code>	선택 항목의 획 스타일을 지정된 스타일로 변경합니다.
<code>document.setTextRectangle()</code>	선택한 텍스트 항목의 경계 사각형을 지정된 크기로 변경합니다.
<code>document.setTextSelection()</code>	현재 선택한 텍스트 필드의 텍스트 선택 영역을 <i>startIndex</i> 및 <i>endIndex</i> 값에 의해 지정된 값으로 설정합니다.
<code>document.setTextString()</code>	텍스트 문자열을 삽입합니다.
<code>document.setTransformationPoint()</code>	현재 선택 항목의 변형점을 이동합니다.
<code>document.skewSelection()</code>	선택 항목을 지정된 양만큼 기울입니다.
<code>document.smoothSelection()</code>	선택한 각 곡선 또는 채우기 외곽선의 곡선을 매끄럽게 만듭니다.

메서드	설명
<code>document.space()</code>	선택 항목 내 각 객체의 간격을 일정하게 만듭니다.
<code>document.straightenSelection()</code>	현재 선택한 획을 곧게 만듭니다. [도구] 패널의 [곧게] 버튼을 사용하는 것과 같습니다.
<code>document.swapElement()</code>	현재 선택 항목을 지정된 항목으로 교체합니다.
<code>document.swapStrokeAndFill()</code>	[획] 및 [채움] 색상을 교체합니다.
<code>document.testMovie()</code>	문서에서 무비 테스트 작업을 실행합니다.
<code>document.testScene()</code>	문서의 현재 장면에서 장면 테스트 작업을 실행합니다.
<code>document.traceBitmap()</code>	현재 선택 항목에서 비트맵 추적을 수행합니다. [수정] > [비트맵] > [비트맵 추적]을 선택하는 것과 같습니다.
<code>document.transformSelection()</code>	인수에 지정된 행렬을 적용하는 방식으로 현재 선택 항목에서 일반적인 변형을 수행합니다.
<code>document.translate3DCenter()</code>	선택 항목이 평행 이동되거나 회전되는 중심 XYZ 위치를 설정합니다.
<code>document.translate3DSelection()</code>	선택 항목에 3D 평행 이동을 적용합니다.
<code>document.unGroup()</code>	현재 선택 항목의 그룹을 해제합니다.
<code>document.union()</code>	선택한 모든 모양을 Drawing 객체로 결합합니다.
<code>document.unlockAllElements()</code>	현재 선택한 프레임에서 모든 잠긴 요소의 잠금을 해제합니다.
<code>document.xmlPanel()</code>	XMLUI 대화 상자를 게시합니다.

속성 요약

Document 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>document.accName</code>	[액세스 가능성] 패널의 [이름] 필드에 해당하는 문자열
<code>document.as3AutoDeclare</code>	스테이지에 배치된 인스턴스가 사용자 정의 타임라인 클래스에 자동으로 추가되는지 여부를 나타내는 부울 값
<code>document.as3Dialect</code>	지정된 문서에서 사용 중인 ActionScript 3.0 "언어"를 설명하는 문자열입니다.
<code>document.as3ExportFrame</code>	ActionScript 3.0 클래스를 내보내는 프레임을 지정하는 정수
<code>document.as3StrictMode</code>	ActionScript 3.0 컴파일러에서 컴파일할 때 엄격 모드 옵션이 켜져 있어야 하는지 아니면 꺼져 있어야 하는지를 지정하는 부울 값
<code>document.as3WarningsMode</code>	ActionScript 3.0 컴파일러에서 컴파일할 때 경고 모드 옵션이 켜져 있어야 하는지 아니면 꺼져 있어야 하는지를 지정하는 부울 값
<code>document.asVersion</code>	지정된 파일에 사용되고 있는 ActionScript의 버전을 지정하는 정수
<code>document.autoLabel</code>	[액세스 가능성] 패널의 [자동 레이블] 체크 상자에 해당하는 부울 값
<code>document.backgroundColor</code>	배경색을 나타내는 문자열, 16진수 값 또는 정수
<code>document.currentPublishProfile</code>	지정된 문서에 대한 활성 제작 프로파일의 이름을 지정하는 문자열
<code>document.currentTimeline</code>	활성 타임라인의 인덱스를 지정하는 정수
<code>document.description</code>	[액세스 가능성] 패널의 [설명] 필드에 해당하는 문자열

속성	설명
<code>document.docClass</code>	문서에 연결된 최상위 ActionScript 3.0 클래스를 지정합니다.
<code>document.externalLibraryPath</code>	런타임 공유 라이브러리로 사용되는 SWC 파일의 위치를 지정하는 문서의 ActionScript 3.0 외부 라이브러리 경로에 있는 항목의 목록이 포함된 문자열
<code>document.forceSimple</code>	지정된 객체의 자식에 액세스할 수 있는지 여부를 지정하는 부울 값
<code>document.frameRate</code>	SWF 파일이 재생될 때 표시되는 초당 프레임 수를 지정하는 부동 소수점 값. 기본 값은 12입니다.
<code>document.height</code>	문서(스태이지)의 높이를 픽셀 단위로 지정하는 정수
<code>document.id</code>	Flash 세션 동안 문서를 식별하는 고유한 정수(자동으로 할당됨)
<code>document.library</code>	읽기 전용, 문서에 대한 library 객체
<code>document.libraryPath</code>	SWC 파일이 포함된 폴더나 SWC 파일의 위치를 지정하는 문서의 ActionScript 3.0 라이브러리 경로에 있는 항목의 목록이 포함된 문자열입니다.
<code>document.livePreview</code>	실시간 미리 보기가 활성화되어 있는지 여부를 지정하는 부울 값
<code>document.name</code>	읽기 전용, 문서(FLA 파일)의 이름을 나타내는 문자열
<code>document.path</code>	읽기 전용, 문서의 경로를 플랫폼별 형식으로 나타내는 문자열
<code>document.pathURI</code>	읽기 전용, 문서의 경로를 나타내는 문자열(file:/// URI로 표시됨)
<code>document.publishProfiles</code>	읽기 전용, 문서에 대한 제작 프로파일 이름의 배열
<code>document.screenOutline</code>	읽기 전용, 문서에 대한 현재 ScreenOutline 객체. Flash Professional CS5부터는 ScreenOutline 객체가 사용되지 않음
<code>document.selection</code>	선택한 문서 객체의 배열
<code>document.silent</code>	객체에 액세스할 수 있는지 여부를 지정하는 부울 값
<code>document.sourcePath</code>	ActionScript 클래스 파일의 위치를 지정하는 문서의 ActionScript 3.0 소스 경로에 있는 항목의 목록이 포함된 문자열
<code>document.timelines</code>	읽기 전용, Timeline 객체의 배열(Timeline 객체 참조)
<code>document.viewMatrix</code>	읽기 전용, Matrix 객체
<code>document.width</code>	문서(스태이지)의 너비를 픽셀 단위로 지정하는 정수
<code>document.zoomFactor</code>	제작 시 스태이지의 확대/축소 퍼센트를 지정합니다.

document.accName

지원 버전
Flash MX 2004

사용법
`document.accName`

설명
속성, [액세스 가능성] 패널의 [이름] 필드에 해당하는 문자열입니다. 화면 판독기에서는 이름을 읽어 객체를 식별합니다.

예제

다음 예제에서는 문서의 액세스 가능성 이름을 "Main Movie"로 설정합니다.

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

다음 예제에서는 문서의 액세스 가능성 이름을 가져옵니다.

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

document.addDataToDocument()

지원 버전

Flash MX 2004

사용법

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

매개 변수

name 추가할 데이터의 이름을 지정하는 문자열

type 추가할 데이터의 유형을 정의하는 문자열. 사용할 수 있는 값에는 "integer", "integerArray", "double", "doubleArray", "string" 및 "byteArray"가 있습니다.

data 추가할 값. 사용할 수 있는 유형은 **type** 매개 변수에 따라 달라집니다.

반환값

없음

설명

메서드, 지정된 데이터를 문서와 함께 저장합니다. 데이터는 FLA 파일에 저장된 후 파일을 다시 열면 JavaScript에 사용할 수 있습니다.

예제

다음 예제에서는 현재 문서에 정수 값 12를 추가합니다.

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

다음 예제에서는 "myData"라는 데이터의 값을 반환하고 [출력] 패널에 결과를 표시합니다.

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

참고 사항

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

document.addDataToSelection()

지원 버전

Flash MX 2004

사용법

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

매개 변수

name 영구 데이터의 이름을 지정하는 문자열

type 데이터의 유형을 정의합니다. 사용할 수 있는 값에는 "integer", "integerArray", "double", "doubleArray", "string" 및 "byteArray"가 있습니다.

data 추가할 값. 사용할 수 있는 유형은 **type** 매개 변수에 따라 달라집니다.

반환값

없음

설명

메서드, 지정된 데이터를 선택한 객체와 함께 저장합니다. 데이터는 FLA 파일에 저장된 후 파일을 다시 열면 JavaScript에 사용할 수 있습니다. 심볼과 비트맵만 영구 데이터를 지원합니다.

예제

다음 예제에서는 선택한 객체에 정수 값 12를 추가합니다.

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

참고 사항

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

document.addFilter()

지원 버전

Flash 8

사용법

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

매개 변수

filterName [필터] 목록에 추가하고 선택한 객체에 대해 활성화할 필터를 지정하는 문자열입니다. 사용할 수 있는 값에는 "adjustColorFilter", "bevelFilter", "blurFilter", "dropShadowFilter", "glowFilter", "gradientBevelFilter" 및 "gradientGlowFilter"가 있습니다.

반환값

없음

설명

메서드, 선택한 객체에 필터를 적용하고 [필터] 목록의 끝에 필터를 추가합니다.

예제

다음 예제에서는 선택한 객체에 광선 필터를 적용합니다.

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

참고 사항

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

document.addItem()

지원 버전

Flash MX 2004

사용법

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

매개 변수

position 항목을 추가할 위치의 **x** 및 **y** 좌표를 지정하는 점으로 심볼의 중심점이나 비트맵 또는 비디오의 왼쪽 위 모서리를 사용합니다.

item 추가할 항목과 추가할 항목이 있는 라이브러리를 지정하는 **Item** 객체([Item 객체 참조](#))

반환값

부울 값, 성공적으로 실행되면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

설명

메서드, 열려 있는 문서 또는 라이브러리의 항목을 지정된 **Document** 객체에 추가합니다.

예제

다음 예제에서는 선택한 심볼, 비트맵 또는 비디오를 위해 지정된 위치의 첫 번째 문서에 첫 번째 라이브러리 항목을 추가합니다.

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

다음 예제에서는 현재 문서의 라이브러리에 있는 심볼 **myMovieClip**을 현재 문서에 추가합니다.

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

다음 예제에서는 문서 배열의 두 번째 문서에 있는 심볼 **myMovieClip**을 문서 배열의 세 번째 문서에 추가합니다.

```
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()

지원 버전

Flash MX 2004

사용법

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

매개 변수

startpoint 줄이 시작되는 **x** 및 **y** 좌표를 지정하는 부동 소수점 숫자 쌍

endpoint 줄이 끝나는 **x** 및 **y** 좌표를 지정하는 부동 소수점 숫자 쌍

반환값

없음

설명

메서드, 두 점 사이에 새 경로를 추가합니다. 이 메서드는 문서의 현재 획 특성을 사용하여 현재 프레임과 레이어에 경로를 추가하며 선 도구를 클릭하고 선을 그리는 것과 동일한 방식으로 작동합니다.

예제

다음 예제에서는 지정된 시작점과 끝점 사이에 선을 추가합니다.

```
f1.getDocumentDOM().addNewLine({x:216.7, y:122.3}, {x:366.8, y:165.8});
```

document.addNewOval()

지원 버전

Flash MX 2004

사용법

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

매개 변수

boundingRectangle 추가할 타원의 경계를 지정하는 사각형. **boundingRectangle**의 형식에 대한 자세한 내용은 [document.addNewRectangle\(\)](#)을 참조하십시오.

bSuppressFill true로 설정된 경우 메서드가 채우기가 적용되지 않은 모양을 만들도록 하는 부울 값. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

bSuppressStroke true로 설정된 경우 메서드가 획이 적용되지 않은 모양을 만들도록 하는 부울 값. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 지정된 경계 사각형에 새 **Oval** 객체를 추가합니다. 이 메서드는 [타원형 도구]와 동일한 작업을 수행하며 문서의 현재 기본 획과 채우기 특성을 사용하여 현재 프레임과 레이어에 타원을 추가합니다. **bSuppressFill**과 **bSuppressStroke**를 모두 true로 설정하면 아무런 영향을 주지 않습니다.

예제

다음 예제에서는 지정된 좌표 내에 새로운 타원을 추가합니다. 이 타원의 너비는 164픽셀이며 높이는 178픽셀입니다.

```
f1.getDocumentDOM().addNewOval({left:72,top:50,right:236,bottom:228});
```

다음 예제에서는 채우기가 적용되지 않은 타원을 그립니다.

```
f1.getDocumentDOM().addNewOval({left:72,top:50,right:236,bottom:228}, true);
```

다음 예제에서는 획이 적용되지 않은 타원을 그립니다.

```
f1.getDocumentDOM().addNewOval({left:72,top:50,right:236,bottom:228}, false, true);
```

참고 사항

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

document.addNewPrimitiveOval()

지원 버전

Flash CS4 Professional

사용법

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

매개 변수

boundingRectangle 새 프리미티브 타원을 추가할 경계를 지정하는 사각형. **boundingRectangle**의 형식에 대한 자세한 내용은 [document.addNewRectangle\(\)](#)을 참조하십시오.

bSuppressFill true로 설정된 경우 메시드가 채우기가 적용되지 않은 타원을 만들도록 하는 부울 값. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

bSuppressStroke true로 설정된 경우 메시드가 획이 적용되지 않은 타원을 만들도록 하는 부울 값. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메시드, 지정된 경계에 맞게 새 프리미티브 타원을 추가합니다. 이 메시드는 [프리미티브 타원 도구]와 동일한 작업을 수행합니다. 프리미티브 타원은 문서의 현재 기본 획과 채우기 특성을 사용하여 현재 프레임과 레이어에 추가됩니다. **bSuppressFill**과 **bSuppressStroke**를 모두 true로 설정하면 아무런 영향을 주지 않습니다.

예제

다음 예제에서는 획과 채우기를 적용하는 방법과 적용하지 않는 방법으로 지정된 좌표 안에 프리미티브 타원을 추가합니다.

```
// 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);
```

참고 사항

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

document.addNewPrimitiveRectangle()

지원 버전

Flash CS4 Professional

사용법

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

매개 변수

rect 새 프리미티브 사각형을 추가할 경계를 지정하는 사각형. **boundingRectangle**의 형식에 대한 자세한 내용은 [document.addNewRectangle\(\)](#)을 참조하십시오.

roundness 모서리의 둥근 정도를 지정하는 데 사용되는 점의 수를 나타내는 0과 999 사이의 정수

bSuppressFill true로 설정된 경우 메시드가 채우기가 적용되지 않은 사각형을 만들도록 하는 부울 값. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

bSuppressStroke true로 설정된 경우 메시드가 획이 적용되지 않은 사각형을 만들도록 하는 부울 값. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메시드, 지정된 경계에 맞게 새 프리미티브 사각형을 추가합니다. 이 메시드는 [프리미티브 사각형 도구]와 동일한 작업을 수행합니다. 프리미티브 사각형은 문서의 현재 기본 획과 채우기 특성을 사용하여 현재 프레임과 레이어에 추가됩니다.

bSuppressFill과 **bSuppressStroke**를 모두 true로 설정하면 아무런 영향을 주지 않습니다.

예제

다음 예제에서는 획과 채우기를 적용하는 방법과 적용하지 않는 방법으로 각기 다른 둥근 정도를 지정하여 지정된 좌표 안에 프리미티브 사각형을 추가합니다.

```
// 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);
```

참고 사항

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

document.addNewPublishProfile()

지원 버전

Flash MX 2004

사용법

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

매개 변수

profileName 새 프로파일의 고유한 이름. 이름을 지정하지 않으면 기본 이름이 사용됩니다. 이 매개 변수는 선택 사항입니다.

반환값

프로파일 목록의 새 프로파일 인덱스를 나타내는 정수. 새 프로파일을 만들 수 없으면 -1을 반환합니다.

설명

메시드, 새 제작 프로파일을 추가하고 이 프로파일을 현재 프로파일로 만듭니다.

예제

다음 예제에서는 기본 이름을 사용하는 새 제작 프로파일을 추가한 다음, [출력] 패널에 이 프로파일의 이름을 표시합니다.

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

다음 예제에서는 이름이 "my profile"인 새 제작 프로파일을 추가합니다.

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

참고 사항

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

document.addNewRectangle()

지원 버전

Flash MX 2004

사용법

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

매개 변수

boundingRectangle 새 사각형을 추가할 경계를 {left:value1,top:value2,right:value3,bottom:value4} 형식으로 지정하는 사각형. **left** 값과 **top** 값은 왼쪽 위 모서리 위치(예: left:0,top:0은 스테이지의 왼쪽 위를 나타냄)를 지정하고 **right** 값과 **bottom** 값은 오른쪽 아래 모서리 위치를 지정합니다. 따라서 사각형의 폭은 **left** 값과 **right** 값의 차이이며 사각형의 높이는 **top** 값과 **bottom** 값의 차이입니다.

즉, 사각형 경계 값이 속성 관리자에 표시된 값과 모두 일치하지는 않습니다. **left** 값과 **top** 값은 각각 속성 관리자의 X 값 및 Y 값과 일치합니다. 그러나 **right** 값과 **bottom** 값은 속성 관리자의 W 값 및 H 값과 일치하지 않습니다. 예를 들어 다음과 같은 경계 값을 가진 사각형을 살펴보겠습니다.

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

속성 관리자에 표시되는 이 사각형의 값은 다음과 같습니다.

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

roundness 모서리에 사용할 둥근 정도를 지정하는 0부터 999까지의 정수 값으로 점의 수로 지정됩니다. 값이 클수록 둥근 정도도 커집니다.

bSuppressFill true로 설정된 경우 메시드가 채우기가 적용되지 않은 모양을 만들도록 하는 부울 값. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

bSuppressStroke true로 설정된 경우 메시드가 획이 적용되지 않은 사각형을 만들도록 하는 부울 값. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메시드, 지정된 경계에 맞게 새 사각형 또는 둥근 사각형을 추가합니다. 이 메시드는 [사각형 도구]와 동일한 작업을 수행하며 문서의 현재 기본 획과 채우기 특성을 사용하여 현재 프레임과 레이어에 사각형을 추가합니다. **bSuppressFill**과 **bSuppressStroke**를 모두 true로 설정하면 아무런 영향을 주지 않습니다.

예제

다음 예제에서는 지정된 좌표 내에서 둥근 모서리가 적용되지 않은 새 사각형을 추가합니다. 이 사각형의 너비와 높이는 각각 100픽셀입니다.

```
f1.getDocumentDOM().addNewRectangle({left:0,top:0,right:100,bottom:100},0);
```

다음 예제에서는 둥근 모서리와 채우기가 적용되지 않은 새 사각형을 추가합니다. 이 사각형의 너비는 100픽셀이며 높이는 200 픽셀입니다.

```
f1.getDocumentDOM().addNewRectangle({left:10,top:10,right:110,bottom:210},0, true);
```

다음 예제에서는 둥근 모서리와 획이 적용되지 않은 새 사각형을 추가합니다. 이 사각형의 너비는 200픽셀이며 높이는 100픽셀입니다.

```
f1.getDocumentDOM().addNewRectangle({left:20,top:20,right:220,bottom:120},0, false, true);
```

참고 사항

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

document.addNewScene()

지원 버전

Flash MX 2004

사용법

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

매개 변수

name 장면의 이름을 지정합니다. 이름을 지정하지 않으면 새 장면 이름이 생성됩니다.

반환값

부울 값, 장면이 성공적으로 추가되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 새 장면([Timeline 객체](#))을 현재 선택한 장면의 다음 장면으로 추가하고 이 장면을 현재 선택한 장면으로 만듭니다. 지정된 장면 이름이 이미 있으면 장면이 추가되지 않고 오류가 반환됩니다.

예제

다음 예제에서는 현재 문서의 현재 장면 뒤에 myScene이라는 새 장면을 추가합니다. 새 장면이 만들어지면 success 변수는 true가 되고 그렇지 않으면 false가 됩니다.

```
var success = f1.getDocumentDOM().addNewScene("myScene");
```

다음 예제에서는 기본 명명 규칙을 사용하여 새 장면을 추가합니다. 장면이 하나만 있을 경우 새로 만들어진 장면의 이름은 "Scene 2"가 됩니다.

```
f1.getDocumentDOM().addNewScene();
```

document.addNewText()

지원 버전

Flash MX 2004, 선택적 **text** 매개 변수가 Flash CS3 Professional에 추가됨

사용법

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

매개 변수

boundingBox 텍스트 필드의 크기와 위치를 지정합니다. **boundingBox**의 형식에 대한 자세한 내용은 [document.addNewRectangle\(\)](#)을 참조하십시오.

text 필드에 배치할 텍스트를 지정하는 선택적 문자열입니다. 이 매개 변수를 생략하면 [도구] 패널의 선택 도구가 [텍스트 도구]로 전환됩니다. 따라서 선택한 도구를 변경하지 않으려면 **text**에 대한 값을 전달합니다.

반환값

없음

설명

메서드, 새 텍스트 필드를 삽입하고 선택적으로 해당 필드에 텍스트를 배치합니다. **text** 매개 변수를 생략한 경우 [document.setTextString\(\)](#)을 호출하여 텍스트 필드를 채웁니다.

예제

다음 예제에서는 스테이지의 왼쪽 위 모서리에 새 텍스트 필드를 만든 다음 텍스트 문자열을 "Hello World"로 설정합니다.

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

참고 사항

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

document.align()

지원 버전

Flash MX 2004

사용법

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

매개 변수

alignmode 선택 항목의 정렬 방법을 지정하는 문자열입니다. 사용할 수 있는 값에는 "left", "right", "top", "bottom", "vertical center" 및 "horizontal center"가 있습니다.

bUseDocumentBounds true로 설정된 경우 메서드가 문서의 경계를 기준으로 문자열을 정렬하도록 하는 부울 값. 그렇지 않으면 선택한 객체의 경계가 사용됩니다. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 선택 항목을 정렬합니다.

예제

다음 예제에서는 왼쪽과 스테이지를 기준으로 객체를 정렬합니다. 이것은 [정렬] 패널에서 [스테이지로] 설정을 선택하고 [왼쪽 정렬] 버튼을 클릭하는 것과 같습니다.

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

참고 사항

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

document.allowScreens()

지원 버전

Flash MX 2004

사용법

```
document.allowScreens()
```

매개 변수

없음

반환값

부울 값, `document.screenOutline`을 안전하게 사용할 수 있으면 `true`를 반환하고 그렇지 않으면 `false`를 반환합니다.

설명

메서드, `document.screenOutline` 속성을 사용하기 전에 사용합니다. 이 메서드가 `true` 값을 반환하면 `document.screenOutline`에 안전하게 액세스할 수 있으며 스크린이 없는 문서에서 `document.screenOutline`에 액세스하면 Flash에 오류가 표시됩니다.

예제

다음 예제에서는 현재 문서에서 `screens` 메서드를 사용할 수 있는지 여부를 나타냅니다.

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

참고 사항

[document.screenOutline](#)

document.arrange()

지원 버전

Flash MX 2004

사용법

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

매개 변수

arrangeMode 선택 항목을 이동할 방향을 지정합니다. 사용할 수 있는 값에는 "back", "backward", "forward" 및 "front"가 있습니다. [수정] > [정렬] 메뉴에서 이러한 옵션이 제공하는 동일한 기능을 제공합니다.

반환값

없음

설명

메서드, 스테이지에서 선택 항목을 정렬합니다. 이 메서드는 모양이 없는 객체에만 적용됩니다.

예제

다음 예제에서는 현재 선택 항목을 맨 앞으로 이동합니다.

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

document.as3AutoDeclare

지원 버전

Flash CS3 Professional

사용법

```
document.as3AutoDeclare
```

설명

속성, 스테이지에 배치된 인스턴스가 사용자 정의 타임라인 클래스에 자동으로 추가되는지 여부를 나타내는 부울 값입니다. 기본값은 true입니다.

예제

다음 예제에서는 현재 문서의 스테이지에 배치된 인스턴스가 사용자 정의 타임라인 클래스에 수동으로 추가되어야 하도록 지정합니다.

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

document.as3Dialect

지원 버전

Flash CS3 Professional

사용법

```
document.as3Dialect
```

설명

속성, 지정된 문서에 사용된 ActionScript 3.0 언어를 설명하는 문자열입니다. 기본값은 "AS3"입니다. 이전 ECMAScript 사양에서 허용된 것처럼 프로토타입 클래스를 허용하려면 이 값을 "ES"로 설정합니다.

예제

다음 예제에서는 현재 문서에 사용되고 있는 언어가 ECMAScript임을 지정합니다.

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

참고 사항

[document.asVersion](#)

document.as3ExportFrame

지원 버전

Flash CS3 Professional

사용법

```
document.as3ExportFrame
```

설명

속성, ActionScript 3.0 클래스를 내보내는 프레임을 지정하는 정수입니다. 기본적으로 클래스는 프레임 1로 내보내집니다.

예제

다음 예제에서는 클래스를 내보내는 프레임을 1(기본값)에서 5로 변경합니다.

```
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

지원 버전

Flash CS3 Professional

사용법

```
document.as3StrictMode
```

설명

속성, ActionScript 3.0에서 컴파일할 때 엄격 모드 옵션이 켜져 있어야 하는지(true), 아니면 꺼져 있어야 하는지(false)를 지정하는 부울 값입니다. 엄격 모드에서는 경고가 오류로 보고되며, 이러한 오류가 있을 경우 컴파일이 실패합니다. 기본값은 true입니다.

예제

다음 예제에서는 엄격 모드 컴파일러 옵션을 끕니다.

```
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);
```

참고 사항

[document.as3WarningsMode](#)

document.as3WarningsMode

지원 버전

Flash CS3 Professional

사용법

`document.as3WarningsMode`

설명

속성, ActionScript 3.0에서 컴파일할 때 경고 모드 옵션이 켜져 있어야 하는지(true), 아니면 꺼져 있어야 하는지(false)를 지정하는 부울 값입니다. 경고 모드에서는 ActionScript 2.0 코드를 ActionScript 3.0으로 업데이트할 때 비호환성을 발견하는 데 유용한 추가 경고가 보고됩니다. 기본값은 true입니다.

예제

다음 예제에서는 경고 모드 컴파일러 옵션을 끕니다.

```
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);
```

참고 사항

[document.as3StrictMode](#)

document.asVersion

지원 버전

Flash CS3 Professional

사용법

`document.asVersion`

설명

속성, 지정된 문서에 사용되고 있는 ActionScript의 버전을 지정하는 정수입니다. 사용할 수 있는 값에는 1, 2 및 3이 있습니다.

특정 문서에 대한 대상 플레이어 버전을 확인하려면 [document.getPlayerVersion\(\)](#)을 사용하십시오. 이 메서드는 문자열을 반환하므로 Flash® Lite™ 플레이어에서 사용할 수 있습니다.

예제

다음 예제에서는 현재 문서의 ActionScript 버전이 ActionScript 1.0으로 설정된 경우 ActionScript 2.0으로 설정합니다.

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


참고 사항

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

document.autoLabel

지원 버전

Flash MX 2004

사용법

```
document.autoLabel
```

설명

속성, [액세스 가능성] 패널의 자동 레이블 체크 상자에 해당하는 부울 값입니다. 이 속성을 사용하면 Flash에서 자동으로 스테이지의 객체에 해당 객체와 연관된 텍스트로 레이블을 지정하도록 할 수 있습니다.

예제

다음 예제에서는 autoLabel 속성의 값을 가져오고 [출력] 패널에 결과를 표시합니다.

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

다음 예제에서는 autoLabel 속성을 true로 설정하여 Flash에서 스테이지의 객체에 자동으로 레이블을 지정하도록 합니다.

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

document.backgroundColor

지원 버전

Flash MX 2004

사용법

```
document.backgroundColor
```

설명

속성, 형식이 다음 중 하나인 배경색입니다.

- "#RRGGBB" 또는 "#RRGGBBAA" 형식의 문자열
- 0xRRGGBB 형식의 16진수
- 16진수 숫자와 같은 십진수를 나타내는 정수

예제

다음 예제에서는 배경색을 검정색으로 설정합니다.

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

document.breakApart()

지원 버전

Flash MX 2004

사용법

```
document.breakApart()
```

매개 변수

없음

반환값

없음

설명

메서드, 현재 선택 영역에서 분리 작업을 수행합니다.

예제

다음 예제에서는 현재 선택 영역을 분리합니다.

```
f1.getDocumentDOM().breakApart();
```

document.canEditSymbol()

지원 버전

Flash MX 2004

사용법

```
document.canEditSymbol()
```

매개 변수

없음

반환값

부울 값, [심볼 편집] 메뉴와 기능을 사용할 수 있으면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, [심볼 편집] 메뉴와 기능이 활성화되어 있는지 여부를 나타냅니다. 이 메서드는 선택 항목의 편집 가능 여부와는 상관이 없으므로 f1.getDocumentDOM().enterEditMode()가 허용되는지 여부를 테스트하는 데 사용하면 안 됩니다.

예제

다음 예제에서는 [심볼 편집] 메뉴와 기능의 상태를 [출력] 패널에 표시합니다.

```
f1.trace("f1.getDocumentDOM().canEditSymbol() returns: " + f1.getDocumentDOM().canEditSymbol());
```

document.canRevert()

지원 버전

Flash MX 2004

사용법

```
document.canRevert()
```

매개 변수

없음

반환값

부울 값, `document.revert()` 또는 `fl.revertDocument()` 메서드를 성공적으로 사용할 수 있으면 `true`를 반환하고 그렇지 않으면 `false`를 반환합니다.

설명

메서드, `document.revert()` 또는 `fl.revertDocument()` 메서드를 성공적으로 사용할 수 있는지 여부를 나타냅니다.

예제

다음 예제에서는 현재 문서를 이전에 저장한 버전으로 되돌릴 수 있는지 여부를 확인합니다. 이전에 저장한 버전으로 되돌릴 수 있으면 `fl.getDocumentDOM().revert()`가 이전에 저장한 버전을 복원합니다.

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

document.canTestMovie()

지원 버전

Flash MX 2004

사용법

```
document.canTestMovie()
```

매개 변수

없음

반환값

부울 값, `document.testMovie()` 메서드를 성공적으로 사용할 수 있으면 `true`를 반환하고 그렇지 않으면 `false`를 반환합니다.

설명

메서드, `document.testMovie()` 메서드를 성공적으로 사용할 수 있는지 여부를 나타냅니다.

예제

다음 예제에서는 `fl.getDocumentDOM().testMovie()`를 사용할 수 있는지 여부를 테스트합니다. 사용할 수 있으면 이 메서드가 호출됩니다.

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

참고 사항

[document.canTestScene\(\)](#), [document.testScene\(\)](#)

document.canTestScene()

지원 버전

Flash MX 2004

사용법

```
document.canTestScene()
```

매개 변수

없음

반환값

부울 값, [document.testScene\(\)](#) 메서드를 성공적으로 사용할 수 있으면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

설명

메서드, [document.testScene\(\)](#) 메서드를 성공적으로 사용할 수 있는지 여부를 나타냅니다.

예제

다음 예제에서는 먼저 [fl.getDocumentDOM\(\).testScene\(\)](#)을 성공적으로 사용할 수 있는지 여부를 테스트합니다. 사용할 수 있으면 이 메서드가 호출됩니다.

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

참고 사항

[document.canTestMovie\(\)](#), [document.testMovie\(\)](#)

document.changeFilterOrder()

지원 버전

Flash 8

사용법

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

매개 변수

oldIndex [필터] 목록에서 위치를 변경할 필터의 0부터 시작하는 현재 인덱스 위치를 나타냅니다.

newIndex 목록에 있는 필터의 새 인덱스 위치를 나타내는 정수

반환값

없음

설명

메서드, [필터] 목록에서 필터의 인덱스를 변경합니다. **newIndex** 위나 아래에 있는 모든 필터가 변경된 인덱스에 맞게 위나 아래로 이동됩니다. 예를 들어 아래에 표시된 필터를 사용하여 `fl.getDocumentDOM().changeFilterOrder(3,0)` 명령을 실행하면 다음과 같이 필터가 다시 정렬됩니다.

실행 전	실행 후
<code>blurFilterdropShadowFilterglowFiltergradientBevelFilter</code>	<code>gradientBevelFilterblurFilterdropShadowFilterglowFilter</code>

그런 다음 `fl.getDocumentDOM().changeFilterOrder(0,2)` 명령을 실행하면 필터가 다음과 같이 다시 정렬됩니다.

실행 전	실행 후
<code>gradientBevelFilterblurFilterdropShadowFilterglowFilter</code>	<code>blurFilterdropShadowFiltergradientBevelFilterglowFilter</code>

예제

다음 예제에서는 현재 [필터] 목록에서 두 번째 위치에 있는 필터를 첫 번째 위치로 이동합니다.

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

참고 사항

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

document.clipCopy()

지원 버전

Flash MX 2004

사용법

```
document.clipCopy()
```

매개 변수

없음

반환값

없음

설명

메서드, 문서의 현재 선택 영역을 클립보드에 복사합니다.

문자열을 클립보드에 복사하려면 [fl.clipCopyString\(\)](#)을 사용합니다.

예제

다음 예제에서는 문서의 현재 선택 영역을 클립보드에 복사합니다.

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

참고 사항

[document.clipCut\(\)](#), [document.clipPaste\(\)](#)

document.clipCut()

지원 버전

Flash MX 2004

사용법

```
document.clipCut()
```

매개 변수

없음

반환값

없음

설명

메서드, 문서의 현재 선택 영역을 잘라내어 클립보드에 복사합니다.

예제

다음 예제에서는 문서의 현재 선택 영역을 잘라내어 클립보드에 복사합니다.

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

참고 사항

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

document.clipPaste()

지원 버전

Flash MX 2004

사용법

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

매개 변수

bInPlace true로 설정된 경우 메서드가 제자리에 붙여넣기 작업을 수행하도록 하는 부울 값. 기본값은 메서드가 문서의 가운데에 붙여넣기 작업을 수행하도록 하는 false입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 클립보드 내용을 문서에 붙여 넣습니다.

예제

다음 예제에서는 클립보드 내용을 문서의 가운데에 붙여 넣습니다.

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

다음 예제에서는 클립보드 내용을 현재 문서의 제자리에 붙여 넣습니다.

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

참고 사항

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

document.close()

지원 버전

Flash MX 2004

사용법

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

매개 변수

bPromptToSaveChanges true로 설정된 경우 메서드가 변경된 문서를 저장하라는 대화 상자를 표시하도록 하는 부울 값.

bPromptToSaveChanges가 false로 설정된 경우에는 변경된 문서를 저장하라는 대화 상자가 표시되지 않습니다. 기본값은 true입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 지정된 문서를 닫습니다.

예제

다음 예제에서는 현재 문서를 닫고 변경 내용을 저장하라는 대화 상자를 표시합니다.

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

다음 예제에서는 변경 내용을 저장하지 않고 현재 문서를 닫습니다.

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

document.convertLinesToFills()

지원 버전

Flash MX 2004

사용법

```
document.convertLinesToFills()
```

매개 변수

없음

반환값

없음

설명

메서드, 선택한 객체에서 선을 채우기로 변환합니다.

예제

다음 예제에서는 현재 선택한 선을 채우기로 변환합니다.

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

document.convertToSymbol()

지원 버전

Flash MX 2004

사용법

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

매개 변수

type 만들 심볼의 유형을 지정하는 문자열입니다. 사용할 수 있는 값에는 "movie clip", "button" 및 "graphic"이 있습니다.

name 새 심볼의 고유한 이름을 지정하는 문자열입니다. 빈 문자열을 전달하면 고유한 심볼 이름이 자동으로 만들어집니다.

registrationPoint 심볼의 0,0 위치를 나타내는 점을 지정합니다. 사용할 수 있는 값에는 "top left", "top center", "top right", "center left", "center", "center right", "bottom left", "bottom center" 및 "bottom right"가 있습니다.

반환값

새로 만든 심볼에 대한 객체를 반환하거나 심볼을 만들 수 없으면 null을 반환합니다.

설명

메서드, 선택한 스테이지 항목을 새 심볼로 변환합니다. 심볼의 링크 및 공유 예셋 속성을 정의하는 방법에 대한 자세한 내용은 [Item 객체](#)를 참조하십시오.

예제

다음 예제에서는 지정된 이름을 사용하는 무비 클립 심볼, 지정된 이름을 사용하는 버튼 심볼 및 기본 이름을 사용하는 무비 클립 심볼을 만듭니다.

```
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()

지원 버전

Flash 8

사용법

```
document.crop()
```

매개 변수

없음

반환값

부울 값, 성공적으로 실행되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 맨 위에 선택된 Drawing 객체를 사용하여 그 아래에 선택되어 있는 모든 Drawing 객체를 자릅니다. 선택한 Drawing 객체가 없거나 선택한 항목 중에 Drawing 객체가 아닌 항목이 있으면 false가 반환됩니다.

예제

다음 예제에서는 현재 선택한 객체를 자릅니다.

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

참고 사항

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

document.currentPublishProfile

지원 버전

Flash MX 2004

사용법

```
document.currentPublishProfile
```

설명

속성, 지정된 문서에 대한 활성 제작 프로파일의 이름을 지정하는 문자열입니다.

예제

다음 예제에서는 기본 이름을 사용하는 새 제작 프로파일을 추가한 다음 [출력] 패널에 이 프로파일의 이름을 표시합니다.

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

다음 예제에서는 선택한 제작 프로파일을 "Default"로 변경합니다.

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

document.currentTimeline

지원 버전

Flash MX 2004

사용법

```
document.currentTimeline
```

설명

속성, 활성 타임라인의 인덱스를 지정하는 정수입니다. 이 속성의 값을 변경하여 활성 타임라인을 설정할 수 있습니다. 그 효과는 [document.editScene\(\)](#)을 호출하는 것과 거의 같습니다. 유일한 차이점은 타임라인의 인덱스가 유효하지 않은 경우 오류 메시지가 표시되지 않는다는 것입니다. 즉, 이 속성이 설정되지 않으므로 오류가 발생해도 오류 메시지가 표시되지 않습니다.

예제

다음 예제에서는 현재 타임라인의 인덱스를 표시합니다.

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

다음 예제에서는 활성 타임라인을 기본 타임라인에서 "myScene"이라는 장면으로 변경합니다.

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

참고 사항

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

document.debugMovie()

지원 버전

Flash Professional CS5

사용법

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

설명

메서드, 문서에서 [동영상 디버그] 명령을 호출합니다.

매개 변수

abortIfErrorsExist 부울, 기본값은 false. true로 설정되면 컴파일러 오류가 있는 경우 디버그 세션이 시작되지 않고 .swf 원도우가 열리지 않습니다. 컴파일러 경고로 인해 명령이 중단되지 않습니다.

예제

다음 예제에서는 현재 문서를 디버그 모드로 열지만 컴파일러 오류가 있는 경우 작업을 중단합니다.

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

document.deleteEnvelope()

지원 버전
Flash 8

사용법

```
document.deleteEnvelope()
```

매개 변수
없음

반환값

부울 값, 성공적으로 실행되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 선택한 객체에서 엔벨로프(하나 이상의 객체를 포함하는 경계 상자)를 삭제합니다.

예제

다음 예제에서는 선택한 객체에서 엔벨로프를 삭제합니다.

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

참고 사항

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

document.deletePublishProfile()

지원 버전
Flash MX 2004

사용법

```
document.deletePublishProfile()
```

매개 변수
없음

반환값

새로운 현재 프로파일의 인덱스를 나타내는 정수. 새 프로파일을 사용할 수 없는 경우 현재 프로파일은 그대로 두고 해당 인덱스를 반환합니다.

설명

메서드, 프로파일이 둘 이상 있는 경우 현재 활성화된 프로파일을 삭제합니다. 프로파일은 최소한 하나 이상 있어야 합니다.

예제

다음 예제에서는 프로파일이 둘 이상 있는 경우 현재 활성화된 프로파일을 삭제하고 현재 활성화되어 있는 새 프로파일의 인덱스를 표시합니다.

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

참고 사항

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

document.deleteScene()

지원 버전

Flash MX 2004

사용법

```
document.deleteScene()
```

매개 변수

없음

반환값

부울 값, 장면이 성공적으로 삭제되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 현재 장면([Timeline 객체](#))을 삭제하고, 삭제된 장면이 마지막 장면이 아닐 경우 그 다음 장면을 현재 [Timeline 객체](#)로 설정합니다. 삭제된 장면이 마지막 장면이면 첫 번째 객체가 현재 [Timeline 객체](#)로 설정됩니다. [Timeline 객체](#)(장면)가 하나만 있으면 false 값이 반환됩니다.

예제

다음 예제에서는 현재 문서에 세 개의 장면(Scene0, Scene1 및 Scene2)이 있다고 가정하고 Scene2를 현재 장면으로 만든 다음 삭제합니다.

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

document.deleteSelection()

지원 버전

Flash MX 2004

사용법

```
document.deleteSelection()
```

매개 변수

없음

반환값

없음

설명

메서드, 스테이지에서 현재 선택 항목을 삭제합니다. 선택 항목이 없으면 오류 메시지가 표시됩니다.

예제

다음 예제에서는 문서에서 현재 선택 영역을 삭제합니다.

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

document.description

지원 버전

Flash MX 2004

사용법

```
document.description
```

설명

속성, [액세스 가능성] 패널의 [설명] 필드에 해당하는 문자열입니다. 이 설명은 화면 판독기에서 읽습니다.

예제

다음 예제에서는 문서에 대한 설명을 설정합니다.

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

다음 예제에서는 문서에 대한 설명을 가져와서 [출력] 패널에 표시합니다.

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

document.disableAllFilters()

지원 버전

Flash 8

사용법

```
document.disableAllFilters()
```

매개 변수

없음

반환값

없음

설명

메서드, 선택한 객체에서 모든 필터를 비활성화합니다.

예제

다음 예제에서는 선택한 객체에서 모든 필터를 비활성화합니다.

```
f1.getDocumentDOM().disableAllFilters();
```

참고 사항

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

document.disableFilter()

지원 버전

Flash 8

사용법

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

매개 변수

filterIndex [필터] 목록에 있는 필터의 0부터 시작하는 인덱스를 나타내는 정수

반환값

없음

설명

메서드, [필터] 목록에서 지정된 필터를 비활성화합니다.

예제

다음 예제에서는 선택한 객체에서 [필터] 목록의 첫 번째 필터와 세 번째 필터(인덱스 값 0 및 2)를 비활성화합니다.

```
f1.getDocumentDOM().disableFilter(0);  
f1.getDocumentDOM().disableFilter(2);
```

참고 사항

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

document.disableOtherFilters()

지원 버전

Flash 8

사용법

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

매개 변수

enabledFilterIndex 다른 필터가 비활성화된 후 활성화된 상태로 유지되어야 하는 필터의 0부터 시작하는 인덱스를 나타내는 정수

반환값

없음

설명

메서드, [필터] 목록에서 지정된 위치의 필터를 제외한 모든 필터를 비활성화합니다.

예제

다음 예제에서는 목록에서 두 번째 필터(인덱스 값 1)를 제외한 모든 필터를 비활성화합니다.

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

참고 사항

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

document.distribute()

지원 버전

Flash MX 2004

사용법

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

매개 변수

distributemode 선택한 객체를 배포할 위치를 지정하는 문자열입니다. 사용할 수 있는 값에는 "left edge", "horizontal center", "right edge", "top edge", "vertical center" 및 "bottom edge"가 있습니다.

bUseDocumentBounds true로 설정된 경우 문서의 경계를 사용하여 선택한 객체를 배포하는 부울 값. 그렇지 않으면 선택한 객체의 경계가 사용됩니다. 기본값은 false입니다.

반환값

없음

설명

메서드, 선택 항목을 배포합니다.

예제

다음 예제에서는 위쪽 가장자리를 기준으로 선택한 객체를 배포합니다.

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

다음 예제에서는 위쪽 가장자리를 기준으로 선택한 객체를 배포하고 **bUseDocumentBounds** 매개 변수를 명시적으로 설정합니다.

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

다음 예제에서는 문서의 경계를 사용하여 위쪽 가장자리를 기준으로 선택한 객체를 배포합니다.

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

참고 사항

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

document.distributeToLayers()

지원 버전

Flash MX 2004

사용법

```
document.distributeToLayers()
```

매개 변수

없음

반환값

없음

설명

메서드, 현재 선택 항목에서 레이어에 배포 작업을 수행합니다. [레이어에 배포]를 선택하는 것과 같습니다. 선택 항목이 없으면 오류 메시지가 표시됩니다.

예제

다음 예제에서는 현재 선택 항목을 레이어에 배포합니다.

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

document.docClass

지원 버전

Flash CS3 Professional

사용법

```
document.docClass
```

설명

속성, 문서에 연결된 최상위 ActionScript 3.0 클래스를 지정하는 문자열입니다. 문서가 ActionScript 3.0을 사용할 수 있도록 구성되지 않은 경우 이 속성은 무시됩니다.

예제

다음 예제에서는 문서에 연결된 ActionScript 3.0 클래스를 com/mycompany/ManagerClass.as에 정의된 com.mycompany.ManagerClass로 지정합니다.

```
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);
```

참고 사항

[item.linkageBaseClass](#)

document.documentHasData()

지원 버전

Flash MX 2004

사용법

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

매개 변수

name 확인할 데이터의 이름을 지정하는 문자열

반환값

부울 값, 문서에 영구 데이터가 있으면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 문서에 지정된 이름을 가진 영구 데이터가 있는지 확인합니다.

예제

다음 예제에서는 문서에 이름이 "myData"인 영구 데이터가 있는지 확인합니다.

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

참고 사항

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

document.duplicatePublishProfile()

지원 버전

Flash MX 2004

사용법

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

매개 변수

profileName 복제된 프로파일의 고유한 이름을 지정하는 문자열입니다. 이름을 지정하지 않으면 기본 이름이 사용됩니다. 이 매개 변수는 선택 사항입니다.

반환값

프로파일 목록의 새 프로파일 인덱스를 나타내는 정수. 프로파일을 복제할 수 없으면 -1을 반환합니다.

설명

메서드, 현재 활성화된 프로파일을 복제하고 이 복제된 프로파일에 포커스를 둡니다.

예제

다음 예제에서는 현재 활성화된 프로파일을 복제하고 [출력] 패널에 새 프로파일의 인덱스를 표시합니다.

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

document.duplicateScene()

지원 버전
Flash MX 2004

사용법
`document.duplicateScene()`

매개 변수
없음

반환값
부울 값, 장면이 성공적으로 복제되면 `true`를 반환하고 그렇지 않으면 `false`를 반환합니다.

설명
메서드, 현재 선택한 장면의 복사본을 만들고 고유한 이름을 지정한 다음 이 장면을 현재 장면으로 만듭니다.

예제
다음 예제에서는 현재 문서의 두 번째 장면을 복제합니다.

```
f1.getDocumentDOM().editScene(1); //Set the middle scene to current scene.  
var success = f1.getDocumentDOM().duplicateScene();
```

document.duplicateSelection()

지원 버전
Flash MX 2004

사용법
`document.duplicateSelection()`

매개 변수
없음

반환값
없음

설명
메서드, 스테이지에서 선택 항목을 복제합니다.

예제
다음 예제에서는 현재 선택 항목을 복제합니다. `Alt` 키를 누른 채 항목을 드래그하는 것과 비슷합니다.

```
f1.getDocumentDOM().duplicateSelection();
```

document.editScene()

지원 버전
Flash MX 2004

사용법
`document.editScene(index)`

매개 변수
`index` 편집할 장면을 지정하는 0부터 시작하는 정수

반환값
없음

설명
메서드, 편집하기 위해 지정된 장면을 선택합니다.

예제
다음 예제에서는 현재 문서에 세 개의 장면(Scene0, Scene1 및 Scene2)이 있다고 가정하고 Scene2를 현재 장면으로 만든 다음 삭제합니다.

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

document.enableAllFilters()

지원 버전
Flash 8

사용법
`document.enableAllFilters()`

매개 변수
없음

반환값
없음

설명
메서드, 선택한 객체에 대한 [필터] 목록에서 모든 필터를 활성화합니다.

예제
다음 예제에서는 선택한 객체에 대한 [필터] 목록에서 모든 필터를 활성화합니다.

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

참고 사항

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

document.enableFilter()

지원 버전

Flash 8

사용법

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

매개 변수

`filterIndex` [필터] 목록에서 활성화할 필터의 0부터 시작하는 인덱스를 지정하는 정수

반환값

없음

설명

메서드, 선택한 객체에 대해 지정된 필터를 활성화합니다.

예제

다음 예제에서는 선택한 객체의 두 번째 필터를 활성화합니다.

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

참고 사항

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

document.enterEditMode()

지원 버전

Flash MX 2004

사용법

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

매개 변수

`editMode` 편집 모드를 지정하는 문자열. 사용할 수 있는 값에는 "inPlace" 또는 "newWindow"가 있습니다. 지정된 매개 변수가 없으면 기본적으로 심볼 편집 모드가 사용됩니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 제작 도구를 매개 변수에 의해 지정된 편집 모드로 전환합니다. 지정된 매개 변수가 없으면 기본적으로 심볼 편집 모드가 사용됩니다. 이 메서드는 심볼을 마우스 오른쪽 버튼으로 클릭하여 컨텍스트 메뉴를 호출한 다음 [편집]을 선택하는 것과 같습니다.

예제

다음 예제에서는 현재 선택한 심볼을 제자리에서 편집 모드로 설정합니다.

```
f1.getDocumentDOM().enterEditMode('inPlace');
```

다음 예제에서는 현재 선택한 심볼을 새 윈도우에서 편집 모드로 설정합니다.

```
f1.getDocumentDOM().enterEditMode('newWindow');
```

참고 사항

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

document.exitEditMode()

지원 버전

Flash MX 2004

사용법

```
document.exitEditMode()
```

매개 변수

없음

반환값

없음

설명

메서드, 심볼 편집 모드를 종료하고 편집 모드보다 한 레벨 위로 포커스를 되돌립니다. 예를 들어 다른 심볼 안에 있는 심볼을 편집하는 경우 편집하고 있는 심볼보다 한 레벨 위에 있는 부모 심볼로 이동합니다.

예제

다음 예제에서는 심볼 편집 모드를 종료합니다.

```
f1.getDocumentDOM().exitEditMode();
```

참고 사항

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

document.exportPNG()

지원 버전

Flash 8

사용법

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

매개 변수

fileURI 내보낸 파일의 이름을 지정하는 문자열(**file:///** URI로 표시됨). **fileURI**가 빈 문자열이거나 지정되지 않은 경우 [무비 내보내기] 대화 상자가 표시됩니다.

bCurrentPNGSettings 현재 PNG 제작 설정을 사용할지(**true**), 아니면 [PNG 내보내기] 대화 상자를 표시할지(**false**)를 지정하는 부울 값. 이 매개 변수는 선택 사항입니다. 기본값은 **false**입니다.

bCurrentFrame 현재 프레임만 내보낼지(**true**) 또는 각 프레임을 별도 PNG 파일로 하여 모든 프레임을 내보낼지(**false**) 여부를 지정하는 부울 값. 이 매개 변수는 선택 사항입니다. 기본값은 **false**입니다.

반환값

파일을 PNG 파일로 성공적으로 내보내면 **true**를 반환하고 그렇지 않으면 **false**를 반환하는 부울 값

설명

메서드, 문서를 하나 이상의 PNG 파일로 내보냅니다. **fileURI**가 지정되었는데 해당 파일이 이미 있으면 경고 메시지를 표시하지 않고 덮어씁니다.

예제

다음 예제에서는 현재 PNG 제작 설정을 사용하여 현재 문서의 현재 프레임을 **myFile.png**로 내보냅니다.

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

document.exportPublishProfile()

지원 버전

Flash MX 2004

사용법

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

매개 변수

fileURI 프로파일을 내보낼 XML 파일의 경로를 지정하는 문자열(**file:///** URI로 표시됨)

반환값

없음

설명

메서드, 현재 활성화된 프로파일을 XML 파일로 내보냅니다.

예제

다음 예제에서는 현재 활성화된 프로파일을 C 드라이브의 /Documents and Settings/username/바탕 화면 폴더에 있는 **profile.xml**이라는 파일로 내보냅니다.

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

참고 사항

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

document.exportPublishProfileString()

지원 버전

Flash CS4 Professional

사용법

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

매개 변수

profileName XML 문자열로 내보낼 프로파일의 이름을 지정하는 문자열. 이 매개 변수는 선택 사항입니다.

반환값

XML 문자열

설명

메서드, 지정된 프로파일을 XML 형식으로 지정하는 문자열을 반환합니다. **profileName**의 값을 전달하지 않으면 현재 프로파일을 내보냅니다.

예제

다음 예제에서는 profileXML이라는 변수에 현재 프로파일을 나타내는 XML 문자열을 저장한 다음 [출력] 패널에 표시합니다.

```
var profileXML=fl.getDocumentDOM().exportPublishProfileString();  
fl.trace(profileXML);
```

참고 사항

[document.exportPublishProfile\(\)](#), [document.importPublishProfileString\(\)](#)

document.exportSWF()

지원 버전

Flash MX 2004

사용법

```
document.exportSWF([fileURI [, bCurrentSettings]])
```

매개 변수

fileURI 내보낸 파일의 이름을 지정하는 문자열(**file:/// URI**로 표시됨). **fileURI**가 비어 있거나 지정되지 않은 경우 [무비 내보내기] 대화 상자가 표시됩니다. 이 매개 변수는 선택 사항입니다.

bCurrentSettings true로 설정된 경우 Flash에서 현재 SWF 제작 설정을 사용하도록 하는 부울 값. 그렇지 않으면 [Flash Player 내보내기] 대화 상자가 표시됩니다. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, Flash SWF 형식으로 문서를 내보냅니다.

예제

다음 예제에서는 현재 제작 설정을 사용하여 지정된 파일 위치로 문서를 내보냅니다.

```
fl.getDocumentDOM().exportSWF("file:///C:/Documents and Settings/joe_user/Desktop/qwerty.swf");
```

다음 예제에서는 [무비 내보내기] 대화 상자와 [Flash Player 내보내기] 대화 상자를 표시한 다음 지정된 설정을 기반으로 문서를 내보냅니다.

```
fl.getDocumentDOM().exportSWF("", true);
```

다음 예제에서는 [무비 내보내기] 대화 상자를 표시한 다음 지정된 설정을 기반으로 문서를 내보냅니다.

```
fl.getDocumentDOM().exportSWF();
```

document.externalLibraryPath

지원 버전

Flash CS4 Professional

사용법

```
document.externalLibraryPath
```

설명

속성, 런타임 공유 라이브러리로 사용되는 SWC 파일의 위치를 지정하는 문서의 ActionScript 3.0 외부 라이브러리 경로에 있는 항목의 목록이 포함된 문자열입니다. 문자열에 있는 항목은 세미콜론으로 구분됩니다. 제작 도구에서 [파일] > [제작 설정]을 선택한 다음 [Flash] 탭에서 [ActionScript 3.0 스크립트 설정]을 선택하여 항목을 지정할 수 있습니다.

예제

다음 예제에서는 문서의 외부 라이브러리 경로를 "." 및 "../mySWCLibrary"로 설정합니다.

```
var myDocument = fl.getDocumentDOM();  
myDocument.externalLibraryPath = ".;../mySWCLibrary";  
fl.trace(myDocument.externalLibraryPath);
```

참고 사항

[document.libraryPath](#), [document.sourcePath](#), [fl.externalLibraryPath](#)

document.forceSimple

지원 버전

Flash MX 2004

사용법

```
document.forceSimple
```


설명

속성, 지정된 객체의 자식에 액세스할 수 있는지 여부를 지정하는 부울 값입니다. [액세스 가능성] 패널에 있는 [자식 객체 액세스 가능] 설정의 역논리에 해당합니다. 즉, `forceSimple`이 `true`이면 [자식 객체 액세스 가능] 옵션이 선택되어 있지 않은 것과 같고 `forceSimple`이 `false`이면 [자식 객체 액세스 가능] 옵션이 선택되어 있는 것과 같습니다.

예제

다음 예제에서는 `areChildrenAccessible` 변수를 `forceSimple` 속성의 값으로 설정합니다. `false` 값은 자식에 액세스할 수 있음을 의미합니다.

```
var areChildrenAccessible = fl.getDocumentDOM().forceSimple;
```

다음 예제에서는 문서의 자식 객체에 액세스할 수 있도록 `forceSimple` 속성을 설정합니다.

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

document.frameRate

지원 버전

Flash MX 2004

사용법

```
document.frameRate
```

설명

속성, SWF 파일이 재생될 때 표시되는 초당 프레임 수를 지정하는 부동 소수점 값입니다. 기본값은 12입니다. 이 속성을 설정하는 것은 FLA 파일의 [문서 속성] 대화 상자([수정] > [문서])에서 기본 프레임 속도를 설정하는 것과 같습니다.

예제

다음 예제에서는 프레임 속도를 초당 25.5프레임으로 설정합니다.

```
fl.getDocumentDOM().frameRate = 25.5;
```

document.getAlignToDocument()

지원 버전

Flash MX 2004

사용법

```
document.getAlignToDocument()
```

매개 변수

없음

반환값

부울 값, 환경 설정이 스테이지를 기준으로 객체를 정렬하도록 설정되어 있으면 `true`를 반환하고 그렇지 않으면 `false`를 반환합니다.

설명

메서드, [정렬] 패널에서 [스테이지로] 버튼의 값을 가져오는 것과 같습니다. 문서의 [document.align\(\)](#), [document.distribute\(\)](#), [document.match\(\)](#) 및 [document.space\(\)](#) 메서드에 사용할 수 있는 환경 설정을 가져옵니다.

예제

다음 예제에서는 [정렬] 패널에서 [스테이지로] 버튼의 값을 가져옵니다. 반환값이 true이면 [스테이지로] 버튼이 활성화되고 그렇지 않으면 활성화되지 않습니다.

```
var isAlignToDoc = fl.getDocumentDOM().getAlignToDocument();  
fl.getDocumentDOM().align("left", isAlignToDoc);
```

참고 사항

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

document.getBlendMode()

지원 버전

Flash 8

사용법

```
document.getBlendMode()
```

매개 변수

없음

반환값

선택한 객체에 대해 혼합 모드를 지정하는 문자열. 객체를 두 개 이상 선택한 경우 이들 객체의 혼합 모드가 서로 다르면 문자열은 심도가 가장 높은 객체의 혼합 모드를 반영합니다.

참고: 반환값은 선택 항목에 혼합 모드를 지원하지 않거나 혼합 모드 값이 "normal"인 객체가 들어 있는 경우 예측할 수 없습니다.

설명

메서드, 선택한 객체에 대해 혼합 모드를 지정하는 문자열을 반환합니다.

예제

다음 예제에서는 [출력] 패널에 혼합 모드의 이름을 표시합니다.

```
fl.trace(fl.getDocumentDom().getBlendMode());
```

document.getCustomFill()

지원 버전

Flash MX 2004

사용법

```
document.getCustomFill([objectToFill])
```

매개 변수

`objectToFill` Fill 객체의 위치를 지정하는 문자열. 사용할 수 있는 값은 다음과 같습니다.

- "toolbar"는 [도구] 패널 및 속성 관리자의 Fill 객체를 반환합니다.
- "selection"은 선택 항목의 Fill 객체를 반환합니다.

이 매개 변수를 생략하면 기본적으로 "selection"이 사용됩니다. 선택 항목이 없으면 `undefined`가 반환됩니다. 이 매개 변수는 선택 사항입니다.

반환값

성공적으로 실행되면 `objectToFill` 매개 변수에 의해 지정된 Fill 객체를 반환하고 그렇지 않으면 `undefined`를 반환합니다.

설명

메서드, 선택한 모양의 Fill 객체를 가져오거나, 지정된 경우 [도구] 패널 및 속성 관리자의 Fill 객체를 가져옵니다.

예제

다음 예제에서는 선택 항목의 Fill 객체를 가져오고 선택 항목의 색상을 흰색으로 변경합니다.

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.color = '#FFFFFF';  
fill.style = "solid";  
fl.getDocumentDOM().setCustomFill(fill);
```

다음 예제에서는 [도구] 패널 및 속성 관리자의 Fill 객체를 반환한 다음 색상 견본을 선형 그라디언트로 변경합니다.

```
var fill = fl.getDocumentDOM().getCustomFill("toolbar");  
fill.style = "linearGradient";  
fill.colorArray = [ 0x00ff00, 0xff0000, 0x0000ff ];  
fill.posArray = [0, 100, 200];  
fl.getDocumentDOM().setCustomFill( fill );
```

참고 사항

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

document.getCustomStroke()

지원 버전

Flash MX 2004

사용법

```
document.getCustomStroke([locationOfStroke])
```

매개 변수

`locationOfStroke` Stroke 객체의 위치를 지정하는 문자열. 사용할 수 있는 값은 다음과 같습니다.

- 설정된 경우 "toolbar"는 [도구] 패널 및 속성 관리자의 Stroke 객체를 반환합니다.
- 설정된 경우 "selection"은 선택 항목의 Stroke 객체를 반환합니다.

이 매개 변수를 생략하면 기본적으로 "selection"이 사용됩니다. 선택 항목이 없으면 `undefined`가 반환됩니다. 이 매개 변수는 선택 사항입니다.

반환값

성공적으로 실행되면 **locationOfStroke** 매개 변수에 의해 지정된 **Stroke** 객체를 반환하고 그렇지 않으면 **undefined**를 반환합니다.

설명

선택한 모양의 **Stroke** 객체를 반환하거나, 지정된 경우 [도구] 패널 및 속성 관리자의 **Stroke** 객체를 반환합니다.

예제

다음 예제에서는 선택 항목의 현재 획 설정을 반환하고 획 두께를 2로 변경합니다.

```
var stroke = fl.getDocumentDOM().getCustomStroke("selection");
stroke.thickness = 2;
fl.getDocumentDOM().setCustomStroke(stroke);
```

다음 예제에서는 [도구] 패널 및 속성 관리자의 현재 획 설정을 반환하고 획 색상을 빨간색으로 변경합니다.

```
var stroke = fl.getDocumentDOM().getCustomStroke("toolbar");
stroke.color = "#FF0000";
fl.getDocumentDOM().setCustomStroke(stroke);
```

참고 사항

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

document.getDataFromDocument()

지원 버전

Flash MX 2004

사용법

```
document.getDataFromDocument (name)
```

매개 변수

name 반환할 데이터의 이름을 지정하는 문자열

반환값

지정된 데이터

설명

메서드, 지정된 데이터의 값을 가져옵니다. 반환되는 유형은 저장된 데이터의 유형에 따라 달라집니다.

예제

다음 예제에서는 현재 문서에 정수 값 12를 추가하고 이 메서드를 사용하여 [출력] 패널에 이 값을 표시합니다.

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

참고 사항

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

document.getElementById()

지원 버전
Flash MX 2004

사용법

```
document.getElementById(propertyName)
```

매개 변수

propertyName 값을 검색해야 하는 Element 속성의 이름을 지정하는 문자열

반환값

지정된 속성의 값. 서로 다른 속성 값을 가진 여러 요소가 선택된 경우처럼 속성이 명확하지 않은 상태이면 null을 반환하고 속성이 선택한 요소의 유효한 속성이 아니면 undefined를 반환합니다.

설명

메서드, 현재 선택 항목의 지정된 Element 속성을 가져옵니다. 사용할 수 있는 값의 목록은 [Element 객체](#)의 속성 요약 표를 참조하십시오.

예제

다음 예제에서는 현재 선택 항목에 대한 Element 속성의 name을 가져옵니다.

```
// elementName = the instance name of the selected object.  
var elementName = fl.getDocumentDOM().getElementProperty("name");
```

참고 사항

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

document.getElementByIdTextAttr()

지원 버전
Flash MX 2004

사용법

```
document.getElementByIdTextAttr(attrName [, startIndex [, endIndex]])
```

매개 변수

attrName 반환할 TextAttrs 속성의 이름을 지정하는 문자열. 속성 이름 및 예상되는 값의 목록은 [TextAttrs 객체](#)의 속성 요약 표를 참조하십시오.

startIndex 첫 번째 문자의 0부터 시작하는 인덱스를 지정하는 정수. 이 매개 변수는 선택 사항입니다.

endIndex 마지막 문자의 인덱스를 지정하는 정수. 이 매개 변수는 선택 사항입니다.

반환값

하나의 텍스트 필드가 선택되어 있는 경우, 텍스트 필드 내에 사용되는 값이 하나만 있으면 그 속성을 반환하고, 텍스트 필드 내에 사용되는 값이 여러 개 있으면 undefined를 반환합니다. 여러 개의 텍스트 필드가 선택되어 있는 경우, 모든 텍스트 정렬 값이 동일하면 그 값이 반환되고 모든 텍스트 정렬 값이 동일하지 않으면 undefined가 반환됩니다. 이러한 규칙은 선택적 인수가 전달되지 않은 경우, 현재 선택한 텍스트 범위에 적용되거나 텍스트가 현재 편집 중이지 않은 경우에는 전체 텍스트 필드에 적용됩니다.

다. **startIndex**만 전달된 경우, 선택한 모든 Text 객체의 값이 일치하면 인덱스 오른쪽에 있는 문자의 속성이 반환되고 **startIndex** 및 **endIndex**가 전달된 경우에는 **startIndex**부터 **endIndex**(포함 안 됨) 사이에 있는 문자의 전체 범위를 나타내는 값이 반환됩니다.

설명

메서드, 선택한 Text 객체의 특정 TextAttrs 속성을 가져옵니다. 선택한 객체 중 텍스트 필드가 아닌 객체는 무시됩니다. 속성 이름 및 예상되는 값의 목록은 [TextAttrs 객체](#)의 속성 요약 표를 참조하십시오. [document.setElementTextAttr\(\)](#)도 참조하십시오.

예제

다음 예제에서는 선택한 텍스트 필드의 크기를 가져옵니다.

```
f1.getDocumentDOM().getElementTextAttr("size");
```

다음 예제에서는 선택한 텍스트 필드의 인덱스 3에 있는 문자의 색상을 가져옵니다.

```
f1.getDocumentDOM().getElementTextAttr("fillColor", 3);
```

다음 예제에서는 선택한 텍스트 필드의 인덱스 2에서 인덱스 10(포함 안 됨) 사이에 있는 텍스트의 글꼴 이름을 가져옵니다.

```
f1.getDocumentDOM().getElementTextAttr("face", 2, 10);
```

document.getFilters()

지원 버전

Flash 8

사용법

```
document.getFilters()
```

매개 변수

없음

반환값

현재 선택되어 있는 객체에 적용된 필터의 목록이 포함된 배열

설명

메서드, 현재 선택되어 있는 객체에 적용된 필터의 목록이 포함된 배열을 반환합니다. 객체를 여러 개 선택한 경우 해당 객체의 필터가 서로 다르면 이 메서드는 선택된 첫 번째 객체에 적용된 필터 목록을 반환합니다.

예제

[document.setFilters\(\)](#)를 참조하십시오.

참고 사항

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.setFilters\(\)](#), [Filter 객체](#)

document.getMetadata()

지원 버전

Flash 8

사용법

```
document.getMetadata()
```

매개 변수

없음

반환값

문서와 연관된 XML 메타데이터가 포함된 문자열을 반환하거나, 메타데이터가 없으면 빈 문자열을 반환합니다.

설명

메서드, 문서와 연관된 XML 메타데이터가 포함된 문자열을 반환하거나 메타데이터가 없으면 빈 문자열을 반환합니다.

예제

다음 예제에서는 현재 문서의 XML 메타데이터를 [출력] 패널에 표시합니다.

```
fl.trace("XML Metadata is :" + fl.getDocumentDOM().getMetadata());
```

참고 사항

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

document.getMobileSettings()

지원 버전

Flash CS3 Professional

사용법

```
document.getMobileSettings()
```

매개 변수

없음

반환값

문서에 대한 XML 설정을 나타내는 문자열. 값을 설정하지 않은 경우 빈 문자열을 반환합니다.

설명

메서드, 문서에 대한 모바일 XML 설정을 반환합니다.

예제

다음 예제에서는 현재 문서에 대한 XML 설정 문자열을 표시합니다.

```
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>"
```

참고 사항

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

document.getPlayerVersion()

지원 버전

Flash CS3 Professional

사용법

```
document.getPlayerVersion()
```

매개 변수

없음

반환값

`document.setPlayerVersion()`을 사용하여 지정된 버전을 나타내는 문자열 값을 설정하지 않은 경우 [제작 설정] 대화 상자에 지정된 값을 반환합니다.

설명

메서드, 지정된 문서의 대상 플레이어 버전을 나타내는 문자열을 반환합니다. 이 메서드가 반환할 수 있는 값 목록은 [document.setPlayerVersion\(\)](#)을 참조하십시오.

지정된 파일의 대상 `ActionScript` 버전을 확인하려면 [document.asVersion](#)을 사용합니다.

예제

다음 예제에서는 지정된 플레이어 버전을 현재 문서에 대한 대상으로 지정한 다음 해당 값을 가져오는 방법을 보여 줍니다.

```
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"
```

참고 사항

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

document.getSelectionRect()

지원 버전

Flash MX 2004

사용법

`document.getSelectionRect()`

매개 변수

없음

반환값

현재 선택 항목의 경계 사각형을 반환하거나, 선택한 항목이 없으면 0을 반환합니다. 반환값의 형식에 대한 자세한 내용은 [document.addNewRectangle\(\)](#)을 참조하십시오.

설명

메서드, 현재 선택 항목의 경계 사각형을 가져옵니다. 선택 항목이 사각형이 아닐 경우 전체 선택 항목을 포함하는 가장 작은 사각형이 반환됩니다. 이 사각형은 문서 공간을 기반으로 하거나, 편집 모드에 있을 경우 편집 중인 심볼의 등록 포인트(원점)를 기반으로 합니다.

예제

다음 예제에서는 현재 선택 항목의 경계 사각형을 가져온 다음 해당 속성을 표시합니다.

```
var newRect = fl.getDocumentDOM().getSelectionRect();
var outputStr = "left: " + newRect.left + " top: " + newRect.top + " right: " + newRect.right + " bottom: " + newRect.bottom;
alert(outputStr);
```

참고 사항

[document.selection](#), [document.setSelectionRect\(\)](#)

document.getTextString()

지원 버전

Flash MX 2004

사용법

`document.getTextString([startIndex [, endIndex]])`

매개 변수

startIndex 가져올 첫 번째 문자의 인덱스를 나타내는 정수. 이 매개 변수는 선택 사항입니다.

endIndex 가져올 마지막 문자의 인덱스를 나타내는 정수. 이 매개 변수는 선택 사항입니다.

반환값

선택한 텍스트가 포함된 문자열

설명

메서드, 현재 선택한 텍스트를 가져옵니다. 선택적 매개 변수가 전달되지 않은 경우 현재 텍스트 선택 영역이 사용됩니다. 텍스트가 현재 편집하기 위해 열려 있지 않은 경우 전체 텍스트 문자열이 반환됩니다. **startIndex**만 전달된 경우 해당 인덱스에서 시작해서 필드의 끝에서 끝나는 문자열이 반환됩니다. **startIndex** 및 **endIndex**가 전달된 경우 **startIndex**부터 **endIndex**(포함 안 됨)까지의 문자열이 반환됩니다.

여러 개의 텍스트 필드가 선택되어 있으면 연결된 모든 문자열이 반환됩니다.

예제

다음 예제에서는 선택한 텍스트 필드에서 문자열을 가져옵니다.

```
f1.getDocumentDOM().getTextString();
```

다음 예제에서는 선택한 텍스트 필드의 문자 인덱스 5에서 문자열을 가져옵니다.

```
f1.getDocumentDOM().getTextString(5);
```

다음 예제에서는 문자 인덱스 2에서 문자 인덱스 10(포함 안 됨) 사이의 문자열을 가져옵니다.

```
f1.getDocumentDOM().getTextString(2, 10);
```

참고 사항

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

document.getTimeline()

지원 버전

Flash MX 2004

사용법

```
document.getTimeline()
```

매개 변수

없음

반환값

현재 Timeline 객체

설명

메서드, 문서에서 현재 [Timeline 객체](#)를 가져옵니다. 현재 Timeline 객체는 현재 장면, 편집 중인 현재 심볼 또는 현재 스크린 일 수 있습니다.

예제

다음 예제에서는 Timeline 객체를 가져오고 가장 긴 레이어에 있는 프레임 수를 반환합니다.

```
var longestLayer = f1.getDocumentDOM().getTimeline().frameCount;  
f1.trace("The longest layer has" + longestLayer + "frames");
```

다음 예제에서는 스테이지에 선택되어 있는 심볼을 제자리에서 편집 모드로 설정하고 이 심볼의 Timeline 객체에 프레임을 삽입합니다.

```
f1.getDocumentDOM().enterEditMode("inPlace");  
f1.getDocumentDOM().getTimeline().insertFrames();
```

다음 예제에서는 Timeline 객체를 가져오고 해당 이름을 표시합니다.

```
var timeline = f1.getDocumentDOM().getTimeline();  
alert(timeline.name);
```

참고 사항

[document.currentTimeline](#), [document.timelines](#), [symbolItem.timeline](#)

document.getTransformationPoint()

지원 버전

Flash MX 2004

사용법

```
document.getTransformationPoint()
```

매개 변수

없음

반환값

선택한 요소의 좌표계 내에서 변형점(또는 원점)의 위치를 지정하는 점(예: {x:10, y:20}, 여기서 x 및 y는 부동 소수점 숫자임)

설명

메서드, 현재 선택 항목의 변형점 위치를 가져옵니다. 변형점은 회전 및 기울기와 같은 변형 작업에 사용할 수 있습니다.

참고: 변형점은 선택한 항목의 유형에 따라 다양한 위치를 기준으로 합니다. 자세한 내용은 [document.setTransformationPoint\(\)](#)를 참조하십시오.

예제

다음 예제에서는 현재 선택 항목의 변형점을 가져옵니다. `transPoint.x` 속성은 변형점의 **x** 좌표를 지정하고 `transPoint.y` 속성은 변형점의 **y** 좌표를 지정합니다.

```
var transPoint = fl.getDocumentDOM().getTransformationPoint();
```

참고 사항

[document.setTransformationPoint\(\)](#), [element.getTransformationPoint\(\)](#)

document.group()

지원 버전

Flash MX 2004

사용법

```
document.group()
```

매개 변수

없음

반환값

없음

설명

메서드, 현재 선택 항목을 그룹으로 변환합니다.

예제

다음 예제에서는 현재 선택 항목의 객체를 그룹으로 변환합니다.

```
fl.getDocumentDOM().group();
```

참고 사항

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

document.height

지원 버전

Flash MX 2004

사용법

```
document.height
```

설명

속성, 문서(스태이지)의 높이를 픽셀 단위로 지정하는 정수입니다.

예제

다음 예제에서는 스태이지의 높이를 400픽셀로 설정합니다.

```
fl.getDocumentDOM().height = 400;
```

참고 사항

[document.width](#)

document.id

지원 버전

Flash CS3 Professional

사용법

```
document.id
```

설명

읽기 전용 속성, Flash 세션 동안 문서를 식별하는 고유한 정수(자동으로 할당됨)입니다. 이 속성을 [fl.findDocumentDOM\(\)](#)과 함께 사용하여 액션에 대한 특정 문서를 지정합니다.

예제

다음 예제에서는 현재 문서의 문서 ID를 표시합니다.

```
fl.trace("Current doc's internal ID is: " + fl.getDocumentDOM().id);
```

참고 사항

[fl.findDocumentDOM\(\)](#)

document.importFile()

지원 버전

Flash 8

사용법

```
document.importFile(fileURI [, importToLibrary])
```

매개 변수

fileURI 가져올 파일의 경로를 지정하는 문자열(file:/// URI로 표시됨)

importToLibrary 파일을 문서의 라이브러리만 가져올지(true), 아니면 스테이지에도 복사본을 배치할지(false)를 지정하는 부울 값. 기본값은 false입니다.

반환값

없음

설명

메서드, 파일을 문서로 가져옵니다. 이 메서드는 [라이브러리로 가져오기] 또는 [스테이지로 가져오기] 메뉴 명령과 동일한 작업을 수행합니다. 제작 프로파일을 가져오려면 [document.importPublishProfile\(\)](#)을 사용합니다.

예제

다음 예제에서는 사용자가 스테이지로 가져올 파일을 탐색할 수 있습니다.

```
var dom = fl.getDocumentDOM();  
var URI = fl.browseForFileURL("select", "Import File");  
dom.importFile(URI);
```

참고 사항

[document.importSWF\(\)](#), [fl.browseForFileURL\(\)](#)

document.importPublishProfile()

지원 버전

Flash MX 2004

사용법

```
document.importPublishProfile( fileURI )
```

매개 변수

fileURI 가져올 프로파일을 정의하는 XML 파일의 경로를 지정하는 문자열(file:/// URI로 표시됨)

반환값

프로파일 목록에서 가져온 프로파일의 인덱스를 나타내는 정수. 프로파일을 가져올 수 없는 경우 -1을 반환합니다.

설명

메서드, 파일에서 프로파일을 가져옵니다.

예제

다음 예제에서는 profile.xml 파일에 들어 있는 프로파일을 가져오고 프로파일 목록에 해당 인덱스를 표시합니다.

```
alert(fl.getDocumentDOM().importPublishProfile('file:///C:/Documents and Settings/janeUser/Desktop/profile.xml'));
```

document.importPublishProfileString()

지원 버전

Flash CS4 Professional

사용법

```
document.importPublishProfileString(xmlString)
```

매개 변수

xmlString 현재 프로파일로 가져올 XML 데이터가 포함된 문자열

반환값

문자열을 성공적으로 가져오면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값

설명

메서드, 제작 프로파일을 나타내는 XML 문자열을 가져오고 현재 프로파일로 설정합니다. 가져올 XML 문자열을 생성하려면 이 메서드를 사용하기 전에 [document.exportPublishProfileString\(\)](#)을 사용합니다.

예제

다음 예제에서는 기본 프로파일을 XML 문자열로 내보냅니다. 표준 JavaScript replace 명령이 XML 문자열을 수정하는 데 사용됩니다. 문자열을 가져오고 기본 ActionScript 3 출력 설정이 ActionScript 1로 설정됩니다.

```
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()

지원 버전

Flash MX 2004

사용법

```
document.importSWF(fileURI)
```

매개 변수

fileURI 가져올 SWF 파일을 지정하는 문자열(file:/// URI로 표시됨)

반환값

없음

설명

메서드, SWF 파일을 문서로 가져옵니다. 이 메서드는 [가져오기] 메뉴 명령을 사용하여 SWF 파일을 지정하는 것과 동일한 작업을 수행합니다. Flash 8 이상에서는 `document.importFile()`을 사용하여 SWF 파일(다른 파일 유형 포함)을 가져올 수도 있습니다.

예제

다음 예제에서는 Flash Configuration 폴더에서 "mySwf.swf" 파일을 가져옵니다.

```
fl.getDocumentDOM().importSWF(fl.configURI+"mySwf.swf");
```

참고 사항

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

document.intersect()

지원 버전

Flash 8

사용법

```
document.intersect()
```

매개 변수

없음

반환값

부울 값, 성공적으로 실행되면 `true`를 반환하고 그렇지 않으면 `false`를 반환합니다.

설명

메서드, 선택한 모든 Drawing 객체에서 교차 Drawing 객체를 만듭니다. 선택한 Drawing 객체가 없거나 선택한 항목 중에 Drawing 객체가 아닌 항목이 있으면 `false`가 반환됩니다.

예제

다음 예제에서는 선택한 모든 Drawing 객체에서 교차 Drawing 객체를 만듭니다.

```
fl.getDocumentDOM().intersect();
```

참고 사항

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

document.library

지원 버전

Flash MX 2004

사용법

```
document.library
```

설명

읽기 전용 속성, 문서에 대한 [library](#) 객체입니다.

예제

다음 예제에서는 현재 포커스가 있는 문서의 라이브러리를 가져옵니다.

```
var myCurrentLib = fl.getDocumentDOM().library;
```

다음 예제에서는 현재 포커스가 있는 문서가 `fl.documents[1]`이 아니라고 가정하고 포커스가 없는 라이브러리 또는 [파일] > [열기]를 사용하여 외부 라이브러리로 연 라이브러리를 가져옵니다.

```
var externalLib = fl.documents[1].library;
```

document.libraryPath

지원 버전

Flash CS4 Professional

사용법

```
document.libraryPath
```

설명

속성, SWC 파일이 포함된 폴더나 SWC 파일의 위치를 지정하는 문서의 ActionScript 3.0 라이브러리 경로에 있는 항목의 목록이 포함된 문자열입니다. 문자열에 있는 항목은 세미콜론으로 구분됩니다. 제작 도구에서 [파일] > [제작 설정]을 선택한 다음 [Flash] 탭에서 [ActionScript 3.0 스크립트 설정]을 선택하여 항목을 지정할 수 있습니다.

예제

다음 예제에서는 ../Files 폴더를 문서의 라이브러리 경로에 추가한 다음 [출력] 패널에 라이브러리 경로를 표시합니다.

```
var myDoc = fl.getDocumentDOM();  
fl.trace(myDoc.libraryPath);  
myDoc.libraryPath = "../Files;" + myDoc.libraryPath;  
fl.trace(myDoc.libraryPath);
```

참고 사항

[document.externalLibraryPath](#), [document.sourcePath](#), [fl.libraryPath](#)

document.livePreview

지원 버전

Flash MX 2004

사용법

```
document.livePreview
```

설명

속성, [실시간 미리 보기]가 활성화되어 있는지 여부를 지정하는 부울 값입니다. true로 설정된 경우 제작된 Flash 내용에 구성 요소가 표시되는 모양을 스테이지에서 실제 크기로 미리 볼 수 있습니다. false로 설정된 경우 구성 요소의 윤곽만 볼 수 있습니다. 기본값은 true입니다.

예제

다음 예제에서는 [실시간 미리 보기]를 `false`로 설정합니다.

```
f1.getDocumentDOM().livePreview = false;
```

document.loadCuepointXML()

지원 버전

Flash Professional CS5

사용법

```
document.loadCuepointXML(String URI)
```

매개 변수

URI 문자열, 큐 포인트 XML 파일의 절대 경로

설명

메서드, 큐 포인트 XML 파일을 로드합니다. XML 파일의 서식과 DTD는 큐 포인트 속성 관리자에서 가져오거나 내보내는 것과 동일합니다. FLVPlayback 구성 요소 인스턴스가 포함된 객체의 큐 포인트 속성으로 직렬화된 문자열 값을 반환합니다.

예제

다음 예제에서는 `C:\testCuePoints.xml`에 큐 포인트 XML 파일이 있습니다.

```
var cuePoints = f1.getDocumentDOM().LoadCuepointXML("c:\testCuePoints.xml");
```

document.match()

지원 버전

Flash MX 2004

사용법

```
document.match(bWidth, bHeight [, bUseDocumentBounds])
```

매개 변수

`bWidth` `true`로 설정된 경우 메서드가 선택한 항목의 너비를 동일하게 만들도록 하는 부울 값

`bHeight` `true`로 설정된 경우 메서드가 선택한 항목의 높이를 동일하게 만들도록 하는 부울 값

`bUseDocumentBounds` `true`로 설정된 경우 메서드가 문서의 경계에 맞게 객체의 크기를 조절하도록 하는 부울 값. 그렇지 않으면 가장 큰 객체의 경계가 사용됩니다. 기본값은 `false`입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 선택한 객체의 크기를 동일하게 만듭니다.

예제

다음 예제에서는 선택한 객체의 너비만 조절합니다.

```
f1.getDocumentDOM().match(true, false);
```

다음 예제에서는 높이만 조절합니다.

```
f1.getDocumentDOM().match(false, true);
```

다음 예제에서는 문서의 경계에 맞게 너비만 조절합니다.

```
f1.getDocumentDOM().match(true, false, true);
```

참고 사항

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

document.mouseClick()

지원 버전

Flash MX 2004

사용법

```
document.mouseClick(position, bToggleSel, bShiftSel)
```

매개 변수

position 클릭의 x 및 y 좌표를 픽셀 단위로 지정하는 부동 소수점 값 쌍

bToggleSel Shift 키의 상태를 지정하는 부울 값으로 Shift 키가 눌러져 있으면 true를 반환하고 그렇지 않으면 false를 반환합니다.

bShiftSel 응용 프로그램 환경 설정 Shift 선택의 상태를 지정하는 부울 값으로 켜져 있으면 true를 반환하고 꺼져 있으면 false를 반환합니다.

반환값

없음

설명

메서드, [선택 도구]에서 마우스 클릭을 수행합니다.

예제

다음 예제에서는 지정된 위치에서 마우스 클릭을 수행합니다.

```
f1.getDocumentDOM().mouseClick({x:300, y:200}, false, false);
```

참고 사항

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

document.mouseDbClick()

지원 버전

Flash MX 2004

사용법

```
document.mouseDbClick(position, bAltDown, bShiftDown, bShiftSelect)
```

매개 변수

position 클릭의 **x** 및 **y** 좌표를 픽셀 단위로 지정하는 부동 소수점 값 쌍

bAltDown 이벤트 발생 시 **Alt** 키가 눌러져 있었는지 여부를 기록하는 부울 값으로 눌러져 있었으면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

bShiftDown 이벤트 발생 시 **Shift** 키가 눌러져 있었는지 여부를 기록하는 부울 값으로 눌러져 있었으면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

bShiftSelect 응용 프로그램 환경 설정 **Shift** 선택의 상태를 나타내는 부울 값으로 켜져 있으면 **true**를 반환하고 꺼져 있으면 **false**를 반환합니다.

반환값

없음

설명

메서드, [선택 도구]에서 마우스 두 번 클릭을 수행합니다.

예제

다음 예제에서는 지정된 위치에서 마우스 두 번 클릭을 수행합니다.

```
fl.getDocumentDOM().mouseDbClick({x:392.9, y:73}, false, false, true);
```

참고 사항

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

document.moveSelectedBezierPointsBy()

지원 버전

Flash MX 2004

사용법

```
document.moveSelectedBezierPointsBy(delta)
```

매개 변수

delta 선택한 베지어 점을 이동하는 데 사용할 **x** 및 **y** 좌표를 픽셀 단위로 지정하는 부동 소수점 값 쌍. 예를 들어 $(\{x:1,y:2\})$ 가 전달되면 현재 위치에서 오른쪽으로 1픽셀, 아래쪽으로 2픽셀 떨어진 위치가 지정됩니다.

반환값

없음

설명

선택 항목에 베지어 점이 최소 하나 이상 선택되어 있는 경로가 적어도 하나 이상 포함되어 있는 경우, 선택한 모든 경로에서 선택한 모든 베지어 점을 지정된 양만큼 이동합니다.

예제

다음 예제에서는 선택한 베지어 점을 오른쪽으로 10픽셀, 아래쪽으로 5픽셀만큼 이동합니다.

```
fl.getDocumentDOM().moveSelectedBezierPointsBy({x:10, y:5});
```

document.moveSelectionBy()

지원 버전

Flash MX 2004

사용법

```
document.moveSelectionBy(distanceToMove)
```

매개 변수

distanceToMove 선택 항목을 이동하는 데 사용할 **x** 및 **y** 좌표 값을 지정하는 부동 소수점 값 쌍. 예를 들어, ({x:1,y:2})가 전달되면 현재 위치에서 오른쪽으로 1픽셀, 아래쪽으로 2픽셀 떨어진 위치가 지정됩니다.

반환값

없음

설명

메서드, 선택한 객체를 지정된 거리만큼 이동합니다.

참고: 사용자가 화살표 키를 사용하여 항목을 이동할 경우 화살표 키를 몇 번 누르든 [작업 내역] 패널에는 하나의 이동으로 처리됩니다. [작업 내역] 패널에서 여러 단계를 실행하는 대신 화살표 키를 반복해서 누르면 메서드는 한 단계만 수행하고 반복된 화살표 키를 반영하기 위해 인수가 업데이트됩니다.

선택 방법에 대한 자세한 내용은 [document.setSelectionRect\(\)](#), [document.mouseClick\(\)](#), [document.mouseDbClk\(\)](#) 및 [Element 객체](#)를 참조하십시오.

예제

다음 예제에서는 선택한 항목을 오른쪽으로 62픽셀, 아래쪽으로 84픽셀만큼 이동합니다.

```
fl.getDocumentDOM().moveSelectionBy({x:62, y:84});
```

document.name

지원 버전

Flash MX 2004

사용법

```
document.name
```

설명

읽기 전용 속성, 문서(FLA 파일)의 이름을 나타내는 문자열입니다.

예제

다음 예제에서는 **fileName** 변수를 문서 배열에 있는 첫 번째 문서의 파일 이름으로 설정합니다.

```
var fileName = flash.documents[0].name;
```

다음 예제에서는 열려 있는 모든 문서의 이름을 [출력] 패널에 표시합니다.

```
var openDocs = fl.documents;  
for(var i=0;i < openDocs.length; i++){  
    fl.trace(i + " " + openDocs[i].name + "\n");  
}
```

document.optimizeCurves()

지원 버전
Flash MX 2004

사용법
document.optimizeCurves(smoothing, bUseMultiplePasses)

매개 변수
smoothing 0부터 100까지의 정수. 0을 지정하면 매끄러움이 적용되지 않고 100을 지정하면 최대 매끄러움이 적용됩니다.
bUseMultiplePasses true로 설정된 경우 실행 속도는 느리지만 더 좋은 결과를 얻을 수 있도록 메서드에서 여러 번 수행해야 하도록 나타내는 부울 값. 이 매개 변수는 [곡선 최적화] 대화 상자에서 [여러 번 수행] 버튼을 클릭하는 것과 같은 효과를 가집니다.

반환값
없음

설명
메서드, 최적의 매끄러움을 얻기 위해 여러 번 수행(지정된 경우)을 통해 현재 선택 항목의 매끄러움을 최적화합니다. 이 메서드는 [수정] > [모양] > [최적화]를 선택하는 것과 같습니다.

예제
다음 예제에서는 여러 번 수행을 통해 현재 선택 항목의 곡선을 50°의 매끄러움으로 최적화합니다.
fl.getDocumentDOM().optimizeCurves(50, true);

document.path

지원 버전
Flash MX 2004

사용법
document.path

설명
읽기 전용 속성, 문서의 경로를 플랫폼별 형식으로 나타내는 문자열입니다. 이전에 문서를 저장한 적이 없으면 이 속성은 undefined가 됩니다.

예제
다음 예제에서는 문서 배열에 있는 첫 번째 문서의 경로를 [출력] 패널에 표시합니다. 이 스크립트를 실행하기 전에 문서를 저장해야 합니다. 이 예제에서는 test.fla라는 파일이 Windows 컴퓨터의 내 문서 폴더에 저장됩니다.

```
var filePath = flash.documents[0].path;  
fl.trace(filePath);  
// displays C:\Documents and Settings\\My Documents\test fla
```

참고 사항

[document.pathURI](#)

document.pathURI

지원 버전

Flash CS4 Professional

사용법

`document.pathURI`

설명

읽기 전용 속성, 문서의 경로를 나타내는 문자열(**file:///** URI로 표시됨)입니다. 이전에 문서를 저장한 적이 없으면 이 속성은 `undefined`가 됩니다.

예제

다음 예제에서는 문서 배열에 있는 첫 번째 문서의 경로를 **file:///** URI 문자열로 [출력] 패널에 표시합니다. 이 스크립트를 실행하기 전에 문서를 저장해야 합니다. 이 예제에서는 **test.fla**라는 파일이 Windows 컴퓨터의 내 문서 폴더에 저장됩니다.

```
var filePathURI = flash.documents[0].pathURI;  
fl.trace(filePathURI);  
// displays file:///C:/Documents%20and%20Settings/<userName>/My%20Documents/test.fla
```

참고 사항

[document.path](#)

document.publish()

지원 버전

Flash MX 2004

사용법

`document.publish()`

매개 변수

없음

반환값

없음

설명

메서드, 활성 제작 설정([파일] > [제작 설정])에 따라 문서를 제작합니다. 이 메서드는 [파일] > [제작]을 선택하는 것과 같습니다.

예제

다음 예제에서는 현재 문서를 제작합니다.

```
fl.getDocumentDOM().publish();
```

document.publishProfiles

지원 버전

Flash MX 2004

사용법

```
document.publishProfiles
```

설명

읽기 전용 속성, 문서에 대한 제작 프로파일 이름의 배열입니다.

예제

다음 예제에서는 문서에 대한 제작 프로파일의 이름을 표시합니다.

```
var myPubProfiles = fl.getDocumentDOM().publishProfiles;
for (var i=0; i < myPubProfiles.length; i++){
    fl.trace(myPubProfiles[i]);
}
```

document.punch()

지원 버전

Flash 8

사용법

```
document.punch()
```

매개 변수

없음

반환값

부울 값, 성공적으로 실행되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 맨 위에 선택된 Drawing 객체를 사용하여 그 아래에 선택되어 있는 모든 Drawing 객체를 펀칭합니다. 선택한 Drawing 객체가 없거나 선택한 항목 중에 Drawing 객체가 아닌 항목이 있으면 false가 반환됩니다.

예제

다음 예제에서는 선택한 Drawing 객체 아래에 있는 Drawing 객체에 펀치 효과를 적용합니다.

```
fl.getDocumentDOM().punch();
```

참고 사항

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

document.removeAllFilters()

지원 버전

Flash 8

사용법

```
document.removeAllFilters()
```

매개 변수

없음

반환값

없음

설명

메서드, 선택한 객체에서 모든 필터를 제거합니다.

예제

다음 예제에서는 선택한 객체에서 모든 필터를 제거합니다.

```
fl.getDocumentDOM().removeAllFilters();
```

참고 사항

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableAllFilters\(\)](#), [document.getFilters\(\)](#),
[document.removeFilter\(\)](#), [Filter 객체](#)

document.removeDataFromDocument()

지원 버전

Flash MX 2004

사용법

```
document.removeDataFromDocument(name)
```

매개 변수

name 제거할 데이터의 이름을 지정하는 문자열

반환값

없음

설명

메서드, 문서에 첨부된 특정 이름을 가진 영구 데이터를 제거합니다.

예제

다음 예제에서는 문서에서 "myData"라는 영구 데이터를 제거합니다.

```
fl.getDocumentDOM().removeDataFromDocument("myData");
```

참고 사항

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

document.removeDataFromSelection()

지원 버전

Flash MX 2004

사용법

```
document.removeDataFromSelection(name)
```

매개 변수

name 제거할 영구 데이터의 이름을 지정하는 문자열

반환값

없음

설명

메서드, 선택 영역에 첨부된 특정 이름을 가진 영구 데이터를 제거합니다.

예제

다음 예제에서는 선택 영역에서 "myData"라는 영구 데이터를 제거합니다.

```
fl.getDocumentDOM().removeDataFromSelection("myData");
```

참고 사항

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

document.removeFilter()

지원 버전

Flash 8

사용법

```
document.removeFilter(filterIndex)
```

매개 변수

filterIndex 선택한 객체에서 제거할 필터의 0부터 시작하는 인덱스를 지정하는 정수

반환값

없음

설명

메서드, 선택한 객체의 [필터] 목록에서 지정된 필터를 제거합니다.

예제

다음 예제에서는 선택한 객체의 [필터] 목록에서 첫 번째 필터(인덱스 값 0)를 제거합니다.

```
fl.getDocumentDOM().removeFilter(0);
```

참고 사항

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableFilter\(\)](#), [document.getFilters\(\)](#),
[document.removeAllFilters\(\)](#), [Filter 객체](#)

document.renamePublishProfile()

지원 버전

Flash MX 2004

사용법

```
document.renamePublishProfile([profileNewName])
```

매개 변수

profileNewName 프로파일의 새 이름을 지정하는 선택적 매개 변수. 새 이름은 고유해야 합니다. 이름을 지정하지 않으면 기본 이름이 사용됩니다.

반환값

부울 값, 이름이 성공적으로 변경되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 현재 프로파일의 이름을 변경합니다.

예제

다음 예제에서는 현재 프로파일의 이름을 기본 이름으로 변경하고 표시합니다.

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

document.renameScene()

지원 버전

Flash MX 2004

사용법

```
document.renameScene(name)
```

매개 변수

name 장면의 새 이름을 지정하는 문자열

반환값

부울 값, 이름이 성공적으로 변경되면 `true`를 반환하고 그렇지 않으면 `false`를 반환합니다. 예를 들어 새 이름이 고유하지 않으면 `false`가 반환됩니다.

설명

메서드, [장면] 패널에서 현재 선택한 장면의 이름을 변경합니다. 선택한 장면의 새 이름은 고유해야 합니다.

예제

다음 예제에서는 현재 장면의 이름을 "new name"으로 변경합니다.

```
var success = fl.getDocumentDOM().renameScene("new name");
```

document.reorderScene()

지원 버전

Flash MX 2004

사용법

```
document.reorderScene(sceneToMove, sceneToPutItBefore)
```

매개 변수

sceneToMove 첫 번째 장면에 0을 지정하는 방식으로 이동할 장면을 지정하는 정수

sceneToPutItBefore **sceneToMove**에 의해 지정된 이동할 장면의 앞에 위치할 장면을 지정하는 정수. 첫 번째 장면에 0을 지정합니다. 예를 들어 **sceneToMove**에 1을 지정하고 **sceneToPutItBefore**에 0을 지정하면 두 번째 장면이 첫 번째 장면 앞으로 이동합니다. 장면을 맨 뒤로 이동하려면 -1을 지정합니다.

반환값

없음

설명

메서드, 특정 장면을 지정된 다른 장면 앞으로 이동합니다.

예제

다음 예제에서는 두 번째 장면을 첫 번째 장면 앞으로 이동합니다.

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

document.resetOvalObject()

지원 버전

Flash CS3 Professional

사용법

```
document.resetOvalObject()
```

매개 변수

없음

반환값

없음

설명

메서드, 속성 관리자의 모든 값을 기본 **Oval** 객체 설정으로 설정합니다. **Oval** 객체를 선택한 경우 해당 객체의 속성도 기본값으로 재설정됩니다.

예제

다음 예제에서는 현재 문서의 **Oval** 객체 속성을 기본값으로 재설정합니다.

```
fl.getDocumentDOM().resetOvalObject();
```

참고 사항

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

document.resetRectangleObject()

지원 버전

Flash CS3 Professional

사용법

```
document.resetRectangleObject();
```

매개 변수

없음

반환값

없음

설명

메서드, 속성 관리자의 모든 값을 기본 **Rectangle** 객체 설정으로 설정합니다. **Rectangle** 객체를 선택한 경우 해당 객체의 속성도 기본값으로 재설정됩니다.

예제

다음 예제에서는 현재 문서의 **Rectangle** 객체 속성을 기본값으로 재설정합니다.

```
fl.getDocumentDOM().resetRectangleObject();
```

참고 사항

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

document.resetTransformation()

지원 버전
Flash MX 2004

사용법

```
document.resetTransformation()
```

매개 변수

없음

반환값

없음

설명

메서드, 변형 행렬을 재설정합니다. 이 메서드는 [수정] > [변형] > [변형 제거]를 선택하는 것과 같습니다.

예제

다음 예제에서는 현재 선택 항목의 변형 행렬을 재설정합니다.

```
fl.getDocumentDOM().resetTransformation();
```

document.revert()

지원 버전
Flash MX 2004

사용법

```
document.revert()
```

매개 변수

없음

반환값

없음

설명

메서드, 지정된 문서를 이전에 저장한 버전으로 되돌립니다. 이 메서드는 [파일] > [복귀]를 선택하는 것과 같습니다.

예제

다음 예제에서는 현재 문서를 이전에 저장한 버전으로 되돌립니다.

```
fl.getDocumentDOM().revert();
```

참고 사항

[document.canRevert\(\)](#), [fl.revertDocument\(\)](#)

document.rotate3DSelection()

지원 버전

Flash CS4 Professional

사용법

```
document.rotate3DSelection(xyzCoordinate, bGlobalTransform)
```

매개 변수

xyzCoordinate 3D 회전의 축을 지정하는 XYZ 좌표점

bGlobalTransform 변형 모드가 전역(true)인지, 아니면 로컬(false)인지를 지정하는 부울 값

반환값

없음

설명

메서드, 선택 항목에 3D 회전을 적용합니다. 이 메서드는 무비 클립에만 사용할 수 있습니다.

예제

다음 예제에서는 선택 항목이 먼저 스테이지를 기준으로(전역적으로) 회전된 다음 자체를 기준으로(로컬로) 회전됩니다.

```
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()

지원 버전

Flash MX 2004

사용법

```
document.rotateSelection(angle [, rotationPoint])
```

매개 변수

angle 회전 각도를 지정하는 부동 소수점 값

rotationPoint 회전할 경계 상자의 면을 지정하는 문자열. 사용할 수 있는 값에는 "top right", "top left", "bottom right", "bottom left", "top center", "right center", "bottom center" 및 "left center"가 있습니다. 이 값을 지정하지 않으면 변형점이 사용됩니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 선택 항목을 지정된 각도만큼 회전합니다. 이 메서드는 [자유 변형 도구]를 사용하여 객체를 회전하는 것과 같은 효과를 가집니다.

예제

다음 예제에서는 선택 항목을 변형점을 중심으로 45° 회전합니다.

```
fl.getDocumentDOM().rotateSelection(45);
```

다음 예제에서는 선택 항목을 왼쪽 아래 모서리를 중심으로 45° 회전합니다.

```
fl.getDocumentDOM().rotateSelection(45, "bottom left");
```

document.save()

지원 버전

Flash MX 2004

사용법

```
document.save([bOkToSaveAs])
```

매개 변수

bOkToSaveAs true이거나 생략된 경우, 파일이 저장된 적이 없으면 [다른 이름으로 저장] 대화 상자를 여는 선택적 매개 변수. false인 경우에는 파일이 저장된 적이 없어도 저장되지 않습니다.

반환값

부울 값, 저장 작업이 성공적으로 완료되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 기본 위치에 문서를 저장합니다. 이 메서드는 [파일] > [저장]을 선택하는 것과 같습니다.

동일한 이름으로 저장하지 않고 파일의 이름을 지정하려면 [fl.saveDocument\(\)](#)를 사용합니다.

참고: 수정 또는 저장되지 않은 새 파일이거나 마지막으로 저장된 이후로 수정되지 않은 파일인 경우 이 메서드를 호출하면 아무 작업도 수행되지 않고 false가 반환됩니다. 저장되지 않거나 수정되지 않은 파일이 저장되도록 하려면 [document.saveAndCompact\(\)](#) 또는 [fl.saveDocumentAs\(\)](#)를 사용합니다.

예제

다음 예제에서는 기본 위치에 현재 문서를 저장합니다.

```
fl.getDocumentDOM().save();
```

참고 사항

[document.saveAndCompact\(\)](#), [fl.saveAll\(\)](#), [fl.saveDocument\(\)](#), [fl.saveDocumentAs\(\)](#)

document.saveAndCompact()

지원 버전

Flash MX 2004

사용법

```
document.saveAndCompact([bOkToSaveAs])
```

매개 변수

bOkToSaveAs true이거나 생략된 경우, 파일이 저장된 적이 없으면 [다른 이름으로 저장] 대화 상자를 여는 선택적 매개 변수. false인 경우에는 파일이 저장된 적이 없어도 저장되지 않습니다. 기본값은 true입니다.

반환값

부울 값, 저장 및 압축 작업이 성공적으로 완료되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 파일을 저장하고 압축합니다. 이 메서드는 [파일] > [저장 및 압축]을 선택하는 것과 같습니다.

참고: 파일을 저장한 적이 없으면 [다른 이름으로 저장] 대화 상자를 취소하더라도 true가 반환됩니다. 파일이 저장되었는지 정확하게 확인하려면 [fl.saveDocumentAs\(\)](#)를 사용합니다.

예제

다음 예제에서는 현재 문서를 저장하고 압축합니다.

```
fl.getDocumentDOM().saveAndCompact();
```

참고 사항

[document.save\(\)](#), [fl.saveDocumentAs\(\)](#), [fl.saveDocument\(\)](#), [fl.saveAll\(\)](#)

document.scaleSelection()

지원 버전

Flash MX 2004

사용법

```
document.scaleSelection(xScale, yScale [, whichCorner])
```

매개 변수

xScale 크기를 조절하는 데 사용할 **x**의 양을 지정하는 부동 소수점 값

yScale 크기를 조절하는 데 사용할 **y**의 양을 지정하는 부동 소수점 값

whichCorner 변형할 가장자리를 지정하는 문자열 값. 생략하면 변형점의 크기가 조절됩니다. 사용할 수 있는 값에는 "bottom left", "bottom right", "top right", "top left", "top center", "right center", "bottom center" 및 "left center"가 있습니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 지정된 양만큼 선택 항목의 크기를 조절합니다. 이 메서드는 [자유 변형 도구]를 사용하여 객체의 크기를 조절하는 것과 같습니다.

예제

다음 예제에서는 현재 선택 항목의 너비를 원래 너비의 두 배로 늘리고 높이를 원래 높이의 절반으로 줄입니다.

```
fl.getDocumentDOM().scaleSelection(2.0, 0.5);
```

다음 예제에서는 선택 항목을 수직 방향으로 뒤집습니다.


```
fl.getDocumentDOM().scaleSelection(1, -1);
```

다음 예제에서는 선택 항목을 수평 방향으로 뒤집습니다.

```
fl.getDocumentDOM().scaleSelection(-1, 1);
```

다음 예제에서는 선택 영역의 크기를 위쪽 가운데를 기준으로 1.9만큼 수직 방향으로 조절합니다.

```
fl.getDocumentDOM().scaleSelection(1, 1.90, 'top center');
```

document.screenOutline

지원 버전

Flash MX 2004

사용법

```
document.screenOutline
```

설명

읽기 전용 속성, 문서에 대한 현재 ScreenOutline 객체입니다. 객체에 처음 액세스하는 경우 먼저 `document.allowScreens()`를 사용하여 속성이 있는지 확인해야 합니다.

예제

다음 예제에서는 `screenOutline` 속성의 값 배열을 표시합니다.

```
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);
```

참고 사항

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

document.selectAll()

지원 버전

Flash MX 2004

사용법

```
document.selectAll()
```

매개 변수

없음

반환값

없음

설명

메서드, 스테이지에 있는 모든 항목을 선택합니다. 이 메서드는 Ctrl+A(Windows) 또는 Command+A(Macintosh)를 누르거나 [편집] > [전체 선택]을 선택하는 것과 같습니다.

예제

다음 예제에서는 현재 표시된 모든 항목을 선택합니다.

```
fl.getDocumentDOM().selectAll();
```

참고 사항

[document.selection](#), [document.selectNone\(\)](#)

document.selection

지원 버전

Flash MX 2004

사용법

`document.selection`

설명

속성, 문서에서 선택한 객체의 배열입니다. 선택한 객체가 없으면 길이가 0인 배열이 반환되고 열려 있는 문서가 없으면 null이 반환됩니다.

배열에 객체를 추가하려면 먼저 다음 방법 중 하나를 사용하여 객체를 선택해야 합니다.

- 스테이지에서 객체를 수동으로 선택합니다.
- [document.setSelectionRect\(\)](#), [document.setSelectionBounds\(\)](#), [document.mouseClick\(\)](#), [document.mouseDblClk\(\)](#) 또는 [document.selectAll\(\)](#) 같은 선택 메서드 중 하나를 사용합니다.
- 수동으로 하나 이상의 프레임을 선택합니다.
- [timeline.getSelectedFrames\(\)](#), [timeline.setSelectedFrames\(\)](#) 또는 [timeline.selectAllFrames\(\)](#) 같은 [Timeline](#) 객체 메서드 중 하나를 사용하여 하나 이상의 프레임을 선택합니다.
- 특정 프레임에서 모든 요소를 지정합니다([Element](#) 객체 참조). 다음 첫 번째 예제를 참조하십시오.
- 하나 이상의 요소로 구성된 배열을 만든 다음 해당 배열을 `document.selection` 배열에 할당합니다. 다음 세 번째 예제를 참조하십시오.

예제

다음 예제에서는 프레임 11에 있는 모든 요소를 현재 선택 항목에 할당합니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
fl.getDocumentDOM().getTimeline().currentFrame = 10;  
fl.getDocumentDOM().selection = fl.getDocumentDOM().getTimeline().layers[0].frames[10].elements;
```

다음 예제에서는 스테이지의 왼쪽 위 모서리에 사각형을 만들고 이 사각형 아래에 텍스트 문자열을 삽입합니다. 그런 다음 [document.setSelectionRect\(\)](#)를 사용하여 두 객체를 모두 선택하고 `document.selection` 배열에 추가한 후 [출력] 패널에 `document.selection`의 내용을 표시합니다.

```
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]);
}
```

다음 예제는 레이어 배열과 요소 배열을 반복하여 특정 심볼의 인스턴스를 찾아서 선택하는 방법을 보여 주는 고급 예제입니다. 이 예제는 여러 개의 프레임이나 장면에 대한 루프를 포함하도록 확장할 수 있습니다. 이 예제에서는 첫 번째 프레임에 있는 myMovieClip 무비 클립의 모든 인스턴스를 현재 선택 항목에 할당합니다.

```
// 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()

지원 버전
Flash MX 2004

사용법
document.selectNone()

매개 변수
없음

반환값

없음

설명

메서드, 선택한 항목을 선택 취소합니다.

예제

다음 예제에서는 선택한 모든 항목을 선택 취소합니다.

```
fl.getDocumentDOM().selectNone();
```

참고 사항

[document.selectAll\(\)](#), [document.selection](#)

document.setAlignToDocument()

지원 버전

Flash MX 2004

사용법

```
document.setAlignToDocument(bToStage)
```

매개 변수

bToStage true로 설정된 경우 스테이지에 객체를 할당하는 부울 값. false로 설정된 경우에는 스테이지에 객체가 할당되지 않습니다.

반환값

없음

설명

메서드, 문서에서 사용할 수 있는 [document.align\(\)](#), [document.distribute\(\)](#), [document.match\(\)](#) 및 [document.space\(\)](#)에 대한 환경 설정을 설정합니다. 이 메서드는 [정렬] 패널에서 [스테이지로] 버튼을 활성화하는 것과 같습니다.

예제

다음 예제에서는 [정렬] 패널에서 [스테이지로] 버튼을 활성화하여 스테이지를 기준으로 객체를 정렬합니다.

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

참고 사항

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

document.setBlendMode()

지원 버전

Flash 8

사용법

```
document.setBlendMode(mode)
```

매개 변수

mode 선택한 객체에 대한 원하는 혼합 모드를 나타내는 문자열. 사용할 수 있는 값에는 "normal", "layer", "multiply", "screen", "overlay", "hardlight", "lighten", "darken", "difference", "add", "subtract", "invert", "alpha" 및 "erase"가 있습니다.

반환값

없음

설명

메서드, 선택한 객체의 혼합 모드를 설정합니다.

예제

다음 예제에서는 선택한 객체의 혼합 모드를 "add"로 설정합니다.

```
fl.getDocumentDOM().setBlendMode("add");
```

참고 사항

[document.addFilter\(\)](#), [document.setFilterProperty\(\)](#), [symbolInstance.blendMode](#)

document.setCustomFill()

지원 버전

Flash MX 2004

사용법

```
document.setCustomFill(fill)
```

매개 변수

fill 사용할 채우기 설정을 지정하는 Fill 객체. [Fill 객체](#)를 참조하십시오.

반환값

없음

설명

메서드, [도구] 패널, 속성 관리자 및 선택한 모양에 대한 채우기 설정을 설정합니다. 이 메서드를 사용하면 객체를 그리고 선택한 다음 채우기 설정을 변경하는 대신, 스크립트를 통해 객체를 그리기 전에 채우기를 설정할 수 있습니다. 또한 [도구] 패널 및 속성 관리자의 채우기 설정을 변경할 수도 있습니다.

예제

다음 예제에서는 [도구] 패널, 속성 관리자 및 선택한 모양에서 채움 색상 견본의 색상을 흰색으로 변경합니다.

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.color = '#FFFFFF';  
fill.style = "solid";  
fl.getDocumentDOM().setCustomFill(fill);
```

참고 사항

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

document.setCustomStroke()

지원 버전

Flash MX 2004

사용법

```
document.setCustomStroke(stroke)
```

매개 변수

stroke [Stroke 객체](#)

반환값

없음

설명

메서드, [도구] 패널, 속성 관리자 및 선택한 모양에 대한 획 설정을 설정합니다. 이 메서드를 사용하면 객체를 그리고 선택한 다음 획 설정을 변경하는 대신, 스크립트를 통해 객체를 그리기 전에 획을 설정할 수 있습니다. 또한 [도구] 패널 및 속성 관리자의 획 설정을 변경할 수도 있습니다.

예제

다음 예제에서는 [도구] 패널, 속성 관리자 및 선택한 모양에서 획 두께 설정을 변경합니다.

```
var stroke = fl.getDocumentDOM().getCustomStroke();  
stroke.thickness += 2;  
fl.getDocumentDOM().setCustomStroke(stroke);
```

참고 사항

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

document.setElementProperty()

지원 버전

Flash MX 2004

사용법

```
document.setElementProperty(property, value)
```

매개 변수

property 설정할 Element 속성의 이름을 지정하는 문자열. 속성과 값의 전체 목록은 [Element 객체](#)의 속성 요약 표를 참조하십시오.

이 메서드를 사용하여 [element.elementType](#), [element.top](#) 또는 [element.left](#)와 같은 읽기 전용 속성에 대한 값을 설정할 수 없습니다.

value 지정된 Element 속성에 설정할 값을 지정하는 정수

반환값

없음

설명

메서드, 문서의 선택한 객체에 지정된 Element 속성을 설정합니다. 선택한 객체가 없으면 아무 작업도 수행되지 않습니다.

예제

다음 예제에서는 선택한 모든 객체의 너비와 높이를 각각 100과 50으로 설정합니다.

```
fl.getDocumentDOM().setElementProperty("width", 100);  
fl.getDocumentDOM().setElementProperty("height", 50);
```

document.setTextElementTextAttr()

지원 버전

Flash MX 2004

사용법

```
document.setTextElementTextAttr(attrName, attrValue [, startIndex [, endIndex]])
```

매개 변수

attrName 변경할 TextAttrs 속성의 이름을 지정하는 문자열

attrValue TextAttrs 속성에 설정할 값. 속성 이름 및 예상되는 값의 목록은 [TextAttrs 객체](#)의 속성 요약 표를 참조하십시오.

startIndex 영향을 받는 첫 번째 문자의 인덱스를 지정하는 정수 값. 이 매개 변수는 선택 사항입니다.

endIndex 영향을 받는 마지막 문자의 인덱스를 지정하는 정수 값. 이 매개 변수는 선택 사항입니다.

반환값

부울 값, 최소한 하나 이상의 텍스트 특성이 변경되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 선택한 텍스트 항목의 지정된 textAttrs 속성을 특정 값으로 설정합니다. 속성 이름 및 허용되는 값의 목록은 [TextAttrs 객체](#)의 속성 요약 표를 참조하십시오. 선택적 매개 변수가 전달되지 않은 경우에는 현재 선택한 텍스트 범위의 스타일이 설정되고, 선택한 텍스트가 없는 경우에는 전체 텍스트 필드가 설정됩니다. **startIndex**만 전달된 경우에는 해당 문자의 특성이 설정됩니다. **startIndex** 및 **endIndex**가 전달된 경우 **startIndex**부터 **endIndex**(포함 안 됨) 사이에 있는 문자의 특성이 설정됩니다. 단락 스타일이 지정되지 않은 경우에는 범위 내의 모든 단락이 영향을 받습니다.

예제

다음 예제에서는 선택한 텍스트 항목의 fillColor, italic 및 bold 텍스트 특성을 설정합니다.

```
var success = fl.getDocumentDOM().setTextElementTextAttr("fillColor", "#00ff00");  
var pass = fl.getDocumentDOM().setTextElementTextAttr("italic", true, 10);  
var ok = fl.getDocumentDOM().setTextElementTextAttr("bold", true, 5, 15);
```

document.setFillColor()

지원 버전

Flash MX 2004

사용법

```
document.setFillColor(color)
```

매개 변수

color 형식이 다음 중 하나인 채우기 색상

- "#RRGGBB" 또는 "#RRGGBBAA" 형식의 문자열
- 0xRRGGBB 형식의 16진수
- 16진수 숫자와 같은 십진수를 나타내는 정수

null로 설정된 경우 채움 색상이 설정되지 않습니다. 이것은 사용자 인터페이스에서 채움 색상 견본을 채움 없음으로 설정하는 것과 같습니다.

반환값

없음

설명

메서드, 선택 항목의 채움 색상을 지정된 색상으로 변경합니다. [도구] 패널 및 속성 관리자에서 채움 색상을 변경하는 방법에 대한 자세한 내용은 [document.setCustomFill\(\)](#)을 참조하십시오.

예제

다음 예제에서 처음 세 개의 명령문은 색상 지정을 위한 각기 다른 형식을 사용하여 채움 색상을 설정합니다. 네 번째 명령문은 채움을 채움 없음으로 설정합니다.

```
fl.getDocumentDOM().setFillColor("#cc00cc");  
fl.getDocumentDOM().setFillColor(0xcc00cc);  
fl.getDocumentDOM().setFillColor(120000);  
fl.getDocumentDOM().setFillColor(null);
```

document.setFilterProperty()

지원 버전

Flash 8

사용법

```
document.setFilterProperty(property, filterIndex, value)
```

매개 변수

property 설정할 속성을 지정하는 문자열. 사용할 수 있는 값에는 "blurX", "blurY", "quality", "angle", "distance", "strength", "knockout", "inner", "bevelType", "color", "shadowColor" 및 "highlightColor"가 있습니다.

filterIndex [필터] 목록에 있는 필터의 0부터 시작하는 인덱스를 지정하는 정수

value 지정된 필터 속성에 대해 설정할 값을 지정하는 숫자 또는 문자열. 사용할 수 있는 값은 설정 중인 속성과 필터에 따라 달라집니다.

반환값

없음

설명

메서드, 현재 선택되어 있는 객체에 대해 지정된 필터 속성을 설정합니다(객체가 지정된 필터를 지원하는 경우).

예제

다음 예제에서는 선택한 객체의 [필터] 목록에서 두 번째 필터(인덱스 값 1)에 대한 `quality` 속성을 2로 설정한 다음 선택한 객체의 [필터] 목록에서 첫 번째 필터의 `shadowColor` 속성을 설정합니다.

```
fl.getDocumentDOM().setFilterProperty("quality", 1, 2);  
fl.getDocumentDOM().setFilterProperty("shadowColor", 0, "#FF00FF");
```

참고 사항

[document.addFilter\(\)](#), [document.getFilters\(\)](#), [document.setBlendMode\(\)](#), [document.setFilters\(\)](#), [Filter 객체](#)

document.setFilters()

지원 버전

Flash 8

사용법

```
document.setFilters(filterArray)
```

매개 변수

`filterArray` 현재 지정된 필터의 배열

반환값

없음

설명

메서드, 선택한 객체에 필터를 적용합니다. `document.getFilters()`를 호출하고 필터를 원하는 대로 변경한 후에 이 메서드를 사용합니다.

예제

다음 예제에서는 선택한 객체의 필터를 가져오고 모든 흐림 필터의 `blurX` 속성을 50으로 설정합니다.

```
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);
```

참고 사항

[document.addFilter\(\)](#), [document.getFilters\(\)](#), [document.setFilterProperty\(\)](#), [Filter 객체](#)

document.setInstanceAlpha()

지원 버전

Flash MX 2004

사용법

```
document.setInstanceAlpha(opacity)
```

매개 변수

opacity 인스턴스의 투명도를 조정하는 0(투명)부터 100(완전 채도)까지의 정수

반환값

없음

설명

메서드, 인스턴스의 불투명도를 설정합니다.

예제

다음 예제에서는 농도의 불투명도 값을 50으로 설정합니다.

```
fl.getDocumentDOM().setInstanceAlpha(50);
```

document.setInstanceBrightness()

지원 버전

Flash MX 2004

사용법

```
document.setInstanceBrightness(brightness)
```

매개 변수

brightness 밝기를 지정하는 -100(검정색)부터 100(흰색)까지의 정수

반환값

없음

설명

메서드, 인스턴스의 밝기를 설정합니다.

예제

다음 예제에서는 인스턴스의 밝기 값을 50으로 설정합니다.

```
fl.getDocumentDOM().setInstanceBrightness(50);
```

document.setInstanceTint()

지원 버전

Flash MX 2004

사용법

```
document.setInstanceTint( color, strength )
```

매개 변수

color 형식이 다음 중 하나인 농도 색상

- "#RRGGBB" 또는 "#RRGGBBAA" 형식의 문자열
- 0xRRGGBB 형식의 16진수
- 16진수 숫자와 같은 십진수를 나타내는 정수

strength 농도의 불투명도를 지정하는 0부터 100까지의 정수

반환값

없음

설명

메서드, 인스턴스의 농도를 설정합니다.

예제

다음 예제에서는 선택한 인스턴스의 농도를 불투명도 값이 50인 빨간색으로 설정합니다.

```
fl.getDocumentDOM().setInstanceTint(0xff0000, 50);
```

document.setMetadata()

지원 버전

Flash 8

사용법

```
document.setMetadata(strMetadata)
```

매개 변수

strMetadata 문서와 연관된 XML 메타데이터가 포함된 문자열. 자세한 내용은 다음 설명을 참조하십시오.

반환값

부울 값, 성공적으로 실행되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 기존 메타데이터를 덮어쓰는 방식으로 지정된 문서에 대한 XML 메타데이터를 설정합니다. **strMetadata**로 전달된 XML은 저장되기 전에 유효성 검사가 수행되며 다시 작성할 수도 있습니다. 유효성 검사 결과 이 XML이 유효한 XML이 아니거나 특정 규칙을 위반하는 경우 XML 메타데이터가 설정되지 않고 false가 반환됩니다. false가 반환되면 보다 자세한 오류 정보를 볼 수 있는 방법이 없습니다.

참고: true가 반환되어도 설정된 XML이 전달한 문자열과 똑같지 않을 수 있습니다. XML이 설정된 정확한 값을 가져오려면 `document.getMetadata()`를 사용합니다.

메타데이터의 형식은 XMP 사양과 호환되는 RDF입니다. RDF와 XMP에 대한 자세한 내용은 아래 자료를 참조하십시오.

- RDF Primer(www.w3.org/TR/rdf-primer/)
- RDF 사양(www.w3.org/TR/1999/REC-rdf-syntax-19990222/)
- XMP 홈 페이지(www.adobe.com/products/xmp/)

예제

다음 예제에서는 같은 데이터를 나타내는 몇 가지 유효한 방법을 보여 줍니다. 두 번째를 제외한 모든 경우에서 `Document.setMetadata()`로 보낸 데이터는 줄 바꿈이 제거될 뿐 다시 쓰여지지 않습니다.

첫 번째 예제에서 메타데이터는 여러 스키마가 각각 별도의 `rdf:Description` 태그에 배치되어 있는 태그에 들어 있습니다.

```
<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>
```

두 번째 예제에서 메타데이터는 태그에 들어 있지만 하나의 `rdf:Description` 태그에 여러 스키마가 모두 포함되어 있습니다. 이 예제에는 `Document.setMetadata()`에서 무시하고 버리는 주석도 포함되어 있습니다.

```
<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>
```

세 번째 예제에서 메타데이터는 특성에 들어 있으며 여러 스키마가 하나의 `rdf:Description` 태그에 모두 포함되어 있습니다.

```
<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>
```

참고 사항

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

document.setMobileSettings()

지원 버전

Flash CS3 Professional

사용법

```
document.setMobileSettings(xmlString)
```

매개 변수

xmlString 모바일 FLA 파일의 XML 설정을 설명하는 문자열

반환값

설정이 성공적으로 설정되었으면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 모바일 FLA 파일에 있는 XML 설정 문자열의 값을 설정합니다. 대부분의 모바일 FLA 파일에는 문서 내의 설정을 설명하는 XML 문자열이 있습니다.

예제

다음 예제에서는 모바일 FLA 파일에 대한 XML 설정 문자열을 설정합니다. 다음 예제는 한 줄로 된 코드입니다.

```
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>");
```

참고 사항

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

document.setOvalObjectProperty()

지원 버전

Flash CS3 Professional

사용법

```
document.setOvalObjectProperty(propertyName, value)
```

매개 변수

propertyName 설정할 속성을 지정하는 문자열. 사용할 수 있는 값은 [Oval 객체](#)의 속성 요약 표를 참조하십시오.

value 속성에 할당될 값. **propertyName**에 지정한 속성에 따라 사용할 수 있는 값이 달라집니다.

반환값

없음

설명

메서드, 프리미티브 Oval 객체의 지정한 속성에 대한 값을 지정합니다.

예제

예제는 [Oval 객체](#)의 개별 속성을 참조하십시오.

참고 사항

[Oval 객체](#), [shape.isOvalObject](#)

document.setPlayerVersion()

지원 버전

Flash CS3 Professional

사용법

```
document.setPlayerVersion(version)
```

매개 변수

version 지정된 문서의 대상 Flash Player 버전을 나타내는 문자열. 사용할 수 있는 값에는 "FlashLite", "FlashLite11", "FlashLite20", "FlashLite30", "1", "2", "3", "4", "5", "6", "7", "8", "9", "FlashPlayer10" 및 "AdobeAIR1_1"이 있습니다.

반환값

플레이어 버전이 성공적으로 설정되었으면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

설명

메서드, 지정된 문서의 대상 Flash Player의 버전을 설정합니다. 이 값은 [제작 설정] 대화 상자에서 설정한 값과 같습니다.

예제

다음 예제에서는 Flash Player 6을 현재 문서의 플레이어 버전으로 지정합니다.

```
fl.getDocumentDOM().setPlayerVersion("6");
```

참고 사항

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

document.setRectangleObjectProperty()

지원 버전

Flash CS3 Professional

사용법

```
document.setRectangleObjectProperty(propertyName, value)
```

매개 변수

propertyName 설정할 속성을 지정하는 문자열. 사용할 수 있는 값은 [Rectangle 객체](#)의 속성 요약 표를 참조하십시오.

value 속성에 할당될 값. **propertyName**에 지정된 속성에 따라 사용할 수 있는 값이 달라집니다.

반환값

없음

설명

메서드, 프리미티브 **Rectangle** 객체의 지정한 속성에 대한 값을 지정합니다.

예제

예제는 [Rectangle](#) 객체의 개별 속성을 참조하십시오.

참고 사항

[Rectangle](#) 객체, [shape.isRectangleObject](#)

document.setSelectionBounds()

지원 버전

Flash MX 2004, Flash 8의 경우 **bContactSensitiveSelection** 매개 변수를 추가로 사용할 수 있습니다.

사용법

```
document.setSelectionBounds(boundingRectangle [, bContactSensitiveSelection])
```

매개 변수

boundingRectangle 선택 항목의 새 위치와 크기를 지정하는 사각형. **boundingRectangle**의 형식에 대한 자세한 내용은 [document.addNewRectangle\(\)](#)을 참조하십시오.

bContactSensitiveSelection 객체 선택 중에 연결 감지 선택 모드가 활성화되어 있는지(**true**), 아니면 비활성화되어 있는지(**false**)를 지정하는 부울 값. 기본값은 **false**입니다.

반환값

없음

설명

메서드, 한 번의 작업으로 선택 항목을 이동하고 크기를 조절합니다.

bContactSensitiveSelection 값을 전달할 경우 이 메서드에만 유효하며 문서의 연결 감지 선택 모드에는 영향을 주지 않습니다([fl.contactSensitiveSelection](#) 참조).

예제

다음 예제에서는 현재 선택 항목을 10, 20으로 이동하고 크기를 100, 200으로 조절합니다.

```
var l = 10;  
var t = 20;  
fl.getDocumentDOM().setSelectionBounds({left:l, top:t, right:(100+l), bottom:(200+t)});
```

참고 사항

[document.selection](#), [document.setSelectionRect\(\)](#)

document.setSelectionRect()

지원 버전

Flash MX 2004, Flash 8의 경우 **bContactSensitiveSelection** 매개 변수를 추가로 사용할 수 있습니다.

사용법

```
document.setSelectionRect(rect [, bReplaceCurrentSelection [, bContactSensitiveSelection]])
```

매개 변수

rect 선택된 것으로 설정할 사각형 객체. **rect**의 형식에 대한 자세한 내용은 [document.addNewRectangle\(\)](#)을 참조하십시오.

bReplaceCurrentSelection 메시드가 현재 선택 항목을 대체하는지(true), 아니면 현재 선택 항목에 추가하는지(false)를 지정하는 부울 값. 기본값은 true입니다.

bContactSensitiveSelection 객체 선택 중에 연결 감지 선택 모드가 활성화되어 있는지(true), 아니면 비활성화되어 있는지(false)를 지정하는 부울 값. 기본값은 false입니다.

반환값

없음

설명

메시드, 지정된 좌표를 사용하여 스테이지를 기준으로 하는 사각형 선택 윤곽을 그립니다. 이 메시드는 사각형이 편집 중인 객체를 기준으로 하는 [document.getSelectionRect\(\)](#)와는 다릅니다.

이 메시드는 [선택 도구]를 사용하여 사각형을 드래그하는 것과 같습니다. 인스턴스를 선택하려면 사각형으로 완전히 둘러싸야 합니다.

bContactSensitiveSelection 값을 전달할 경우 이 메시드에만 유효하며 문서의 연결 감지 선택 모드에는 영향을 주지 않습니다([fl.contactSensitiveSelection](#) 참조).

참고: [작업 내역] 패널이나 메뉴 항목을 사용하여 [setSelectionRect\(\)](#)를 반복하면 [setSelectionRect\(\)](#) 작업 전의 단계도 반복됩니다.

예제

다음 예제에서는 두 번째 선택 항목으로 첫 번째 항목을 바꿉니다.

```
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);
```

다음 예제에서는 첫 번째 선택 항목에 두 번째 선택 항목을 추가합니다. 이것은 Shift 키를 누른 채 두 번째 객체를 선택하는 수동 작업과 같습니다.

```
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);
```

참고 사항

[document.getSelectionRect\(\)](#), [document.selection](#), [document.setSelectionBounds\(\)](#)

document.setStageVanishingPoint()

지원 버전

Flash CS4 Professional

사용법

```
document.setStageVanishingPoint(point)
```

매개 변수

point 3D 객체를 보기 위한 소실점을 설정하는 위치의 x 및 y 좌표를 지정하는 점

반환값

없음

설명

3D 객체를 보기 위한 소실점을 지정합니다.

예제

다음 예제에서는 스테이지 소실점을 설정합니다.

```
fl.getDocumentDOM().setStageVanishingPoint({x:45, y:45});
```

document.setStageViewAngle()

지원 버전

Flash CS4 Professional

사용법

```
document.setStageViewAngle(angle)
```

매개 변수

angle 0.0부터 179.0까지의 부동 소수점 값

반환값

없음

설명

3D 객체를 보기 위한 원근감 각도를 지정합니다.

예제

다음 예제에서는 스테이지 원근감 각도를 70도로 설정합니다.

```
fl.getDocumentDOM().setStageViewAngle(70);
```

document.setStroke()

지원 버전

Flash MX 2004

사용법

```
document.setStroke(color, size, strokeType)
```

매개 변수

color 형식이 다음 중 하나인 획의 색상

- "#RRGGBB" 또는 "#RRGGBBAA" 형식의 문자열
- 0xRRGGBB 형식의 16진수
- 16진수 숫자와 같은 십진수를 나타내는 정수

size 선택 항목의 새 획 크기를 지정하는 부동 소수점 값

strokeType 선택 항목의 새 획 유형을 지정하는 문자열. 사용할 수 있는 값에는 "hairline", "solid", "dashed", "dotted", "ragged", "stipple" 및 "hatched"가 있습니다.

반환값

없음

설명

메서드, 선택한 획의 색상, 폭 및 스타일을 설정합니다. [도구] 패널 및 속성 관리자에서 획을 변경하는 방법에 대한 자세한 내용은 [document.setCustomStroke\(\)](#)를 참조하십시오.

예제

다음 예제에서는 획의 색상, 크기 및 유형을 각각 빨간색, 3.25 및 파선으로 설정합니다.

```
fl.getDocumentDOM().setStroke("#ff0000", 3.25, "dashed");
```

document.setStrokeColor()

지원 버전

Flash MX 2004

사용법

```
document.setStrokeColor(color)
```

매개 변수

color 형식이 다음 중 하나인 획의 색상

- "#RRGGBB" 또는 "#RRGGBBAA" 형식의 문자열
- 0xRRGGBB 형식의 16진수
- 16진수 숫자와 같은 십진수를 나타내는 정수

반환값

없음

설명

메서드, 선택 항목의 획 색상을 지정된 색상으로 변경합니다. [도구] 패널 및 속성 관리자에서 획을 변경하는 방법에 대한 자세한 내용은 [document.setCustomStroke\(\)](#)를 참조하십시오.

예제

다음 예제에서 처음 세 개의 명령문은 색상 지정을 위한 각기 다른 포맷을 사용하여 획 색상을 설정합니다.

```
fl.getDocumentDOM().setStrokeColor("#cc00cc");  
fl.getDocumentDOM().setStrokeColor(0xcc00cc);  
fl.getDocumentDOM().setStrokeColor(120000);
```

document.setStrokeSize()

지원 버전
Flash MX 2004

사용법
document.setStrokeSize(size)

매개 변수
size 획 크기를 지정하는 0.25부터 10까지의 부동 소수점 값. 2자리 소수 자릿수보다 큰 정밀도는 무시됩니다.

반환값
없음

설명
메서드, 선택 항목의 획 크기를 지정된 크기로 변경합니다. [도구] 패널 및 속성 관리자에서 획을 변경하는 방법에 대한 자세한 내용은 [document.setCustomStroke\(\)](#)를 참조하십시오.

예제
다음 예제에서는 선택 항목의 획 크기를 5로 변경합니다.

```
fl.getDocumentDOM().setStrokeSize(5);
```

document.setStrokeStyle()

지원 버전
Flash MX 2004

사용법
document.setStrokeStyle(strokeType)

매개 변수
strokeType 현재 선택 항목의 획 스타일을 지정하는 문자열. 사용할 수 있는 값에는 "hairline", "solid", "dashed", "dotted", "ragged", "stipple" 및 "hatched"가 있습니다.

반환값
없음

설명
메서드, 선택 항목의 획 스타일을 지정된 스타일로 변경합니다. [도구] 패널 및 속성 관리자에서 획을 변경하는 방법에 대한 자세한 내용은 [document.setCustomStroke\(\)](#)를 참조하십시오.

예제

다음 예제에서는 선택 항목의 획 스타일을 "dashed"로 변경합니다.

```
fl.getDocumentDOM().setStrokeStyle("dashed");
```

document.setTextRectangle()

지원 버전

Flash MX 2004

사용법

```
document.setTextRectangle(boundingBox)
```

매개 변수

boundingRectangle 텍스트 항목을 전개할 새 크기를 지정하는 사각형. **boundingRectangle**의 형식에 대한 자세한 내용은 [document.addNewRectangle\(\)](#)을 참조하십시오.

반환값

부울 값, 하나 이상의 텍스트 필드 크기가 변경되면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

설명

메서드, 선택한 텍스트 항목의 경계 사각형을 지정된 크기로 변경합니다. 이 메서드는 텍스트가 새 사각형 안에서 다시 전개되도록 합니다. 이 경우 텍스트 항목은 크기가 조절되거나 변형되지 않습니다. **boundingRectangle**로 전달된 값은 다음과 같이 사용됩니다.

- 수평 및 정적 텍스트의 경우 메서드는 **boundingRectangle**로 전달된 너비 값만 고려하며 높이는 전체 텍스트에 맞게 자동으로 계산됩니다.
- 수직 텍스트(정적 텍스트)의 경우 메서드는 **boundingRectangle**로 전달된 높이 값만 고려하며 너비는 전체 텍스트에 맞게 자동으로 계산됩니다.
- 동적 또는 입력 텍스트의 경우 메서드는 **boundingRectangle**로 전달된 너비와 높이 값을 모두 고려하며 이 결과 생성되는 사각형은 전체 텍스트에 맞게 생성되는 사각형보다 클 수 있습니다. 그러나 매개 변수가 크기가 너무 작아 전체 텍스트에 맞출 수 없는 사각형을 지정하는 경우 메서드는 **boundingRectangle**로 전달된 너비 값만 고려합니다. 높이는 전체 텍스트에 맞게 자동으로 계산됩니다.

예제

다음 예제에서는 경계 텍스트 상자의 크기를 지정된 크기로 변경합니다.

```
fl.getDocumentDOM().setTextRectangle({left:0, top:0, right:50, bottom:200})
```

document.setTextSelection()

지원 버전

Flash MX 2004

사용법

```
document.setTextSelection(startIndex, endIndex)
```

매개 변수

startIndex 선택할 첫 번째 문자의 위치를 지정하는 정수. 첫 번째 문자 위치는 0입니다.

endIndex **endIndex**(포함 안 됨) 내에서 선택 항목의 끝 위치를 지정하는 정수. 첫 번째 문자 위치는 0입니다.

반환값

부울 값, 텍스트 선택 영역을 성공적으로 설정할 수 있으면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

설명

메서드, 현재 선택한 텍스트 필드의 텍스트 선택 영역을 **startIndex** 및 **endIndex** 값에 의해 지정된 값으로 설정합니다. 텍스트 편집이 아직 활성화되어 있지 않으면 활성화됩니다.

예제

다음 예제에서는 6번째 문자부터 25번째 문자까지의 텍스트를 선택합니다.

```
fl.document.setTextSelection(5, 25);
```

document.setTextString()

지원 버전

Flash MX 2004

사용법

```
document.setTextString(text [, startIndex [, endIndex]])
```

매개 변수

text 텍스트 필드에 삽입할 문자열

startIndex 바꿀 첫 번째 문자를 지정하는 정수. 첫 번째 문자 위치는 0입니다. 이 매개 변수는 선택 사항입니다.

endIndex 바꿀 마지막 문자를 지정하는 정수. 이 매개 변수는 선택 사항입니다.

반환값

부울 값, 하나 이상의 텍스트 문자열의 텍스트가 설정된 경우 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

설명

메서드, 텍스트 문자열을 삽입합니다. 선택적 매개 변수가 전달되지 않은 경우에는 기존 텍스트 선택 영역이 바뀌고, **Text** 객체가 현재 편집 중이지 않은 경우에는 전체 텍스트 문자열이 바뀝니다. **startIndex**만 전달된 경우 전달된 문자열이 이 위치에 삽입됩니다. **startIndex** 및 **endIndex**가 전달된 경우 **startIndex**부터 **endIndex**(포함 안 됨)까지의 텍스트 세그먼트는 전달된 문자열로 바뀝니다.

예제

다음 예제에서는 현재 텍스트 선택 영역을 "Hello World"로 바꿉니다.

```
var success = fl.getDocumentDOM().setTextString("Hello World!");
```

다음 예제에서는 현재 텍스트 선택 영역의 위치 6에 "hello"를 삽입합니다.

```
var pass = fl.getDocumentDOM().setTextString("hello", 6);
```

다음 예제에서는 현재 텍스트 선택 영역의 위치 2에서 위치 7(포함 안 됨) 사이에 "Howdy"를 삽입합니다.

```
var ok = fl.getDocumentDOM().setTextString("Howdy", 2, 7);
```

참고 사항

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

document.setTransformationPoint()

지원 버전

Flash MX 2004

사용법

```
document.setTransformationPoint( transformationPoint )
```

매개 변수

transformationPoint 다음 각 요소의 변형점에 대한 값을 지정하는 점(예: {x:10,y:20}, 여기서 x 및 y는 부동 소수점 숫자임)

- 모양: **transformationPoint**가 문서를 기준으로 설정됩니다(0,0이 스테이지의 왼쪽 위 모서리임).
- 심볼: **transformationPoint**가 심볼의 등록 포인트를 기준으로 설정됩니다(0,0이 등록 포인트임).
- 텍스트: **transformationPoint**가 텍스트 필드를 기준으로 설정됩니다(0,0이 텍스트 필드의 왼쪽 위 모서리임).
- 비트맵/비디오편 **transformationPoint**가 비트맵/비디오를 기준으로 설정됩니다(0,0이 비트맵 또는 비디오의 왼쪽 위 모서리임).
- 드로잉 객체, 프리미티브 타원형 및 사각형, 그룹: **transformationPoint**가 문서를 기준으로 설정됩니다(0,0이 스테이지의 왼쪽 위 모서리임). 객체, 프리미티브 또는 그룹의 중심점을 기준으로 **transformationPoint**를 설정하려면 [element.setTransformationPoint\(\)](#)를 사용합니다.

반환값

없음

설명

메서드, 현재 선택 항목의 변형점 위치를 설정합니다.

예제

다음 예제에서는 현재 선택 항목의 변형점을 100, 200으로 설정합니다.

```
fl.getDocumentDOM().setTransformationPoint({x:100, y:200});
```

참고 사항

[document.getTransformationPoint\(\)](#), [element.setTransformationPoint\(\)](#)

document.silent

지원 버전

Flash MX 2004

사용법

```
document.silent
```

설명

속성, 객체에 액세스할 수 있는지 여부를 지정하는 부울 값입니다. 이 속성은 [액세스 가능성] 패널에 있는 [무비를 액세스 가능하게 만들기] 설정의 역논리에 해당합니다. 즉, `document.silent`가 `true`이면 [무비를 액세스 가능하게 만들기] 옵션이 선택되어 있지 않은 것과 같고 `false`이면 [무비를 액세스 가능하게 만들기] 옵션이 선택되어 있는 것과 같습니다.

예제

다음 예제에서는 `isSilent` 변수를 `silent` 속성의 값으로 설정합니다.

```
var isSilent = fl.getDocumentDOM().silent;
```

다음 예제에서는 문서에 액세스할 수 있도록 `silent` 속성을 `false`로 설정합니다.

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

document.skewSelection()

지원 버전

Flash MX 2004

사용법

```
document.skewSelection(xSkew, ySkew [, whichEdge])
```

매개 변수

`xSkew` 기울일 `x`의 양을 각도 단위로 지정하는 부동 소수점 숫자

`ySkew` 기울일 `y`의 양을 각도 단위로 지정하는 부동 소수점 숫자

`whichEdge` 변형할 모서리를 지정하는 문자열. 생략하면 변형점이 기울어집니다. 사용할 수 있는 값에는 "top center", "right center", "bottom center" 및 "left center"가 있습니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 선택 항목을 지정된 양만큼 기울입니다. 이 메서드는 [자유 변형 도구]를 사용하여 객체를 기울이는 것과 같은 효과를 가집니다.

예제

다음 예제에서는 선택한 객체를 수직으로 2.0, 수평으로 1.5만큼 기울입니다. 두 번째 예제에서는 위쪽 가운데 가장자리에 있는 객체를 변형합니다.

```
fl.getDocumentDOM().skewSelection(2.0, 1.5);  
fl.getDocumentDOM().skewSelection(2.0, 1.5, "top center");
```

document.smoothSelection()

지원 버전

Flash MX 2004

사용법

```
document.smoothSelection()
```

매개 변수

없음

반환값

없음

설명

메서드, 선택한 각 곡선 또는 채우기 외곽선의 곡선을 매끄럽게 만듭니다. 이 메서드는 [도구] 패널의 [매끄럽게] 버튼과 동일한 동작을 수행합니다.

예제

다음 예제에서는 현재 선택 항목의 곡선을 매끄럽게 만듭니다.

```
fl.getDocumentDOM().smoothSelection();
```

document.sourcePath

지원 버전

Flash CS4 Professional

사용법

```
document.sourcePath
```

설명

속성, ActionScript 클래스 파일의 위치를 지정하는 문서의 ActionScript 3.0 소스 경로에 있는 항목의 목록이 포함된 문자열입니다. 문자열에 있는 항목은 세미콜론으로 구분됩니다. 제작 도구에서 [파일] > [제작 설정]을 선택한 다음 [Flash] 탭에서 [ActionScript 3.0 스크립트 설정]을 선택하여 항목을 지정할 수 있습니다.

예제

다음 예제는 ./Class 파일 폴더를 문서의 소스 경로에 추가합니다.

```
var myDoc = fl.getDocumentDOM();  
fl.trace(myDoc.sourcePath);  
myDoc.sourcePath = "./Class files;" + myDoc.sourcePath;  
fl.trace(myDoc.sourcePath);
```

참고 사항

[document.externalLibraryPath](#), [document.libraryPath](#), [fl.sourcePath](#)

document.space()

지원 버전

Flash MX 2004

사용법

```
document.space(direction [, bUseDocumentBounds])
```

매개 변수

direction 선택 항목 내 각 객체의 간격 방향을 지정하는 문자열. 사용할 수 있는 값에는 "horizontal" 또는 "vertical"이 있습니다.
bUseDocumentBounds true로 설정된 경우 문서의 경계를 기준으로 각 객체의 간격을 지정하는 부울 값. 그렇지 않으면 선택한 객체의 경계가 사용됩니다. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 선택 항목 내 각 객체의 간격을 일정하게 만듭니다.

예제

다음 예제에서는 스테이지를 기준으로 각 객체의 수평 간격을 지정합니다.

```
fl.getDocumentDOM().space("horizontal", true);
```

다음 예제에서는 서로를 기준으로 각 객체의 수평 간격을 지정합니다.

```
fl.getDocumentDOM().space("horizontal");
```

다음 예제에서는 **bUseDocumentBounds**를 명시적으로 false로 설정하여 서로를 기준으로 각 객체의 수평 간격을 지정합니다.

```
fl.getDocumentDOM().space("horizontal", false);
```

참고 사항

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

document.straightenSelection()

지원 버전

Flash MX 2004

사용법

```
document.straightenSelection()
```

매개 변수

없음

반환값

없음

설명

메서드, 현재 선택한 획을 곧게 만듭니다. 이 메서드는 [도구] 패널의 [곧게] 버튼을 사용하는 것과 같습니다.

예제

다음 예제에서는 현재 선택 항목의 곡선을 곧게 만듭니다.

```
fl.getDocumentDOM().straightenSelection();
```

document.swapElement()

지원 버전

Flash MX 2004

사용법

```
document.swapElement(name)
```

매개 변수

name 사용할 라이브러리 항목의 이름을 지정하는 문자열

반환값

없음

설명

메서드, 현재 선택 항목을 지정된 항목으로 교체합니다. 선택 항목에는 그래픽, 버튼, 무비 클립, 비디오 또는 비트맵이 포함되어 있어야 합니다. 객체를 선택하지 않았거나 지정된 객체를 찾을 수 없으면 오류 메시지가 표시됩니다.

예제

다음 예제에서는 현재 선택 항목을 라이브러리의 Symbol 1로 교체합니다.

```
fl.getDocumentDOM().swapElement('Symbol 1');
```

document.swapStrokeAndFill()

지원 버전

Flash 8

사용법

```
document.swapStrokeAndFill()
```

매개 변수

없음

반환값

없음

설명

메서드, [획] 및 [채움] 색상을 변경합니다.

예제

다음 예제에서는 현재 문서의 [획] 및 [채움] 색상을 변경합니다.

```
fl.getDocumentDOM().swapStrokeAndFill();
```

document.testMovie()

지원 버전

Flash MX 2004

사용법

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

매개 변수

abortIfErrorsExist 부울, 기본값은 false입니다. true로 설정되면 컴파일러 오류가 있는 경우 동영상 테스트 세션이 시작되지 않고 .swf 윈도우가 열리지 않습니다. 컴파일러 경고로 인해 명령이 중단되지 않습니다. 이 매개 변수는 Flash Professional CS5에서 추가되었습니다.

반환값

없음

설명

메서드, 문서에서 무비 테스트 작업을 실행합니다.

예제

다음 예제에서는 현재 문서에 대해 동영상을 테스트하지만 컴파일러 오류가 있는 경우 동영상 테스트를 중단합니다.

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

참고 사항

[document.canTestMovie\(\)](#), [document.testScene\(\)](#)

document.testScene()

지원 버전

Flash MX 2004

사용법

```
document.testScene()
```

매개 변수

없음

반환값

없음

설명

메서드, 문서의 현재 장면에서 장면 테스트 작업을 실행합니다.

예제

다음 예제에서는 문서의 현재 장면을 테스트합니다.

```
fl.getDocumentDOM().testScene();
```

참고 사항

[document.canTestScene\(\)](#), [document.testMovie\(\)](#)

document.timelines

지원 버전

Flash MX 2004

사용법

```
document.timelines
```

설명

읽기 전용 속성, **Timeline** 객체의 배열([Timeline 객체 참조](#))입니다.

예제

다음 예제에서는 활성 문서의 현재 **Timelines** 객체 배열을 가져오고 해당 이름을 [출력] 패널에 표시합니다.

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

참고 사항

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

document.traceBitmap()

지원 버전

Flash MX 2004

사용법

```
document.traceBitmap(threshold, minimumArea, curveFit, cornerThreshold)
```

매개 변수

threshold 추적된 비트맵의 색상 수를 제어하는 정수. 사용할 수 있는 값은 0부터 500까지의 정수입니다.

minimumArea 픽셀 단위로 측정되는 반경을 지정하는 정수. 사용할 수 있는 값은 1부터 1000까지의 정수입니다.

curveFit 외곽선의 매끄러움 정도를 지정하는 문자열. 사용할 수 있는 값에는 "pixels", "very tight", "tight", "normal", "smooth" 및 "very smooth"가 있습니다.

cornerThreshold **curveFit**와 비슷하지만 비트맵 이미지의 모서리에 적용되는 문자열. 사용할 수 있는 값에는 "many corners", "normal" 및 "few corners"가 있습니다.

반환값

없음

설명

메서드, 현재 선택 항목에서 비트맵 추적을 수행합니다. 이 메서드는 [수정] > [비트맵] > [비트맵 추적]을 선택하는 것과 같습니다.

예제

다음 예제에서는 지정된 매개 변수를 사용하여 선택한 비트맵을 추적합니다.

```
fl.getDocumentDOM().traceBitmap(0, 500, 'normal', 'normal');
```

document.translate3DCenter()

지원 버전

Flash CS4 Professional

사용법

```
document.translate3DCenter(xyzCoordinate)
```

매개 변수

xyzCoordinate 3D 회전 또는 평행 이동의 중심점을 지정하는 XYZ 좌표

반환값

없음

설명

메서드, 선택 항목이 평행 이동되거나 회전되는 중심 XYZ 위치를 설정합니다. 이 메서드는 무비 클립에만 사용할 수 있습니다.

예제

다음 예제에서는 3D 평행 이동의 XYZ 축을 지정합니다.

```
fl.getDocumentDOM().translate3DCenter({x:180, y:18, z:-30});
```

document.translate3DSelection()

지원 버전

Flash CS4 Professional

사용법

```
document.translate3DSelection(xyzCoordinate, bGlobalTransform)
```

매개 변수

xyzCoordinate 3D 평행 이동의 축을 지정하는 XYZ 좌표

bGlobalTransform 변형 모드가 전역(true)인지, 아니면 로컬(false)인지를 지정하는 부울 값

반환값

없음

설명

메서드, 선택 항목에 3D 평행 이동을 적용합니다. 이 메서드는 무비 클립에만 사용할 수 있습니다.

예제

다음 예제에서는 선택 항목이 먼저 스테이지를 기준으로(전역적으로) 평행 이동된 다음 자체를 기준으로(로컬로) 평행 이동됩니다.

```
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);
```

참고 사항

[document.translate3DCenter\(\)](#)

document.transformSelection()

지원 버전

Flash MX 2004

사용법

```
document.transformSelection(a, b, c, d)
```

매개 변수

- a 변형 행렬의 (0,0) 요소를 지정하는 부동 소수점 숫자
- b 변형 행렬의 (0,1) 요소를 지정하는 부동 소수점 숫자
- c 변형 행렬의 (1,0) 요소를 지정하는 부동 소수점 숫자
- d 변형 행렬의 (1,1) 요소를 지정하는 부동 소수점 숫자

반환값

없음

설명

메서드, 인수에 지정된 행렬을 적용하는 방식으로 현재 선택 영역에서 일반적인 변형을 수행합니다. 자세한 내용은 [element.matrix](#) 속성을 참조하십시오.

예제

다음 예제에서는 선택 항목을 x 방향으로 2의 배율로 늘립니다.

```
fl.getDocumentDOM().transformSelection(2.0, 0.0, 0.0, 1.0);
```

document.unGroup()

지원 버전
Flash MX 2004

사용법
`document.unGroup()`

매개 변수
없음

반환값
없음

설명
메서드, 현재 선택 항목의 그룹을 해제합니다.

예제
다음 예제에서는 현재 선택 항목의 요소를 그룹 해제합니다.

```
fl.getDocumentDOM().unGroup();
```

참고 사항
[document.group\(\)](#)

document.union()

지원 버전
Flash 8

사용법
`document.union()`

매개 변수
없음

반환값
부울 값, 성공적으로 실행되면 `true`를 반환하고 그렇지 않으면 `false`를 반환합니다.

설명
메서드, 선택한 모든 모양을 Drawing 객체로 결합합니다.

예제
다음 예제에서는 선택한 모든 모양을 Drawing 객체로 결합합니다.

```
fl.getDocumentDOM().union();
```

참고 사항

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

document.unlockAllElements()

지원 버전

Flash MX 2004

사용법

```
document.unlockAllElements()
```

매개 변수

없음

반환값

없음

설명

메서드, 현재 선택한 프레임에서 잠겨 있는 모든 요소의 잠금을 해제합니다.

예제

다음 예제에서는 현재 프레임에서 잠겨 있는 모든 객체의 잠금을 해제합니다.

```
fl.getDocumentDOM().unlockAllElements();
```

참고 사항

[element.locked](#)

document.viewMatrix

지원 버전

Flash MX 2004

사용법

```
document.viewMatrix
```

설명

읽기 전용 속성, Matrix 객체입니다. viewMatrix는 문서가 편집 모드에 있을 때 객체 공간을 문서 공간으로 변형하는 데 사용됩니다. 도구에 전달되는 마우스 위치는 현재 편집 중인 객체를 기준으로 합니다. Matrix 객체를 참조하십시오.

예를 들어 심볼을 만들고 편집할 심볼을 두 번 클릭한 다음 [다각형 별] 도구를 사용하여 심볼을 그리면 (0,0) 점이 심볼의 등록 포인트에 위치하게 됩니다. 그러나 drawingLayer 객체에는 문서 공간의 값이 전달되어야 하기 때문에 drawingLayer를 사용하여 (0,0)에서 선을 그리면 스테이지의 왼쪽 위 모서리에서 선이 시작됩니다. viewMatrix 속성을 사용하면 편집 중인 객체 공간을 문서 공간으로 변형할 수 있습니다.

예제

다음 예제에서는 viewMatrix 속성의 값을 가져옵니다.

```
var mat = fl.getDocumentDOM().viewMatrix;
```

document.width

지원 버전

Flash MX 2004

사용법

```
document.width
```

설명

속성, 문서(스태이지)의 너비를 픽셀 단위로 지정하는 정수입니다.

예제

다음 예제에서는 스태이지의 너비를 400픽셀로 설정합니다.

```
fl.getDocumentDOM().width= 400;
```

참고 사항

[document.height](#)

document.xmlPanel()

지원 버전

Flash MX 2004

사용법

```
document.xmlPanel(fileURI)
```

매개 변수

fileURI 패널 내의 컨트롤을 정의하는 XML 파일의 경로를 지정하는 문자열(file:/// URI로 표시됨). 전체 경로가 필요합니다.

반환값

XML 파일에 정의된 모든 컨트롤에 대해 정의된 속성이 있는 객체. 속성은 모두 문자열로 반환됩니다. 반환된 객체에는 문자열 값 "accept" 또는 "cancel"이 포함된 "dismiss"라는 미리 정의된 속성이 하나 있습니다.

설명

메서드, XMLUI 대화 상자를 게시합니다. [fl.xmlui](#)를 참조하십시오.

예제

다음 예제에서는 Test.xml 파일을 로드하고 이 파일에 포함되어 있는 각 속성을 표시합니다.

```
var obj = fl.getDocumentDOM().xmlPanel(fl.configURI + "Commands/Test.xml");  
for (var prop in obj) {  
    fl.trace("property " + prop + " = " + obj[prop]);  
}
```

document.zoomFactor

지원 버전
Flash 8

사용법
document.zoomFactor

설명
속성, 제작 시 스테이지의 확대/축소 퍼센트를 지정합니다. 확대/축소 값 1은 100%, 8은 800%, .5는 50%를 나타내는 방식으로 지정됩니다.

예제
다음 예제에서는 스테이지의 확대/축소 배율을 200%로 설정합니다.

```
fl.getDocumentDOM().zoomFactor = 2;
```

12장: drawingLayer 객체

지원 버전

Flash MX 2004

설명

`drawingLayer` 객체는 `flash` 객체의 자식으로 JavaScript에서 액세스할 수 있습니다. `drawingLayer` 객체는 선택 윤곽을 만들 때처럼 드래그하는 동안 모양을 임시로 그리려고 할 때 확장 가능한 도구에 사용됩니다. 다른 `drawingLayer` 메서드를 호출하려면 먼저 `drawingLayer.beginFrame()`을 호출해야 합니다.

메서드 요약

`drawingLayer` 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
<code>drawingLayer.beginDraw()</code>	Flash를 드로잉 모드로 전환합니다.
<code>drawingLayer.beginFrame()</code>	<code>drawingLayer</code> 를 사용하여 이전에 그린 내용을 지우고 추가 드로잉 명령을 준비합니다.
<code>drawingLayer.cubicCurveTo()</code>	매개 변수를 3차원 선분의 좌표로 사용하여 현재 펜 위치에서 3차원 곡선을 그립니다.
<code>drawingLayer.curveTo()</code>	현재 드로잉 위치에서 지정된 점까지 2차원 곡선을 그립니다.
<code>drawingLayer.drawPath()</code>	지정된 패스를 그립니다.
<code>drawingLayer.endDraw()</code>	드로잉 모드를 종료합니다.
<code>drawingLayer.endFrame()</code>	드로잉 명령 그룹의 끝을 알립니다.
<code>drawingLayer.lineTo()</code>	현재 드로잉 위치에서 (x,y) 점까지 선을 그립니다.
<code>drawingLayer.moveTo()</code>	현재 드로잉 위치를 설정합니다.
<code>drawingLayer.newPath()</code>	새 <code>Path</code> 객체를 반환합니다.
<code>drawingLayer.setColor()</code>	순차적으로 그려진 데이터의 색상을 설정합니다.
<code>drawingLayer.setFill()</code>	이 메서드는 사용할 수 없습니다.
<code>drawingLayer.setStroke()</code>	이 메서드는 사용할 수 없습니다.

drawingLayer.beginDraw()

지원 버전

Flash MX 2004

사용법

```
drawingLayer.beginDraw([persistentDraw])
```

매개 변수

`persistentDraw` 부울 값(선택 사항). `true`로 설정된 경우 `beginDraw()` 또는 `beginFrame()`을 새로 호출할 때까지 마지막 프레임에 있는 드로잉이 스테이지에 남아 있습니다. 이 컨텍스트에서 프레임은 타임라인 프레임이 아닌 드로잉이 시작되고 끝나는 위치를

나타납니다. 예를 들어 사각형을 그릴 때 마우스를 드래그하는 동안 모양의 외곽선을 미리 볼 수 있습니다. 마우스 버튼을 놓은 후에도 이 미리 보기 모양을 계속 표시하려면 `persistentDraw`를 `true`로 설정하면 됩니다.

반환값

없음

설명

메서드, Flash를 드로잉 모드로 전환합니다. 드로잉 모드는 마우스 버튼을 누른 채 모양을 임시로 그릴 때 사용됩니다. 일반적으로 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

예제

다음 예제에서는 Flash를 드로잉 모드로 전환합니다.

```
fl.drawingLayer.beginDraw();
```

drawingLayer.beginFrame()

지원 버전

Flash MX 2004

사용법

```
drawingLayer.beginFrame()
```

매개 변수

없음

반환값

없음

설명

메서드, `drawingLayer`를 사용하여 이전에 그린 내용을 지우고 추가 드로잉 명령을 준비합니다. `drawingLayer.beginDraw()` 다음에 호출해야 합니다. `drawingLayer.beginFrame()`과 `drawingLayer.endFrame()` 사이에 그려진 모든 내용은 다음 `beginFrame()` 및 `endFrame()`을 호출할 때까지 스테이지에 남아 있습니다. 이 컨텍스트에서 프레임은 타임라인 프레임이 아닌 드로잉이 시작되고 끝나는 위치를 나타냅니다. 일반적으로 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다. [drawingLayer.beginDraw\(\)](#)를 참조하십시오.

drawingLayer.cubicCurveTo()

지원 버전

Flash MX 2004

사용법

```
drawingLayer.cubicCurveTo(x1Ctrl, y1Ctrl, x2Ctl, y2Ctl, xEnd, yEnd)
```

매개 변수

x1Ctl 첫 번째 제어점의 x 위치를 지정하는 부동 소수점 값

y1Ctl 첫 번째 제어점의 **y** 위치를 지정하는 부동 소수점 값

x2Ctl 중간 제어점의 **x** 위치를 지정하는 부동 소수점 값

y2Ctl 중간 제어점의 **y** 위치를 지정하는 부동 소수점 값

xEnd 끝 제어점의 **x** 위치를 지정하는 부동 소수점 값

yEnd 끝 제어점의 **y** 위치를 지정하는 부동 소수점 값

반환값

없음

설명

메서드, 매개 변수를 3차원 선분의 좌표로 사용하여 현재 펜 위치에서 3차원 곡선을 그립니다. 일반적으로 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

예제

다음 예제에서는 지정된 제어점을 사용하여 3차원 곡선을 그립니다.

```
fl.drawingLayer.cubicCurveTo(0, 0, 1, 1, 2, 0);
```

drawingLayer.curveTo()

지원 버전

Flash MX 2004

사용법

```
drawingLayer.curveTo(xCtl, yCtl, xEnd, yEnd)
```

매개 변수

xCtl 제어점의 **x** 위치를 지정하는 부동 소수점 값

yCtl 제어점의 **y** 위치를 지정하는 부동 소수점 값

xEnd 끝 제어점의 **x** 위치를 지정하는 부동 소수점 값

yEnd 끝 제어점의 **y** 위치를 지정하는 부동 소수점 값

반환값

없음

설명

메서드, 현재 드로잉 위치에서 지정된 점까지 이차 곡선을 그립니다. 일반적으로 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

예제

다음 예제에서는 지정된 제어점을 사용하여 이차 곡선을 그립니다.

```
fl.drawingLayer.curveTo(0, 0, 2, 0);
```

drawingLayer.drawPath()

지원 버전
Flash MX 2004

사용법
`drawingLayer.drawPath(path)`

매개 변수
path 그릴 [Path 객체](#)

반환값
없음

설명
메서드, **path** 매개 변수에 지정된 패스를 그립니다. 일반적으로 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

예제
다음 예제에서는 `gamePath`라는 `Path` 객체에 지정된 패스를 그립니다.

```
fl.drawingLayer.drawPath(gamePath);
```

drawingLayer.endDraw()

지원 버전
Flash MX 2004

사용법
`drawingLayer.endDraw()`

매개 변수
없음

반환값
없음

설명
메서드, 드로잉 모드를 종료합니다. 드로잉 모드는 마우스 버튼을 누른 채 모양을 임시로 그릴 때 사용됩니다. 일반적으로 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

예제
다음 예제에서는 드로잉 모드를 종료합니다.

```
fl.drawingLayer.endDraw();
```

drawingLayer.endFrame()

지원 버전

Flash MX 2004

사용법

```
drawingLayer.endFrame ()
```

매개 변수

없음

반환값

없음

설명

메서드, 드로잉 명령 그룹의 끝을 알립니다. 드로잉 명령 그룹은 `drawingLayer.beginFrame()`과 `drawingLayer.endFrame()` 사이에 그려진 모든 내용을 나타냅니다. `drawingLayer.beginFrame()`을 다음에 호출하면 이 드로잉 명령 그룹에 그려진 모든 내용이 지워집니다. 일반적으로 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

drawingLayer.lineTo()

지원 버전

Flash MX 2004

사용법

```
drawingLayer.lineTo(x, y)
```

매개 변수

x 그릴 선의 끝점에 대한 x 좌표를 지정하는 부동 소수점 값

y 그릴 선의 끝점에 대한 y 좌표를 지정하는 부동 소수점 값

반환값

없음

설명

메서드, 현재 드로잉 위치에서 (x,y) 점까지 선을 그립니다. 일반적으로 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

예제

다음 예제에서는 현재 드로잉 위치에서 (20,30) 점까지 선을 그립니다.

```
fl.drawingLayer.lineTo(20, 30);
```

drawingLayer.moveTo()

지원 버전

Flash MX 2004

사용법

```
drawingLayer.moveTo(x, y)
```

매개 변수

x 드로잉을 시작할 위치의 **x** 좌표를 지정하는 부동 소수점 값

y 드로잉을 시작할 위치의 **y** 좌표를 지정하는 부동 소수점 값

반환값

없음

설명

메서드, 현재 드로잉 위치를 설정합니다. 일반적으로 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

예제

다음 예제에서는 (10,15) 점에 현재 드로잉 위치를 설정합니다.

```
fl.drawingLayer.moveTo(10, 15);
```

drawingLayer.newPath()

지원 버전

Flash MX 2004

사용법

```
drawingLayer.newPath()
```

매개 변수

없음

반환값

Path 객체

설명

메서드, 새 Path 객체를 반환합니다. 일반적으로 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다. [Path 객체](#)를 참조하십시오.

예제

다음 예제에서는 새 Path 객체를 반환합니다.

```
fl.drawingLayer.newPath();
```


drawingLayer.setColor()

지원 버전

Flash MX 2004

사용법

```
drawingLayer.setColor (color)
```

매개 변수

color 형식이 다음 중 하나인 순차적으로 그려진 데이터의 색상

- "#RRGGBB" 또는 "#RRGGBBAA" 형식의 문자열
- 0xRRGGBB 형식의 16진수
- 16진수 숫자와 같은 십진수를 나타내는 정수

반환값

없음

설명

메서드, 순차적으로 그려진 데이터의 색상을 설정합니다. 영구 데이터에만 적용됩니다. 이 메서드를 사용하려면 drawingLayer.beginDraw()에 전달된 매개 변수를 true로 설정해야 합니다. 일반적으로 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다. drawingLayer.beginDraw()를 참조하십시오.

예제

다음 예제에서는 스테이지에 빨간색 선을 그립니다.

```
f1.drawingLayer.beginDraw( true );  
f1.drawingLayer.beginFrame();  
f1.drawingLayer.setColor( "#ff0000" );  
f1.drawingLayer.moveTo(0,0);  
f1.drawingLayer.lineTo(100,100);  
f1.drawingLayer.endFrame();  
f1.drawingLayer.endDraw();
```

drawingLayer.setFill()

이 메서드는 사용할 수 없습니다.

drawingLayer.setStroke()

이 메서드는 사용할 수 없습니다.

13장: Edge 객체

지원 버전

Flash MX 2004

설명

Edge 객체는 스테이지에 있는 모양의 가장자리를 나타냅니다.

메서드 요약

Edge 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
<code>edge.getControl()</code>	가장자리의 지정된 제어점 위치로 설정된 Point 객체를 가져옵니다.
<code>edge.getHalfEdge()</code>	HalfEdge 객체를 반환합니다.
<code>edge.setControl()</code>	가장자리의 제어점 위치를 설정합니다.
<code>edge.splitEdge()</code>	가장자리를 두 조각으로 분할합니다.

속성 요약

Edge 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>edge.cubicSegmentIndex</code>	가장자리의 삼차 선분에 대한 인덱스 값을 지정하는 정수
<code>edge.id</code>	읽기 전용, 가장자리의 고유한 식별자를 나타내는 정수
<code>edge.isLine</code>	읽기 전용, 0 또는 1의 정수 값
<code>edge.stroke</code>	Stroke 객체

edge.cubicSegmentIndex

지원 버전

Flash CS4 Professional

사용법

```
edge.cubicSegmentIndex
```

설명

읽기 전용 속성, 가장자리의 삼차 선분에 대한 인덱스 값을 지정하는 정수(`shape.getCubicSegmentPoints()` 참조)입니다.

예제

다음 코드에서는 지정된 가장자리에 있는 모든 삼차 선분의 인덱스 값을 표시합니다.

```
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()

지원 버전
Flash MX 2004

사용법
edge.getControl(i)

매개 변수
i 반환할 가장자리의 제어점을 지정하는 정수. 시작 제어점에는 0, 중간 제어점에는 1, 끝 제어점에는 2를 지정합니다. [edge.isLine](#) 속성이 true인 경우 중간 제어점은 시작 및 끝 제어점을 연결하는 선분의 중간 지점으로 설정됩니다.

반환값
지정된 제어점

설명
메서드, 가장자리의 지정된 제어점 위치로 설정된 Point 객체를 가져옵니다.

예제
다음 예제에서는 지정된 모양의 시작 제어점을 pt 변수에 저장합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
var pt = shape.edges[0].getControl(0);
```

edge.getHalfEdge()

지원 버전
Flash MX 2004

사용법
edge.getHalfEdge(index)

매개 변수
index 반환할 반쪽 가장자리를 지정하는 정수. **index**의 값은 첫 번째 반쪽 가장자리를 나타내는 0과 두 번째 반쪽 가장자리를 나타내는 1 중 하나여야 합니다.

반환값
HalfEdge 객체

설명
메서드, [HalfEdge](#) 객체를 반환합니다.

예제

다음 예제에서는 지정된 가장자리의 반쪽 가장자리를 hEdge0 및 hEdge1 변수에 저장합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
var edge = shape.edges[0];  
var hEdge0 = edge.getHalfEdge(0);  
var hEdge1 = edge.getHalfEdge(1);
```

edge.id

지원 버전

Flash MX 2004

사용법

edge.id

설명

읽기 전용 속성, 가장자리의 고유한 식별자를 나타내는 정수입니다.

예제

다음 예제에서는 지정된 가장자리의 고유한 식별자를 my_shape_id 변수에 저장합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
var my_shape_id = shape.edges[0].id;
```

edge.isLine

지원 버전

Flash MX 2004

사용법

edge.isLine

설명

읽기 전용 속성, 0 또는 1의 정수 값입니다. 값이 1이면 가장자리가 직선임을 나타냅니다. 이 경우 중간 제어점에 의해 양쪽 끝 제어점을 연결하는 선이 이등분됩니다.

예제

다음 예제에서는 지정된 가장자리가 직선인지 여부를 확인하고 [출력] 패널에 값 1(직선일 경우) 또는 값 0(직선이 아닐 경우)을 표시합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
fl.trace(shape.edges[0].isLine);
```

edge.setControl()

지원 버전

Flash MX 2004

사용법

```
edge.setControl(index, x, y)
```

매개 변수

index 설정할 제어점을 지정하는 정수. 값 0, 1 및 2를 사용하여 각각 시작, 중간 및 끝 제어점을 지정합니다.

x 제어점의 수평 위치를 지정하는 부동 소수점 값. 스테이지가 편집 또는 제자리에서 편집 모드이면 점 좌표는 편집된 객체를 기준으로 하고 그렇지 않으면 스테이지를 기준으로 합니다.

y 제어점의 수직 위치를 지정하는 부동 소수점 값. 스테이지가 편집 또는 제자리에서 편집 모드이면 점 좌표는 편집된 객체를 기준으로 하고 그렇지 않으면 스테이지를 기준으로 합니다.

반환값

없음

설명

메서드, 가장자리의 제어점 위치를 설정합니다. 이 메서드를 사용하려면 먼저 `shape.beginEdit()`을 호출해야 합니다. [shape.beginEdit\(\)](#)을 참조하십시오.

예제

다음 예제에서는 지정된 가장자리의 시작 제어점을 (0,1) 좌표로 설정합니다.

```
x = 0; y = 1;  
var shape = fl.getDocumentDOM().selection[0];  
shape.beginEdit();  
shape.edges[0].setControl(0, x, y);  
shape.endEdit();
```

edge.splitEdge()

지원 버전

Flash MX 2004

사용법

```
edge.splitEdge(t)
```

매개 변수

t 가장자리를 분할할 위치를 지정하는 0과 1 사이의 부동 소수점 값. 값 0은 한 쪽 끝점을 나타내고 값 1은 다른 쪽 끝점을 나타냅니다. 예를 들어 값 0.5를 지정하면 가장자리가 정확히 한 가운데에서 분할됩니다. 가장자리가 곡선일 경우 0.5는 곡선의 파라메트릭 가운데를 나타냅니다.

반환값

없음

설명

메서드, 가장자리를 두 조각으로 분할합니다. 이 메서드를 사용하려면 먼저 [shape.beginEdit\(\)](#)을 호출해야 합니다.

예제

다음 예제에서는 지정된 가장자리를 반으로 분할합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
shape.beginEdit()  
shape.edges[0].splitEdge( 0.5 );  
shape.endEdit()
```

edge.stroke

지원 버전

Flash CS4 Professional

사용법

`edge.stroke`

설명

속성, [Stroke 객체](#)입니다.

예제

다음 예제에서는 선택한 객체의 첫 번째 가장자리 획 색상을 표시합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
fl.trace(shape.edges[0].stroke.color);
```

14장: Element 객체

지원 버전

Flash MX 2004

설명

스태이지에 표시되는 모든 항목의 유형은 **Element**입니다. 다음은 요소를 선택할 수 있는 코드 예제입니다.

```
var e1 = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];
```

메서드 요약

Element 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
element.getPersistentData()	<i>name</i> 매개 변수에 지정된 데이터의 값을 가져옵니다.
element.getTransformationPoint()	지정된 요소의 변형점 값을 가져옵니다.
element.hasPersistentData()	특정 데이터가 지정된 요소에 첨부되었는지 여부를 나타냅니다.
element.removePersistentData()	객체에 첨부된 특정 이름을 가진 영구 데이터를 제거합니다.
element.setPersistentData()	요소와 함께 데이터를 저장합니다.
element.setTransformationPoint()	요소의 변형점 위치를 설정합니다.

속성 요약

Element 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
element.depth	읽기 전용, 뷰에 있는 객체의 깊이를 나타내는 0보다 큰 정수 값
element.elementType	읽기 전용, 지정된 요소의 유형을 나타내는 문자열
element.height	요소의 높이를 픽셀 단위로 지정하는 부동 소수점 값
element.layer	읽기 전용, 요소가 있는 Layer 객체 를 나타냅니다.
element.left	읽기 전용, 요소의 왼쪽을 나타내는 부동 소수점 값
element.locked	부울 값, 요소가 잠겨 있으면 true를 반환하고 그렇지 않으면 false를 반환합니다.
element.matrix	Matrix 객체 입니다. Matrix에는 a, b, c, d, tx 및 ty 속성이 있고 a, b, c 및 d는 부동 소수점 값이며 tx와 ty는 좌표입니다.
element.name	일반적으로 인스턴스 이름이라고 하는 요소의 이름을 지정하는 문자열
element.rotation	객체의 시계 방향 회전을 각도로 지정하는 -180부터 180 사이의 정수 또는 부동 소수점 값
element.scaleX	심볼, 드로잉 객체 및 프리미티브 사각형 및 타원형의 x 배율 값을 지정하는 부동 소수점 값
element.scaleY	심볼, 드로잉 객체 및 프리미티브 사각형 및 타원형의 y 배율 값을 지정하는 부동 소수점 값
element.selected	요소가 선택되어 있는지 여부를 지정하는 부울 값
element.skewX	심볼, 드로잉 객체 및 프리미티브 사각형 및 타원형의 x 기울기 값을 지정하는 -180부터 180 사이의 부동 소수점 값

속성	설명
<code>element.skewY</code>	심볼, 드로잉 객체 및 프리미티브 사각형 및 타원형의 y 기울기 값을 지정하는 -180부터 180 사이의 부동 소수점 값
<code>element.top</code>	읽기 전용, 요소의 위쪽
<code>element.transformX</code>	상위 요소의 좌표에서 선택한 요소의 변형점에 대한 x 값을 지정하는 부동 소수점 숫자
<code>element.transformY</code>	상위 요소의 좌표에서 선택한 요소의 변형점에 대한 y 값을 지정하는 부동 소수점 숫자
<code>element.width</code>	요소의 폭을 픽셀 단위로 지정하는 부동 소수점 값
<code>element.x</code>	선택된 요소의 등록 포인트에 대한 x 값을 지정하는 부동 소수점 값
<code>element.y</code>	선택된 요소의 등록 포인트에 대한 y 값을 지정하는 부동 소수점 값

element.depth

지원 버전

Flash MX 2004

사용법

`element.depth`

설명

읽기 전용 속성, 뷰에 있는 객체의 깊이를 나타내는 0보다 큰 정수 값입니다. 객체 순서는 스테이지에 그려진 순서에 따라 지정됩니다. 객체 순서는 또한 [수정] > [정렬] 메뉴 항목으로 관리할 수도 있습니다.

예제

다음 예제에서는 지정된 요소의 깊이를 [출력] 패널에 표시합니다.

```
// Select an object and run this script.  
fl.trace("Depth of selected object: " + fl.getDocumentDOM().selection[0].depth);
```

`element.elementType`의 예제를 참조하십시오.

element.elementType

지원 버전

Flash MX 2004

사용법

`element.elementType`

설명

읽기 전용 속성, 지정된 요소의 유형을 나타내는 문자열입니다. 사용할 수 있는 값에는 "shape", "text", "instance" 또는 "shapeObj"가 있습니다. "shapeObj"는 확장 가능한 도구에서 만듭니다.

예제

다음 예제에서는 첫 번째 요소의 유형을 `eType` 변수에 저장합니다.


```
// 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
```

다음 예제에서는 현재 레이어나 프레임에 있는 모든 요소의 다양한 속성을 표시합니다.

```
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()

지원 버전

Flash MX 2004

사용법

```
element.getPersistentData(name)
```

매개 변수

name 반환할 데이터를 식별하는 문자열

반환값

name 매개 변수에 지정된 데이터를 반환하거나 데이터가 존재하지 않을 경우 0을 반환합니다.

설명

메서드, **name** 매개 변수에 지정된 데이터의 값을 가져옵니다. 데이터의 유형은 저장된 데이터의 유형에 따라 달라집니다 ([element.setPersistentData\(\)](#) 참조). 심볼과 비트맵만 영구 데이터를 지원합니다.

예제

다음 예제에서는 지정된 요소에 대한 데이터를 설정하거나 가져오고 [출력] 패널에 해당 값을 표시한 다음 데이터를 제거합니다.

```
// 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()

지원 버전

Flash CS3 Professional

사용법

```
element.getTransformationPoint()
```

매개 변수

없음

반환값

요소의 좌표계 내에서 변형점(또는 원점)의 위치를 지정하는 점(예: {x:10, y:20}, 여기서 x 및 y는 부동 소수점 숫자임)

설명

메서드, 지정된 요소의 변형점 값을 가져옵니다.

변형점은 선택한 항목의 유형에 따라 다양한 위치를 기준으로 합니다. 자세한 내용은 [element.setTransformationPoint\(\)](#)를 참조하십시오.

예제

다음 예제에서는 문서의 첫 번째 레이어에 있는 아홉 번째 프레임의 세 번째 요소에 대한 변형점을 가져옵니다. `transPoint.x` 속성은 변형점의 `x` 좌표를 지정하고 `transPoint.y` 속성은 변형점의 `y` 좌표를 지정합니다.

```
var transPoint =  
fl.getDocumentDOM().getTimeline().layers[0].frames[8].elements[2].getTransformationPoint();
```

참고 사항

[document.getTransformationPoint\(\)](#), [element.setTransformationPoint\(\)](#), [element.transformX](#), [element.transformY](#)

element.hasPersistentData()

지원 버전

Flash MX 2004

사용법

```
element.hasPersistentData(name)
```

매개 변수

`name` 테스트할 데이터 항목의 이름을 지정하는 문자열

반환값

부울 값, 지정된 데이터가 객체에 첨부된 경우 `true`를 반환하고 그렇지 않으면 `false`를 반환합니다.

설명

메서드, 지정된 데이터가 지정된 요소에 첨부되었는지 여부를 나타냅니다. 심볼과 비트맵만 영구 데이터를 지원합니다.

예제

[element.getPersistentData\(\)](#)를 참조하십시오.

element.height

지원 버전

Flash MX 2004

사용법

`element.height`

설명

속성, 요소의 높이를 픽셀 단위로 지정하는 부동 소수점 값입니다.

이 속성을 사용하여 텍스트 필드의 크기를 조절하지 마십시오. 대신 텍스트 필드를 선택하고 [document.setTextRectangle\(\)](#)을 사용하십시오. 텍스트 필드에서 이 속성을 사용하면 텍스트의 크기가 변경됩니다.

예제

다음 예제에서는 지정된 요소의 높이를 100으로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].height = 100;
```

element.layer

지원 버전

Flash 8

사용법

`element.layer`

설명

읽기 전용 속성, 요소가 있는 [Layer 객체](#)를 나타냅니다.

예제

다음 예제에서는 요소가 포함된 Layer 객체를 theLayer 변수에 저장합니다.

```
var theLayer = element.layer;
```

element.left

지원 버전

Flash MX 2004

사용법

`element.left`

설명

읽기 전용 속성, 요소의 왼쪽을 나타내는 부동 소수점 값입니다. `element.left`의 값은 요소가 장면에 있는 경우 스테이지의 왼쪽 위를 기준으로 하고 요소가 심볼 내에 저장된 경우 심볼의 등록 포인트(원점)를 기준으로 합니다. 이 속성을 설정하려면 [document.setSelectionBounds\(\)](#) 또는 [document.moveSelectionBy\(\)](#)를 사용합니다.

예제

다음 예제에서는 요소가 이동할 때 이 속성의 값이 어떻게 변경되는지 보여 줍니다.

```
// 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);
```

[element.elementType](#) 예제를 참조하십시오.

element.locked

지원 버전

Flash MX 2004

사용법

`element.locked`

설명

속성, 요소가 잠겨 있으면 `true`를 반환하고 그렇지 않으면 `false`를 반환하는 부울 값입니다. [element.elementType](#)의 값이 "shape" 이면 이 속성은 무시됩니다.

예제

다음 예제에서는 최상위 레이어의 첫 번째 프레임에 있는 첫 번째 요소를 잠급니다.

```
// Similar to Modify > Arrange > Lock:  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].locked = true;
```

element.matrix

지원 버전

Flash MX 2004

사용법

`element.matrix`

설명

속성, `Matrix` 객체입니다. `Matrix`에는 속성 `a`, `b`, `c`, `d`, `tx` 및 `ty`가 있습니다. `a`, `b`, `c` 및 `d` 속성은 부동 소수점 값이고 `tx` 및 `ty` 속성은 좌표입니다. [Matrix 객체](#)를 참조하십시오.

예제

다음 예제에서는 지정된 요소를 `x` 방향으로 10픽셀, `y` 방향으로 20픽셀만큼 이동합니다.

```
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

지원 버전

Flash MX 2004

사용법

`element.name`

설명

속성, 일반적으로 인스턴스 이름이라고 하는 요소의 이름을 지정하는 문자열입니다. `element.elementType`의 값이 "shape"이면 이 속성은 무시됩니다. [element.elementType](#)을 참조하십시오.

예제

다음 예제에서는 최상위 레이어의 프레임 1에 있는 첫 번째 요소의 인스턴스 이름을 "clip_mc"로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].name = "clip_mc";
```

[element.elementType](#) 예제를 참조하십시오.

element.removePersistentData()

지원 버전

Flash MX 2004

사용법

`element.removePersistentData(name)`

매개 변수

`name` 제거할 데이터의 이름을 지정하는 문자열

반환값

없음

설명

메서드, 객체에 첨부된 특정 이름을 가진 영구 데이터를 제거합니다. 심볼과 비트맵만 영구 데이터를 지원합니다.

예제

[element.getPersistentData\(\)](#)를 참조하십시오.

element.rotation

지원 버전

Flash CS3 Professional

사용법

`element.rotation`

설명

속성, 객체의 시계 방향 회전을 각도로 지정하는 -180부터 180 사이의 정수 또는 부동 소수점 값입니다.

예제

다음 예제에서는 현재 선택된 요소의 회전을 45도로 설정합니다.

```
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

지원 버전

Flash CS3 Professional

사용법

`element.scaleX`

설명

속성, 심볼, 드로잉 객체 및 프리미티브 사각형 및 타원형의 **x** 배율 값을 지정하는 부동 소수점 값입니다. 값 1은 100% 배율을 의미합니다.

예제

다음 예제에서는 현재 선택된 항목의 **x** 배율 값을 2(값의 2배)로 설정합니다.

```
var element = fl.getDocumentDOM().selection[0];
element.scaleX = 2;
```

참고 사항

[element.scaleY](#)

element.scaleY

지원 버전

Flash CS3 Professional

사용법

`element.scaleY`

설명

속성, 심볼, 드로잉 객체 및 프리미티브 사각형 및 타원의 **y** 배율 값을 지정하는 부동 소수점 값입니다. 값 1은 100% 배율을 의미합니다.

예제

다음 예제에서는 현재 선택된 항목의 **y** 배율 값을 2(값의 2배)로 설정합니다.

```
var element = fl.getDocumentDOM().selection[0];  
element.scaleY = 2;
```

참고 사항

[element.scaleX](#)

element.selected

지원 버전

Flash 8

사용법

```
element.selected
```

설명

속성, 요소가 선택되었는지(true), 아니면 선택되지 않았는지(false)를 나타내는 부울 값입니다.

예제

다음 예제에서는 요소를 선택합니다.

```
element.selected = true;
```

element.setPersistentData()

지원 버전

Flash MX 2004

사용법

```
element.setPersistentData(name, type, value)
```

매개 변수

name 데이터와 연관된 이름을 지정하는 문자열. 이 이름은 데이터를 검색하는 데 사용됩니다.

type 데이터의 유형을 정의하는 문자열. 사용할 수 있는 값에는 "integer", "integerArray", "double", "doubleArray", "string" 및 "byteArray"가 있습니다.

value 객체와 연관된 값을 지정합니다. **value**의 데이터 유형은 **type** 매개 변수의 값에 따라 달라집니다. 지정된 값은 **type** 매개 변수에 지정된 데이터 유형에 적합해야 합니다.

반환값

없음

설명

메서드, 요소와 함께 데이터를 저장합니다. 이 데이터는 요소가 포함된 FLA 파일을 다시 열면 사용할 수 있습니다. 심볼과 비트맵만 영구 데이터를 지원합니다.

예제

[element.getPersistentData\(\)](#)를 참조하십시오.

element.setTransformationPoint()

지원 버전

Flash CS3 Professional

사용법

```
element.setTransformationPoint(transformationPoint)
```

매개 변수

transformationPoint 요소 또는 그룹의 변형점에 대한 값을 지정하는 점(예: {x:10, y:20}, 여기서 x 및 y는 부동 소수점 숫자임)

- 모양: **transformationPoint**가 문서를 기준으로 설정됩니다(0,0이 스테이지의 왼쪽 위 모서리임).
- 심볼: **transformationPoint**가 심볼의 등록 포인트를 기준으로 설정됩니다(0,0이 등록 포인트임).
- 텍스트: **transformationPoint**가 텍스트 필드를 기준으로 설정됩니다(0,0이 텍스트 필드의 왼쪽 위 모서리임).
- 비트맵/비디오: **transformationPoint**가 비트맵/비디오를 기준으로 설정됩니다(0,0이 비트맵 또는 비디오의 왼쪽 위 모서리임).
- 드로잉 객체, 프리미티브 객체 및 그룹: **transformationPoint**가 요소나 그룹의 중심을 기준으로 설정됩니다(0,0이 요소 또는 그룹의 중심점임).

반환값

없음

설명

메서드, 요소의 변형점 위치를 설정합니다.

이 메서드는 [document.setTransformationPoint\(\)](#)와 거의 동일하지만 다음과 같은 점이 다릅니다.

- 드로잉 객체, 프리미티브 객체 및 그룹의 변형점이 스테이지 대신 요소 또는 그룹의 중심을 기준으로 설정됩니다.
- 요소를 먼저 선택하지 않고도 요소의 변형점을 설정할 수 있습니다.

이 메서드는 변형점을 이동하지만 요소를 이동하지는 않습니다. 이와는 반대로 [element.transformX](#) 및 [element.transformY](#) 속성은 요소를 이동합니다.

예제

다음 예제에서는 스테이지에 있는 세 번째 요소의 변형점을 100, 200으로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[2].setTransformationPoint({x:100, y:200});
```

참고 사항

[document.setTransformationPoint\(\)](#), [element.getTransformationPoint\(\)](#), [element.transformX](#), [element.transformY](#)

element.skewX

지원 버전

Flash CS3 Professional

사용법

```
element.skewX
```


설명

속성, 심볼, 드로잉 객체 및 프리미티브 사각형 및 타원형의 **x** 기울기 값을 지정하는 -180부터 180 사이의 부동 소수점 값입니다.

예제

다음 예제에서는 현재 선택 항목의 **x** 기울기 값을 10으로 설정합니다.

```
var element = fl.getDocumentDOM().selection[0];  
element.skewX = 10;
```

참고 사항

[document.setTransformationPoint\(\)](#), [element.skewY](#)

element.skewY

지원 버전

Flash CS3 Professional

사용법

```
element.skewY
```

설명

속성, 심볼, 드로잉 객체 및 프리미티브 사각형 및 타원형의 **y** 기울기 값을 지정하는 -180부터 180 사이의 부동 소수점 값입니다.

예제

다음 예제에서는 현재 선택 항목의 **y** 기울기 값을 10으로 설정합니다.

```
var element = fl.getDocumentDOM().selection[0];  
element.skewY = 10;
```

참고 사항

[document.setTransformationPoint\(\)](#), [element.skewX](#)

element.top

지원 버전

Flash MX 2004

사용법

```
element.top
```

설명

읽기 전용 속성, 요소의 위쪽입니다. `element.top`의 값은 장면에 있는 요소의 경우 스테이지의 왼쪽 위를 기준으로 하고 심볼 내에 저장된 요소의 경우 심볼의 등록 포인트를 기준으로 합니다. 이 속성을 설정하려면 [document.setSelectionBounds\(\)](#) 또는 [document.moveSelectionBy\(\)](#)를 사용합니다.

예제

다음 예제에서는 요소가 이동할 때 이 속성의 값이 어떻게 변경되는지 보여 줍니다.

```
// 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);
```

[element.elementType](#) 예제를 참조하십시오.

element.transformX

지원 버전

Flash CS3 Professional

사용법

```
element.transformX
```

설명

속성, 상위 요소의 좌표계에서 선택한 요소의 변형점에 대한 **x** 값을 지정하는 부동 소수점 숫자입니다. 이 속성을 새 값으로 설정하면 요소가 이동합니다. 이와 반대로 [element.setTransformationPoint\(\)](#) 메서드는 변형점을 이동하지만 요소를 이동하지는 않습니다.

예제

참고 사항

[element.getTransformationPoint\(\)](#), [element.setTransformationPoint\(\)](#), [element.transformY](#)

element.transformY

지원 버전

Flash CS3 Professional

사용법

```
element.transformY
```

설명

속성, 상위 요소의 좌표계에서 선택한 요소의 변형점에 대한 **y** 값을 지정하는 부동 소수점 숫자입니다. 이 속성을 새 값으로 설정하면 요소가 이동합니다. 이와 반대로 [element.setTransformationPoint\(\)](#) 메서드는 변형점을 이동하지만 요소를 이동하지는 않습니다.

참고 사항

[element.getTransformationPoint\(\)](#), [element.setTransformationPoint\(\)](#), [element.transformX](#)

element.width

지원 버전

Flash MX 2004

사용법

`element.width`

설명

속성, 요소의 폭을 픽셀 단위로 지정하는 부동 소수점 값입니다.

이 속성을 사용하여 텍스트 필드의 크기를 조절하지 마십시오. 대신 텍스트 필드를 선택하고 [document.setTextRectangle\(\)](#)을 사용하십시오. 텍스트 필드에서 이 속성을 사용하면 텍스트의 크기가 변경됩니다.

예제

다음 예제에서는 지정된 요소의 폭을 100으로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].width= 100;
```

element.x

지원 버전

Flash CS3 Professional

사용법

`element.x`

설명

속성, 선택된 요소의 등록 포인트에 대한 x 값을 지정하는 부동 소수점 값입니다.

예제

다음 예제에서는 지정된 요소의 등록 포인트 값을 100, 200으로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].x= 100;  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].y= 200;
```

참고 사항

[element.y](#)

element.y

지원 버전

Flash CS3 Professional

사용법

`element.y`

설명

속성, 선택된 요소의 등록 포인트에 대한 y 값을 지정하는 부동 소수점 값입니다.

예제

[element.x](#)를 참조하십시오.

15장: Fill 객체

지원 버전

Flash MX 2004

설명

이 객체에는 [도구] 패널 또는 선택한 모양의 채움 색상 설정에 대한 모든 속성이 포함되어 있습니다. Fill 객체를 가져오려면 `document.getCustomFill()`을 사용합니다.

속성 요약

Fill 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>fill.bitmapsClipped</code>	비트맵보다 큰 모양의 비트맵 채우기가 잘리는지 아니면 반복되는지를 지정하는 부울 값
<code>fill.bitmapPath</code>	라이브러리에 있는 비트맵 채우기의 경로와 이름을 지정하는 문자열
<code>fill.color</code>	채움 색상을 나타내는 문자열, 16진수 값 또는 정수
<code>fill.colorArray</code>	그라디언트의 색상 배열
<code>fill.focalPoint</code>	변형점을 기준으로 하는 그라디언트 초점의 수평 오프셋을 지정합니다.
<code>fill.linearRGB</code>	채우기를 선형 또는 방사형 RGB 그라디언트 중 어느 것으로 렌더링할지를 지정하는 부울 값
<code>fill.matrix</code>	그라디언트 채우기의 배치, 방향 및 크기를 정의하는 Matrix 객체
<code>fill.overflow</code>	그라디언트 오버플로의 비헤이비어를 지정하는 문자열
<code>fill.posArray</code>	해당 색상의 위치를 나타내고 범위가 각각 0에서 255까지인 정수 배열
<code>fill.style</code>	채우기 스타일을 지정하는 문자열

fill.bitmapsClipped

지원 버전

Flash CS4 Professional

사용법

```
fill.bitmapsClipped
```

설명

속성, 비트맵보다 큰 모양의 비트맵 채우기가 잘리는지(`true`), 아니면 반복되는지(`false`)를 지정하는 부울 값입니다. 이 속성은 `fill.style` 속성의 값이 "bitmap"인 경우에만 사용할 수 있습니다. 모양이 비트맵보다 작으면 이 값은 `false`입니다.

예제

다음 예제에서는 해당되는 경우 비트맵 채우기가 잘리는지 여부에 대한 정보를 [출력] 패널에 표시합니다.

```
var fill = fl.getDocumentDOM().getCustomFill();
if (fill.style == "bitmap")
    fl.trace("Fill image is clipped: " + fill.bitmapsClipped);
```

참고 사항

[fill.bitmapPath](#)

fill.bitmapPath

지원 버전

Flash CS4 Professional

사용법

`fill.bitmapPath`

설명

속성, 라이브러리에 있는 비트맵 채우기의 경로와 이름을 지정하는 문자열입니다. 이 속성은 [fill.style](#) 속성의 값이 "bitmap"인 경우에만 사용할 수 있습니다.

예제

다음 예제에서는 지정된 항목의 채우기 스타일을 라이브러리에 있는 비트맵 이미지로 설정합니다.

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.style = "bitmap";  
fill.bitmapPath = "myBitmap.jpg";  
fl.getDocumentDOM().setCustomFill(fill);
```

참고 사항

[fill.bitmapIsClipped](#)

fill.color

지원 버전

Flash MX 2004

사용법

`fill.color`

설명

속성, 형식이 다음 중 하나인 채우기 색상입니다.

- "#RRGGBB" 또는 "#RRGGBBAA" 형식의 문자열
- 0xRRGGBB 형식의 16진수
- 16진수 숫자와 같은 십진수를 나타내는 정수

예제

다음 예제에서는 현재 선택 항목의 채움 색상을 설정합니다.

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.color = "#FFFFFF";  
fl.getDocumentDOM().setCustomFill( fill );
```

fill.colorArray

지원 버전
Flash MX 2004

사용법
fill.colorArray

설명
속성, 정수로 나타내는 그래디언트의 색상 배열입니다. 이 속성은 fill.style 속성의 값이 "radialGradient" 또는 "linearGradient"인 경우에만 사용할 수 있습니다. fill.style을 참조하십시오.

예제
다음 예제에서는 해당되는 경우 현재 선택 항목의 색상 배열을 [출력] 패널에 표시합니다.

```
var fill = fl.getDocumentDOM().getCustomFill();  
if(fill.style == "linearGradient" || fill.style == "radialGradient")  
    alert(fill.colorArray);
```

다음 예제에서는 채움을 지정된 선형 그래디언트로 설정합니다.

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.style = "linearGradient";  
fill.colorArray = ["#00ff00", "#ff00ff"];  
fill.posArray = [0, 255];  
fl.getDocumentDOM().setCustomFill(fill);
```

fill.focalPoint

지원 버전
Flash 8

사용법
fill.focalPoint

설명
속성, 변형점을 기준으로 하는 그래디언트 초점의 수평 오프셋을 지정하는 정수입니다. 예를 들어, 이 값이 10이면 변형점에서 그래디언트 가장자리까지 거리의 10/255에 해당하는 위치에 초점이 설정되고 이 값이 -255면 그래디언트의 왼쪽 경계에 초점이 설정됩니다. 기본값은 0입니다.

이 속성은 fill.style 속성의 값이 "radialGradient"인 경우에만 사용할 수 있습니다.

예제
다음 예제에서는 모양의 중심에서 오른쪽으로 100픽셀 떨어진 위치에 현재 선택 항목에 대한 방사형 그래디언트의 초점을 설정합니다.

```
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

지원 버전

Flash 8

사용법

`fill.linearRGB`

설명

속성, 채우기를 선형 또는 방사형 RGB 그라디언트 중 어느 것으로 렌더링할지를 지정하는 부울 값입니다. 그라디언트의 선형 보간을 지정하려면 `true`로 설정하고 그라디언트의 방사형 보간을 지정하려면 `false`로 설정합니다. 기본값은 `false`입니다.

예제

다음 예제에서는 현재 선택 항목의 그라디언트를 선형 RGB로 렌더링하도록 지정합니다.

```
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

지원 버전

Flash MX 2004

사용법

`fill.matrix`

설명

속성, 그라디언트 채우기의 배치, 방향 및 크기를 정의하는 [Matrix 객체](#)입니다.

예제

다음 예제에서는 `fill.matrix` 속성을 사용하여 현재 선택 항목에 대한 그라디언트 채우기를 지정합니다.


```
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

지원 버전
Flash 8

사용법

```
fill.overflow
```

설명

속성, 그래디언트 오버플로의 비헤이비어를 지정하는 문자열입니다. 사용할 수 있는 값에는 "extend", "repeat" 및 "reflect"가 있으며 문자열은 대/소문자를 구분하지 않습니다. 기본값은 "extend"입니다.

예제

다음 예제에서는 현재 선택 항목에 대한 오버플로의 비헤이비어가 "extend"이도록 지정합니다.

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.overflow = "extend";
fl.getDocumentDOM().setCustomFill(fill);
```

fill.posArray

지원 버전
Flash MX 2004

사용법

```
fill.posArray
```

설명

속성, 해당 색상의 위치를 나타내고 범위가 각각 0에서 255까지인 정수 배열입니다. 이 속성은 [fill.style](#) 속성의 값이 "radialGradient" 또는 "linearGradient"인 경우에만 사용할 수 있습니다.

예제

다음 예제에서는 현재 선택 항목의 선형 그래디언트에 사용할 색상을 지정합니다.

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.style = "linearGradient";  
fill.colorArray = [ 0x00ff00, 0xff0000, 0x0000ff ];  
fill.posArray= [0,100, 200];  
fl.getDocumentDOM().setCustomFill( fill );
```

fill.style

지원 버전

Flash MX 2004. "bitmap" 값이 Flash CS4 Professional에서 추가됨

사용법

`fill.style`

설명

속성, 채우기 스타일을 지정하는 문자열입니다. 사용할 수 있는 값에는 "bitmap", "solid", "linearGradient", "radialGradient" 및 "noFill"이 있습니다.

이 값이 "linearGradient" 또는 "radialGradient"이면 `fill.colorArray` 및 `fill.posArray` 속성도 사용할 수 있습니다. 이 값이 "bitmap"이면 `fill.bitmapIsClipped` 및 `fill.bitmapPath` 속성도 사용할 수 있습니다.

예제

다음 예제에서는 현재 선택 항목의 선형 그래디언트에 사용할 색상을 지정합니다.

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.style= "linearGradient";  
fill.colorArray = [ 0x00ff00, 0xff0000, 0x0000ff ];  
fill.posArray= [0,100, 200];  
fl.getDocumentDOM().setCustomFill( fill );
```

16장: Filter 객체

지원 버전
Flash 8

설명

이 객체에는 모든 필터에 대한 속성이 모두 들어 있습니다. `filter.name` 속성은 필터의 유형을 지정하고 각 필터에 적용할 수 있는 속성을 결정합니다. `filter.name`을 참조하십시오.

객체에 대한 필터 목록을 반환하려면 `document.getFilters()`를 사용합니다. 객체에 필터를 적용하려면 `document.setFilters()`를 사용합니다. `document.getFilters()` 및 `document.setFilters()`를 참조하십시오.

속성 요약

Filter 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>filter.angle</code>	그림자 또는 강조 색상 적용 각도를 도 단위로 지정하는 부동 소수점 값
<code>filter.blurX</code>	x 방향으로 흐리게 표시할 양을 픽셀 단위로 지정하는 부동 소수점 값
<code>filter.blurY</code>	y 방향으로 흐리게 표시할 양을 픽셀 단위로 지정하는 부동 소수점 값
<code>filter.brightness</code>	필터의 밝기를 지정하는 부동 소수점 값
<code>filter.color</code>	필터 색상을 나타내는 문자열, 16진수 값 또는 정수
<code>filter.contrast</code>	필터의 대비 값을 지정하는 부동 소수점 값
<code>filter.distance</code>	필터 효과와 객체 간의 거리를 픽셀 단위로 지정하는 부동 소수점 값
<code>filter.enabled</code>	지정된 필터의 활성화 여부를 지정하는 부울 값
<code>filter.hideObject</code>	소스 이미지를 숨길지 여부를 지정하는 부울 값
<code>filter.highlightColor</code>	강조 색상을 나타내는 문자열, 16진수 값 또는 정수
<code>filter.hue</code>	필터의 색조를 지정하는 부동 소수점 값
<code>filter.inner</code>	그림자가 내부 그림자인지 여부를 지정하는 부울 값
<code>filter.knockout</code>	필터가 녹아웃 필터인지 지정하는 부울 값
<code>filter.name</code>	읽기 전용, 필터의 유형을 지정하는 문자열
<code>filter.quality</code>	흐림 품질을 지정하는 문자열
<code>filter.saturation</code>	필터의 채도 값을 지정하는 부동 소수점 값
<code>filter.shadowColor</code>	그림자 색상을 나타내는 문자열, 16진수 값 또는 정수
<code>filter.strength</code>	필터의 백분율 강도를 지정하는 정수
<code>filter.type</code>	경사 또는 광선의 유형을 지정하는 문자열

filter.angle

지원 버전

Flash 8

사용법

`filter.angle`

설명

속성, 그림자 또는 강조 색상 적용 각도를 도 단위로 지정하는 부동 소수점 값입니다. 사용할 수 있는 값은 0에서 360 사이입니다. 이 속성은 `filter.name` 속성 값이 "bevelFilter", "dropShadowFilter", "gradientBevelFilter" 또는 "gradientGlowFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체의 경사 필터에 대해 각도를 120으로 설정합니다.

```
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);
```

참고 사항

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

filter.blurX

지원 버전

Flash 8

사용법

`filter.blurX`

설명

속성, **x** 방향으로 흐리게 표시할 양을 픽셀 단위로 지정하는 부동 소수점 값입니다. 사용할 수 있는 값은 0에서 255 사이입니다. 이 속성은 `filter.name` 속성 값이 "bevelFilter", "blurFilter", "dropShadowFilter", "glowFilter", "gradientBevelFilter" `filter.name` "gradientGlowFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체에서 흐림 필터에 대해 blurX 값을 30으로 설정하고 blurY 값을 20으로 설정합니다.

```
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);
```

참고 사항

[document.setFilterProperty\(\)](#), [filter.blurY](#)

filter.blurY

지원 버전

Flash 8

사용법

`filter.blurY`

설명

속성, y 방향으로 흐리게 표시할 양을 픽셀 단위로 지정하는 부동 소수점 값입니다. 사용할 수 있는 값은 0에서 255 사이입니다. 이 속성은 `filter.name` 속성 값이 "bevelFilter", "blurFilter", "dropShadowFilter", "glowFilter", "gradientBevelFilter"`filter.name`"gradientGlowFilter"인 Filter 객체에 대해 정의됩니다.

예제

[filter.blurX](#)를 참조하십시오.

참고 사항

[document.setFilterProperty\(\)](#), [filter.blurX](#)

filter.brightness

지원 버전

Flash 8

사용법

`filter.brightness`

설명

속성, 필터의 밝기를 지정하는 부동 소수점 값입니다. 사용할 수 있는 값은 -100에서 100 사이입니다. 이 속성은 [filter.name](#) 속성 값이 "adjustColorFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체에서 색상 조정 필터에 대해 밝기를 30.5로 설정합니다.

```
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

지원 버전

Flash 8

사용법

`filter.color`

설명

속성, 형식이 다음 중 하나인 필터의 색상입니다.

- "#RRGGBB" 또는 "#RRGGBBAA" 형식의 문자열
- 0xRRGGBB 형식의 16진수
- 16진수 숫자와 같은 십진수를 나타내는 정수

이 속성은 `filter.name` 속성 값이 "dropShadowFilter" 또는 "glowFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체에서 그림자 필터에 대해 색상을 "#ff0003e"로 설정합니다.

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'dropShadowFilter'){
        myFilters[i].color = '#ff0003e';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

참고 사항

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

filter.contrast

지원 버전

Flash 8

사용법

`filter.contrast`

설명

속성, 필터의 대비 값을 지정하는 부동 소수점 값입니다. 사용할 수 있는 값은 -100에서 100 사이입니다. 이 속성은 `filter.name` 속성 값이 "adjustColorFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체에서 색상 조정 필터에 대해 대비 값을 -15.5로 설정합니다.

```
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

지원 버전
Flash 8

사용법

`filter.distance`

설명

속성, 필터 효과와 객체 간의 거리를 픽셀 단위로 지정하는 부동 소수점 값입니다. 사용할 수 있는 값은 -255부터 255까지입니다. 이 속성은 `filter.name` 속성 값이 "bevelFilter", "dropShadowFilter", "gradientBevelFilter" 또는 "gradientGlowFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체에서 그림자 필터에 대해 거리를 10픽셀로 설정합니다.

```
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);
```

참고 사항

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

filter.enabled

지원 버전
Flash CS3 Professional

사용법

`filter.enabled`

설명

속성, 필터가 활성화되어 있는지(true), 아니면 비활성화되어 있는지(false)를 지정하는 부울 값입니다.

예제

다음 예제에서는 선택한 객체의 색상 필터를 비활성화합니다.

```
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

지원 버전
Flash 8

사용법
filter.hideObject

설명
속성, 소스 이미지를 숨길지(true), 아니면 표시할지(false)를 지정하는 부울 값입니다. 이 속성은 [filter.name](#) 속성 값이 "dropShadowFilter"인 Filter 객체에 대해 정의됩니다.

예제
다음 예제에서는 선택한 객체의 그림자 필터에 대해 hideObject 값을 true로 설정합니다.

```
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

지원 버전
Flash 8

사용법
filter.highlightColor

설명
속성, 형식이 다음 중 하나인 강조 색상입니다.

- "#RRGGBB" 또는 "#RRGGBBAA" 형식의 문자열
- 0xRRGGBB 형식의 16진수
- 16진수 숫자와 같은 십진수를 나타내는 정수

이 속성은 [filter.name](#) 속성 값이 "bevelFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체의 경사 필터에 대해 강조 색상을 "#ff00003e"로 설정합니다.

```
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

지원 버전

Flash 8

사용법

filter.hue

설명

속성, 필터의 색조를 지정하는 부동 소수점 값입니다. 사용할 수 있는 값은 -180에서 180 사이입니다. 이 속성은 [filter.name](#) 속성 값이 "adjustColorFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체의 색상 조정 필터에 대해 색조를 120으로 설정합니다.

```
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

지원 버전

Flash 8

사용법

filter.inner

설명

속성, 그림자가 내부 그림자인지(true), 아니면 내부 그림자가 아닌지(false)를 지정하는 부울 값입니다. 이 속성은 [filter.name](#) 속성 값이 "dropShadowFilter" 또는 "glowFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체의 광선 필터에 대해 inner 속성의 값을 true로 설정합니다.

```
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);
```

참고 사항

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

filter.knockout

지원 버전

Flash 8

사용법

`filter.knockout`

설명

속성, 필터가 녹아웃 필터인지(true), 아니면 녹아웃 필터가 아닌지(false)를 지정하는 부울 값입니다. 이 속성은 [filter.name](#) 속성 값이 "bevelFilter", "dropShadowFilter", "glowFilter", "gradientBevelFilter" 또는 "gradientGlowFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체의 광선 필터에 대해 knockout 속성을 true로 설정합니다.

```
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);
```

참고 사항

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

filter.name

지원 버전

Flash 8

사용법

`filter.name`

설명

읽기 전용 속성, 필터의 유형을 지정하는 문자열입니다. 이 속성의 값은 사용할 수 있는 **Filter** 객체의 기타 속성을 결정합니다. 사용할 수 있는 값에는 "adjustColorFilter", "bevelFilter", "blurFilter", "dropShadowFilter", "glowFilter", "gradientBevelFilter" 또는 "gradientGlowFilter"가 있습니다.

예제

다음 예제에서는 [출력] 패널에 필터 이름과 인덱스 위치를 표시합니다.

```
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);
```

참고 사항

[document.getFilters\(\)](#), [document.setFilterProperty\(\)](#)

filter.quality

지원 버전

Flash 8

사용법

filter.quality

설명

속성, 흐림 품질을 지정하는 문자열입니다. 사용할 수 있는 값에는 "low", "medium" 및 "high" 가 있습니다. 여기서 "high"는 가우시안 흐림과 비슷합니다. 이 속성은 [filter.name](#) 속성 값이 "bevelFilter", "blurFilter", "dropShadowFilter", "glowFilter", "gradientGlowFilter" 또는 "gradientBevelFilter"인 **Filter** 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체의 광선 필터에 대해 흐림 품질을 "medium"으로 설정합니다.

```
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);
```

참고 사항

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

filter.saturation

지원 버전

Flash 8

사용법

`filter.saturation`

설명

속성, 필터의 채도 값을 지정하는 부동 소수점 값입니다. 사용할 수 있는 값은 -100부터 100까지입니다. 이 속성은 `filter.name` 속성 값이 "adjustColorFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체에서 색상 조정 필터에 대해 채도 값을 -100(회색 음영)으로 설정합니다.

```
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);
```

참고 사항

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

filter.shadowColor

지원 버전

Flash 8

사용법

`filter.shadowColor`

설명

속성, 형식이 다음 중 하나인 그림자 색상입니다.

- "#RRGGBB" 또는 "#RRGGBBAA" 형식의 문자열
- 0xRRGGBB 형식의 16진수
- 16진수 숫자와 같은 십진수를 나타내는 정수

이 속성은 `filter.name` 속성 값이 "bevelFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체의 경사 필터에 대해 그림자 색상을 "#ff00003e"로 설정합니다.

```
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);
```

참고 사항

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

filter.strength

지원 버전

Flash 8

사용법

`filter.strength`

설명

속성, 필터의 백분율 강도를 지정하는 정수입니다. 사용할 수 있는 값은 0에서 25,500 사이입니다. 이 속성은 `filter.name` 속성 값이 "bevelFilter", "dropShadowFilter", "glowFilter", "gradientGlowFilter" 또는 "gradientBevelFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체의 광선 필터에 대해 강도를 50으로 설정합니다.

```
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);
```

참고 사항

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

filter.type

지원 버전

Flash 8

사용법

`filter.type`

설명

속성, 경사 또는 광선의 유형을 지정하는 문자열입니다. 사용할 수 있는 값에는 "inner", "outer" 및 "full"이 있습니다. 이 속성은 `filter.name` 속성 값이 "bevelFilter", "gradientGlowFilter" 또는 "gradientBevelFilter"인 Filter 객체에 대해 정의됩니다.

예제

다음 예제에서는 선택한 객체의 모든 경사 필터에 대해 유형을 "full"로 설정합니다.

```
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);
```

참고 사항

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

17장: Flash 객체(fl)

지원 버전

Flash MX 2004

설명

flash 객체는 Flash 응용 프로그램을 나타냅니다. 이 객체는 flash 또는 fl을 사용하여 나타낼 수 있습니다. 이 문서의 코드 샘플 전반에서는 fl이 사용됩니다.

메서드 요약

flash 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
fl.addEventListener()	특정 이벤트가 수신될 때 호출할 함수를 등록합니다.
fl.browseForFileURL()	[파일 열기] 또는 [파일 저장] 시스템 대화 상자를 열고, 열거나 저장할 파일을 지정할 수 있도록 합니다.
fl.browseForFolderURL()	[Browse For Folder] 대화 상자를 표시하고 사용자가 폴더를 선택할 수 있도록 합니다.
212페이지의 " fl.clearPublishCache() "	게시 캐시를 지웁니다.
fl.clipCopyString()	지정된 문자열을 클립보드에 복사합니다.
fl.closeAll()	열려 있는 모든 파일을 닫고 이전에 저장하지 않은 문서가 있으면 [다른 이름으로 저장] 대화 상자를 표시합니다.
fl.closeAllPlayerDocuments()	[컨트롤] > [무비 테스트]를 통해 열려 있던 모든 SWF 파일을 닫습니다.
fl.closeDocument()	지정된 문서를 닫습니다.
fl.createDocument()	새 문서를 열고 선택합니다.
220페이지의 " fl.exportPublishProfileString() "	제작 설정을 내보내는 URI(Uniform Resource Identifier)입니다.
fl.fileExists()	디스크에 파일이 이미 있는지 여부를 확인합니다.
fl.findDocumentDOM()	특정 파일을 해당 파일의 고유한 식별자를 사용하여 대상으로 지정할 수 있게 합니다.
fl.findDocumentIndex()	문서 위치를 fl.documents 배열로 나타내는 정수 배열을 반환합니다.
fl.findObjectInDocByName()	지정된 텍스트와 일치하는 인스턴스 이름을 가진 요소를 표시합니다.
fl.findObjectInDocByType()	지정된 요소 유형의 요소를 문서에 표시합니다.
fl.getAppMemoryInfo()	Flash.exe 메모리의 지정된 영역에 사용되고 있는 바이트 수를 나타내는 정수를 반환합니다.
fl.getDocumentDOM()	현재 활성화된 문서의 DOM(Document 객체)을 가져옵니다.
fl.getSwfPanel()	패널의 지역화된 이름 또는 SWF 파일 이름을 기준으로 SWFPanel 객체를 반환합니다.
fl.isFontInstalled()	지정된 글꼴이 설치되어 있는지 여부를 확인합니다.
fl.mapPlayerURL()	이스케이프된 유니코드 URL을 UTF-8 또는 MBCS URL에 매핑합니다.

메서드	설명
<code>fl.openDocument()</code>	새 Flash 문서 윈도우에서 Flash(FLA) 문서를 편집하기 위해 열고 이 문서에 포커스를 둡니다.
<code>fl.openScript()</code>	Flash 텍스트 편집기에서 스크립트(JSFL, AS, ASC) 또는 기타 파일(XML, TXT)을 엽니다.
<code>fl.quit()</code>	Flash를 종료할 때 변경된 문서를 저장하라는 대화 상자를 표시합니다.
<code>fl.reloadTools()</code>	<code>toolconfig.xml</code> 파일에서 [도구] 패널을 다시 구성합니다. 확장 가능한 도구를 만들 때만 사용됩니다.
<code>fl.removeEventListener()</code>	<code>fl.addEventListener()</code> 를 사용하여 등록된 함수의 등록을 해제합니다.
<code>fl.resetAS3PackagePaths()</code>	[ActionScript 3.0 설정] 대화 상자의 전역 클래스 경로 설정을 기본값으로 재설정합니다.
<code>fl.resetPackagePaths()</code>	[ActionScript 2.0 설정] 대화 상자의 전역 클래스 경로 설정을 기본값으로 재설정합니다.
<code>fl.runScript()</code>	JavaScript 파일을 실행합니다.
<code>fl.saveAll()</code>	열려 있는 모든 파일을 저장하고 이전에 저장하지 않은 문서가 있으면 [다른 이름으로 저장] 대화 상자를 표시합니다.
<code>fl.saveDocument()</code>	지정된 문서를 FLA 문서로 저장합니다.
<code>fl.saveDocumentAs()</code>	지정된 문서를 저장하기 위한 [다른 이름으로 저장] 대화 상자를 표시합니다.
<code>fl.selectElement()</code>	요소의 선택 또는 편집을 활성화합니다.
<code>fl.selectTool()</code>	[도구] 패널에서 지정된 도구를 선택합니다.
<code>fl.setActiveWindow()</code>	활성 윈도우를 지정된 문서가 되도록 설정합니다.
<code>fl.showIdleMessage()</code>	너무 오래 실행되는 스크립트에 대한 경고 메시지가 표시되지 않도록 합니다.
<code>fl.toggleBreakpoint()</code>	지정된 행에서 지정된 .as 파일의 중단점을 전환합니다.
<code>fl.trace()</code>	텍스트 문자열을 [출력] 패널에 보냅니다.

속성 요약

Flash 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>fl.actionsPanel</code>	읽기 전용, <code>actionsPanel</code> 객체
<code>fl.as3PackagePaths</code>	[ActionScript 3.0 설정] 대화 상자의 전역 클래스 경로 설정에 해당하는 문자열
<code>fl.compilerErrors</code>	읽기 전용, <code>compilerErrors</code> 객체
<code>fl.componentsPanel</code>	읽기 전용, [구성 요소] 패널을 나타내는 <code>componentsPanel</code> 객체
<code>fl.configDirectory</code>	읽기 전용, 로컬 사용자의 Configuration 폴더에 대한 전체 경로를 플랫폼별 경로로 지정하는 문자열
<code>fl.configURI</code>	읽기 전용, 로컬 사용자의 Configuration 디렉토리에 대한 전체 경로를 <code>file:/// URI</code> 로 지정하는 문자열
<code>fl.contactSensitiveSelection</code>	[연결 감지] 선택 모드의 활성 여부를 지정하는 부울 값
<code>fl.createNewDocList</code>	읽기 전용, 만들 수 있는 문서의 다양한 유형을 나타내는 문자열 배열

속성	설명
fl.createNewDocListType	읽기 전용, 만들 수 있는 문서 유형의 파일 확장명을 나타내는 문자열 배열
fl.createNewTemplateList	읽기 전용, 만들 수 있는 템플릿의 다양한 유형을 나타내는 문자열 배열
fl.documents	읽기 전용, 편집하려고 현재 열어 놓은 문서(FLA 파일)를 나타내는 Document 객체 (Document 객체 참조)의 배열
fl.drawingLayer	읽기 전용, 드래그하는 동안 모양을 임시로 그리려고 할 때 확장 가능한 도구에서 사용해야 하는 drawingLayer 객체
fl.externalLibraryPath	런타임 공유 라이브러리로 사용되는 SWC 파일의 위치를 지정하는 전역 ActionScript 3.0 외부 라이브러리 경로에 있는 항목의 목록이 포함된 문자열
fl.flexSDKPath	bin, frameworks, lib 및 기타 폴더가 포함된 Flex SDK 폴더의 경로를 지정하는 문자열
fl.installedPlayers	문서 속성 관리자에서 설치된 Flash Player 목록에 해당하는 일반 객체의 배열을 반환합니다.
fl.languageCode	응용 프로그램의 사용자 인터페이스의 로케를 식별하는 다섯 자리 문자 코드를 반환합니다.
fl.libraryPath	SWC 파일이 포함된 폴더나 SWC 파일의 위치를 지정하는 전역 ActionScript 3.0 라이브러리 경로에 있는 항목의 목록이 포함된 문자열
fl.Math	읽기 전용, 행렬 및 소수점 연산을 위한 메서드를 제공하는 Math 객체
fl.mruRecentFileList	읽기 전용, Flash 제작 도구에서 관리하는 가장 최근에 사용한 문서(MRU) 목록에 있는 전체 파일 이름의 배열
fl.mruRecentFileListType	읽기 전용, Flash 제작 도구에서 관리하는 MRU 목록에 있는 파일 유형의 배열
fl.packagePaths	[ActionScript 2.0 설정] 대화 상자의 전역 클래스 경로 설정에 해당하는 문자열
fl.publishCacheDiskSizeMax	디스크 캐시 크기 제한 환경 설정을 구성하는 정수입니다.
fl.publishCacheEnabled	게시 캐시가 사용되는지 여부를 설정하는 부울 값입니다.
fl.publishCacheMemoryEntrySizeLimit	메모리 캐시 항목 환경 설정의 최대 크기를 설정하는 정수 속성입니다.
fl.publishCacheMemorySizeMax	메모리 캐시 크기 제한 환경 설정을 구성하는 정수입니다.
fl.objectDrawingMode	활성화된 객체 드로잉 모드를 나타내는 정수
fl.outputPanel	읽기 전용, outputPanel 객체에 대한 참조
fl.presetPanel	읽기 전용, presetPanel 객체
fl.scriptURI	읽기 전용, 현재 실행 중인 JSFL 스크립트의 경로를 나타내는 문자열(file:///URI로 표시됨)
fl.sourcePath	ActionScript 클래스 파일의 위치를 지정하는 전역 ActionScript 3.0 소스 경로에 있는 항목의 목록이 포함된 문자열
fl.swfPanels	등록된 swfPanel 객체의 배열(swfPanel 객체 참조)
fl.tools	읽기 전용, Tools 객체의 배열
fl.version	읽기 전용, 플랫폼을 포함하는 Flash 제작 도구의 긴 문자열 버전
fl.xmlui	읽기 전용, XMLUI 객체

fl.actionsPanel

지원 버전

Flash CS3 Professional

사용법

```
fl.actionsPanel
```

설명

읽기 전용 속성, 현재 표시된 [액션] 패널을 나타내는 actionsPanel 객체입니다. 이 속성에 대한 자세한 내용은 [actionsPanel 객체](#)를 참조하십시오.

fl.addEventListener()

지원 버전

Flash CS3 Professional

사용법

```
fl.addEventListener(eventType, callbackFunction)
```

매개 변수

eventType 이 콜백 함수에 전달할 이벤트 유형을 지정하는 문자열. 사용할 수 있는 값에는 "documentNew", "documentOpened", "documentClosed", "mouseMove", "documentChanged", "layerChanged" 및 "frameChanged"가 있습니다.

documentChanged 값은 문서의 내용이 변경되었음을 의미하는 것이 아니라 현재 전경에 다른 문서가 있음을 의미합니다. 즉, fl.getDocumentDOM()은 이 이벤트가 발생하기 전과는 다른 값을 반환합니다.

callbackFunction 이벤트가 발생할 때마다 실행할 함수의 이름

반환값

없음

설명

메서드, 특정 이벤트가 발생할 때 호출할 함수를 등록합니다.

이 메서드를 사용할 경우에는 이벤트가 자주 발생하고(예: mouseMove) 함수의 실행 시간이 오래 걸리면 응용 프로그램이 중지되거나 오류 상태로 들어갈 수 있다는 점에 주의해야 합니다.

예제

다음 예제에서는 문서가 닫힐 때 [출력] 패널에 메시지를 표시합니다.

```
myFunction = function () {  
    fl.trace('document was closed');  
}  
fl.addEventListener("documentClosed", myFunction);
```

참고 사항

[fl.removeEventListener\(\)](#)

fl.as3PackagePaths

지원 버전

Flash CS3 Professional

사용법

```
fl.as3PackagePaths
```

설명

속성, [ActionScript 3.0 설정] 대화 상자의 전역 클래스 경로 설정에 해당하는 문자열입니다. 문자열에 있는 항목은 세미콜론으로 구분됩니다. ActionScript 2.0 클래스 경로 설정을 보거나 변경하려면 [fl.packagePaths](#)를 사용합니다.

예제

다음 예제에서는 ActionScript 3.0 클래스 경로 설정을 변경하는 방법을 보여 줍니다.

```
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
```

참고 사항

[fl.resetAS3PackagePaths\(\)](#)

fl.browseForFileURL()

지원 버전

Flash MX 2004

사용법

```
fl.browseForFileURL(browseType [, title [, previewArea]])
```

매개 변수

browseType 파일 탐색 작업의 유형을 지정하는 문자열. 사용할 수 있는 값에는 "open", "select" 또는 "save"가 있습니다. "open" 및 "select" 값은 [파일 열기] 시스템 대화 상자를 엽니다. 이 두 값은 Dreamweaver와의 호환성을 위해 제공됩니다. "save" 값은 [파일 저장] 시스템 대화 상자를 엽니다.

title [파일 열기] 또는 [파일 저장] 대화 상자의 제목을 지정하는 문자열. 이 매개 변수를 생략하면 기본값이 사용됩니다. 이 매개 변수는 선택 사항입니다.

previewArea Flash 및 Fireworks에서는 더 이상 사용되지 않고 Dreamweaver와의 호환성을 위해서만 제공되는 선택적 매개 변수

반환값

file:/// URI로 표현된 파일의 URL을 반환하고 사용자가 취소하여 대화 상자를 벗어난 경우에는 null을 반환합니다.

설명

메서드, [파일 열기] 또는 [파일 저장] 시스템 대화 상자를 연 다음 열거나 저장할 파일을 지정할 수 있도록 합니다.

예제

다음 예제에서는 사용자가 열 FLA 파일을 선택할 수 있도록 한 다음 해당 파일을 엽니다. `fl.browseForFileURL()` 메서드는 모든 유형의 파일을 탐색할 수 있지만 `fl.openDocument()`는 FLA 파일만 열 수 있습니다.

```
var fileURL = fl.browseForFileURL("open", "Select file");  
var doc = fl.openDocument(fileURL);
```

참고 사항

[fl.browseForFolderURL\(\)](#)

fl.browseForFolderURL()

지원 버전

Flash 8

사용법

```
fl.browseForFolderURL([description])
```

매개 변수

description [폴더 찾아보기] 대화 상자에 대한 설명을 지정하는 선택적 문자열. 이 매개 변수를 생략하면 설명 영역에 아무 것도 표시되지 않습니다.

반환값

file:/// URI로 표현된 폴더의 URL을 반환하고 사용자가 취소하여 대화 상자를 벗어난 경우에는 null을 반환합니다.

설명

메서드, [폴더 찾아보기] 대화 상자를 표시하고 사용자가 폴더를 선택할 수 있도록 합니다.

참고: 대화 상자의 제목은 항상 [폴더 찾아보기]입니다. `description` 매개 변수를 사용하여 제목 아래의 설명 영역에 "폴더 선택" 또는 "가져올 프로파일이 포함된 경로 선택" 등의 추가 정보를 추가합니다.

예제

다음 예제에서는 사용자가 폴더를 선택한 다음 해당 폴더 안에 들어 있는 파일 목록을 표시할 수 있도록 합니다.

```
var folderURI = fl.browseForFolderURL("Select a folder.");  
var folderContents = FLfile.listFolder(folderURI);
```

참고 사항

[fl.browseForFileURL\(\)](#), [FLfile 객체](#)

fl.clearPublishCache()

지원 버전

Flash CS5.5 Professional

사용법

```
fl.clearPublishCache()
```

매개 변수

없음

반환값

없음

설명

메서드, 게시 캐시를 비웁니다.

예제

다음 코드는 게시 캐시를 비웁니다.

```
fl.clearPublishCache()
```

참고 사항

[fl.publishCacheDiskSizeMax](#), [fl.publishCacheEnabled](#), 235페이지의 "[fl.publishCacheMemoryEntrySizeLimit](#)", [fl.publishCacheMemorySizeMax](#)

fl.clipCopyString()

지원 버전

Flash CS3 Professional

사용법

```
fl.clipCopyString(string)
```

매개 변수

string 클립보드에 복사할 문자열

반환값

없음

설명

메서드, 지정된 문자열을 클립보드에 복사합니다.

현재 선택 항목을 클립보드에 복사하려면 [document.clipCopy\(\)](#)를 사용합니다.

예제

다음 예제에서는 현재 문서의 경로를 클립보드에 복사합니다.

```
var documentPath = fl.getDocumentDOM().path;  
fl.clipCopyString(documentPath);
```

fl.closeAll()

지원 버전

Flash MX 2004

사용법

```
fl.closeAll([bPromptToSave])
```

매개 변수

bPromptToSave (선택 사항) 이전에 저장한 이후로 변경된 파일에 대해 [저장] 대화 상자를 표시할지, 아니면 저장하지 않은 파일에 대해 [다른 이름으로 저장] 대화 상자를 표시할지를 지정하는 부울 값입니다. 기본값은 true입니다.

반환값

없음

설명

메서드, 열려 있는 모든 파일(FLA 파일, SWF 파일, JSFL 파일 등)을 닫습니다. 변경 내용을 저장하지 않고 열려 있는 모든 파일을 닫으려면 bPromptToSave에 대해 false를 전달합니다. 이 메서드는 응용 프로그램을 종료하지는 않습니다.

예제

다음 코드에서는 열려 있는 모든 파일을 닫을 때 변경되었거나 새로운 파일을 저장하라는 대화 상자를 표시합니다.

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

참고 사항

[fl.closeAllPlayerDocuments\(\)](#), [fl.closeDocument\(\)](#)

fl.closeAllPlayerDocuments()

지원 버전

Flash CS3 Professional

사용법

```
fl.closeAllPlayerDocuments()
```

매개 변수

없음

반환값

부울 값, 열려 있는 무비 윈도우가 하나 이상 있으면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, [컨트롤] > [무비 테스트]를 통해 열려 있던 모든 SWF 파일을 닫습니다.

예제

다음 예제에서는 [컨트롤] > [무비 테스트]를 통해 열려 있던 모든 SWF 파일을 닫습니다.

```
fl.closeAllPlayerDocuments();
```

참고 사항

[fl.closeAll\(\)](#), [fl.closeDocument\(\)](#)

fl.closeDocument()

지원 버전

Flash MX 2004

사용법

```
fl.closeDocument (documentObject [, bPromptToSaveChanges])
```

매개 변수

documentObject [Document](#) 객체. documentObject가 활성 문서를 나타내면 이 메시지를 호출하는 스크립트의 실행이 끝날 때까지 문서 윈도우가 닫히지 않을 수 있습니다.

bPromptToSaveChanges 부울 값. bPromptToSaveChanges가 false로 설정된 경우 문서에 저장되지 않은 변경 내용이 있어도 변경 내용을 저장하라는 대화 상자가 표시되지 않습니다. 즉, 파일을 닫을 때 변경 내용이 저장되지 않습니다.

bPromptToSaveChanges가 true로 설정된 경우 문서에 저장되지 않은 변경 내용이 있으면 변경 내용을 저장하라는 [예]/[아니오] 버튼이 있는 표준 대화 상자가 표시됩니다. 기본값은 true입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 지정된 문서를 닫습니다.

예제

다음 예제에서는 문서를 닫는 두 가지 방법을 보여 줍니다.

```
// 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);
```

참고 사항

[fl.closeAll\(\)](#)

fl.compilerErrors

지원 버전

Flash CS3 Professional

사용법

```
fl.compilerErrors
```

설명

읽기 전용 속성, [오류] 패널을 나타내는 compilerErrors 객체입니다. 이 속성을 사용하는 방법에 대한 자세한 내용은 [compilerErrors](#) 객체를 참조하십시오.

fl.componentsPanel

지원 버전
Flash MX 2004

사용법
`fl.componentsPanel`

설명
읽기 전용 속성, [구성 요소] 패널을 나타내는 `componentsPanel` 객체입니다.

예제
다음 예제에서는 `componentsPanel` 객체를 `comPanel` 변수에 저장합니다.

```
var comPanel = fl.componentsPanel;
```

fl.configDirectory

지원 버전
Flash MX 2004

사용법
`fl.configDirectory`

설명
읽기 전용 속성, 로컬 사용자의 Configuration 디렉토리에 대한 전체 경로를 플랫폼별 형식으로 지정하는 문자열입니다. 이 경로를 플랫폼과 관련이 없는 `file:///` URI로 지정하려면 `fl.configURI`를 사용합니다.

예제
다음 예제에서는 Configuration 디렉토리를 [출력] 패널에 표시합니다.

```
fl.trace("My local configuration directory is " + fl.configDirectory);
```

fl.configURI

지원 버전
Flash MX 2004

사용법
`fl.configURI`

설명
읽기 전용 속성, 로컬 사용자의 Configuration 디렉토리에 대한 전체 경로를 `file:///` URI로 지정하는 문자열입니다. `fl.configDirectory`도 참조하십시오.

예제

다음 예제에서는 지정된 스크립트를 실행합니다. fl.configURI를 사용하면 스크립트가 실행되고 있는 플랫폼을 몰라도 스크립트의 위치를 지정할 수 있습니다.

```
// 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

지원 버전

Flash 8

사용법

```
fl.contactSensitiveSelection
```

설명

연결 감지 선택 모드가 활성화되어 있는지(true), 아니면 비활성화되어 있는지(false)를 지정하는 부울 값

예제

다음 예제에서는 선택 전에 연결 감지 선택 모드를 비활성화한 다음 선택 후 원래 값으로 다시 설정하는 방법을 보여 줍니다.

```
var contact = fl.contactSensitiveSelection;  
fl.contactSensitiveSelection = false;  
// Insert selection code here.  
fl.contactSensitiveSelection = contact;
```

fl.createDocument()

지원 버전

Flash MX 2004

사용법

```
fl.createDocument([docType])
```

매개 변수

docType 만들 문서의 유형을 지정하는 문자열. 사용할 수 있는 값에는 "timeline", "presentation" 및 "application"이 있습니다. 기본값은 "timeline"이며, [파일] > [새로 만들기] > [Flash 파일(ActionScript 3.0)]을 선택한 것과 동일한 효과를 냅니다. 이 매개 변수는 선택 사항입니다.

반환값

메서드가 성공적으로 실행된 경우 새로 만든 문서에 대한 Document 객체. 오류가 발생하면 값은 undefined가 됩니다.

설명

메서드, 새 문서를 열고 선택합니다. 크기, 해상도 및 색상에 대한 값은 현재 기본값과 같습니다.

예제

다음 예제에서는 다양한 유형의 문서를 만듭니다.

```
// 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

지원 버전
Flash MX 2004

사용법
fl.createNewDocList

설명
읽기 전용 속성, 만들 수 있는 문서의 다양한 유형을 나타내는 문자열 배열입니다.

예제
다음 예제에서는 만들 수 있는 문서의 유형을 [출력] 패널에 표시합니다.

```
fl.trace("Number of choices " + fl.createNewDocList.length);  
for (i = 0; i < fl.createNewDocList.length; i++)  
    fl.trace("choice: " + fl.createNewDocList[i]);
```

fl.createNewDocListType

지원 버전
Flash MX 2004

사용법
fl.createNewDocListType

설명
읽기 전용 속성, 만들 수 있는 문서 유형의 파일 확장명을 나타내는 문자열 배열입니다. 이 배열의 항목은 인덱스별로 [fl.createNewDocList](#) 배열의 항목에 직접 대응됩니다.

예제
다음 예제에서는 만들 수 있는 문서 유형의 확장명을 [출력] 패널에 표시합니다.

```
fl.trace("Number of types " + fl.createNewDocListType.length);  
for (i = 0; i < fl.createNewDocListType.length; i++) fl.trace("type: " + fl.createNewDocListType[i]);
```

fl.createNewTemplateList

지원 버전
Flash MX 2004

사용법

```
fl.createNewTemplateList
```

설명

읽기 전용 속성, 만들 수 있는 템플릿의 다양한 유형을 나타내는 문자열 배열입니다.

예제

다음 예제에서는 만들 수 있는 템플릿의 유형을 [출력] 패널에 표시합니다.

```
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

지원 버전
Flash MX 2004

사용법

```
fl.documents
```

설명

읽기 전용 속성, 편집하려고 현재 열어 놓은 문서(FLA 파일)를 나타내는 Document 객체([Document 객체 참조](#))의 배열입니다.

예제

다음 예제에서는 열려 있는 문서의 배열을 docs 변수에 저장합니다.

```
var docs = fl.documents;
```

다음 예제에서는 현재 열려 있는 문서의 이름을 [출력] 패널에 표시합니다.

```
for (doc in fl.documents) {  
    fl.trace(fl.documents[doc].name);  
}
```

fl.drawingLayer

지원 버전
Flash MX 2004

사용법

```
fl.drawingLayer
```

설명

읽기 전용 속성, 선택 윤곽을 만들 때처럼 드래그하는 동안 모양을 임시로 그리려고 할 때 확장 가능한 도구에서 사용해야 하는 [drawingLayer 객체](#)입니다.

예제

[drawingLayer.setColor\(\)](#)를 참조하십시오.

fl.exportPublishProfileString()

지원 버전

Flash Professional CS5

사용법

```
fl.exportPublishProfileString( ucfURI [, profileName] )
```

매개 변수

ucfURI 제작 설정을 내보낼 파일 URI(Uniform Resource Identifier)를 지정하는 문자열
profileName 내보낼 프로파일 이름을 지정하는 문자열. 이 매개 변수는 선택 사항입니다.

반환값

문자열

설명

파일을 열지 않고도 특정 문서의 제작 프로파일을 반환합니다. 제작 프로파일을 지정할 수도 있지만, 이는 선택 사항입니다.

예제

다음 예제에서는 제작 프로파일 문자열을 읽습니다.

```
var ppXML = "";  
var ucfURI = fl.browseForFileURL("open", "select a FLA");  
if (ucfURI && ucfURI.length > 0)  
ppXML = fl.exportPublishProfileString(ucfURI);  
fl.trace(ppXML);
```

fl.externalLibraryPath

지원 버전

Flash CS4 Professional

사용법

```
fl.externalLibraryPath
```

설명

속성, 런타임 공유 라이브러리로 사용되는 SWC 파일의 위치를 지정하는 전역 ActionScript 3.0 외부 라이브러리 경로에 있는 항목의 목록이 포함된 문자열입니다. 문자열에 있는 항목은 세미콜론으로 구분됩니다. 제작 도구에서 [편집] > [환경 설정] > [ActionScript] > [ActionScript 3.0 설정]을 선택하여 항목을 지정할 수 있습니다.

예제

다음 예제에서는 /SWC_runtime 폴더를 전역 ActionScript 3.0 외부 라이브러리 경로에 추가합니다.

```
fl.trace(fl.externalLibraryPath);  
fl.externalLibraryPath = "/SWC_runtime;" + fl.externalLibraryPath;  
fl.trace(fl.externalLibraryPath);
```

참고 사항

[fl.flexSDKPath](#), [fl.libraryPath](#), [fl.sourcePath](#), [document.externalLibraryPath](#)

fl.fileExists()

지원 버전

Flash MX 2004

사용법

```
fl.fileExists(fileURI)
```

매개 변수

fileURI 파일에 대한 경로가 포함된 문자열(file:/// URI로 표시됨)

반환값

부울 값, 파일이 디스크에 있으면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 디스크에 파일이 이미 있는지 여부를 확인합니다.

예제

다음 예제에서는 지정된 각 파일의 존재 여부를 [출력] 패널에 true 또는 false로 표시합니다.

```
alert(fl.fileExists("file:///C:/example fla"));  
alert(fl.fileExists("file:///C:/example.jsfl"));  
alert(fl.fileExists(""));
```

fl.findDocumentDOM()

지원 버전

Flash CS3 Professional

사용법

```
fl.findDocumentDOM(id)
```

매개 변수

id 문서의 고유 식별자를 나타내는 정수

반환값

Document 객체를 반환하며, 지정된 id를 가진 문서가 없으면 null을 반환합니다.

설명

메서드, 파일의 인덱스 값 등을 사용하는 대신 파일의 고유 식별자를 사용하여 특정 파일을 대상으로 지정할 수 있습니다. 이 메서드는 [document.id](#)와 함께 사용됩니다.

예제

다음 예제에서는 문서의 ID를 읽은 다음 이 ID를 사용하여 해당 문서를 대상으로 지정하는 방법을 보여 줍니다.

```
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;
```

참고 사항

[fl.findDocumentIndex\(\)](#)

fl.findDocumentIndex()

지원 버전

Flash MX 2004

사용법

```
fl.findDocumentIndex(name)
```

매개 변수

name 인덱스를 찾을 문서 이름. 문서가 열려 있어야 합니다.

반환값

fl.documents 배열에서 name 문서의 위치를 나타내는 정수 배열

설명

메서드, **fl.documents** 배열에서 name 문서의 위치를 나타내는 정수 배열을 반환합니다. 문서가 서로 다른 폴더에 있을 경우 같은 이름의 문서를 둘 이상 열 수 있습니다.

예제

다음 예제에서는 [출력] 패널의 test.fla라는 열려 있는 파일의 인덱스 위치에 대한 정보를 표시합니다.

```
var filename = "test.fla"
var docIndex = fl.findDocumentIndex(filename);
for (var index in docIndex)
    fl.trace(filename + " is open at index " + docIndex[index]);
```

참고 사항

[fl.documents](#), [fl.findDocumentDOM\(\)](#)

fl.findObjectInDocByName()

지원 버전

Flash CS3 Professional

사용법

```
fl.findObjectInDocByName(instanceName, document)
```

매개 변수

instanceName 지정된 문서에 있는 항목의 인스턴스 이름을 지정하는 문자열

document 지정된 항목을 검색할 [Document 객체](#)

반환값

일반 객체의 배열. 배열에 있는 각 항목의 .obj 속성을 사용하여 객체를 가져옵니다. 객체에는 **keyframe**, **layer**, **timeline** 및 **parent** 속성이 있습니다. 이러한 속성을 사용하여 객체의 계층 구조에 액세스할 수 있습니다. 이 속성 및 속성에 액세스하는 방법에 대한 자세한 내용은 [fl.findObjectInDocByType\(\)](#)을 참조하십시오.

layer 및 timeline 값(각각 [Layer 객체](#) 및 [Timeline 객체](#)에 해당)에 대한 메서드 및 속성에 액세스할 수도 있습니다.

설명

메서드, 지정된 텍스트와 일치하는 인스턴스 이름을 가진 요소를 문서에 표시합니다.

참고: 이 메서드는 JSFL 파일을 현재 보고 있거나 편집하고 있는 경우가 아닌 FLA 파일에서 명령을 실행될 때에만 작동합니다.

예제

다음 예제에서는 현재 문서에서 인스턴스 이름이 "instance01"인 요소를 검색합니다.

```
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");  
}
```

참고 사항

[fl.findObjectInDocByType\(\)](#)

fl.findObjectInDocByType()

지원 버전

Flash CS3 Professional

사용법

```
fl.findObjectInDocByType(elementType, document)
```

매개 변수

elementType 검색할 요소의 유형을 나타내는 문자열. 사용할 수 있는 값은 [element.elementType](#)을 참조하십시오.

document 지정된 항목을 검색할 **Document** 객체

반환값

일반 객체의 배열. 배열에 있는 각 항목의 .obj 속성을 사용하여 요소 객체를 가져옵니다. 각 객체에는 keyframe, layer, timeline 및 parent 속성이 있습니다. 이러한 속성을 사용하여 객체의 계층 구조에 액세스할 수 있습니다.

layer 및 timeline 값(각각 **Layer** 객체 및 **Timeline** 객체에 해당)에 대한 메서드 및 속성에 액세스할 수도 있습니다.

예제 단원의 두 번째 및 세 번째 예제에서는 이러한 속성에 액세스하는 방법을 보여 줍니다.

설명

메서드, 지정된 요소 유형의 요소를 문서에 표시합니다.

참고: 이 메서드는 JSFL 파일을 현재 보고 있거나 편집하고 있는 경우가 아닌 FLA 파일에서 명령을 실행될 때에만 작동합니다.

예제

다음 예제에서는 현재 문서에서 텍스트 필드를 검색한 다음 텍스트 필드의 내용을 변경합니다.

```
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");
}
```

다음 예제에서는 이 메서드에서 반환된 객체의 특수한 속성에 액세스하는 방법을 보여 줍니다.

```
var doc = fl.getDocumentDOM();
var resultsArray = 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.
}
```

다음 예제에서는 resultArray.obj 객체를 사용하여 DOM 백업을 통해 텍스트 필드가 발견된 레이어의 이름을 찾는 방법을 보여 줍니다.


```
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");
}
```

참고 사항

[fl.findObjectInDocByName\(\)](#)

fl.flexSDKPath

지원 버전

Flash CS4 Professional

사용법

`fl.flexSDKPath`

설명

속성, bin, frameworks, lib 및 기타 폴더가 포함된 Flex SDK 폴더의 경로를 지정하는 문자열입니다. 제작 도구에서 [편집] > [환경 설정] > [ActionScript] > [ActionScript 3.0 설정]을 선택하여 항목을 지정할 수 있습니다.

예제

다음 코드에서는 Flex SDK 경로를 [출력] 패널에 표시합니다.

```
fl.trace(fl.flexSDKPath);
```

참고 사항

[fl.externalLibraryPath](#), [fl.libraryPath](#), [fl.sourcePath](#)

fl.getAppMemoryInfo()

지원 버전

Flash 8(Windows 전용)

사용법

`fl.getAppMemoryInfo(memType)`

매개 변수

`memType` 조회할 메모리 사용 영역을 지정하는 정수. 사용할 수 있는 값의 목록을 보려면 다음 설명을 참조하십시오.

반환값

Flash.exe 메모리의 지정된 영역에 사용되고 있는 바이트 수를 나타내는 정수

설명

메서드(Windows 전용), Flash.exe 메모리의 지정된 영역에 사용되고 있는 바이트 수를 나타내는 정수를 반환합니다. 다음 표에서 **memType**으로 전달할 값을 확인할 수 있습니다.

memType	리소스 데이터
0	PAGEFAULTCOUNT
1	PEAKWORKINGSETSIZE
2	WORKINGSETSIZE
3	QUOTAPEAKPAGEDPOOLUSAGE
4	QUOTAPAGEDPOOLUSAGE
5	QUOTAPEAKNONPAGEDPOOLUSAGE
6	QUOTANONPAGEDPOOLUSAGE
7	PAGEFILEUSAGE
8	PEAKPAGEFILEUSAGE

예제

다음 예제에서는 현재 작업 메모리 사용량을 표시합니다.

```
var memsize = fl.getAppMemoryInfo(2);  
fl.trace("Flash current memory consumption is " + memsize + " bytes or " + memsize/1024 + " KB");
```

fl.getDocumentDOM()

지원 버전

Flash MX 2004

사용법

```
fl.getDocumentDOM()
```

매개 변수

없음

반환값

Document 객체를 반환하거나 열려 있는 문서가 없으면 null을 반환합니다.

설명

메서드, 현재 활성화된 문서(FLA 파일)의 DOM(Document 객체)을 가져옵니다. 하나 이상의 문서가 열려 있지만 현재 포커스를 가지고 있는 문서가 없으면(예를 들어, JSFL 파일에 포커스가 있는 경우) 가장 최근에 활성화된 문서의 DOM을 가져옵니다.

예제

다음 예제에서는 현재 또는 가장 최근에 활성화된 문서의 이름을 [출력] 패널에 표시합니다.

```
var currentDoc = fl.getDocumentDOM();  
fl.trace(currentDoc.name);
```

fl.getSwfPanel()

지원 버전

Flash CS5.5 Professional

사용법

```
fl.getSwfPanel(panelName, [useLocalizedPanelName])
```

매개 변수

panelName 지역화된 패널 이름 또는 패널 SWF 파일의 루트 파일 이름입니다. 후자를 사용하는 경우 **false**를 두 번째 매개 변수로 전달합니다.

useLocalizedPanelName 선택 사항입니다. 기본값은 **true**입니다. **false**일 경우에는 **panelName** 매개 변수가 패널의 영어(지역화되지 않음) 이름인 것으로 가정합니다. 이 이름은 파일 확장자가 없는 SWF 파일 이름에 해당합니다.

반환값

swfPanel 객체

설명

메서드, 패널의 지역화된 이름 또는 SWF 파일 이름(파일 이름 확장자 없이)을 기준으로 SWFPanel 객체를 반환합니다.

예제

다음 예제에서는 '프로젝트'로 참조된 패널의 이름을 [출력] 패널에 표시합니다.

```
fl.trace('name of panel is: ' + fl.getSwfPanel('Project').name);
```

fl.installedPlayers

지원 버전

Flash CS5.5 Professional

사용법

```
fl.installedPlayers()
```

매개 변수

없음

반환값

문서 속성 관리자에서 설치된 Flash Player 목록에 해당하는 일반 객체의 배열입니다.

설명

읽기 전용 속성, 문서 속성 관리자에서 설치된 Flash Player 목록에 해당하는 일반 객체의 배열입니다.

배열의 각 객체는 다음 속성을 포함합니다.

name 문서의 문자열 이름입니다.

version Document.setPlayerVersion() 함수를 사용하여 문서의 현재 플레이어를 설정하는 데 사용할 수 있습니다.

minASVersion 문서에 필요한 최소 ActionScript 버전입니다. Document.asVersion 속성을 통해 minASVersion 및 maxASVersion 사이(양쪽 끝 포함)의 정수를 사용하여 문서의 ActionScript 버전을 설정할 수 있습니다.

maxASVersion 문서에서 지원하는 최대 ActionScript 버전입니다.

stageWidth 주어진 대상의 Stage 폭 기본값을 픽셀 단위로 나타낸 것입니다. 예를 들어 iPhone은 기본 크기가 320 x 480 픽셀입니다. Android의 경우 기본 크기는 480 x 800입니다.

stageHeight 주어진 대상의 Stage 높이 기본값을 픽셀 단위로 나타낸 것입니다. 예를 들어 iPhone은 기본 크기가 320 x 480 픽셀입니다. Android의 경우 기본 크기는 480 x 800입니다.

예제

다음 예제에서는 installedPlayers 배열에 있는 모든 객체의 속성을 추적하여 출력 윈도우에 표시합니다.

```
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()

지원 버전

Flash CS4 Professional

사용법

```
fl.isFontInstalled(fontName)
```

매개 변수

fontName 장치 글꼴의 이름을 지정하는 문자열

반환값

지정된 글꼴이 설치되어 있으면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값

설명

메서드, 지정된 글꼴이 설치되어 있는지 여부를 확인합니다.

예제

다음 코드에서는 Times 글꼴이 설치되어 있는 경우 "true"를 [출력] 패널에 표시합니다.

```
fl.trace(fl.isFontInstalled("Times"));
```

fl.languageCode

지원 버전

Flash CS5 Professional

사용법

`fl.languageCode`

설명

속성, 응용 프로그램의 사용자 인터페이스의 로케를 식별하는 다섯 자리 문자 코드를 반환하는 문자열입니다.

예제

다음 예제에서는 Flash 응용 프로그램의 지역화된 사용자 인터페이스에 지정된 다섯 자리 언어 코드를 반환합니다.

```
locConfigURI = fl.applicationURI + fl.languageCode + "/Configuration";
```

fl.libraryPath

지원 버전

Flash CS4 Professional

사용법

`fl.libraryPath`

설명

속성, SWC 파일이 포함된 폴더나 SWC 파일의 위치를 지정하는 전역 ActionScript 3.0 라이브러리 경로에 있는 항목의 목록이 포함된 문자열입니다. 문자열에 있는 항목은 세미콜론으로 구분됩니다. 제작 도구에서 [편집] > [환경 설정] > [ActionScript] > [ActionScript 3.0 설정]을 선택하여 항목을 지정할 수 있습니다.

예제

다음 예제에서는 /SWC 폴더를 전역 ActionScript 3.0 라이브러리 경로에 추가합니다.

```
fl.trace(fl.libraryPath);  
fl.libraryPath = "/SWC;" + fl.libraryPath;  
fl.trace(fl.libraryPath);
```

참고 사항

[fl.externalLibraryPath](#), [fl.flexSDKPath](#), [fl.sourcePath](#), [document.libraryPath](#)

fl.mapPlayerURL()

지원 버전

Flash MX 2004

사용법

```
fl.mapPlayerURL(URI [, returnMBCS])
```

매개 변수

URI 매핑할 이스케이프된 유니코드 URL을 포함하는 문자열

returnMBCS 이스케이프된 MBCS 경로를 반환하려는 경우 true로 설정해야 하는 부울 값. 기본값은 false입니다. 이 경우 UTF-8이 반환됩니다. 이 매개 변수는 선택 사항입니다.

반환값

변환된 URL을 나타내는 문자열

설명

메서드, 이스케이프된 유니코드 URL을 UTF-8 또는 MBCS URL에 매핑합니다. 외부 리소스에 액세스하기 위해 이 문자열이 ActionScript에서 사용될 때 이 메서드를 사용합니다. 멀티바이트 문자열을 처리해야 하는 경우에는 이 메서드를 사용해야 합니다.

예제

다음 예제에서는 플레이어에 로드할 수 있도록 URL을 UTF-8로 변환합니다.

```
var url = MMExecute( "fl.mapPlayerURL(" + myURL + ", false);" );  
mc.loadMovie( url);
```

fl.Math

지원 버전

Flash MX 2004

사용법

fl.Math

설명

읽기 전용 속성, 행렬 및 소수점 연산을 위한 메서드를 제공하는 [Math 객체](#)입니다.

예제

다음 예제에서는 선택한 객체의 변형 행렬과 그 역행렬을 보여 줍니다.

```
// 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

지원 버전

Flash MX 2004

사용법

```
fl.mruRecentFileList
```

설명

읽기 전용 속성, Flash 제작 도구에서 관리하는 가장 최근에 사용한 문서(MRU) 목록에 있는 전체 파일 이름의 배열입니다.

예제

다음 예제에서는 최근에 연 파일의 개수와 각 파일의 이름을 [출력] 패널에 표시합니다.

```
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.mruRecentFileListType

지원 버전

Flash MX 2004

사용법

```
fl.mruRecentFileListType
```

설명

읽기 전용 속성, Flash 제작 도구에서 관리하는 MRU 목록에 있는 파일 유형의 배열입니다. 이 배열은 [fl.mruRecentFileList](#) 속성의 배열에 해당합니다.

예제

다음 예제에서는 최근에 연 파일의 개수와 각 파일의 유형을 [출력] 패널에 표시합니다.

```
fl.trace("Number of recently opened files: " + fl.mruRecentFileListType.length);  
for (i = 0; i < fl.mruRecentFileListType.length; i++) fl.trace("type: " + fl.mruRecentFileListType[i]);
```

fl.objectDrawingMode

지원 버전

Flash 8

사용법

```
fl.objectDrawingMode
```

설명

속성, 객체 드로잉 모드가 활성화되어 있는지(true), 아니면 병합 드로잉 모드가 활성화되어 있는지(false)를 지정하는 부울 값입니다.

예제

다음 예제에서는 객체 드로잉 모드의 상태를 전환합니다.

```
var toggleMode = fl.objectDrawingMode;
if (toggleMode) {
    fl.objectDrawingMode = false;
} else {
    fl.objectDrawingMode = true;
}
```

fl.openDocument()

지원 버전

Flash MX 2004

사용법

```
fl.openDocument(fileURI)
```

매개 변수

fileURI 열 파일의 이름을 지정하는 문자열(file:/// URI로 표시됨)

반환값

메서드가 성공적으로 실행된 경우 새로 연 문서에 대한 **Document** 객체. 파일이 없거나 유효한 FLA 파일이 아니면 오류가 보고되고 스크립트가 취소됩니다.

설명

메서드, 새 Flash 문서 윈도우에서 Flash 문서(FLA 파일)를 편집하기 위해 열고 이 문서에 포커스를 둡니다. 이 메서드는 [파일]>[열기]를 선택한 다음 파일을 선택하는 것과 같은 효과를 가집니다. 지정된 파일이 이미 열려 있으면 해당 문서가 들어 있는 윈도우가 맨 앞으로 나오고 지정된 파일을 포함하는 윈도우가 현재 선택된 문서가 됩니다.

예제

다음 예제에서는 C 드라이브의 루트 디렉토리에 저장되어 있는 **Document.fla**라는 파일을 엽니다. 이 코드에서는 해당 문서를 나타내는 **Document** 객체를 doc 변수에 저장한 다음 이 문서를 현재 선택된 문서가 되게 설정합니다. 즉, 포커스가 변경될 때 까지 fl.getDocumentDOM()에서 이 문서를 참조합니다.

```
var doc = fl.openDocument("file:///c:/Document.fla");
```

fl.openScript()

지원 버전

Flash MX 2004. Flash Professional CS5에서 추가된 선택적 매개 변수입니다.

사용법

```
fl.openScript(fileURI [, createExtension, className])
```

매개 변수

fileURI Flash 텍스트 편집기에 로드해야 하는 JSFL, AS, ASC, XML, TXT 또는 기타 파일의 경로를 지정하는 문자열(file:///URI로 표시됨). 이 매개 변수는 null일 수 있습니다. null일 경우 이 메서드는 createExtension 매개 변수에 의해 지정된 유형의 새 스크립트를 엽니다.

createExtension fileURI가 null일 경우 만들 문서 유형을 결정하는 데 사용하는 문자열. 기본값은 'AS'이며, 허용 가능한 값은 'JSFL', 'AS', 'ASC', 'XML', 'TXT', 'AS3_CLASS' 또는 'AS3_INTERFACE'입니다. 이 매개 변수는 Flash Professional CS5에서 추가되었습니다.

className 클래스 또는 인터페이스를 만드는 경우 정규화된 클래스 이름을 지정하는 데 사용하는 문자열(createExtension 매개 변수에 의해 결정됨). 이 매개 변수는 Flash Professional CS5에서 추가되었습니다.

반환값

없음

설명

메서드, Flash 텍스트 편집기에서 기존 파일을 열거나 새 스크립트(JSFL, AS, ASC) 또는 다른 파일(XML, TXT)을 만듭니다.

예제

다음 예제에서는 C 드라이브의 /temp 디렉토리에 저장되어 있는 my_test.jsfl이라는 파일을 엽니다.

```
fl.openScript("file:///c:/temp/my_test.jsfl");
```

예제

다음 예제에서는 빈 AS3 클래스 정의를 갖는 새 .as 파일을 만듭니다.

```
fl.openScript(null, 'AS3_CLASS');
```

fl.outputPanel

지원 버전

Flash MX 2004

사용법

```
fl.outputPanel
```

설명

읽기 전용 속성, [outputPanel 객체](#)에 대한 참조입니다.

예제

[outputPanel 객체](#)를 참조하십시오.

fl.packagePaths

지원 버전

Flash CS3 Professional

사용법

```
fl.packagePaths
```

설명

속성, [ActionScript 2.0 설정] 대화 상자의 전역 클래스 경로 설정에 해당하는 문자열입니다. 문자열 내의 클래스 경로는 세미 콜론(;)으로 구분됩니다. ActionScript 3.0 클래스 경로 설정을 보거나 변경하려면 [fl.as3PackagePaths](#)를 사용합니다.

예제

다음 예제에서는 ActionScript 2.0 클래스 경로 설정을 변경하는 방법을 보여 줍니다.

```
fl.trace(fl.packagePaths);  
// Output (assuming started with default value)  
// .;$ (LocalData)/Classes  
fl.packagePaths="buying;selling";  
fl.trace(fl.packagePaths);  
// Output  
// buying; selling
```

참고 사항

[fl.resetPackagePaths\(\)](#)

fl.presetPanel

지원 버전

Flash CS4 Professional

사용법

```
fl.presetPanel
```

설명

읽기 전용 속성, [presetPanel](#) 객체입니다.

fl.publishCacheDiskSizeMax

지원 버전

Flash CS5.5 Professional

사용법

```
fl.publishCacheDiskSizeMax
```

설명

속성: 디스크의 최대 게시 캐시 크기를 메가바이트 단위로 설정하는 정수입니다.

예제

다음 코드는 디스크의 최대 게시 캐시 크기를 1메가바이트로 설정합니다.

```
fl.publishCacheDiskSizeMax = 1
```

참고 사항

[fl.clearPublishCache\(\)](#), [fl.publishCacheEnabled](#), 235페이지의 "[fl.publishCacheMemoryEntrySizeLimit](#)", 236페이지의 "[fl.publishCacheMemorySizeMax](#)"

fl.publishCacheEnabled

지원 버전

Flash CS5.5 Professional

사용법

```
fl.publishCacheEnabled
```

설명

속성: 게시 캐시가 사용되는지 여부를 설정하는 부울 값입니다.

예제

다음 코드는 게시된 캐시가 사용되는지 여부를 출력 윈도우에 표시합니다.

```
fl.trace(fl.publishCacheEnabled);
```

참고 사항

[fl.publishCacheDiskSizeMax](#), [fl.clearPublishCache\(\)](#), 235페이지의 "[fl.publishCacheMemoryEntrySizeLimit](#)", 236페이지의 "[fl.publishCacheMemorySizeMax](#)"

fl.publishCacheMemoryEntrySizeLimit

지원 버전

Flash CS5.5 Professional

사용법

```
fl.publishCacheMemoryEntrySizeLimit
```

설명

속성: 메모리의 게시 캐시에 추가할 수 있는 항목의 최대 크기를 킬로바이트 단위로 설정하는 정수입니다. 이 크기 이하의 항목은 메모리에 저장되며 이보다 큰 항목은 디스크에 기록됩니다.

사용할 수 있는 메모리가 많은 사용자는 성능을 향상시키기 위해 이 값을 높여야 할 수 있지만 메모리가 적은 사용자는 게시 캐시에서 너무 많은 메모리를 사용하지 않도록 하기 위해 이 값을 낮춰야 할 수 있습니다.

예제

다음 코드에서는 메모리에 저장할 수 있는 최대 게시 캐시 항목 크기를 100킬로바이트로 설정합니다.

```
fl.publishCacheMemoryEntrySizeLimit = 100
```

참고 사항

[fl.publishCacheDiskSizeMax](#), [fl.publishCacheEnabled](#), 212페이지의 "[fl.clearPublishCache\(\)](#)", 236페이지의 "[fl.publishCacheMemorySizeMax](#)"

fl.publishCacheMemorySizeMax

지원 버전

Flash CS5.5 Professional

사용법

```
fl.publishCacheMemorySizeMax
```

설명

속성: 메모리의 최대 게시 캐시 크기를 메가바이트 단위로 설정하는 정수입니다.

예제

다음 코드는 메모리의 최대 게시 캐시 크기를 1메가바이트로 설정합니다.

```
fl.publishCacheMemorySizeMax = 1
```

참고 사항

[fl.publishCacheDiskSizeMax](#), [fl.publishCacheEnabled](#), 235페이지의 "[fl.publishCacheMemoryEntrySizeLimit](#)", 212페이지의 "[fl.clearPublishCache\(\)](#)"

fl.publishDocument()

지원 버전

Flash CS5 Professional

사용법

```
fl.publishDocument( flaURI [, publishProfile] )
```

매개 변수

flaURI 자동으로 제작되는 FLA 파일의 경로를 지정하는 문자열(file:/// URI로 표시됨)

publishProfile 제작 시 사용할 제작 프로파일을 지정하는 문자열. 이 문자열을 생략하면 기본 제작 프로파일이 사용됩니다.

반환값

부울

설명

메서드, FLA 파일을 열지 않고 제작합니다. 이 API는 FLA를 headless 모드로 열고 SWF(또는 프로파일이 설정된 형식)를 제작합니다. 두 번째 매개 변수(publishProfile)는 선택 사항이며 프로파일을 찾을 수 있는지 여부를 나타내는 부울 값을 반환합니다. 두 번째 매개 변수를 지정하지 않는 경우 항상 true 값을 반환합니다.

예제

다음 예제에서는 사용자에게 FLA 파일을 선택하라는 메시지를 표시하고 "Default" 제작 프로파일을 사용하여 자동으로 제작합니다.

```
var uri = fl.browseForFileURL("select", "select a FLA file to publish");  
var publishProfileName = "Default";  
fl.publishDocument(uri, publishProfileName);
```

fl.quit()

지원 버전
Flash MX 2004

사용법

```
fl.quit([bPromptIfNeeded])
```

매개 변수

bPromptIfNeeded true(기본값)로 설정된 경우 수정된 문서를 저장하라는 대화 상자가 표시되는 부울 값. 수정된 문서를 저장하라는 대화 상자를 표시하지 않으려면 이 매개 변수를 **false**로 설정하십시오. 이 경우 열려 있는 문서의 수정 내용을 저장하지 않고 응용 프로그램이 바로 종료됩니다. 이 메서드는 일괄 처리 작업에 유용하지만 주의해서 사용해야 합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, Flash를 종료할 때 변경된 문서를 저장하라는 대화 상자를 표시합니다.

예제

다음 예제에서는 수정된 문서를 저장하라는 대화 상자를 표시하지 않고 Flash를 종료하는 방법을 보여 줍니다.

```
// 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()

지원 버전
Flash MX 2004

사용법

```
fl.reloadEffects()
```

매개 변수

없음

반환값

없음

설명

메서드, 사용자의 **Configuration Effects** 폴더에 정의된 모든 효과 설명자를 다시 로드합니다. 이 메서드는 스크립트를 개발하는 동안 스크립트를 빠르게 변경할 수 있도록 하고 응용 프로그램을 다시 시작하지 않고도 효과를 향상시킬 수 있는 메커니즘을 제공합니다. 이 메서드는 **Commands** 폴더에 배치된 명령에 사용할 때 가장 효과적입니다.

예제

다음 예제는 Commands 폴더에 배치할 수 있는 한 줄 스크립트입니다. 효과를 다시 로드해야 하는 경우에는 [명령] 메뉴에서 스크립트를 실행합니다.

```
fl.reloadEffects();
```

fl.reloadTools()

지원 버전

Flash MX 2004

사용법

```
fl.reloadTools();
```

매개 변수

없음

반환값

없음

설명

메서드, toolconfig.xml 파일을 사용하여 [도구] 패널을 다시 구성합니다. 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다. 예를 들어 해당 패널에 이미 있는 도구를 정의하는 JSFL 파일을 수정한 후 [도구] 패널을 다시 로드해야 할 때 이 메서드를 사용합니다.

예제

다음 예제는 Commands 폴더에 배치할 수 있는 한 줄 스크립트입니다. [도구] 패널을 다시 로드해야 할 때 [명령] 메뉴에서 이 스크립트를 실행하십시오.

```
fl.reloadTools();
```

fl.removeEventListener()

지원 버전

Flash CS3 Professional

사용법

```
fl.removeEventListener(eventType)
```

매개 변수

eventType 이 콜백 함수에서 제거할 이벤트 유형을 지정하는 문자열. 사용할 수 있는 값에는 "documentNew", "documentOpened", "documentClosed", "mouseMove", "documentChanged", "layerChanged" 및 "frameChanged"가 있습니다.

반환값

이벤트 리스너가 성공적으로 제거되면 true를 반환하고 fl.addEventListener() 메서드를 사용하여 함수가 목록에 추가되지 않으면 false를 반환합니다.

설명

[fl.addEventListener\(\)](#)를 사용하여 등록한 함수의 등록을 해제합니다.

예제

다음 예제에서는 documentClosed 이벤트에 연결된 이벤트 리스너를 제거합니다.

```
fl.removeEventListener("documentClosed");
```

참고 사항

[fl.addEventListener\(\)](#)

fl.resetAS3PackagePaths()

지원 버전

Flash CS3 Professional

사용법

```
fl.resetAS3PackagePaths();
```

매개 변수

없음

설명

메서드, [ActionScript 3.0 설정] 대화 상자의 전역 클래스 경로 설정을 기본값으로 재설정합니다. ActionScript 2.0 전역 클래스 경로를 재설정하려면 [fl.resetPackagePaths\(\)](#)를 사용합니다.

예제

다음 예제에서는 ActionScript3.0 클래스 경로 설정을 기본값으로 재설정하는 방법을 보여 줍니다.

```
fl.resetAS3PackagePaths();
```

참고 사항

[fl.as3PackagePaths](#)

fl.resetPackagePaths()

지원 버전

Flash CS3 Professional

사용법

```
fl.resetPackagePaths();
```

매개 변수

없음

설명

메서드, [ActionScript 2.0 설정] 대화 상자의 전역 클래스 경로 설정을 기본값으로 재설정합니다. ActionScript 3.0 전역 클래스 경로를 재설정하려면 [fl.resetAS3PackagePaths\(\)](#)를 사용합니다.

예제

다음 예제에서는 ActionScript 2.0 클래스 경로 설정을 기본값으로 재설정하는 방법을 보여 줍니다.

```
fl.resetPackagePaths();
```

참고 사항

[fl.packagePaths](#)

fl.revertDocument()

지원 버전

Flash MX 2004

사용법

```
fl.revertDocument (documentObject)
```

매개 변수

documentObject [Document](#) 객체. **documentObject**가 활성 문서를 나타내면 이 메서드를 호출하는 스크립트의 실행이 끝날 때까지 문서 윈도우가 복귀되지 않을 수 있습니다.

반환값

부울 값, 복귀 작업이 성공적으로 완료되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 지정된 FLA 문서를 가장 최근에 저장된 버전으로 되돌립니다. [파일] > [복귀] 메뉴 옵션과 달리 이 메서드는 작업을 정말로 실행할 것인지를 묻는 경고 윈도우를 표시하지 않습니다. [document.revert\(\)](#) 및 [document.canRevert\(\)](#)도 참조하십시오.

예제

다음 예제에서는 현재 FLA 문서를 가장 최근에 저장된 버전으로 되돌립니다. 따라서 가장 최근에 저장된 이후에 변경된 모든 내용은 손실됩니다.

```
fl.revertDocument (fl.getDocumentDOM());
```

fl.runScript()

지원 버전

Flash MX 2004

사용법

```
fl.runScript (fileURI [, funcName [, arg1, arg2, ...]])
```

매개 변수

fileURI 실행할 스크립트 파일의 이름을 지정하는 문자열(file:/// URI로 표시됨)

funcName **fileURI**에 지정된 JSFL 파일에서 실행할 함수를 식별하는 문자열. 이 매개 변수는 선택 사항입니다.

arg **funcname**에 전달할 하나 이상의 인수를 지정하는 선택적 매개 변수

반환값

funcName이 지정된 경우 함수의 결과를 문자열로 반환하고 그렇지 않으면 아무 것도 반환하지 않습니다.

설명

메서드, JavaScript 파일을 실행합니다. 함수가 인수 중 하나로 지정된 경우 이 함수가 실행될 때 스크립트에 있는 코드 중 이 함수에 포함되지 않은 코드도 함께 실행됩니다. 스크립트에 있는 나머지 코드는 이 함수가 실행되기 전에 실행됩니다.

예제

C 드라이브의 루트 디렉토리에 testScript.jsfl이라는 스크립트 파일이 있고 그 내용이 다음과 같다고 가정해 봅니다.

```
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
```

다음 명령을 실행합니다.

```
fl.runScript("file:///C:/testScript.jsfl", "testFunc", 10, 1);
```

다음 정보가 [출력] 패널에 표시됩니다.

```
in for loop i=0  
in for loop i=1  
end of for loop  
in testFunc: 1st arg: 10 2nd arg: 1
```

또한 다음과 같이 함수를 실행하지 않고 testScript.jsfl만 호출할 수도 있습니다.

```
fl.runScript("file:///C:/testScript.jsfl");
```

그러면 [출력] 패널에 다음이 표시됩니다.

```
in for loop i=0  
in for loop i=1  
end of for loop
```

fl.saveAll()

지원 버전

Flash MX 2004

사용법

```
fl.saveAll()
```

매개 변수

없음

반환값

없음

설명

메서드, 열려 있는 모든 문서를 저장합니다.

파일을 저장한 적이 없거나 마지막으로 저장한 후 수정하지 않은 경우 파일이 저장되지 않습니다. 저장되지 않거나 수정되지 않은 파일이 저장되도록 하려면 [fl.saveDocumentAs\(\)](#)를 사용합니다.

예제

다음 예제에서는 이전에 저장된 적이 있고 마지막으로 저장된 이후 수정된 열려 있는 문서를 모두 저장합니다.

```
fl.saveAll();
```

참고 사항

[document.save\(\)](#), [document.saveAndCompact\(\)](#), [fl.saveDocument\(\)](#), [fl.saveDocumentAs\(\)](#)

fl.saveDocument()

지원 버전

Flash MX 2004

사용법

```
fl.saveDocument(document [, fileURI])
```

매개 변수

document 저장할 문서를 지정하는 [Document](#) 객체. **document**가 null이면 활성 문서가 저장됩니다.

fileURI 저장할 문서의 이름을 지정하는 문자열([file:/// URI](#)로 표시됨). **fileURI** 매개 변수가 null이거나 생략되면 문서가 현재 이름으로 저장됩니다. 이 매개 변수는 선택 사항입니다.

반환값

부울 값, 저장 작업이 성공적으로 완료되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

파일을 저장한 적이 없거나 마지막으로 저장한 후 수정하지 않은 경우 파일이 저장되지 않고 false가 반환됩니다. 저장되지 않거나 수정되지 않은 파일이 저장되도록 하려면 [fl.saveDocumentAs\(\)](#)를 사용합니다.

설명

메서드, 지정된 문서를 FLA 문서로 저장합니다.

예제

다음 예제에서는 현재 문서와 지정된 두 문서를 저장합니다.

```
// 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"));
```

참고 사항

[document.save\(\)](#), [document.saveAndCompact\(\)](#), [fl.saveAll\(\)](#), [fl.saveDocumentAs\(\)](#)

fl.saveDocumentAs()

지원 버전
Flash MX 2004

사용법

```
fl.saveDocumentAs (document)
```

매개 변수

document 저장할 문서를 지정하는 **Document** 객체. **document**가 null이면 활성 문서가 저장됩니다.

반환값

부울 값, 다른 이름으로 저장 작업이 성공적으로 완료되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 지정된 문서를 저장하기 위한 [다른 이름으로 저장] 대화 상자를 표시합니다.

예제

다음 예제에서는 지정된 문서를 저장하라는 대화 상자를 표시한 후 이 문서가 저장되었는지 여부를 나타내는 경고 메시지를 표시합니다.

```
alert (fl.saveDocumentAs (fl.documents [1]));
```

참고 사항

[document.save\(\)](#), [document.saveAndCompact\(\)](#), [fl.saveAll\(\)](#), [fl.saveDocument\(\)](#)

fl.scriptURI

지원 버전
Flash CS3 Professional

사용법

```
fl.scriptURI
```

설명

읽기 전용 속성, 현재 실행 중인 JSFL 스크립트의 경로를 나타내는 문자열(file:/// URI로 표시됨)입니다. 스크립트가 [fl.runScript\(\)](#)에서 호출된 경우 이 속성은 바로 위 부모 스크립트의 경로를 나타냅니다. 즉, 원래 호출 스크립트의 경로를 찾기 위해 여러 [fl.runScript\(\)](#) 호출을 순회하지 않습니다.

예제

다음 예제에서는 현재 실행 중인 JSFL 스크립트의 경로를 [출력] 패널에 표시합니다.

```
fl.trace (fl.scriptURI);
```

참고 사항

[fl.runScript\(\)](#)

fl.selectElement()

지원 버전

Flash CS3 Professional

사용법

```
fl.selectElement(elementObject, editMode)
```

매개 변수

elementObject 선택할 [Element](#) 객체

editMode 요소를 편집할 것인지(true), 아니면 선택만 할 것인지(false)를 지정하는 부울 값

반환값

요소가 성공적으로 선택되면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값

설명

메서드, 요소의 선택 또는 편집을 활성화합니다. 일반적으로 [fl.findObjectInDocByName\(\)](#) 또는 [fl.findObjectInDocByType\(\)](#)에서 반환된 객체에서 이 메서드를 사용합니다.

예제

다음 예제에서는 문서에 "second text field"라는 요소가 있으면 이 요소를 선택합니다.

```
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");  
}
```

참고 사항

[fl.findObjectInDocByName\(\)](#), [fl.findObjectInDocByType\(\)](#)

fl.selectTool()

지원 버전

Flash CS3 Professional

사용법

```
fl.selectTool(toolName)
```

매개 변수

toolName 선택할 도구의 이름을 지정하는 문자열. 이 매개 변수에 사용할 수 있는 값을 보려면 다음 "설명"을 참조하십시오.

설명

메서드, [도구] 패널에서 지정된 도구를 선택합니다. **toolName**에 사용할 수 있는 기본값에는 "arrow", "bezierSelect", "freeXform", "fillXform", "lasso", "pen", "penplus", "penminus", "penmodify", "text", "line", "rect", "oval", "rectPrimitive", "ovalPrimitive", "polystar", "pencil", "brush", "inkBottle", "bucket", "eyeDropper", "eraser", "hand" 및 "magnifier"가 있습니다.

사용자가 만든 사용자 정의 도구가 있는 경우에는 해당 도구의 이름도 **toolName** 매개 변수로 전달할 수 있습니다. 도구 이름 목록은 다음 파일에 있습니다.

- Windows Vista:

부트 드라이브\Users\사용자 이름\Local Settings\Application Data\Adobe\Flash CS3\언어
\Configuration\Tools\toolConfig.xml

- Windows XP:

부트 드라이브\Documents and Settings\사용자 이름\Local Settings\Application Data\Adobe\Flash CS3\언어
\Configuration\Tools\toolConfig.xml

- Mac OS X:

Macintosh HD/Users/사용자 이름/Library/Application Support/Adobe/Flash CS3/언어
/Configuration/Tools/toolConfig.xml

예제

다음 예제에서는 [펜 도구]를 선택합니다.

```
fl.selectTool("pen");
```

참고 사항

[Tools 객체](#), [ToolObj 객체](#)

fl.setActiveWindow()

지원 버전

Flash MX 2004

사용법

```
fl.setActiveWindow(document [, bActivateFrame])
```

매개 변수

document 활성 윈도우로 선택할 문서를 지정하는 [Document 객체](#)

bActivateFrame Flash 및 Fireworks에서는 더 이상 사용되지 않고 Dreamweaver와의 호환성을 위해서만 제공되는 선택적 매개 변수

반환값

없음

설명

메서드, 활성 윈도우를 지정된 문서가 되도록 설정합니다. 이 메서드는 Dreamweaver 및 Fireworks에서도 지원됩니다. [윈도우] > [창 복제]로 만든 뷰가 문서에 여러 개 있는 경우 가장 최근에 활성화된 뷰가 선택됩니다.

예제

다음 예제에서는 지정된 문서를 활성화하는 두 가지 방법을 보여 줍니다.

```
fl.setActiveWindow(fl.documents[0]);  
  
var theIndex = fl.findDocumentIndex("myFile.fla");  
fl.setActiveWindow(fl.documents[theIndex]);
```

fl.showIdleMessage()

지원 버전

Flash 8

사용법

```
fl.showIdleMessage(show)
```

매개 변수

show 너무 오래 실행되는 스크립트에 대한 경고 메시지를 표시할 것인지 여부를 지정하는 부울 값

반환값

없음

설명

메서드, show에 대해 **false**를 전달하여 너무 오래 실행되는 스크립트에 대한 경고 메시지가 표시되지 않도록 합니다. 이 메서드는 완료하는 데 시간이 오래 걸리는 일괄 처리 작업을 수행할 때 사용할 수 있습니다. 경고 메시지가 다시 표시되도록 하려면 명령을 다시 실행하고 이번에는 show에 대해 **true**를 전달합니다.

예제

다음 예제에서는 너무 오래 실행되는 스크립트에 대한 경고가 표시되지 않도록 했다가 다시 표시되도록 하는 방법을 보여 줍니다.

```
fl.showIdleMessage(false);  
var result = timeConsumingFunction();  
fl.showIdleMessage(true); ;  
var result = timeConsumingFunction();
```

fl.sourcePath

지원 버전

Flash CS4 Professional

사용법

```
fl.sourcePath
```

설명

속성, ActionScript 클래스 파일의 위치를 지정하는 전역 ActionScript 3.0 소스 경로에 있는 항목의 목록이 포함된 문자열입니다. 문자열에 있는 항목은 세미콜론으로 구분됩니다. 제작 도구에서 [편집] > [환경 설정] > [ActionScript] > [ActionScript 3.0 설정]을 선택하여 항목을 지정할 수 있습니다.

예제

다음 예제에서는 /Classes 폴더를 전역 ActionScript 3.0 소스 경로에 추가합니다.

```
fl.trace(fl.sourcePath);  
fl.sourcePath = "/Classes;" + fl.sourcePath;  
fl.trace(fl.sourcePath);
```

참고 사항

[fl.flexSDKPath](#), [fl.externalLibraryPath](#), [fl.libraryPath](#), [document.sourcePath](#)

fl.swfPanels

지원 버전

Flash CS4 Professional

사용법

```
fl.swfPanels
```

설명

읽기 전용 속성, 등록된 swfPanel 객체의 배열([swfPanel 객체 참조](#))입니다. swfPanel 객체는 한 번 이상 열린 경우 등록됩니다.

배열에서 패널의 위치는 패널이 열린 순서를 나타냅니다. 첫 번째로 열린 패널이 TraceBitmap이고 두 번째로 열린 패널이 AnotherFunction인 경우 fl.swfPanels[0]는 TraceBitmap swfPanel 객체이고, fl.swfPanels[1]는 AnotherFunction swfPanel 객체입니다.

예제

다음 코드에서는 등록된 모든 윈도우 SWF 패널의 이름과 경로를 [출력] 패널에 표시합니다.

```
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()

지원 버전

Flash Professional CS5

사용법

```
fl.toggleBreakPoint(String fileURI, int line, Boolean enable)
```

매개 변수

fileURI 문자열, 중단점을 전환할 AS 파일의 URI

행 정수, 중단점을 전환할 행 번호

`enable` 부울, `true`로 설정되면 중단점이 활성화되고 `false`로 설정되면 중단점이 비활성화됩니다.

설명

지정된 행에서 지정된 `.as` 파일의 중단점을 전환합니다. `enable`이 `false`이면 해당 행에 현재 저장된 중단점이 삭제됩니다.

예제

다음 예제에서는 `C:\AS\breakpointTest.as`에 있는 AS 파일의 10행에서 중단점을 활성화합니다.

```
fl.toggleBreakPoint("file:///C:/AS/breakpointTest.as", 10, 1);
```

fl.tools

지원 버전

Flash MX 2004

사용법

```
fl.tools
```

설명

읽기 전용 속성, Tools 객체의 배열([Tools 객체](#))입니다. 이 속성은 확장 가능한 도구를 만들 때만 사용됩니다.

fl.trace()

지원 버전

Flash MX 2004

사용법

```
fl.trace(message)
```

매개 변수

`message` [출력] 패널에 표시되는 문자열

반환값

없음

설명

메서드, 새 행으로 끝나는 텍스트 문자열을 [출력] 패널로 보내고 아직 표시되지 않은 경우 [출력] 패널에 표시합니다. 이 메서드는 [outputPanel.trace\(\)](#)와 동일하며 `ActionScript`의 `trace()` 문과 같은 방식으로 작동합니다.

빈 행을 보내려면 `fl.trace("")` 또는 `fl.trace("\n")`를 사용합니다. 두 번째 명령 인라인을 사용하여 `\n`을 `message` 문자열의 일부로 만들 수 있습니다.

예제

다음 예제에서는 여러 텍스트 행을 [출력] 패널에 표시합니다.


```
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

지원 버전
Flash MX 2004

사용법
fl.version

설명
읽기 전용 속성, 플랫폼을 포함하는 Flash 제작 도구의 긴 문자열 버전입니다.

예제
다음 예제에서는 Flash 제작 도구의 버전을 [출력] 패널에 표시합니다.

```
alert(fl.version); // For example, WIN 10,0,0,540
```

fl.xmlui

지원 버전
Flash MX 2004

사용법
fl.xmlui

설명
읽기 전용 속성, [XMLUI 객체](#)입니다. 이 속성을 사용하면 XMLUI 대화 상자의 XMLUI 속성을 가져와서 설정할 수 있으며 대화 상자를 프로그래밍 방식으로 적용하거나 취소할 수 있습니다.

예제
[XMLUI 객체](#)를 참조하십시오.

18장: FLfile 객체

지원 버전

Flash MX 2004 7.2

설명

FLfile 객체를 사용하면 로컬 파일 시스템에 있는 파일과 폴더를 액세스, 수정 및 제거할 수 있는 Flash 확장을 작성할 수 있습니다. FLfile API는 JavaScript API에 대한 확장 형태로 제공됩니다. 공유 라이브러리라고도 하는 이 확장은 다음 폴더에 있습니다.

- Windows Vista:

부트 드라이브\Users\사용자 이름\Local Settings\Application Data\Adobe\Flash CS3\언어\Configuration\External Libraries\FLfile.dll

- Windows XP:

부트 드라이브\Documents and Settings\사용자 이름\Local Settings\Application Data\Adobe\Flash CS3\언어\Configuration\External Libraries\FLfile.dll

- Mac OS X:

Macintosh HD/Users/사용자 이름/Library/Application Support/Adobe/Flash CS3/언어/Configuration/External Libraries/FLfile.dll

참고: Flash 문서의 심볼을 포함하는 공유 라이브러리와 JavaScript API 공유 라이브러리는 서로 다른 것이므로 혼동하면 안 됩니다.

FLfile 메서드는 디스크에 있는 파일이나 폴더(디렉토리)에 사용되므로 각 FLfile 메서드에는 파일 또는 폴더의 위치를 지정하는 매개 변수가 하나 이상 있습니다. 파일 또는 폴더의 위치는 웹 사이트 URL과 매우 유사한 형식의 문자열로 표시됩니다. 이러한 문자열은 파일URI(Uniform Resource Identifier)라고 하며 다음과 같은 형식(따옴표포함)으로 되어 있습니다.

```
"file:///drive|/folder 1/folder 2/.../filename"
```

예를 들어 C드라이브에 config라는 폴더를 만들고 이 폴더를 Program Files/MyApp 폴더에 배치하려면 다음 명령을 사용합니다.

```
FLfile.createFolder("file:///C:/Program Files/MyApp/config");
```

config.ini라는 파일을 config 폴더에 배치하려면 다음 명령을 사용합니다.

```
FLfile.write("file:///C:/Program Files/MyApp/config/config.ini", "");
```

Macintosh에 폴더를 만들려면 다음 명령을 사용할 수 있습니다.

```
FLfile.createFolder("file:///Macintosh/MyApp/config");
```

메서드 요약

FLfile 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
FLfile.copy()	파일을 복사합니다.
FLfile.createFolder()	하나 이상의 폴더를 만듭니다.
FLfile.exists()	파일이나 폴더가 있는지 여부를 확인합니다.
FLfile.getAttributes()	파일이 쓰기 가능, 읽기 전용, 숨김, 표시 또는 시스템 폴더인지를 확인합니다.

메서드	설명
<code>FLfile.getCreationDate()</code>	1970년 1월 1일부터 파일이나 폴더를 만든 날짜까지 경과된 시간(초)을 지정합니다.
<code>FLfile.getCreationDateObj()</code>	파일이나 폴더를 만든 날짜를 가져옵니다.
<code>FLfile.getModificationDate()</code>	1970년 1월 1일부터 파일이나 폴더를 가장 최근 수정한 날짜까지 경과된 시간(초)을 지정합니다.
<code>FLfile.getModificationDateObj()</code>	파일이나 폴더를 가장 최근 수정한 날짜를 가져옵니다.
<code>FLfile.getSize()</code>	파일의 크기를 가져옵니다.
<code>FLfile.listFolder()</code>	폴더의 내용을 나열합니다.
<code>FLfile.platformPathToURI()</code>	플랫폼별 형식으로 된 파일 이름을 <code>file:/// URI</code> 형식으로 변환합니다.
<code>FLfile.read()</code>	파일의 내용을 읽습니다.
<code>FLfile.remove()</code>	파일이나 폴더를 삭제합니다.
<code>FLfile.setAttributes()</code>	파일이나 폴더를 읽기 전용, 쓰기 가능, 숨김 또는 표시로 만듭니다.
<code>FLfile.uriToPlatformPath()</code>	<code>file:/// URI</code> 로 표현된 파일 이름을 플랫폼별 형식으로 변환합니다.
<code>FLfile.write()</code>	파일을 만들거나, 파일에 쓰거나, 파일에 추가합니다.

FLfile.copy()

지원 버전
Flash MX 2004 7.2

사용법

`FLfile.copy(fileURI, copyURI)`

매개 변수

`fileURI` 복사할 파일을 지정하는 문자열(`file:/// URI`로 표시됨)

`copyURI` 복사한 파일의 위치와 이름을 지정하는 문자열(`file:/// URI`로 표시됨)

반환값

성공적으로 실행되면 `true`를 반환하고 그렇지 않으면 `false`를 반환하는 부울 값

설명

메서드, 한 위치에서 다른 위치로 파일을 복사합니다. `copyURI`가 이미 있으면 `false`가 반환됩니다.

예제

다음 예제에서는 `config.ini`라는 구성 파일의 백업 복사본을 만들고 이 파일을 원래 파일이 있는 동일한 폴더에 새 이름으로 저장합니다.

```
var originalFileURI="file:///C:/Program Files/MyApp/config.ini";
var newFileURI="file:///C:/Program Files/MyApp/config_backup.ini";
FLfile.copy(originalFileURI, newFileURI);
```

원하는 경우 이 작업을 하나의 명령으로 수행할 수 있습니다.

```
FLfile.copy("file:///C|:/Program Files/MyApp/config.ini", file:///C|/Program  
Files/MyApp/config_backup.ini");
```

FLfile.createFolder()

지원 버전

Flash MX 2004 7.2

사용법

```
FLfile.createFolder(folderURI)
```

매개 변수

folderURI 만들 폴더 구조를 지정하는 폴더 URI

반환값

성공적으로 실행되면 **true**를 반환하고 folderURI가 이미 있으면 **false**를 반환하는 부울 값

설명

메서드, 지정된 위치에 하나 이상의 폴더를 만듭니다.

동시에 여러 개의 폴더를 만들 수 있습니다. 예를 들어 다음 명령은 MyData 및 TempData 폴더가 아직 없는 경우 이 두 폴더를 만듭니다.

```
FLfile.createFolder("file:///c|/MyData/TempData")
```

예제

다음 예제에서는 Configuration 폴더(fl.configURI)에 폴더와 하위 폴더를 만듭니다.

```
fl.trace(FLfile.createFolder(fl.configURI+"folder01/subfolder01"));
```

다음 예제에서는 C드라이브의 루트에 tempFolder라는 폴더를 만들려고 시도하고 작업이 성공했는지 나타내는 경고 상자를 표시합니다.

```
var folderURI = "file:///c|/tempFolder";  
if (FLfile.createFolder(folderURI)) {  
    alert("Created " + folderURI);  
}  
else {  
    alert(folderURI + " already exists");  
}
```

참고 사항

[FLfile.remove\(\)](#), [FLfile.write\(\)](#)

FLfile.exists()

지원 버전

Flash MX 2004 7.2

사용법

`FLfile.exists(fileURI)`

매개 변수

fileURI 확인할 파일을 지정하는 문자열(file:/// URI로 표시됨)

반환값

성공적으로 실행되면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값

설명

메서드, 지정된 파일이 있는지 여부를 확인합니다. 폴더 또는 파일 이름을 지정하려면 폴더가 이미 있어야 합니다. 폴더를 만들려면 [FLfile.createFolder\(\)](#)를 참조하십시오.

예제

다음 예제에서는 Temp 폴더에 mydata.txt라는 파일이 있는지 여부를 확인하고 그 결과를 나타내는 경고 상자를 표시합니다.

```
var fileURI = "file:///c:/temp/mydata.txt";
if (FLfile.exists(fileURI)) {
    alert( fileURI + " exists.");
}
else {
    alert( fileURI + " does not exist.");
}
```

다음 예제에서는 MyApplication 폴더에 필요한 구성 파일이 있는지 여부를 확인하고 파일이 없으면 자동으로 만듭니다.

```
var configFile = "file:///C:/MyApplication/config.ini";
if (!FLfile.exists(configFile)) {
    FLfile.write(configFile, "");
}
}
```

참고 사항

[FLfile.write\(\)](#)

FLfile.getAttributes()

지원 버전

Flash MX 2004 7.2

사용법

`FLfile.getAttributes(fileOrFolderURI)`

매개 변수

fileOrFolderURI 특성을 가져올 파일이나 폴더를 지정하는 문자열(file:/// URI로 표시됨)

반환값

지정된 파일 또는 폴더의 특성을 나타내는 문자열

파일이나 폴더가 없으면 예기치 못한 결과가 발생할 수 있습니다. 이 메서드를 사용하려면 먼저 [FLfile.exists\(\)](#)를 사용해야 합니다.

설명

메서드, 지정된 파일 또는 폴더의 특성을 나타내는 문자열을 반환하거나 이 파일에 특정 특성이 없을 경우(즉, 읽기 전용 또는 숨김이 아닌 경우) 빈 문자열을 반환합니다. 이 메서드를 사용하기 전에는 항상 [FLfile.exists\(\)](#)를 사용하여 파일 또는 폴더가 있는지 테스트해야 합니다.

문자열 내의 각 문자는 다음과 같은 특성을 나타냅니다.

- R - **fileOrFolderURI**는 읽기 전용입니다.
- D - **fileOrFolderURI**는 폴더(디렉토리)입니다.
- H - **fileOrFolderURI**는 숨겨져 있습니다(Windows에만 해당).
- S - **fileOrFolderURI**는 시스템 파일 또는 폴더입니다(Windows에만 해당).
- A - **fileOrFolderURI**는 압축할 준비가 되어 있습니다(Windows에만 해당).

예를 들어 **fileOrFolderURI**가 숨겨진 폴더이면 "DH" 문자열이 반환됩니다.

예제

다음 예제에서는 `mydata.txt` 파일의 특성을 가져오고 파일이 읽기 전용이면 경고 상자를 표시합니다.

```
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!");
    }
}
```

참고 사항

[FLfile.setAttributes\(\)](#)

FLfile.getCreationDate()

지원 버전

Flash MX 2004 7.2

사용법

```
FLfile.getCreationDate(fileOrFolderURI)
```

매개 변수

`fileOrFolderURI` 만든 날짜 및 시간을 16진수 문자열로 가져올 파일 또는 폴더를 지정하는 문자열(`file:/// URI`로 표시됨)

반환값

1970년 1월 1일부터 파일이나 폴더를 만든 날짜까지 경과된 시간(초)을 나타내는 16진수를 포함하거나 파일이나 폴더가 없는 경우 "00000000"을 포함하는 문자열

설명

메서드, 1970년 1월 1일부터 파일 또는 폴더를 만든 날짜까지 경과된 시간(초)을 지정합니다. 이 메서드는 주로 파일 또는 폴더의 수정 날짜나 만든 날짜를 비교하기 위해 사용됩니다.

예제

다음 예제에서는 파일이 만들어진 후 수정되었는지 여부를 확인합니다.

```
// 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.");
}
```

참고 사항

[FLfile.getCreationDateObj\(\)](#), [FLfile.getModificationDate\(\)](#)

FLfile.getCreationDateObj()

지원 버전

Flash MX 2004 7.2

사용법

```
FLfile.getCreationDateObj(fileOrFolderURI)
```

매개 변수

fileOrFolderURI 만든 날짜 및 시간을 JavaScript Date 객체로 가져올 파일 또는 폴더를 지정하는 문자열(file:/// URI로 표시됨)

반환값

지정된 파일 또는 폴더를 만든 날짜와 시간을 나타내는 JavaScript Date 객체. 파일이 없으면 객체에 1969년 12월 31일 자정(그리니치 표준시)에 만든 파일 또는 폴더를 나타내는 정보가 포함됩니다.

설명

메서드, 지정된 파일 또는 폴더를 만든 날짜 및 시간을 나타내는 JavaScript Date 객체를 반환합니다.

예제

다음 예제에서는 사용자가 읽을 수 있는 형식으로 [출력] 패널에 파일을 만든 날짜를 표시합니다.

```
// Make sure the specified file exists.
var file1Date = FLfile.getCreationDateObj("file:///c:/temp/file1.txt");
fl.trace(file1Date);
```

참고 사항

[FLfile.getCreationDate\(\)](#), [FLfile.getModificationDateObj\(\)](#)

FLfile.getModificationDate()

지원 버전

Flash MX 2004 7.2

사용법

FLfile.getModificationDate(fileOrFolderURI)

매개 변수

fileOrFolderURI 수정 날짜 및 시간을 16진수 문자열로 가져올 파일을 지정하는 문자열(file:/// URI로 표시됨)

반환값

1970년 1월 1일부터 파일이나 폴더를 마지막으로 수정한 날짜까지 경과된 시간(초)을 나타내는 16진수를 포함하거나 파일이 없는 경우 "00000000"을 포함하는 문자열

설명

메서드, 1970년 1월 1일부터 파일 또는 폴더를 마지막으로 수정한 날짜까지 경과된 시간(초)을 지정합니다. 이 메서드는 주로 파일 또는 폴더의 수정 날짜나 만든 날짜를 비교하기 위해 사용됩니다.

예제

다음 예제에서는 두 파일의 수정 날짜를 비교하고 둘 중 가장 최근에 수정된 파일이 어느 것인지 확인합니다.

```
// 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") ;
}
```

참고 사항

[FLfile.getCreationDate\(\)](#), [FLfile.getModificationDateObj\(\)](#)

FLfile.getModificationDateObj()

지원 버전

Flash MX 2004 7.2

사용법

FLfile.getModificationDateObj(fileOrFolderURI)

매개 변수

fileOrFolderURI 수정 날짜 및 시간을 JavaScript Date 객체로 가져올 파일 또는 폴더를 지정하는 문자열(file:/// URI로 표시됨)

반환값

지정된 파일 또는 폴더를 마지막으로 수정한 날짜와 시간을 나타내는 JavaScript Date 객체. 파일이나 폴더가 없으면 객체에 1969년 12월 31일 자정(그리니치 표준시)에 만든 파일 또는 폴더를 나타내는 정보가 포함됩니다.

설명

메서드, 지정된 파일 또는 폴더를 마지막으로 수정한 날짜와 시간을 나타내는 JavaScript Date 객체를 반환합니다.

예제

다음 예제에서는 [출력] 패널에 파일을 마지막으로 수정한 날짜를 사용자가 읽을 수 있는 형식으로 표시합니다.

```
// Make sure the specified file exists.  
var file1Date = FLfile.getModificationDateObj("file:///c:/temp/file1.txt");  
trace(file1Date);
```

참고 사항

[FLfile.getCreationDateObj\(\)](#), [FLfile.getModificationDate\(\)](#)

FLfile.getSize()

지원 버전

Flash MX 2004 7.2

사용법

```
FLfile.getSize(fileURI)
```

매개 변수

fileURI 크기를 가져올 파일을 지정하는 문자열(file:/// URI로 표시됨)

반환값

지정된 파일의 크기를 바이트 단위로 나타내거나 파일이 없으면 0으로 나타나는 정수

설명

메서드, 지정된 파일의 크기를 바이트 단위로 나타내거나 파일이 없으면 0으로 나타나는 정수를 반환합니다. 반환값이 0이면 [FLfile.exists\(\)](#)를 사용하여 파일이 0바이트인지, 아니면 파일이 없는지를 확인할 수 있습니다.

이 메서드는 크기가 2GB 이하인 파일에 대해서만 올바른 파일 크기를 반환합니다.

예제

다음 예제에서는 fileSize 변수에 mydata.txt 파일의 크기를 저장합니다.

```
var URL = "file:///c:/temp/mydata.txt";  
var fileSize = FLfile.getSize(URL);
```

FLfile.listFiles()

지원 버전

Flash MX 2004 7.2

사용법

```
FLfile.listFiles(folderURI [, filesOrDirectories])
```

매개 변수

folderURI 내용을 검색할 폴더를 지정하는 문자열(**file:/// URI**로 표시됨). 와일드카드 마스크를 **folderURI**의 일부로 포함할 수 있습니다. 유효한 와일드카드는 *(하나 이상의 문자와 일치) 및 ?(한 문자와 일치)입니다.

filesOrDirectories 파일 이름만 반환할지, 아니면 폴더(디렉토리) 이름만 반환할지를 지정하는 선택적 문자열. 생략하면 파일 이름과 폴더 이름이 모두 반환됩니다. 사용할 수 있는 값에는 "files" 및 "directories"가 있습니다.

반환값

폴더의 내용을 나타내는 문자열 배열입니다. 폴더가 없거나 지정된 조건에 맞는 파일이나 폴더가 없는 경우 빈 배열을 반환합니다.

설명

메서드, 폴더의 내용을 나타내는 문자열 배열을 반환합니다.

예제

다음 예제에서는 세 가지 배열을 반환합니다. 첫 번째 배열은 C:\temp 폴더의 모든 파일을, 두 번째 배열은 C:\temp 폴더의 모든 폴더를, 세 번째 배열은 C:\temp 폴더의 파일 및 폴더를 나타냅니다.

```
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);
```

다음 예제에서는 temp 폴더에 있는 모든 텍스트(.txt) 파일의 배열을 반환하고 경고 상자에 이 목록을 표시합니다.

```
var folderURI = "file:///c:/temp";
var fileMask = "*.txt";
var list = FLfile.listFolder(folderURI + "/" + fileMask, "files");
if (list) {
    alert(folderURI + " contains: " + list.join(" "));
}
```

다음 예제에서는 지정된 **folderURI**에서 파일 마스크를 사용하여 Windows 응용 프로그램 폴더에 있는 모든 실행 파일의 이름을 반환합니다.

```
var executables = FLfile.listFolder("file:///C:/WINDOWS/*.exe", "files");
alert(executables.join("\n"));
```

FLfile.platformPathToURI()

지원 버전

Flash CS4 Professional

사용법

```
FLfile.platformPathToURI(fileName)
```

매개 변수

fileName 문자열, 변환할 파일 이름을 지정하는 플랫폼별 형식으로 표현됩니다.

반환값

file:/// URI로 표현된 문자열

설명

메서드, 플랫폼별 형식으로 된 파일 이름을 file:/// URI 형식으로 변환합니다.

예제

다음 예제에서는 플랫폼별 형식으로 된 파일 이름을 file:/// URI 형식으로 변환합니다. 이 파일 이름은 `outputPanel.save()`에 전달됩니다.

```
var myFilename = "C:\\outputPanel.txt";  
var myURI=FLfile.platformPathToURI(myFilename);  
fl.outputPanel.save(myURI);
```

참고 사항

[FLfile.uriToPlatformPath\(\)](#)

FLfile.read()

지원 버전

Flash MX 2004 7.2

사용법

```
FLfile.read()
```

매개 변수

fileOrFolderURI 특성을 가져올 파일이나 폴더를 지정하는 문자열(file:/// URI로 표시됨)

반환값

지정된 파일의 내용을 문자열로 반환하거나 파일 읽기에 실패하면 null을 반환합니다.

설명

메서드, 지정된 파일의 내용을 문자열로 반환하거나 파일 읽기에 실패하면 null을 반환합니다.

예제

다음 예제에서는 `mydata.txt` 파일을 읽고 파일 읽기에 성공하면 경고 상자에 파일의 내용을 표시합니다.

```
var fileURI = "file:///c:/temp/mydata.txt";  
var str = FLfile.read( fileURI);  
if (str) {  
    alert( fileURL + " contains: " + str);  
}
```

다음 예제에서는 클래스 파일에서 `ActionScript` 코드를 읽고 이를 `code` 변수에 저장합니다.

```
var classFileURI = "file:///C:/MyApplication/TextCarousel.as";  
var code = FLfile.read(classFileURI);
```

FLfile.remove()

지원 버전

Flash MX 2004 7.2

사용법

```
FLfile.remove(fileOrFolderURI)
```

매개 변수

fileOrFolderURI 제거(삭제)할 파일이나 폴더를 지정하는 문자열(file:/// URI로 표시됨)

반환값

성공적으로 실행되면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값

설명

메서드, 지정된 파일이나 폴더를 삭제합니다. 폴더에 파일이 포함되어 있으면 파일도 함께 삭제됩니다. R(읽기 전용) 특성을 가진 파일은 제거할 수 없습니다.

예제

다음 예제에서는 파일이 있으면 경고 메시지를 표시한 다음 사용자가 삭제하도록 선택하면 파일을 삭제합니다.

```
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");
}
```

다음 예제에서는 응용 프로그램에서 만든 구성 파일을 삭제합니다.

```
if(FLfile.remove("file:///C:/MyApplication/config.ini")) {
    alert("Configuration file deleted");
}
```

다음 예제에서는 Configuration 폴더와 해당 내용을 삭제합니다.

```
FLfile.remove("file:///C:/MyApplication/Configuration/");
```

참고 사항

[FLfile.createFolder\(\)](#), [FLfile.getAttributes\(\)](#)

FLfile.setAttributes()

지원 버전

Flash MX 2004 7.2

사용법

```
FLfile.setAttributes(fileURI, strAttrs)
```

매개 변수

fileURI 특성을 설정할 파일을 지정하는 문자열(**file:///** URI로 표시됨)

strAttrs 설정할 특성의 값을 지정하는 문자열. **strAttrs**에 사용할 수 있는 값에 대한 자세한 내용은 다음 "설명" 단원을 참조하십시오.

반환값

성공적으로 실행되면 **true**를 반환하는 부울 값

참고: 파일이나 폴더가 없으면 예기치 못한 결과가 발생할 수 있습니다. 이 메서드를 사용하려면 먼저 [FLfile.exists\(\)](#)를 사용해야 합니다.

설명

메서드, 지정된 파일의 시스템 레벨 특성을 지정합니다.

strAttrs에 사용할 수 있는 값은 다음과 같습니다.

- N - 특정 특성 없음(읽기 전용, 숨김 등이 아닌 경우)
- A - 압축 준비(Windows에만 해당)
- R - 읽기 전용(Macintosh에서 읽기 전용은 "잠김"을 의미함)
- W - 쓰기 가능(R보다 우선함)
- H - 숨김(Windows에만 해당)
- V - 표시(H보다 우선함, Windows에만 해당)

strAttrs에 R과 W를 둘 다 포함하면 R은 무시되고 파일은 쓰기 가능으로 설정됩니다. 마찬가지로 H와 V를 전달하면 H는 무시되고 파일은 표시로 설정됩니다.

Archive 특성이 설정되지 않도록 하려면 특성을 설정하기 전에 N 매개 변수와 함께 이 명령을 사용합니다. 즉, Archive 특성을 해제하는 A에 직접 대응하는 것은 없습니다.

예제

다음 예제에서는 **mydata.txt** 파일을 읽기 전용과 숨김으로 설정합니다. Archive 특성에는 영향을 주지 않습니다.

```
var URI = "file:///c:/temp/mydata.txt";  
if (FLfile.exists(URI)) {  
    FLfile.setAttributes(URI, "RH");  
}
```

다음 예제에서는 **mydata.txt** 파일을 읽기 전용과 숨김으로 설정합니다. 또한 Archive 특성이 설정되지 않도록 합니다.

```
var URI = "file:///c:/temp/mydata.txt";  
  
if (FLfile.exists(URI)) {  
    FLfile.setAttributes(URI, "N");  
    FLfile.setAttributes(URI, "RH");  
}
```

참고 사항

[FLfile.getAttributes\(\)](#)

FLfile.uriToPlatformPath()

지원 버전

Flash CS4 Professional

사용법

```
FLfile.uriToPlatformPath(fileURI)
```

매개 변수

fileURI 변환할 파일 이름을 지정하는 문자열(file:/// URI로 표시됨)

반환값

플랫폼별 경로를 나타내는 문자열

설명

메서드, file:/// URI로 표현된 파일 이름을 플랫폼별 형식으로 변환합니다.

예제

다음 예제에서는 file:/// URI 형식을 플랫폼별 형식으로 변환합니다.

```
var dir = (fl.configDirectory);  
var URI = FLfile.platformPathToURI(dir);  
fl.trace(URI == fl.configURI); // displays "true"
```

참고 사항

[FLfile.platformPathToURI\(\)](#)

FLfile.write()

지원 버전

Flash MX 2004 7.2

사용법

```
FLfile.write(fileURI, textToWrite, [ , strAppendMode])
```

매개 변수

fileURI 쓸 파일을 지정하는 문자열(file:/// URI로 표시됨)

textToWrite 파일에 쓸 텍스트를 나타내는 문자열

strAppendMode 기존 파일에 textToWrite를 추가할지를 지정하는 "append" 값이 포함된 선택적 문자열. 생략하면 textToWrite가 fileURI를 덮어쓰게 됩니다.

반환값

성공적으로 실행되면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값

설명

메서드, 지정된 파일(예: UTF-8)에 특정 문자열을 쓰고 지정된 파일이 없으면 자동으로 만듭니다. 그러나 이 메서드를 사용하려면 파일을 배치할 폴더가 있어야 합니다. 폴더를 만들려면 [FLfile.createFolder\(\)](#)를 사용합니다.

예제

다음 예제에서는 mydata.txt 파일에 "xxx" 문자열을 쓰려고 시도하고 쓰기가 성공하면 경고 메시지를 표시합니다. 그런 다음 파일에 "aaa"를 추가하려고 시도하고 쓰기 작업이 성공하면 두 번째 경고 메시지를 표시합니다. 이 스크립트를 실행하고 나면 mydata.txt 파일에 "xxxaaa" 텍스트만 남습니다.

```
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);
}
```

참고 사항

[FLfile.createFolder\(\)](#), [FLfile.exists\(\)](#)

19장: folderItem 객체

상속 [Item 객체](#) > folderItem 객체

지원 버전

Flash MX 2004

설명

folderItem 객체는 Item 객체의 하위 클래스입니다. folderItem의 고유한 메서드나 속성은 없습니다. [Item 객체](#)를 참조하십시오.

20장: fontItem 객체

상속 [Item 객체](#) > fontItem 객체

지원 버전
Flash MX 2004

설명
fontItem 객체는 Item 객체의 하위 클래스입니다([Item 객체](#) 참조).

속성 요약
Item 객체 속성 외에도 fontItem 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
fontItem.bitmap	글꼴 항목이 비트맵 글꼴인지 여부를 지정합니다.
fontItem.bold	글꼴 항목이 굵은체인지 여부를 지정합니다.
fontItem.embeddedCharacters	포함할 문자를 지정합니다.
fontItem.embedRanges	[글꼴 포함] 대화 상자에서 선택할 수 있는 항목을 지정합니다.
fontItem.embedVariantGlyphs	SWF 파일을 제작할 때 변형 글리프가 글꼴로 출력되어야 하는지 여부를 지정합니다.
fontItem.font	글꼴 항목과 연관된 장치 글꼴의 이름입니다.
fontItem.isDefineFont4Symbol	SWF 파일을 제작할 때 출력되는 글꼴 형식을 지정합니다.
fontItem.italic	글꼴 항목이 기울임체인지 여부를 지정합니다.
fontItem.size	글꼴 항목의 크기(포인트 단위)입니다.

fontItem.bitmap

지원 버전
Flash CS4 Professional

사용법
`fontItem.bitmap`

설명
속성, 글꼴 항목이 비트맵 글꼴인지(true), 아니면 비트맵 글꼴이 아닌지(false)를 지정하는 부울 값입니다.

예제
다음 코드에서는 라이브러리의 첫 번째 항목이 글꼴 항목이라고 가정하고 이 항목이 비트맵 글꼴이면 [출력] 패널에 true를 표시하고 그렇지 않으면 false를 표시합니다.

```
var theItem = fl.getDocumentDOM().library.items[0];
fl.trace("bitmap: "+ theItem.bitmap);
```

fontItem.bold

지원 버전

Flash CS4 Professional

사용법

fontItem.bold

설명

속성, 글꼴 항목이 굵은체인지(true), 아니면 굵은체가 아닌지(false)를 지정하는 부울 값입니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목이 글꼴 항목이라고 가정하고 이 항목이 굵은체이면 [출력] 패널에 true를 표시하고 그렇지 않으면 false를 표시한 다음 해당 항목을 굵은체로 설정합니다.

```
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

지원 버전

Flash CS5 Professional

사용법

fontItem.embeddedCharacters

설명

속성, SWF 파일에 포함할 문자를 지정하여 장치에 문자가 설치되어 있지 않아도 SWF 파일이 재생되도록 하는 문자열 값입니다. 이 속성은 [글꼴 포함] 대화 상자와 동일한 기능을 제공합니다.

또한 이 속성을 읽으면 [글꼴 포함] 대화 상자에서 해당 글꼴 항목에 대해 어떤 문자가 지정되었는지를 찾을 수 있습니다.

예제

라이브러리의 첫 번째 항목은 글꼴 항목이며 뒤에 오는 코드는 문자 a, b 및 c를 포함하는 것으로 가정합니다.

```
fl.getDocumentDOM().library.items[0].embeddedCharacters = "abc";
```

fontItem.embedRanges

지원 버전

Flash CS5 Professional

사용법

fontItem.embedRanges

설명

속성, [글꼴 포함] 대화 상자에서 선택할 수 있는 항목에 해당하는 일련의 정수(구분 기호로 분리됨)를 지정하는 문자열 값입니다. 또한 이 속성을 읽으면 [글꼴 포함] 대화 상자에서 해당 글꼴 항목에 대해 어떤 문자가 지정되었는지를 찾을 수 있습니다.

참고: 범위 숫자는 구성 폴더의 FontEmbedding/UnicodeTables.xml 파일에 해당합니다.

예제

라이브러리의 첫 번째 항목은 글꼴 항목이며 뒤에 오는 코드는 정수 1, 2 및 3으로 식별되는 범위를 포함하는 것으로 가정합니다.

```
fl.getDocumentDOM().library.items[0].embedRanges = "1|3|7";
```

라이브러리의 첫 번째 항목은 글꼴 항목이며 뒤에 오는 코드는 포함할 범위를 다시 설정합니다.

```
fl.getDocumentDOM().library.items[0].embedRanges = "";
```

fontItem.embedVariantGlyphs

지원 버전

Flash CS4 Professional

사용법

```
fontItem.embedVariantGlyphs
```

설명

참고: 이 속성은 Flash CS5 Professional에서 사용 가능하지만 TLF(Text Layout Framework)에 적용될 경우 아무런 효과도 발생하지 않습니다. Flash Professional CS5에서 시작하는 경우 다양한 글리프는 항상 TLF 텍스트를 사용하여 글꼴에 포함됩니다. 아래에서 참조되는 flash.text.engine(FTE)는 Flash Professional CS4에서만 사용 가능합니다.

속성, SWF 파일을 제작할 때 변형 글리프가 글꼴로 출력되어야 하는지(true), 아니면 글꼴로 출력되지 않아도 되는지(false) 지정하는 부울 값입니다. 이 값을 true로 설정하면 SWF 파일 크기가 커집니다. 기본값은 false입니다.

태국어, 아랍어, 히브리어 및 그리스어 등 일부 언어에서는 사용자가 입력하는 대로 문자 글리프를 동적으로 대체합니다. 이러한 유형의 언어에 텍스트를 배치하거나 입력하는 경우 이 속성을 true로 설정하십시오.

예제

flash.text API와 호환되는 글꼴 심볼은 라이브러리에 표시되며 사용자는 이를 직접 관리할 수 있습니다. 단, flash.text.engine(FTE) API와 호환되는 글꼴 심볼은 라이브러리에 나타나지 않으므로 수동으로 관리해야 합니다. 다음 함수는 FTE API와 함께 사용할 수 있는 라이브러리에 새 글꼴을 추가합니다.

```
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;
        }
    }
}
```

다음 함수는 [출력] 패널의 모든 글꼴 심볼을 표시합니다.

```
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);
            }
        }
    }
}
```

참고 사항

[fontItem.isDefineFont4Symbol](#), [text.embedVariantGlyphs](#)

fontItem.font

지원 버전

Flash CS4 Professional

사용법

fontItem.font

설명

속성, 글꼴 항목과 연관된 장치 글꼴의 이름을 지정하는 문자열입니다. 설치된 장치 글꼴에 해당하지 않는 문자열을 입력하면 오류 메시지가 표시됩니다. 글꼴이 시스템에 있는지 확인하려면 [fl.isFontInstalled\(\)](#)를 사용합니다.

참고: 이 값을 설정하면 생성되는 속성 값이 입력한 문자열과 다를 수 있습니다. 다음 예제를 참조하십시오.

예제

다음 코드에서는 라이브러리의 첫 번째 항목이 글꼴 항목이라고 가정하고 글꼴 항목과 현재 연관된 장치 글꼴의 이름을 표시한 다음 글꼴의 이름을 Times로 변경합니다.

```
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

지원 버전

Flash CS4 Professional

사용법

fontItem.isDefineFont4Symbol

설명

속성, SWF 파일을 제작할 때 출력되는 글꼴 형식을 지정하는 부울 값입니다. 이 값이 true이면 Flash는 flash.text.engine(FTE) API와 함께 사용할 수 있는 글꼴을 출력합니다. 반대로 이 값이 false이면 텍스트 필드와 같은 flash.text API와 함께 글꼴을 사용할 수 있습니다. 기본값은 false입니다.

예제

[fontItem.embedVariantGlyphs](#)를 참조하십시오.

fontItem.italic

지원 버전

Flash CS4 Professional

사용법

fontItem.italic

설명

속성, 글꼴 항목이 기울임체인지(true), 아니면 기울임체가 아닌지(false)를 지정하는 부울 값입니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목이 글꼴 항목이라고 가정하고 이 항목이 기울임체이면 [출력] 패널에 true를 표시하고 그렇지 않으면 false를 표시한 다음 해당 항목을 기울임체로 설정합니다.

```
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

지원 버전

Flash CS4 Professional

사용법

fontItem.size

설명

속성, 글꼴 항목의 크기를 포인트 단위로 나타내는 정수입니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목이 글꼴 항목이라고 가정하고 이 항목의 포인트 크기를 [출력] 패널에 표시한 다음 포인트 크기를 24로 설정합니다.

```
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);
```

21장: Frame 객체

지원 버전
Flash MX 2004

설명
Frame 객체는 레이어 내의 프레임을 나타냅니다.

메서드 요약
Frame 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
<code>frame.convertMotionObjectTo2D()</code>	선택한 모션 객체를 2D 모션 객체로 변환합니다.
<code>frame.convertMotionObjectTo3D()</code>	선택한 모션 객체를 3D 모션 객체로 변환합니다.
<code>frame.getCustomEase()</code>	JavaScript 객체의 배열을 반환합니다. 각 객체에는 x 및 y 속성이 있습니다.
<code>frame.getMotionObjectXML()</code>	선택한 모션 객체에서 모션 XML을 반환합니다.
<code>frame.hasMotionPath()</code>	현재 선택 항목에 모션 트윈이 있는지 여부를 알려 줍니다.
<code>frame.is3DMotionObject()</code>	현재 선택 항목에 3D 모션 객체가 있는지 여부를 알려 줍니다.
<code>frame.isMotionObject()</code>	현재 선택 항목에 모션 객체가 있는지 여부를 알려 줍니다.
<code>frame.selectMotionPath()</code>	현재 모션 객체의 모션 경로를 선택하거나 선택 취소합니다.
<code>frame.setCustomEase()</code>	사용자 정의 여유 곡선으로 사용할 3차원 베지어 곡선을 지정합니다.
<code>frame.setMotionObjectDuration()</code>	현재 선택한 모션 객체의 지속 기간(트윈 범위 길이)을 지정합니다.
<code>frame.setMotionObjectXML()</code>	선택한 모션 객체에 선택한 모션 XML을 적용합니다.

속성 요약
Frame 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>frame.actionScript</code>	ActionScript 코드를 나타내는 문자열
<code>frame.duration</code>	읽기 전용, 프레임 시퀀스의 프레임 수를 나타내는 정수
<code>frame.elements</code>	읽기 전용, Element 객체의 배열(Element 객체 참조).
<code>frame.hasCustomEase</code>	프레임이 사용자 정의 여유 곡선에서 해당 여유 정보를 가져오는지 여부를 지정하는 부울 값
<code>frame.labelType</code>	프레임 이름의 형식을 지정하는 문자열
<code>frame.motionTweenOrientToPath</code>	트위닝된 요소가 경로를 따라 이동할 때 경로에 있는 각 점과의 각도를 유지하기 위해 요소를 회전할지 여부를 지정하는 부울 값
<code>frame.motionTweenRotate</code>	트위닝된 요소의 회전 방법을 지정하는 문자열
<code>frame.motionTweenRotateTimes</code>	트위닝된 요소가 시작 키프레임과 다음 키프레임 사이에서 회전하는 횟수를 지정하는 정수

속성	설명
frame.motionTweenScale	트위닝된 요소의 크기가 트윈에 있는 각 프레임만큼 크기를 늘리는 방식으로 다음 키프레임에 있는 객체의 크기에 맞게 조절되는지(true), 아니면 조절되지 않는지(false)를 지정하는 부울 값
frame.motionTweenSnap	트위닝된 요소가 해당 프레임의 레이어와 연관된 모션 안내선 레이어에 있는 가장 가까운 점에 자동으로 물리는지(true), 아니면 자동으로 물리지 않는지(false)를 지정하는 부울 값
frame.motionTweenSync	true로 설정된 경우 트위닝된 객체의 애니메이션이 기본 타임라인과 동기화되는 부울 값
frame.name	프레임의 이름을 지정하는 문자열
frame.shapeTweenBlend	모양 트윈이 트윈의 시작 부분에 있는 키프레임의 모양과 다음 키프레임의 모양 사이에서 블렌드되는 방법을 지정하는 문자열
frame.soundEffect	프레임(frame.soundLibraryItem)에 직접 첨부된 사운드에 대한 효과를 지정하는 문자열
frame.soundLibraryItem	사운드를 만드는 데 사용되는 라이브러리 항목(SoundItem 객체 참조)
frame.soundLoop	프레임(frame.soundLibraryItem)에 직접 첨부된 사운드가 재생되는 횟수를 지정하는 정수 값
frame.soundLoopMode	프레임(frame.soundLibraryItem)에 직접 첨부된 사운드를 특정 횟수만큼 재생할지, 아니면 무한히 반복할지를 지정하는 문자열
frame.soundName	라이브러리에 저장된 것처럼 프레임(frame.soundLibraryItem)에 직접 첨부된 사운드의 이름을 지정하는 문자열
frame.soundSync	프레임(frame.soundLibraryItem)에 직접 첨부된 사운드의 동기화 비헤이비어를 지정하는 문자열
frame.startFrame	읽기 전용, 시퀀스에 있는 첫 번째 프레임의 인덱스
frame.tweenEasing	트위닝된 객체에 적용해야 하는 여유의 양을 지정하는 정수
frame.tweenInstanceName	지정한 모션 객체에 인스턴스 이름을 할당합니다.
frame.tweenType	트윈의 유형을 지정하는 문자열
frame.useSingleEaseCurve	하나의 사용자 정의 여유 곡선이 모든 속성의 여유 정보에 사용되는지 여부를 지정하는 부울 값

frame.convertMotionObjectTo2D()

지원 버전

Flash Professional CS5

사용법

```
frame.convertMotionObjectTo2D()
```

설명

메서드, 선택한 모션 객체를 2D 모션 객체로 변환합니다.

예제

다음 예제는 선택한 모션 객체를 2D 모션 객체로 변환합니다.

```
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()

지원 버전

Flash Professional CS5

사용법

```
frame.convertMotionObjectTo3D()
```

설명

메서드, 선택한 모션 객체를 3D 모션 객체로 변환합니다.

예제

다음 예제는 선택한 모션 객체를 3D 모션 객체로 변환합니다.

```
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

지원 버전

Flash MX 2004

사용법

```
frame.actionScript
```

설명

속성, ActionScript 코드를 나타내는 문자열입니다. 새 행 문자를 삽입하려면 "\n"을 사용합니다.

예제

다음 예제에서는 최상위 레이어의 첫 번째 프레임 액션에 stop()을 할당합니다.

```
f1.getDocumentDOM().getTimeline().layers[0].frames[0].actionScript = 'stop();';
```

frame.duration

지원 버전

Flash MX 2004

사용법

```
frame.duration
```

설명

읽기 전용 속성, 프레임 시퀀스의 프레임 수를 나타내는 정수입니다.

예제

다음 예제에서는 최상위 레이어의 첫 번째 프레임에서 시작하는 프레임 시퀀스의 프레임 수를 frameSpan 변수에 저장합니다.

```
var frameSpan = f1.getDocumentDOM().getTimeline().layers[0].frames[0].duration;
```

frame.elements

지원 버전

Flash MX 2004

사용법

```
frame.elements
```

설명

읽기 전용 속성, Element 객체의 배열([Element 객체 참조](#))입니다. 요소 순서는 FLA 파일에 저장된 순서에 따라 지정됩니다. 스테이지에 여러 개의 모양이 있고 각 모양이 그룹 해제되어 있으면 모두 하나의 요소로 취급됩니다. 스테이지에 여러 개의 모양이 있고 각 모양이 그룹화되어 있으면 모두 별개의 요소로 취급됩니다. 즉, 스테이지에 있는 개별 모양의 수와 상관없이 그룹화되지 않은 프리미티브 모양은 단일 요소로 취급됩니다. 예를 들어, 프레임에 세 개의 그룹화되지 않은 프리미티브 모양이 있을 경우 해당 프레임의 elements.length에서 값 1을 반환합니다. 이 문제를 해결하려면 각 모양을 개별적으로 선택하고 그룹화합니다.

예제

다음 예제에서는 최상위 레이어의 첫 번째 프레임에 있는 현재 요소의 배열을 myElements 변수에 저장합니다.

```
var myElements = f1.getDocumentDOM().getTimeline().layers[0].frames[0].elements;
```

frame.getCustomEase()

지원 버전

Flash 8

사용법

```
Frame.getCustomEase([property])
```

매개 변수

property 사용자 정의 여유 값을 반환할 속성을 지정하는 선택적 문자열. 사용할 수 있는 값에는 "all", "position", "rotation", "scale", "color" 및 "filters"가 있습니다. 기본값은 "all"입니다.

반환값

JavaScript 객체의 배열을 반환합니다. 각 객체에는 **x** 및 **y** 속성이 있습니다.

설명

메서드, 여유 곡선을 정의하는 3차원 베지어 곡선에 대한 제어점을 나타내는 객체의 배열을 반환합니다.

예제

다음 예제에서는 최상위 레이어의 첫 번째 프레임에 대한 **position** 속성의 사용자 정의 여유 값을 반환합니다.

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]
var easeArray = theFrame.getCustomEase("position");
```

참고 사항

[frame.hasCustomEase](#), [frame.setCustomEase\(\)](#), [frame.useSingleEaseCurve](#)

frame.getMotionObjectXML()

지원 버전

Flash Professional CS5

사용법

```
Frame.getMotionObjectXML()
```

설명

선택한 모션 객체에서 모션 XML의 문자열을 반환합니다.

예제

다음 예제는 선택한 모션 객체에서 모션 XML을 반환합니다.

```
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

지원 버전

Flash 8

사용법

`frame.hasCustomEase`

설명

속성, 부울 값입니다. `true`로 설정된 경우 프레임이 사용자 정의 여유 곡선에서 여유 정보를 가져오고 `false`로 설정된 경우 여유 값에서 여유 정보를 가져옵니다.

예제

다음 예제에서는 최상위 레이어의 첫 번째 프레임이 사용자 정의 여유 곡선 대신 여유 값에서 여유 정보를 가져오도록 지정합니다.

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]
theFrame.hasCustomEase = false;
```

참고 사항

[frame.getCustomEase\(\)](#), [frame.setCustomEase\(\)](#), [frame.useSingleEaseCurve](#)

frame.hasMotionPath()

지원 버전

Flash Professional CS5

사용법

`Frame.hasMotionPath()`

설명

메서드, 부울 값입니다. 현재 선택 항목에 모션 경로가 있는지 여부를 알려 줍니다.

예제

다음 예제는 현재 선택 항목에 모션 경로가 있는지 여부를 알려 주는 `trace` 문을 반환합니다.

```
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()

지원 버전

Flash Professional CS5

사용법

Frame.is3DMotionObject()

설명

메서드, 부울값입니다. 현재 선택 항목에 3D 모션 객체가 있는지 여부를 알려 줍니다.

예제

다음 예제는 현재 선택 항목이 3D 모션 객체인지를 알려 주는 trace 문을 반환합니다.

```
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()

지원 버전

Flash Professional CS5

사용법

Frame.isMotionObject()

설명

메서드, 부울값입니다. 현재 선택한 항목에 모션 객체가 있는지 여부를 알려 줍니다.

예제

다음 예제는 현재 선택 항목이 모션 객체인지를 알려 주는 `trace` 문을 반환합니다.

```
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

지원 버전

Flash MX 2004

사용법

`frame.labelType`

설명

속성, 프레임 이름의 형식을 지정하는 문자열입니다. 사용할 수 있는 값에는 "none", "name", "comment" 및 "anchor"가 있습니다. 레이블을 "none"으로 설정하면 `frame.name` 속성이 지워집니다.

예제

다음 예제에서는 최상위 레이어의 첫 번째 프레임 이름을 "First Frame"으로 설정한 다음 해당 레이블을 "comment"로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].name = 'First Frame';
fl.getDocumentDOM().getTimeline().layers[0].frames[0].labelType = 'comment';
```

frame.motionTweenOrientToPath

지원 버전

Flash MX 2004

사용법

`frame.motionTweenOrientToPath`

설명

속성, `true`로 설정된 경우 트위닝된 요소가 경로를 따라 이동할 때 경로에 있는 각 점과의 각도를 유지하기 위해 요소를 회전시키고 `false`로 설정된 경우 회전시키지 않는 부울 값입니다.

이 속성의 값을 지정하려면 `frame.motionTweenRotate`를 "none"으로 설정해야 합니다.

frame.motionTweenRotate

지원 버전
Flash MX 2004

사용법

`frame.motionTweenRotate`

설명

속성, 트위닝된 요소의 회전 방법을 지정하는 문자열입니다. 사용할 수 있는 값에는 "none", "auto", "clockwise" 및 "counter-clockwise"가 있습니다. 이 값이 "auto"이면 다음 키프레임에 있는 객체의 회전과 일치시켜야 하는 모션이 가장 적은 방향으로 객체가 회전합니다.

[frame.motionTweenOrientToPath](#)의 값을 지정하려면 이 속성을 "none"으로 설정합니다.

예제

[frame.motionTweenRotateTimes](#)를 참조하십시오.

frame.motionTweenRotateTimes

지원 버전
Flash MX 2004

사용법

`frame.motionTweenRotateTimes`

설명

속성, 트위닝된 요소가 시작 키프레임과 다음 키프레임 사이에서 회전하는 횟수를 지정하는 정수입니다.

예제

다음 예제에서는 다음 키프레임에 도달할 때까지 이 프레임의 요소를 시계 반대 방향으로 세 번 회전합니다.

```
f1.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenRotate = "counter-clockwise";  
f1.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenRotateTimes = 3;
```

frame.motionTweenScale

지원 버전
Flash MX 2004

사용법

`frame.motionTweenScale`

설명

속성, 트위닝된 요소의 크기가 트윈에 있는 각 프레임만큼 크기를 늘리는 방식으로 다음 키프레임에 있는 객체의 크기에 맞게 조절되는지(true), 아니면 조절되지 않는지(false)를 지정하는 부울 값입니다.

예제

다음 예제에서는 트위닝된 요소의 크기가 트윈에 있는 각 프레임만큼 크기를 늘리는 방식으로 다음 키프레임에 있는 객체의 크기에 맞게 조절되도록 지정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenScale = true;
```

frame.motionTweenSnap

지원 버전

Flash MX 2004

사용법

```
frame.motionTweenSnap
```

설명

속성, 트위닝된 요소가 해당 프레임의 레이어와 연관된 모션 안내선 레이어에 있는 가장 가까운 점에 자동으로 물리는지(true), 아니면 자동으로 물리지 않는지(false)를 지정하는 부울 값입니다.

frame.motionTweenSync

지원 버전

Flash MX 2004

사용법

```
frame.motionTweenSync
```

설명

속성, true로 설정된 경우 트위닝된 객체의 애니메이션이 기본 타임라인과 동기화되는 부울 값입니다.

예제

다음 예제에서는 트위닝된 객체가 타임라인과 동기화되도록 지정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenSync = true;
```

frame.name

지원 버전

Flash MX 2004

사용법

```
frame.name
```

설명

속성, 프레임의 이름을 지정하는 문자열입니다.

예제

다음 예제에서는 최상위 레이어의 첫 번째 프레임 이름을 "First Frame"으로 설정한 다음 name 값을 frameLabel 변수에 저장합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].name = 'First Frame';  
var frameLabel = fl.getDocumentDOM().getTimeline().layers[0].frames[0].name;
```

frame.selectMotionPath()

지원 버전

Flash Professional CS5

사용법

```
Frame.selectMotionPath()
```

설명

메서드, 부울값입니다. 현재 모션 객체의 모션 경로를 선택(true)하거나 선택 취소(false)합니다.

예제

다음 예제는 현재 모션 객체의 모션 경로를 선택하거나 선택 취소합니다.

```
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()

지원 버전

Flash 8

사용법

```
frame.setCustomEase(property, easeCurve)
```

매개 변수

property 여유 곡선을 사용해야 하는 속성을 지정하는 문자열. 사용할 수 있는 값에는 "all", "position", "rotation", "scale", "color" 및 "filters"가 있습니다.

`easeCurve` 여유 곡선을 정의하는 객체의 배열. 각 배열 요소는 **x** 및 **y** 속성을 가진 JavaScript 객체여야 합니다.

반환값

없음

설명

메서드, 사용자 정의 여유 곡선으로 사용할 3차원 베지어 곡선을 설명하는 제어점 및 접선 끝점 좌표의 배열을 지정합니다. 이 배열은 제어점 및 접선 끝점의 가로(순서: 왼쪽에서 오른쪽) 위치로 구성됩니다.

예제

다음 예제에서는 첫 번째 레이어의 첫 번째 프레임에 있는 모든 속성의 여유 곡선을 `easeCurve` 배열에 지정된 베지어 곡선으로 설정합니다.

```
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 );
```

참고 사항

[frame.getCustomEase\(\)](#), [frame.hasCustomEase](#), [frame.useSingleEaseCurve](#)

frame.setMotionObjectDuration()

지원 버전

Flash Professional CS5

사용법

```
Frame.setMotionObjectDuration( duration [, stretchExistingKeyframes] )
```

매개 변수

duration 선택한 모션 객체의 트윈 범위에 대한 프레임 수를 지정합니다.

stretchExistingKeyframes 마지막 프레임의 끝으로 트윈 범위가 확장 되었는지 또는 프레임이 추가되었는지 여부를 결정하는 부울 값

설명

메서드, 현재 선택한 모션 객체의 지속 기간(트윈 범위 길이)을 설정합니다.

예제

다음 예제는 선택한 모션 객체에 대해 11프레임의 지속 기간을 지정합니다.

```
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()

지원 버전

Flash Professional CS5

사용법

```
Frame.setMotionObjectXML( xmlstr [, endAtCurrentLocation] )
```

매개 변수

xmlstr XML 문자열을 지정하는 문자열 값

endAtCurrentLocation 트윈이 현재 위치에서 시작하거나 끝나는지를 결정하는 부울 값

설명

메서드, 선택한 모션 객체에 선택한 모션 XML을 적용합니다.

예제

이 예제는 선택한 모션 객체에 myMotionXML로 식별된 모션 XML이 적용되도록 지정합니다.

```
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

지원 버전

Flash MX 2004

사용법

```
frame.shapeTweenBlend
```

설명

속성, 모양 트윈이 트윈의 시작 부분에 있는 키프레임의 모양과 다음 키프레임의 모양 사이에서 블렌드되는 방법을 지정하는 문자열입니다. 사용할 수 있는 값에는 "distributive" 및 "angular"가 있습니다.

frame.soundEffect

지원 버전

Flash MX 2004

사용법

```
frame.soundEffect
```

설명

속성, 프레임([frame.soundLibraryItem](#))에 직접 첨부된 사운드에 대한 효과를 지정하는 문자열입니다. 사용할 수 있는 값에는 "none", "left channel", "right channel", "fade left to right", "fade right to left", "fade in", "fade out" 및 "custom"이 있습니다.

예제

다음 예제에서는 첫 번째 프레임에 첨부된 사운드가 페이드 인되도록 지정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundEffect = "fade in";
```

frame.soundLibraryItem

지원 버전

Flash MX 2004

사용법

```
frame.soundLibraryItem
```

설명

속성, 사운드를 만드는 데 사용되는 라이브러리 항목([SoundItem 객체](#) 참조)입니다. 사운드는 프레임에 직접 첨부됩니다.

예제

다음 예제에서는 라이브러리의 첫 번째 항목을 첫 번째 프레임의 `soundLibraryItem` 속성에 할당합니다.

```
// 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

지원 버전

Flash MX 2004

사용법

`frame.soundLoop`

설명

속성, 프레임([frame.soundLibraryItem](#))에 직접 첨부된 사운드가 재생되는 횟수를 지정하는 정수 값입니다. 이 속성의 값을 지정하려면 [frame.soundLoopMode](#)를 "repeat"로 설정합니다.

예제

[frame.soundLoopMode](#)를 참조하십시오.

frame.soundLoopMode

지원 버전

Flash MX 2004

사용법

`frame.soundLoopMode`

설명

속성, 프레임([frame.soundLibraryItem](#))에 직접 첨부된 사운드를 특정 횟수만큼 재생할지, 아니면 무한히 반복할지를 지정하는 문자열입니다. 사용할 수 있는 값에는 "repeat" 및 "loop"가 있습니다. 사운드의 재생 횟수를 지정하려면 [frame.soundLoop](#)의 값을 설정합니다.

예제

다음 예제에서는 사운드가 두 번 재생되도록 지정합니다.

```
f1.getDocumentDOM().getTimeline().layers[0].frames[0].soundLoopMode = "repeat";  
f1.getDocumentDOM().getTimeline().layers[0].frames[0].soundLoop = 2;
```

frame.soundName

지원 버전

Flash MX 2004

사용법

`frame.soundName`

설명

속성, 라이브러리에 저장된 것처럼 프레임([frame.soundLibraryItem](#))에 직접 첨부된 사운드의 이름을 지정하는 문자열입니다.

예제

다음 예제에서는 첫 번째 프레임의 `soundName` 속성을 "song1.mp3"로 변경합니다. 이 경우 라이브러리에 `song1.mp3`가 있어야 합니다.

```
f1.getDocumentDOM().getTimeline().layers[0].frames[0].soundName = "song1.mp3";
```

frame.soundSync

지원 버전
Flash MX 2004

사용법
`frame.soundSync`

설명
속성, 프레임([frame.soundLibraryItem](#))에 직접 첨부된 사운드의 동기화 비헤이비어를 지정하는 문자열입니다. 사용할 수 있는 값에는 "event", "stop", "start" 및 "stream"이 있습니다.

예제
다음 예제에서는 사운드가 스트리밍되도록 지정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundSync = 'stream';
```

frame.startFrame

지원 버전
Flash MX 2004

사용법
`frame.startFrame`

설명
읽기 전용 속성, 시퀀스에 있는 첫 번째 프레임의 인덱스입니다.

예제
다음 예제에서는 `stFrame`이 프레임 시퀀스에 있는 첫 번째 프레임의 인덱스입니다. 이 예제에서는 프레임 시퀀스가 프레임 5부터 프레임 10까지 여섯 개의 프레임에 걸쳐 있습니다. 따라서 프레임 5에서 프레임 10 사이에 있는 모든 프레임의 `stFrame` 값은 4입니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
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

지원 버전
Flash MX 2004

사용법
`frame.tweenEasing`

설명

속성, 트위닝된 객체에 적용해야 하는 여유의 양을 지정하는 정수입니다. 사용할 수 있는 값은 -100부터 100까지입니다. 모션 트윈을 느리게 시작하여 애니메이션 끝으로 갈수록 트위닝되는 속도를 빠르게 하려면 -1부터 -100 사이의 값을 사용합니다. 모션 트윈을 빠르게 시작하여 애니메이션 끝으로 갈수록 트위닝되는 속도를 느리게 하려면 1부터 100 사이의 양수 값을 사용합니다.

예제

다음 예제에서는 트위닝된 객체의 모션이 매우 빠르게 시작되어 애니메이션 끝으로 갈수록 액션 속도가 느려지도록 지정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].tweenEasing = 50;
```

frame.tweenInstanceName

지원 버전

Flash Professional CS5

사용법

```
Frame.tweenInstanceName ()
```

설명

속성, 선택한 모션 객체에 인스턴스 이름을 할당합니다.

예제

다음 예제는 선택한 모션 객체에 인스턴스 이름 MyMotionTween을 지정합니다.

```
theFrame.tweenInstanceName = "MyMotionTween";
```

frame.tweenType

지원 버전

Flash MX 2004

사용법

```
frame.tweenType
```

설명

속성, 트윈의 유형을 지정하는 문자열입니다. 사용할 수 있는 값에는 "motion", "shape" 또는 "none"이 있습니다. "none" 값은 모션 트윈을 제거합니다. 모션 트윈을 만들려면 [timeline.createMotionTween\(\)](#) 메서드를 사용합니다.

"motion"을 지정하는 경우에는 프레임에 있는 객체가 심볼, 텍스트 필드 또는 그룹화된 객체여야 합니다. 이 값을 지정하면 해당 객체의 위치가 현재 키프레임에서 다음 키프레임으로 트위닝됩니다.

"shape"를 지정하는 경우에는 프레임에 있는 객체가 모양이어야 합니다. 이 값을 지정하면 현재 키프레임에 있는 모양이 다음 키프레임에 있는 모양으로 블렌드됩니다.

예제

다음 예제에서는 객체가 모션 트윈이므로 해당 위치가 현재 키프레임에서 다음 키프레임으로 트위닝되도록 지정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].tweenType = "motion";
```

frame.useSingleEaseCurve

지원 버전

Flash 8

사용법

```
frame.useSingleEaseCurve
```

설명

속성, 부울 값입니다. **true**로 설정된 경우 하나의 사용자 정의 여유 곡선이 모든 속성의 여유 정보에 사용되고 **false**로 설정된 경우 각 속성에 고유한 여유 곡선이 사용됩니다.

프레임에 사용자 정의 여유 기능이 적용되지 않은 경우 이 속성은 무시됩니다.

예제

다음 예제에서는 최상위 레이어에 있는 첫 번째 프레임의 모든 속성에 하나의 사용자 정의 여유 곡선이 사용되도록 지정합니다.

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]  
theFrame.useSingleEaseCurve = true;
```

참고 사항

[frame.getCustomEase\(\)](#), [frame.hasCustomEase](#), [frame.setCustomEase\(\)](#)

22장: HalfEdge 객체

지원 버전

Flash MX 2004

설명

HalfEdge 객체는 [Shape 객체](#) 가장자리의 지정된 반쪽입니다. 가장자리는 두 개의 반쪽 가장자리로 되어 있습니다. 이러한 반쪽 가장자리를 따라 이동하면 모양의 윤곽선을 순회할 수 있습니다. 예를 들어 두 반쪽 가장자리 중 어느 한 쪽에서 출발해도 모양의 윤곽선 주위를 거쳐 원래 위치로 돌아올 수 있습니다.

반쪽 가장자리에는 순서가 지정됩니다. 하나는 가장자리의 한 쪽을 나타내고 다른 하나는 가장자리의 다른 한 쪽을 나타냅니다.

메서드 요약

HalfEdge 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
halfEdge.getEdge()	HalfEdge 객체에 대한 Edge 객체 를 가져옵니다.
halfEdge.getNext()	현재 윤곽선에서 다음 반쪽 가장자리를 가져옵니다.
halfEdge.getOppositeHalfEdge()	반대쪽 가장자리에서 HalfEdge 객체를 가져옵니다.
halfEdge.getPrev()	현재 윤곽선에서 이전 HalfEdge 객체를 가져옵니다.
halfEdge.getVertex()	HalfEdge 객체의 머리에서 Vertex 객체 를 가져옵니다.

속성 요약

HalfEdge 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
halfEdge.id	읽기 전용, HalfEdge 객체에 대한 고유한 정수 식별자
halfEdge.index	부모 가장자리에 이 HalfEdge 객체에 대한 인덱스를 지정하며 0 또는 1의 값을 가지는 정수

halfEdge.getEdge()

지원 버전

Flash MX 2004

사용법

```
halfEdge.getEdge()
```

매개 변수

없음

반환값

[Edge 객체](#)

설명

메서드, HalfEdge 객체에 대한 Edge 객체를 가져옵니다. [Edge 객체](#) 참조

예제

다음 예제에서는 지정된 모양의 가장자리와 반쪽 가장자리를 가져오는 방법을 보여 줍니다.

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge(0);  
var edge = hEdge.getEdge();
```

halfEdge.getNext()

지원 버전

Flash MX 2004

사용법

```
halfEdge.getNext()
```

매개 변수

없음

반환값

HalfEdge 객체

설명

메서드, 현재 윤곽선에서 다음 반쪽 가장자리를 가져옵니다.

참고: 방향과 순서는 반쪽 가장자리에만 있고 가장자리에는 없습니다.

예제

다음 예제에서는 지정된 윤곽선의 다음 반쪽 가장자리를 nextHalfEdge 변수에 저장합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge( 0 );  
var nextHalfEdge = hEdge.getNext();
```

halfEdge.getOppositeHalfEdge()

지원 버전

Flash MX 2004

사용법

```
halfEdge.getOppositeHalfEdge()
```

매개 변수

없음

반환값

HalfEdge 객체

설명

메서드, 반대쪽 가장자리에서 HalfEdge 객체를 가져옵니다.

예제

다음 예제에서는 반대쪽 반쪽 가장자리의 hEdge를 otherHalfEdge 변수에 저장합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge(0);  
var otherHalfEdge = hEdge.getOppositeHalfEdge();
```

halfEdge.getPrev()

지원 버전

Flash MX 2004

사용법

```
halfEdge.getPrev()
```

매개 변수

없음

반환값

HalfEdge 객체

설명

메서드, 현재 윤곽선에서 이전 HalfEdge 객체를 가져옵니다.

참고: 방향과 순서는 반쪽 가장자리에만 있고 가장자리에는 없습니다.

예제

다음 예제에서는 지정된 윤곽선의 이전 반쪽 가장자리를 prevHalfEdge 변수에 저장합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge( 0 );  
var prevHalfEdge = hEdge.getPrev();
```

halfEdge.getVertex()

지원 버전

Flash MX 2004

사용법

```
halfEdge.getVertex()
```

매개 변수
없음

반환값
[Vertex 객체](#)

설명
메서드, HalfEdge 객체의 머리에서 Vertex 객체를 가져옵니다. [Vertex 객체](#) 참조

예제
다음 예제에서는 hEdge의 머리에 있는 Vertex 객체를 vertex 변수에 저장합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
var edge = shape.edges[0];  
var hEdge = edge.getHalfEdge(0);  
var vertex = hEdge.getVertex();
```

halfEdge.id

지원 버전
Flash MX 2004

사용법
halfEdge.id

설명
읽기 전용 속성, HalfEdge 객체에 대한 고유한 정수 식별자입니다.

예제
다음 예제에서는 지정된 반쪽 가장자리의 고유한 식별자를 [출력] 패널에 표시합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
alert(shape.contours[0].getHalfEdge().id);
```

halfEdge.index

지원 버전
Flash MX 2004

사용법
halfEdge.index

설명
읽기 전용 속성, 부모 가장자리에 이 HalfEdge 객체에 대한 인덱스를 지정하며 0 또는 1의 값을 가지는 정수입니다.

예제

다음 예제에서는 지정된 반쪽 가장자리의 인덱스 값을 [출력] 패널에 표시합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge(0);  
var heIndex = hEdge.index;
```

23장: Instance 객체

상속 [Element 객체](#) > Instance 객체

지원 버전
Flash MX 2004

설명
Instance 객체는 [Element 객체](#)의 하위 클래스입니다.

속성 요약
모든 Element 객체 속성 외에도 Instance 객체에는 다음과 같은 속성이 있습니다.

속성	설명
instance.instanceType	읽기 전용, 인스턴스의 유형을 나타내는 문자열
instance.libraryItem	이 인스턴스를 인스턴스화하는 데 사용되는 라이브러리 항목

instance.instanceType

지원 버전
Flash MX 2004, Flash 8의 경우 "video" 값을 추가로 사용할 수 있습니다.

사용법
`instance.instanceType`

설명
읽기 전용 속성, 인스턴스의 유형을 나타내는 문자열입니다. 사용할 수 있는 값에는 "symbol", "bitmap", "embedded video", "linked video", "video" 및 "compiled clip"이 있습니다.

Flash MX 2004에서 `library.addItem("video")`을 사용하여 라이브러리에 추가된 항목의 `instance.instanceType` 값은 "embedded_video"입니다. Flash 8 이상에서 이 값은 "video"입니다. [library.addItem\(\)](#)을 참조하십시오.

예제
다음 예제에서는 무비 클립의 인스턴스 유형이 symbol임을 보여 줍니다.

```
// 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

지원 버전
Flash MX 2004

사용법

`instance.libraryItem`

설명

속성, 이 인스턴스를 인스턴스화하는 데 사용되는 라이브러리 항목입니다. 이 속성은 동일한 유형의 다른 라이브러리 항목으로만 변경할 수 있습니다. 즉, `symbol` 인스턴스를 비트맵을 참조하도록 설정할 수 없습니다. [library 객체](#) 참조

예제

다음 예제에서는 선택한 심볼을 라이브러리의 첫 번째 항목을 참조하도록 변경합니다.

```
f1.getDocumentDOM().selection[0].libraryItem = f1.getDocumentDOM().library.items[0];
```

24장: Item 객체

지원 버전

Flash MX 2004

설명

Item 객체는 기본 추상 클래스입니다. Item 객체에서는 모든 라이브러리 항목이 파생됩니다. 자세한 내용은 [library 객체](#)를 참조하십시오.

메서드 요약

Item 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
item.addData()	라이브러리 항목에 지정된 데이터를 추가합니다.
item.getData()	지정된 데이터의 값을 가져옵니다.
item.hasData()	라이브러리 항목에 이름이 지정된 데이터가 있는지 여부를 확인합니다.
item.removeData()	라이브러리 항목에서 영구 데이터를 제거합니다.

속성 요약

Item 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
item.itemType	읽기 전용, 요소의 유형을 지정하는 문자열입니다.
item.linkageBaseClass	심볼과 연관시킬 ActionScript 3.0 클래스를 지정하는 문자열입니다.
item.linkageClassName	심볼과 연관시킬 ActionScript 2.0 클래스를 지정하는 문자열입니다.
item.linkageExportForAS	부울 값. true로 설정된 경우 항목이 ActionScript에 대해 내보내집니다.
item.linkageExportForRS	부울 값. true로 설정된 경우 항목이 런타임 공유에 내보내집니다.
item.linkageExportInFirstFrame	부울 값. true로 설정된 경우 항목이 첫 번째 프레임으로 내보내집니다.
item.linkageIdentifier	Flash에서 대상 SWF 파일에 연결할 때 예셋을 식별하는 데 사용할 이름을 지정하는 문자열입니다.
item.linkageImportForRS	부울 값. true로 설정된 경우 런타임 공유에 항목을 가져옵니다.
item.linkageURL	공유 예셋이 포함된 SWF 파일이 있는 URL을 지정하는 문자열입니다.
item.name	폴더 구조가 포함된 라이브러리 항목의 이름을 지정하는 문자열입니다.

item.addData()

지원 버전

Flash MX 2004

사용법

```
item.addData(name, type, data)
```

매개 변수

name 데이터의 이름을 지정하는 문자열.

type 데이터의 유형을 지정하는 문자열. 사용할 수 있는 유형에는 "integer", "integerArray", "double", "doubleArray", "string" 및 "byteArray"가 있습니다.

data 지정된 라이브러리 항목을 추가할 데이터. 데이터의 유형은 **type** 매개 변수의 값에 따라 달라집니다. 예를 들어, 유형이 "integer"이면 데이터의 값은 정수여야 합니다.

반환값

없음

설명

메서드, 라이브러리 항목에 지정된 데이터를 추가합니다.

예제

다음 예제에서는 정수 값 12를 가진 **myData**라는 데이터를 첫 번째 라이브러리 항목에 추가합니다.

```
f1.getDocumentDOM().library.items[0].addData("myData", "integer", 12);
```

item.getData()

지원 버전

Flash MX 2004

사용법

```
item.getData(name)
```

매개 변수

name 가져올 데이터의 이름을 지정하는 문자열.

반환값

name 매개 변수에 의해 지정된 데이터. 반환되는 데이터의 유형은 저장된 데이터의 유형에 따라 달라집니다.

설명

메서드, 지정된 데이터의 값을 가져옵니다.

예제

다음 예제에서는 첫 번째 라이브러리 항목에서 **myData**라는 데이터의 값을 가져와서 **libData** 변수에 저장합니다.

```
var libData = f1.getDocumentDOM().library.items[0].getData("myData");
```

item.hasData()

지원 버전
Flash MX 2004

사용법
`item.hasData(name)`

매개 변수
name 라이브러리 항목에서 확인할 데이터의 이름을 지정하는 문자열.

반환값
부울 값, 지정된 데이터가 있으면 `true`를 반환하고 그렇지 않으면 `false`를 반환합니다.

설명
메서드, 라이브러리 항목에 지정된 데이터가 있는지 여부를 확인합니다.

예제
다음 예제에서는 첫 번째 라이브러리 항목에 `myData`라는 데이터가 있으면 [출력] 패널에 메시지를 표시합니다.

```
if (fl.getDocumentDOM().library.items[0].hasData("myData")) {  
    fl.trace("Yep, it's there!");  
}
```

item.itemType

지원 버전
Flash MX 2004

사용법
`item.itemType`

설명
읽기 전용 속성, 요소의 유형을 지정하는 문자열입니다. 값은 다음 중 하나입니다. "undefined", "component", "movie clip", "graphic", "button", "folder", "font", "sound", "bitmap", "compiled clip", "screen" 또는 "video". 이 속성이 "video"면 비디오 유형을 결정할 수 있습니다. 자세한 내용은 [videoItem.videoType](#)을 참조하십시오.

예제
다음 예제에서는 지정된 라이브러리 항목의 유형을 [출력] 패널에 표시합니다.

```
fl.trace(fl.getDocumentDOM().library.items[0].itemType);
```

item.linkageBaseClass

지원 버전
Flash CS3 Professional

사용법

`item.linkageBaseClass`

설명

속성, 심볼과 연관시킬 **ActionScript 3.0** 클래스를 지정하는 문자열입니다. 여기에 지정된 값은 제작 환경의 [링크] 대화 상자 및 [심볼 속성] 대화 상자과 같은 [링크] 대화 상자 컨트롤이 있는 다른 대화 상자에서 나타납니다. **ActionScript 2.0** 클래스에 이 값을 지정하려면 `item.linkageClassName`을 사용합니다.

기본 클래스가 심볼 유형의 디폴트인 경우(예: 무비 클립의 경우 `flash.display.MovieClip`, 버튼의 경우 `flash.display.SimpleButton` 등) 이 속성은 빈 문자열("")입니다. 마찬가지로 항목을 디폴트 기본 클래스로 만들려면 이 값을 빈 문자열로 설정합니다.

[링크] 대화 상자에서 수행되는 검사가 수행되지 않도록 하고 **Flash**에서 기본 클래스를 지정된 값으로 설정할 수 없을 때 오류가 **throw**되지 않도록 하려면 이 값을 설정합니다. 예를 들어 [링크] 대화 상자에서 이 값을 설정하면 **FLA** 파일의 클래스 경로에서 기본 클래스를 검색할 수 있도록 확인하는 검사를 강제로 수행합니다. 이 기능을 사용하면 [제작 설정] 대화 상자의 [Flash] 탭에서 **ActionScript 3.0**이 선택되도록 하는 등의 작업을 수행할 수 있습니다. 이 속성을 스크립트에 설정한 경우에는 이러한 검사가 수행되지 않습니다.

예제

다음 코드 행에서는 이 속성을 사용하는 몇 가지 방법을 보여 줍니다.

```
// 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);
```

참고 사항

[document.docClass](#)

item.linkageClassName

지원 버전

Flash MX 2004

사용법

`item.linkageClassName`

설명

속성, 심볼과 연관시킬 **ActionScript 2.0** 클래스를 지정하는 문자열입니다. **ActionScript 3.0** 클래스에 이 값을 지정하려면 `item.linkageBaseClass`를 사용합니다.

이 속성을 정의하려면 `item.linkageExportForAS` 및/또는 `item.linkageExportForRS` 속성을 `true`로 설정해야 하며 `item.linkageImportForRS` 속성을 `false`로 설정해야 합니다.

예제

다음 예제에서는 첫 번째 라이브러리 항목과 연관된 **ActionScript 2.0** 클래스 이름을 `myClass`로 지정합니다.

```
fl.getDocumentDOM().library.items[0].linkageClassName = "myClass";
```

item.linkageExportForAS

지원 버전

Flash MX 2004

사용법

```
item.linkageExportForAS
```

설명

속성, 부울 값입니다. 이 속성이 true로 설정된 경우 항목이 ActionScript에 대해 내보내집니다. 또한 [item.linkageExportForRS](#) 및 [item.linkageExportInFirstFrame](#) 속성을 true로 설정할 수 있습니다.

이 속성을 true로 설정하면 [item.linkageImportForRS](#) 속성은 false로 설정해야 합니다. 또한 식별자([item.linkageIdentifier](#)) 및 URL([item.linkageURL](#))도 지정해야 합니다.

예제

다음 예제에서는 지정된 라이브러리 항목에 대해 이 속성을 설정합니다.

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

item.linkageExportForRS

지원 버전

Flash MX 2004

사용법

```
item.linkageExportForRS
```

설명

속성, 부울 값입니다. 이 속성이 true로 설정된 경우 항목이 런타임 공유에 내보내집니다. 또한 [item.linkageExportForAS](#) 및 [item.linkageExportInFirstFrame](#) 속성을 true로 설정할 수 있습니다.

이 속성을 true로 설정하면 [item.linkageImportForRS](#) 속성은 false로 설정해야 합니다. 또한 식별자([item.linkageIdentifier](#)) 및 URL([item.linkageURL](#))도 지정해야 합니다.

예제

다음 예제에서는 지정된 라이브러리 항목에 대해 이 속성을 설정합니다.

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

item.linkageExportInFirstFrame

지원 버전

Flash MX 2004

사용법

```
item.linkageExportInFirstFrame
```

설명

속성, 부울 값입니다. `true`로 설정된 경우 항목이 첫 번째 프레임으로 내보내지고 `false`로 설정된 경우 항목이 첫 번째 인스턴스의 프레임으로 내보내집니다. 이 때 스테이지에 표시되지 않는 항목은 내보내지지 않습니다.

이 속성을 `true`로 설정할 수 있는 경우는 `item.linkageExportForAS` 및/또는 `item.linkageExportForRS`가 `true`로 설정되어 있을 때뿐입니다.

예제

다음 예제에서는 지정된 라이브러리 항목이 첫 번째 프레임으로 내보내지도록 지정합니다.

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

item.linkageIdentifier

지원 버전

Flash MX 2004

사용법

```
item.linkageIdentifier
```

설명

속성, Flash에서 대상 SWF 파일에 연결할 때 예셋을 식별하는 데 사용할 이름을 지정하는 문자열입니다.

`item.linkageImportForRS`, `item.linkageExportForAS` 및 `item.linkageExportForRS`가 `false`로 설정되어 있으면 이 속성은 무시됩니다. 반대로 이러한 속성 중 하나가 `true`로 설정된 경우 이 속성을 설정해야 합니다.

예제

다음 예제에서는 내보낼 대상 SWF 파일에 연결할 때 라이브러리 항목을 식별하는 데 `my_mc` 문자열이 사용되도록 지정합니다.

```
fl.getDocumentDOM().library.items[0].linkageIdentifier = "my_mc";
```

참고 사항

[item.linkageURL](#)

item.linkageImportForRS

지원 버전

Flash MX 2004

사용법

```
item.linkageImportForRS
```

설명

속성, 부울 값입니다. `true`로 설정된 경우 런타임 공유에 항목을 가져옵니다. 이 속성을 `true`로 설정하는 경우에는

`item.linkageExportForAS` 및/또는 `item.linkageExportForRS`를 모두 `true`로 설정해야 합니다. 또한 식별자(`item.linkageIdentifier`) 및 URL(`item.linkageURL`)도 지정해야 합니다.

예제

다음 예제에서는 지정된 라이브러리 항목에 대해 이 속성을 true로 설정합니다.

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

item.linkageURL

지원 버전

Flash MX 2004

사용법

item.linkageURL

설명

속성, 공유 예셋이 포함된 SWF 파일이 있는 URL을 지정하는 문자열입니다. [item.linkageImportForRS](#), [item.linkageExportForAS](#) 및 [item.linkageExportForRS](#)가 false로 설정되어 있으면 이 속성은 무시됩니다. 반대로 이러한 속성 중 하나가 true로 설정된 경우 이 속성을 설정해야 합니다. 웹 URL 또는 파일 이름은 플랫폼의 형식에 따라 다르게 지정할 수 있습니다. 즉, 플랫폼에 따라 순방향 슬래시[/] 또는 역방향 슬래시[\]를 사용합니다.

예제

다음 예제에서는 지정된 라이브러리 항목에 대한 링크 URL을 지정합니다.

```
fl.getDocumentDOM().library.items[0].linkageURL = "theShareSWF.swf";
```

참고 사항

[item.linkageIdentifier](#)

item.name

지원 버전

Flash MX 2004

사용법

item.name

설명

메서드, 폴더 구조가 포함된 라이브러리 항목의 이름을 지정하는 문자열입니다. 예를 들어, Symbol_1이 Folder_1이라는 폴더 안에 있으면 Symbol_1의 name 속성은 "Folder_1/Symbol_1"이 됩니다.

예제

다음 예제에서는 지정된 라이브러리 항목의 이름을 [출력] 패널에 표시합니다.

```
fl.trace(fl.getDocumentDOM().library.items[0].name);
```

item.removeData()

지원 버전

Flash MX 2004

사용법

```
item.removeData(name)
```

매개 변수

name 라이브러리 항목에서 제거할 데이터의 이름을 지정합니다.

반환값

없음

설명

속성, 라이브러리 항목에서 영구 데이터를 제거합니다.

예제

다음 예제에서는 첫 번째 라이브러리 항목에서 **myData**라는 데이터를 제거합니다.

```
fl.getDocumentDOM().library.items[0].removeData("myData");
```

25장: Layer 객체

지원 버전

Flash MX 2004

설명

Layer 객체는 타임라인 내의 레이어를 나타냅니다. `timeline.layers` 속성에는 `fl.getDocumentDOM().getTimeline().layers`를 통해 액세스할 수 있는 Layer 객체의 배열이 포함됩니다.

속성 요약

Layer 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>layer.color</code>	레이어의 외곽선을 그리기 위해 할당된 색상을 지정하는 문자열, 16진수 값 또는 정수
<code>layer.frameCount</code>	읽기 전용, 레이어의 프레임 수를 지정하는 정수
<code>layer.frames</code>	읽기 전용, Frame 객체의 배열
<code>layer.height</code>	레이어 높이의 백분율을 지정하는 정수. [레이어 속성] 대화 상자의 레이어 높이 값에 해당합니다.
<code>layer.layerType</code>	레이어의 현재 용도를 지정하는 문자열. [레이어 속성] 대화 상자의 유형 설정에 해당합니다.
<code>layer.locked</code>	레이어의 잠긴 상태를 지정하는 부울 값
<code>layer.name</code>	레이어의 이름을 지정하는 문자열
<code>layer.outline</code>	레이어에 있는 모든 객체의 외곽선 상태를 지정하는 부울 값
<code>layer.parentLayer</code>	레이어의 상위 폴더, 안내 또는 마스크 레이어를 나타내는 Layer 객체
<code>layer.visible</code>	스테이지에 레이어의 객체를 표시할지 여부를 지정하는 부울 값

layer.color

지원 버전

Flash MX 2004

사용법

`layer.color`

설명

속성, 형식이 다음 중 하나인 레이어의 외곽선을 그리기 위해 할당된 색상입니다.

- "#RRGGBB" 또는 "#RRGGBBAA" 형식의 문자열
- 0xRRGGBB 형식의 16진수
- 16진수 숫자와 같은 십진수를 나타내는 정수

이 속성은 [레이어 속성] 대화 상자의 외곽선 색상 설정에 해당합니다.

예제

다음 예제에서는 첫 번째 레이어의 값을 `colorValue` 변수에 저장합니다.

```
var colorValue = fl.getDocumentDOM().getTimeline().layers[0].color;
```

다음 예제에서는 첫 번째 레이어의 색상을 빨간색으로 설정하는 세 가지 방법을 보여 줍니다.

```
fl.getDocumentDOM().getTimeline().layers[0].color=16711680;  
fl.getDocumentDOM().getTimeline().layers[0].color="#ff0000";  
fl.getDocumentDOM().getTimeline().layers[0].color=0xFF0000;
```

layer.frameCount

지원 버전

Flash MX 2004

사용법

```
layer.frameCount
```

설명

읽기 전용 속성, 레이어의 프레임 수를 지정하는 정수입니다.

예제

다음 예제에서는 첫 번째 레이어의 프레임 수를 `fcNum` 변수에 저장합니다.

```
var fcNum = fl.getDocumentDOM().getTimeline().layers[0].frameCount;
```

layer.frames

지원 버전

Flash MX 2004

사용법

```
layer.frames
```

설명

읽기 전용 속성, `Frame` 객체의 배열([Frame 객체](#) 참조)입니다.

예제

다음 예제에서는 `frameArray` 변수를 현재 문서의 프레임에 대한 `Frame` 객체의 배열로 설정합니다.

```
var frameArray = fl.getDocumentDOM().getTimeline().layers[0].frames;
```

프레임이 키프레임인지 확인하려면 다음 예제에서처럼 `frame.startFrame` 속성이 배열 인덱스와 일치하는지 확인합니다.

```
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

지원 버전
Flash MX 2004

사용법
`layer.height`

설명
속성, 레이어 높이의 백분율을 지정하는 정수입니다. [레이어 속성] 대화 상자의 레이어 높이 값에 해당합니다. 사용할 수 있는 값은 100, 200 또는 300이며 기본 높이의 백분율을 나타냅니다.

예제
다음 예제에서는 첫 번째 레이어 높이 설정의 백분율 값을 저장합니다.

```
var layerHeight = fl.getDocumentDOM().getTimeline().layers[0].height;
```

다음 예제에서는 첫 번째 레이어의 높이를 300퍼센트로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].height = 300;
```

layer.layerType

지원 버전
Flash MX 2004

사용법
`layer.layerType`

설명
속성, 레이어의 현재 용도를 지정하는 문자열입니다. [레이어 속성] 대화 상자의 유형 설정에 해당합니다. 사용할 수 있는 값에는 "normal", "guide", "guided", "mask", "masked" 및 "folder"가 있습니다.

예제
다음 예제에서는 타임라인의 첫 번째 레이어를 folder 유형으로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].layerType = "folder";
```

layer.locked

지원 버전
Flash MX 2004

사용법
`layer.locked`

설명
속성, 레이어의 잠긴 상태를 지정하는 부울 값입니다. true로 설정된 경우 레이어가 잠깁니다. 기본값은 false입니다.

예제

다음 예제에서는 첫 번째 레이어의 상태 부울 값을 `lockStatus` 변수에 저장합니다.

```
var lockStatus = fl.getDocumentDOM().getTimeline().layers[0].locked;
```

다음 예제에서는 첫 번째 레이어의 상태를 잠금 해제로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].locked = false;
```

layer.name

지원 버전

Flash MX 2004

사용법

```
layer.name
```

설명

속성, 레이어의 이름을 지정하는 문자열입니다.

예제

다음 예제에서는 현재 문서의 첫 번째 레이어 이름을 `foreground`로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].name = "foreground";
```

layer.outline

지원 버전

Flash MX 2004

사용법

```
layer.outline
```

설명

속성, 레이어에 있는 모든 객체의 외곽선 상태를 지정하는 부울 값입니다. `true`로 설정된 경우 레이어의 모든 객체가 외곽선만으로 표시되고 `false`로 설정된 경우 처음 만들어진 상태로 표시됩니다.

예제

다음 예제에서는 첫 번째 레이어의 모든 객체를 외곽선만으로 표시합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].outline = true;
```

layer.parentLayer

지원 버전

Flash MX 2004

사용법

`layer.parentLayer`

설명

속성, 레이어의 상위 폴더, 안내 또는 마스크 레이어를 나타내는 Layer 객체입니다. 부모 레이어는 레이어 앞에 오는 폴더, 안내 선 또는 마스크 레이어이거나 이전 또는 이후 레이어의 `parentLayer`이어야 합니다. 레이어의 `parentLayer`를 설정해도 목록에서 해당 레이어의 위치가 이동하지는 않습니다. 따라서 레이어의 `parentLayer`를 이동이 필요한 레이어로 설정하려고 하면 아무런 영향을 주지 않습니다. 최상위 레이어에는 `null`을 사용합니다.

예제

다음 예제에서는 동일한 타임라인의 동일한 레벨에 있는 두 레이어를 사용합니다. 첫 번째 레이어(`layers[0]`)는 폴더로 변환된 다음 두 번째 레이어(`layers[1]`)의 부모 폴더로 설정됩니다. 이 액션은 두 번째 레이어를 첫 번째 레이어 안으로 이동합니다.

```
var parLayer = fl.getDocumentDOM().getTimeline().layers[0];
parLayer.layerType = "folder";
fl.getDocumentDOM().getTimeline().layers[1].parentLayer = parLayer;
```

layer.visible

지원 버전

Flash MX 2004

사용법

`layer.visible`

설명

속성, 스테이지에 레이어의 객체를 표시할지 여부를 지정하는 부울 값입니다. `true`로 설정된 경우 레이어의 모든 객체가 표시되고 `false`로 설정된 경우 레이어의 모든 객체가 숨겨집니다. 기본값은 `true`입니다.

예제

다음 예제에서는 첫 번째 레이어의 모든 객체를 숨깁니다.

```
fl.getDocumentDOM().getTimeline().layers[0].visible = false;
```

26장: library 객체

지원 버전

Flash MX 2004

설명

library 객체는 [라이브러리] 패널을 나타냅니다. library 객체는 Document 객체(`document.library` 참조)의 속성이며 `fl.getDocumentDOM().library`를 통해 액세스할 수 있습니다.

library 객체에는 심볼, 비트맵, 사운드, 비디오 등 다양한 유형의 항목 배열이 포함됩니다.

메서드 요약

library 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
<code>library.addItemToDocument()</code>	현재 또는 지정된 항목을 스테이지의 지정된 위치에 추가합니다.
<code>library.addNewItem()</code>	[라이브러리] 패널에서 지정된 유형의 새 항목을 만들고 새 항목을 현재 선택한 항목으로 설정합니다.
<code>library.deleteItem()</code>	[라이브러리] 패널에서 현재 항목 또는 지정된 항목을 삭제합니다.
<code>library.duplicateItem()</code>	현재 선택한 항목 또는 지정된 항목의 복사본을 만듭니다.
<code>library.editItem()</code>	현재 선택한 항목 또는 지정된 항목을 편집 모드로 엽니다.
<code>library.expandFolder()</code>	라이브러리에서 현재 선택한 폴더 또는 지정된 폴더를 확장하거나 축소합니다.
<code>library.findItemIndex()</code>	라이브러리 항목의 인덱스 값(0부터 시작)을 반환합니다.
<code>library.getItemProperty()</code>	선택한 항목의 속성을 가져옵니다.
<code>library.getItemType()</code>	라이브러리 경로를 통해 현재 선택한 객체 또는 지정된 객체의 유형을 가져옵니다.
<code>library.getSelectedItems()</code>	현재 선택한 모든 라이브러리 항목의 배열을 가져옵니다.
<code>library.importEmbeddedSWF()</code>	SWF 파일을 컴파일된 클립으로 라이브러리에 가져옵니다.
<code>library.itemExists()</code>	지정된 항목이 라이브러리에 있는지 확인합니다.
<code>library.moveToFolder()</code>	현재 선택한 라이브러리 항목 또는 지정된 라이브러리 항목을 지정된 폴더로 이동합니다.
<code>library.newFolder()</code>	현재 선택한 폴더에 지정된 이름을 가진 새 폴더를 만들거나 <code>folderName</code> 매개 변수가 제공되지 않은 경우 기본 이름("untitled folder #")이 지정된 새 폴더를 만듭니다.
<code>library.renameItem()</code>	[라이브러리] 패널에서 현재 선택한 라이브러리 항목의 이름을 변경합니다.
<code>library.selectAll()</code>	모든 라이브러리 항목을 선택하거나 선택 취소합니다.
<code>library.selectItem()</code>	지정된 라이브러리 항목을 선택합니다.
<code>library.selectNone()</code>	모든 라이브러리 항목을 선택 취소합니다.
<code>library.setItemProperty()</code>	선택한 모든 라이브러리 항목(폴더 제외)의 속성을 설정합니다.
<code>library.updateItem()</code>	지정된 항목을 업데이트합니다.

library 객체의 속성 요약

library 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
library.items	라이브러리의 <code>Item</code> 객체의 배열

library.addItemToDocument()

지원 버전
Flash MX 2004

사용법

```
library.addItemToDocument(position [, namePath])
```

매개 변수

position 스테이지에 있는 항목의 중심 **x,y** 위치를 지정하는 점

namePath 항목의 이름을 지정하는 문자열. 항목이 폴더 안에 있으면 슬래시 표기법을 사용하여 해당 이름과 경로를 지정할 수 있습니다. **namePath**를 지정하지 않으면 현재 선택한 라이브러리 항목이 사용됩니다. 이 매개 변수는 선택 사항입니다.

반환값

부울 값, 항목이 문서에 성공적으로 추가되면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

설명

메서드, 현재 또는 지정된 항목을 스테이지의 지정된 위치에 추가합니다.

예제

다음 예제에서는 스테이지의 (3, 60) 위치에 현재 선택한 항목을 추가합니다.

```
fl.getDocumentDOM().library.addItemToDocument({x:3, y:60});
```

다음 예제에서는 (550, 485) 위치의 스테이지에 라이브러리의 폴더 1에 있는 심볼1 항목을 추가합니다.

```
fl.getDocumentDOM().library.addItemToDocument({x:550.0, y:485.0}, "folder1/Symbol1");
```

library.addItem()

지원 버전
Flash MX 2004

사용법

```
library.addItem(type [, namePath])
```

매개 변수

type 만들 항목의 유형을 지정하는 문자열. **type**에는 "video", "movie clip", "button", "graphic", "bitmap", "screen" 및 "folder" 값만 사용할 수 있습니다. 따라서 이 메서드를 사용하여 라이브러리에 사운드를 추가하는 등의 작업은 수행할 수 없습니다. 폴더 경로를 지정하는 것은 이 메서드를 호출하기 전에 [library.newFolder\(\)](#)를 사용하는 것과 같습니다.

namePath 추가할 항목의 이름을 지정하는 문자열. 항목이 폴더 안에 있으면 슬래시 표기법을 사용하여 해당 이름과 경로를 지정합니다. 이 매개 변수는 선택 사항입니다.

반환값

부울 값, 항목이 성공적으로 만들어지면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, [라이브러리] 패널에서 지정된 유형의 새 항목을 만들고 새 항목을 현재 선택한 항목으로 설정합니다. 사운드 등의 항목을 라이브러리로 가져오는 방법에 대한 자세한 내용은 [document.importFile\(\)](#)을 참조하십시오.

예제

다음 예제에서는 folderTwo라는 새 폴더에 start라는 새 버튼 항목을 만듭니다.

```
fl.getDocumentDOM().library.addItem("button", "folderTwo/start");
```

library.deleteItem()

지원 버전

Flash MX 2004

사용법

```
library.deleteItem([namePath])
```

매개 변수

namePath 삭제할 항목의 이름을 지정하는 문자열. 항목이 폴더 안에 있으면 슬래시 표기법을 사용하여 해당 이름과 경로를 지정할 수 있습니다. 폴더 이름을 지정하면 해당 폴더와 그 안에 포함된 모든 항목이 삭제되고 이름을 지정하지 않으면 현재 선택한 항목이 삭제됩니다. [라이브러리] 패널에서 모든 항목을 삭제하려면 이 메서드를 사용하기 전에 모든 항목을 선택합니다. 이 매개 변수는 선택 사항입니다.

반환값

부울 값, 항목이 성공적으로 삭제되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, [라이브러리] 패널에서 현재 항목 또는 지정된 항목을 삭제합니다. 이 메서드는 여러 항목을 선택한 경우 여러 항목에 영향을 줄 수 있습니다.

예제

다음 예제에서는 현재 선택한 항목을 삭제합니다.

```
fl.getDocumentDOM().library.deleteItem();
```

다음 예제에서는 Folder_1 라이브러리 폴더에서 Symbol_1 항목을 삭제합니다.

```
fl.getDocumentDOM().library.deleteItem("Folder_1/Symbol_1");
```

library.duplicateItem()

지원 버전

Flash MX 2004

사용법

```
library.duplicateItem( [ namePath ] )
```

매개 변수

namePath 복제할 항목의 이름을 지정하는 문자열. 항목이 폴더 안에 있으면 슬래시 표기법을 사용하여 해당 이름과 경로를 지정할 수 있습니다. 이 매개 변수는 선택 사항입니다.

반환값

부울 값, 항목이 성공적으로 복제되면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다. 둘 이상의 항목을 선택해도 **false**를 반환합니다.

설명

메서드, 현재 선택한 항목 또는 지정된 항목의 복사본을 만듭니다. 새 항목은 기본 이름(예: **item copy**)을 사용하게 되고 현재 선택한 항목으로 설정됩니다. 둘 이상의 항목을 선택하면 명령이 실패합니다.

예제

다음 예제에서는 **test** 라이브러리 폴더에 있는 **square** 항목의 복사본을 만듭니다.

```
fl.getDocumentDOM().library.duplicateItem("test/square");
```

library.editItem()

지원 버전

Flash MX 2004

사용법

```
library.editItem( [namePath] )
```

매개 변수

namePath 항목의 이름을 지정하는 문자열. 항목이 폴더 안에 있으면 슬래시 표기법을 사용하여 해당 이름과 경로를 지정할 수 있습니다. **namePath**를 지정하지 않으면 선택한 라이브러리 항목 하나만 편집 모드로 열립니다. 라이브러리에 현재 선택한 항목이 없거나 둘 이상 있는 경우 기본 타임라인의 첫 번째 장면이 편집을 위해 표시됩니다. 이 매개 변수는 선택 사항입니다.

반환값

부울 값, 지정된 항목이 있고 편집할 수 있으면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

설명

메서드, 현재 선택한 항목 또는 지정된 항목을 편집 모드로 엽니다.

예제

다음 예제에서는 **test** 라이브러리 폴더에 있는 **circle** 항목을 편집하기 위해 엽니다.

```
fl.getDocumentDOM().library.editItem("test/circle");
```


library.expandFolder()

지원 버전
Flash MX 2004

사용법

```
library.expandFolder(bExpand [, bRecurseNestedParents [, namePath]])
```

매개 변수

bExpand 부울 값, true로 설정한 경우 폴더가 확장되고 false(기본값)로 설정된 경우 폴더가 축소됩니다.

bRecurseNestedParents 부울 값, true로 설정된 경우 **bExpand**의 값에 따라 지정된 폴더 내의 모든 폴더가 확장되거나 축소됩니다. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

namePath 확장 또는 축소할 폴더의 이름과 선택적으로 경로를 지정하는 문자열. 이 매개 변수를 지정하지 않으면 현재 선택한 폴더에 이 메서드가 적용됩니다. 이 매개 변수는 선택 사항입니다.

반환값

부울 값, 항목이 성공적으로 확장되거나 축소되면 true를 반환하고 실패하거나 지정된 항목이 폴더가 아니면 false를 반환합니다.

설명

메서드, 라이브러리에서 현재 선택한 폴더 또는 지정된 폴더를 확장하거나 축소합니다.

예제

다음 예제에서는 test 라이브러리 폴더와 그 안에 들어 있는 모든 폴더(있을 경우)를 축소합니다.

```
fl.getDocumentDOM().library.expandFolder(false, true, "test");
```

library.findItemIndex()

지원 버전
Flash MX 2004

사용법

```
library.findItemIndex(namePath)
```

매개 변수

namePath 항목의 이름을 지정하는 문자열. 항목이 폴더 안에 있으면 슬래시 표기법을 사용하여 해당 이름과 경로를 지정할 수 있습니다.

반환값

항목의 0부터 시작하는 인덱스 값을 나타내는 정수 값.

설명

메서드, 라이브러리 항목의 인덱스 값(0부터 시작)을 반환합니다. 라이브러리 인덱스는 고정이기 때문에 폴더는 기본 인덱스의 일부로 간주됩니다. 폴더 경로는 중첩된 항목을 지정하는 데 사용될 수 있습니다.

예제

다음 예제에서는 test 폴더에 있는 square 라이브러리 항목의 0부터 시작하는 인덱스 값을 sqIndex 변수에 저장한 다음 대화 상자에 이 인덱스 값을 표시합니다.

```
var sqIndex = fl.getDocumentDOM().library.findItemIndex("test/square");  
alert(sqIndex);
```

library.getItemProperty()

지원 버전

Flash MX 2004

사용법

```
library.getItemProperty(property)
```

매개 변수

property 문자열입니다. property 매개 변수로 사용할 수 있는 값 목록은 [Item 객체](#)의 속성 요약 표 및 해당 하위 클래스의 속성 요약을 참조하십시오.

반환값

속성의 문자열 값

설명

메서드, 선택한 항목의 속성을 가져옵니다.

예제

다음 예제에서는 **ActionScript**를 사용하여 참조할 때나 런타임 공유에 심볼의 링크 식별자 값이 포함된 대화 상자를 표시합니다.

```
alert(fl.getDocumentDOM().library.getItemProperty("linkageIdentifier"));
```

library.getItemType()

지원 버전

Flash MX 2004

사용법

```
library.getItemType([namePath])
```

매개 변수

namePath 항목의 이름을 지정하는 문자열. 항목이 폴더 안에 있으면 슬래시 표기법을 사용하여 해당 이름과 경로를 지정합니다. namePath를 지정하지 않으면 현재 선택한 항목의 유형이 제공됩니다. 현재 선택한 항목이 둘 이상 있고 namePath가 제공되지 않은 경우 이 명령은 무시됩니다. 이 매개 변수는 선택 사항입니다.

반환값

객체의 유형을 지정하는 문자열 값. 사용할 수 있는 반환값에 대한 내용은 [item.itemType](#)을 참조하십시오.

설명

메서드, 라이브러리 경로를 통해 현재 선택한 객체 또는 지정된 객체의 유형을 가져옵니다.

예제

다음 예제에서는 Folder_1/Folder_2 폴더에 있는 Symbol_1의 항목 유형이 포함된 대화 상자를 표시합니다.

```
alert(fl.getDocumentDOM().library.getItemType("Folder_1/Folder_2/Symbol_1"));
```

library.getSelectedItems()

지원 버전

Flash MX 2004

매개 변수

없음

반환값

현재 선택한 모든 라이브러리 항목의 값 배열

설명

메서드, 현재 선택한 모든 라이브러리 항목의 배열을 가져옵니다.

예제

다음 예제에서는 현재 선택한 라이브러리 항목의 배열(이 경우 여러 개의 오디오 파일)을 selItems 변수에 저장한 다음 배열에 있는 첫 번째 오디오 파일의 sampleRate 속성을 11 kHz로 변경합니다.

```
var selItems = fl.getDocumentDOM().library.getSelectedItems();  
selItems[0].sampleRate = "11 kHz";
```

library.importEmbeddedSWF()

지원 버전

Flash MX 2004

사용법

```
library.importEmbeddedSWF(linkageName, swfData [, libName])
```

매개 변수

linkageName 루트 무비 클립의 SWF 링크 이름을 제공하는 문자열

swfData 외부 라이브러리 또는 DLL에서 가져온 이진 SWF 데이터의 배열

libName 만든 항목의 라이브러리 이름을 지정하는 문자열. 이름이 이미 사용되고 있으면 다른 이름이 만들어집니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, SWF 파일을 컴파일된 클립으로 라이브러리에 가져옵니다. [파일] > [가져오기] > [SWF]와 달리 이 메서드를 사용하면 컴파일된 SWF 파일을 라이브러리에 포함할 수 있습니다. 사용자 인터페이스 기능 중에는 이 메서드에 해당하는 것이 없으며 이 메서드는 외부 라이브러리 또는 DLL과 함께 사용해야 합니다(501페이지의 “C 레벨 확장성” 참조).

가져올 SWF 파일에는 모든 내용이 포함된 하나의 최상위 무비 클립이 있어야 합니다. 이 무비 클립에는 이 메서드에 전달된 **linkageName** 매개 변수와 같은 값으로 설정된 링크 식별자가 있어야 합니다.

예제

다음 예제에서는 **linkageName** 값으로 MyMovie가 지정된 SWF 파일을 Intro라는 컴파일된 클립으로 라이브러리에 추가합니다.

```
fl.getDocumentDOM().library.importEmbeddedSWF("MyMovie", swfData, "Intro");
```

library.itemExists()

지원 버전

Flash MX 2004

사용법

```
library.itemExists(namePath)
```

매개 변수

namePath 항목의 이름을 지정하는 문자열. 항목이 폴더 안에 있으면 슬래시 표기법을 사용하여 해당 이름과 경로를 지정합니다.

반환값

부울 값, 라이브러리에 지정된 항목이 있으면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

설명

메서드, 지정된 항목이 라이브러리에 있는지 확인합니다.

예제

다음 예제에서는 Folder_1 라이브러리 폴더에 Symbol_1 항목이 있는지 여부에 따라 대화 상자에 **true** 또는 **false**를 표시합니다.

```
alert(fl.getDocumentDOM().library.itemExists('Folder_1/Symbol_1'));
```

library.items

지원 버전

Flash MX 2004

사용법

```
library.items
```

설명

속성, 라이브러리 항목 객체의 배열입니다.

예제

다음 예제에서는 모든 라이브러리 항목의 배열을 `itemArray` 변수에 저장합니다.

```
var itemArray = fl.getDocumentDOM().library.items;
```

library.moveToFolder()

지원 버전

Flash MX 2004

사용법

```
library.moveToFolder(folderPath [, itemToMove [, bReplace]])
```

매개 변수

folderPath "FolderName" 또는 "FolderName/FolderName" 형식으로 폴더에 대한 경로를 지정하는 문자열. 항목을 최상위로 이동하려면 **folderPath**에 빈 문자열("")을 지정합니다.

itemToMove 이동할 항목의 이름을 지정하는 문자열. **itemToMove**를 지정하지 않으면 현재 선택한 항목이 이동합니다. 이 매개 변수는 선택 사항입니다.

bReplace 부울 값. 같은 이름을 가진 항목이 이미 있는 경우 **bReplace** 매개 변수에 **true**를 지정하면 기존 항목이 이동하려는 항목으로 바뀌고 **false**를 지정하면 이동하려는 항목의 이름이 고유한 이름으로 변경됩니다. 기본값은 **false**입니다. 이 매개 변수는 선택 사항입니다.

반환값

부울 값, 항목이 성공적으로 이동하면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

설명

메서드, 현재 선택한 라이브러리 항목 또는 지정된 라이브러리 항목을 지정된 폴더로 이동합니다. **folderPath** 매개 변수가 비어 있으면 항목이 최상위로 이동합니다.

예제

다음 예제에서는 `Symbol_1` 항목을 `new` 라이브러리 폴더로 이동하고 이 항목으로 해당 폴더에 있는 같은 이름의 항목을 대체합니다.

```
fl.getDocumentDOM().library.moveToFolder("new", "Symbol_1", true);
```

library.newFolder()

지원 버전

Flash MX 2004

사용법

```
library.newFolder([folderPath])
```

매개 변수

folderPath 만들 폴더의 이름을 지정하는 문자열. 경로로 지정된 경우 해당 경로가 없으면 자동으로 만들어집니다. 이 매개 변수는 선택 사항입니다.

반환값

부울 값, 폴더가 성공적으로 만들어지면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

설명

메서드, 현재 선택한 폴더에 지정된 이름을 가진 새 폴더를 만들거나 **folderName** 매개 변수가 제공되지 않은 경우 기본 이름 ("untitled folder #")이 지정된 새 폴더를 만듭니다.

예제

다음 예제에서는 두 개의 새 라이브러리 폴더를 만듭니다. 두 번째 폴더는 첫 번째 폴더의 하위 폴더입니다.

```
fl.getDocumentDOM().library.newFolder("first/second");
```

library.renameItem()

지원 버전

Flash MX 2004

사용법

```
library.renameItem(name)
```

매개 변수

name 라이브러리 항목의 새 이름을 지정하는 문자열

반환값

항목의 이름이 성공적으로 변경되면 **true**를 반환하고 그렇지 않으면 **false**를 반환하는 부울 값. 여러 항목이 선택되어 있고 변경된 이름이 없으면 **false**를 반환합니다(사용자 인터페이스 비헤이비어와 같음).

설명

메서드, [라이브러리] 패널에서 현재 선택한 라이브러리 항목의 이름을 변경합니다.

예제

다음 예제에서는 현재 선택한 라이브러리 항목의 이름을 **new name**으로 변경합니다.

```
fl.getDocumentDOM().library.renameItem("new name");
```

library.selectAll()

지원 버전

Flash MX 2004

사용법

```
library.selectAll([bSelectAll])
```

매개 변수

bSelectAll 모든 라이브러리 항목을 선택할 것인지 아니면 선택 취소할 것인지를 지정하는 부울 값. 모든 라이브러리 항목을 선택하려면 이 매개 변수를 생략하거나 기본값인 **true**를 사용하고 모든 라이브러리 항목을 선택 취소하려면 **false**를 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 모든 라이브러리 항목을 선택하거나 선택 취소합니다.

예제

다음 예제에서는 모든 라이브러리 항목을 선택합니다.

```
fl.getDocumentDOM().library.selectAll();  
fl.getDocumentDOM().library.selectAll(true);
```

다음 예제에서는 모든 라이브러리 항목을 선택 취소합니다.

```
fl.getDocumentDOM().library.selectAll(false);  
fl.getDocumentDOM().library.selectNone();
```

library.selectItem()

지원 버전

Flash MX 2004

사용법

```
library.selectItem(namePath [, bReplaceCurrentSelection [, bSelect]])
```

매개 변수

namePath 항목의 이름을 지정하는 문자열. 항목이 폴더 안에 있으면 슬래시 표기법을 사용하여 해당 이름과 경로를 지정할 수 있습니다.

bReplaceCurrentSelection 현재 선택한 항목을 바꿀 것인지 아니면 현재 선택한 항목에 항목을 추가할 것인지를 지정하는 부울 값. 기본값은 현재 선택한 항목을 바꾸는 **true**입니다. 이 매개 변수는 선택 사항입니다.

bSelect 항목을 선택할 것인지 아니면 선택 취소할 것인지를 지정하는 부울 값. 기본값은 **true**(선택)입니다. 이 매개 변수는 선택 사항입니다.

반환값

부울 값, 지정된 항목이 있으면 **true**를 반환하고 그렇지 않으면 **false**를 반환합니다.

설명

메서드, 지정된 라이브러리 항목을 선택합니다.

예제

다음 예제에서는 현재 선택한 라이브러리 항목을 제목이 없는 Folder_1에 있는 Symbol_1로 변경합니다.

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1");
```

다음 예제에서는 현재 선택한 라이브러리 항목을 제목이 없는 Folder_1에 있는 Symbol_1을 포함하도록 확장합니다.

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1", false);
```

다음 예제에서는 선택한 다른 항목은 그대로 두고 제목이 없는 Folder_1에 있는 Symbol_1만 선택 취소합니다.

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1", true, false);
```

library.selectNone()

지원 버전
Flash MX 2004

사용법

```
library.selectNone()
```

매개 변수
없음

반환값
없음

설명

메서드, 모든 라이브러리 항목을 선택 취소합니다.

예제

다음 예제에서는 모든 라이브러리 항목을 선택 취소합니다.

```
fl.getDocumentDOM().library.selectNone();  
fl.getDocumentDOM().library.selectAll(false);
```

library.setItemProperty()

지원 버전
Flash MX 2004

사용법

```
library.setItemProperty(property, value)
```

매개 변수

property 설정할 속성의 이름을 나타내는 문자열. 속성의 목록은 [Item 객체](#)의 속성 요약 표 및 해당 하위 클래스의 속성 요약을 참조하십시오. 어떤 객체가 Item 객체의 하위 클래스인지 확인하려면 10페이지의 “[DOM 구조의 요약](#)”을 참조하십시오.

value 지정된 속성에 할당할 값

반환값
없음

설명

메서드, 선택한 모든 라이브러리 항목(폴더 제외)의 속성을 설정합니다.

예제

다음 예제에서는 선택한 라이브러리 항목의 **symbolType** 속성에 값 **button**을 할당합니다. 이 경우 항목이 [SymbolItem 객체](#)여야 하기 때문에 **symbolType**은 SymbolItem 객체의 유효한 속성입니다.

```
fl.getDocumentDOM().library.setItemProperty("symbolType", "button");
```


library.updateItem()

지원 버전

Flash MX 2004

사용법

```
library.updateItem([namePath])
```

매개 변수

namePath 항목의 이름을 지정하는 문자열. 항목이 폴더 안에 있으면 슬래시 표기법을 사용하여 해당 이름과 경로를 지정합니다. 이것은 항목을 마우스 오른쪽 버튼으로 클릭하고 사용자 인터페이스 메뉴에서 [업데이트]를 선택하는 것과 같습니다. 이름을 제공하지 않으면 현재 선택한 항목이 업데이트됩니다. 이 매개 변수는 선택 사항입니다.

반환값

부울 값, Flash에서 항목이 성공적으로 업데이트되면 true를 반환하고 그렇지 않으면 false를 반환합니다.

설명

메서드, 지정된 항목을 업데이트합니다.

예제

다음 예제에서는 현재 선택한 항목이 업데이트되었는지 여부에 따라 true 또는 false를 표시하는 대화 상자를 보여 줍니다.

```
alert(fl.getDocumentDOM().library.updateItem());
```

27장: Math 객체

지원 버전

Flash MX 2004

설명

Math 객체는 flash 객체의 읽기 전용 속성으로 사용할 수 있습니다([fl.Math](#) 참조). 이 객체는 일반적인 수학 연산을 수행하는 메서드를 제공합니다.

메서드 요약

Math 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
Math.concatMatrix()	행렬 연결을 수행하고 그 결과를 반환합니다.
Math.invertMatrix()	지정된 행렬의 역행렬을 반환합니다.
Math.pointDistance()	두 점 사이의 거리를 계산합니다.

Math.concatMatrix()

지원 버전

Flash MX 2004

사용법

```
Math.concatMatrix(mat1, mat2)
```

매개 변수

mat1, mat2 연결할 Matrix 객체를 지정합니다(see [Matrix 객체](#) 참조). 각 매개 변수는 a, b, c, d, tx 및 ty 필드가 있는 객체여야 합니다.

반환값

연결된 객체 행렬

설명

메서드, 행렬 연결을 수행하고 그 결과를 반환합니다.

예제

다음 예제에서는 현재 선택한 객체를 elt 변수에 저장하고 객체 행렬을 뷰 행렬에 곱한 다음 그 값을 mat 변수에 저장합니다.

```
var elt = fl.getDocumentDOM().selection[0];
var mat = fl.Math.concatMatrix( elt.matrix , fl.getDocumentDOM().viewMatrix );
```

Math.invertMatrix()

지원 버전
Flash MX 2004

사용법

```
Math.invertMatrix(mat)
```

매개 변수

행렬 반전시킬 Matrix 객체를 나타냅니다([Matrix 객체](#) 참조). 이 객체에는 a, b, c, d, tx 및 ty 필드가 있어야 합니다.

반환값

원행렬의 역행렬인 Matrix 객체

설명

메서드, 지정된 행렬의 역행렬을 반환합니다.

예제

다음 예제에서는 현재 선택한 객체를 elt 변수에 저장하고 해당 행렬을 mat 변수에 할당한 다음 그 역행렬을 inv 변수에 저장합니다.

```
var elt = fl.getDocumentDOM().selection[0];  
var mat = elt.matrix;  
var inv = fl.Math.invertMatrix( mat );
```

Math.pointDistance()

지원 버전
Flash MX 2004

사용법

```
Math.pointDistance(pt1, pt2)
```

매개 변수

pt1, pt2 거리를 측정할 두 점을 지정합니다.

반환값

두 점 사이의 거리를 나타내는 부동 소수점 값

설명

메서드, 두 점 사이의 거리를 계산합니다.

예제

다음 예제에서는 pt1과 pt2 사이의 거리 값을 dist 변수에 저장합니다.

```
var pt1 = {x:10, y:20}  
var pt2 = {x:100, y:200}  
var dist = fl.Math.pointDistance(pt1, pt2);
```

28장: Matrix 객체

지원 버전
Flash MX 2004

설명
Matrix 객체는 변형 행렬을 나타냅니다.

속성 요약
Matrix 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
matrix.a	변형 행렬의 (0,0) 요소를 지정하는 부동 소수점 값
matrix.b	행렬의 (0,1) 요소를 지정하는 부동 소수점 값
matrix.c	행렬의 (1,0) 요소를 지정하는 부동 소수점 값
matrix.d	행렬의 (1,1) 요소를 지정하는 부동 소수점 값
matrix.tx	심볼의 등록 포인트 또는 모양의 중심에 대한 x 축 위치를 지정하는 부동 소수점 값
matrix.ty	심볼의 등록 포인트 또는 모양의 중심에 대한 y 축 위치를 지정하는 부동 소수점 값

matrix.a

지원 버전
Flash MX 2004

사용법
matrix.a

설명
속성, 변형 행렬의 (0,0) 요소를 지정하는 부동 소수점 값입니다. 이 값은 객체의 **x**축에 대한 배율 인수를 나타냅니다.

예제
행렬의 a 및 d 속성은 배율을 나타냅니다. 다음 예제에서는 값을 각각 2와 3으로 설정하여 선택한 객체의 폭과 높이를 각각 2배와 3배로 조절합니다.

```
var mat = fl.getDocumentDOM().selection[0].matrix;
mat.a = 2;
mat.d = 3;
fl.getDocumentDOM().selection[0].matrix = mat;
```

a = d 및 b = -c와 같이 서로 연관된 a, b, c 및 d 행렬 속성을 설정하여 객체를 회전시킬 수 있습니다. 예를 들어, 값 0.5, 0.8, -0.8 및 0.5는 객체를 60° 회전시킵니다.

```
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;
```

객체를 원래 모양으로 되돌리려면 $a = d = 1$ 및 $c = b = 0$ 을 설정하면 됩니다.

matrix.b

지원 버전
Flash MX 2004

사용법
matrix.b

설명

속성, 행렬의 (0,1) 요소를 지정하는 부동 소수점 값입니다. 이 값은 모양을 수직으로 기울입니다. 즉, 모양의 오른쪽 가장자리를 수직 축을 따라 이동시킵니다.

행렬의 matrix.b 및 matrix.c 속성은 기울임 정도를 나타냅니다([matrix.c](#) 참조).

예제

다음 예제에서는 b와 c를 각각 -1과 0으로 설정할 수 있습니다. 그러면 객체가 수직 축을 기준으로 45° 기울어집니다.

```
var mat = fl.getDocumentDOM().selection[0].matrix;  
mat.b = -1;  
mat.c = 0;  
fl.getDocumentDOM().selection[0].matrix = mat;
```

객체의 기울기를 원래대로 되돌리려면 b와 c를 0으로 설정하면 됩니다.

자세한 내용은 [matrix.a](#) 예제를 참조하십시오.

matrix.c

지원 버전
Flash MX 2004

사용법
matrix.c

설명

속성, 행렬의 (1,0) 요소를 지정하는 부동 소수점 값입니다. 이 값은 객체의 아래쪽 가장자리를 수평 축을 따라 이동시켜 객체를 기울입니다.

행렬의 matrix.b 및 matrix.c 속성은 기울임 정도를 나타냅니다.

예제

자세한 내용은 [matrix.b](#) 예제를 참조하십시오.

matrix.d

지원 버전

Flash MX 2004

사용법

`matrix.d`

설명

속성, 행렬의 (1,1) 요소를 지정하는 부동 소수점 값입니다. 이 값은 객체의 **y**축에 대한 배율 인수를 나타냅니다.

예제

자세한 내용은 [matrix.a](#) 예제를 참조하십시오.

matrix.tx

지원 버전

Flash MX 2004

사용법

`matrix.tx`

설명

속성, 심볼의 등록 포인트(원점) 또는 모양의 중심에 대한 **x**축 위치를 지정하는 부동 소수점 값입니다. 이 값은 변형을 위한 **x** 이동 값을 정의합니다.

`matrix.tx` 및 `matrix.ty` 속성을 설정하여 객체를 이동할 수 있습니다([matrix.ty](#) 참조).

예제

다음 예제에서는 `tx` 및 `ty`를 0으로 설정하여 객체의 등록 포인트를 문서의 0,0 위치로 이동합니다.

```
var mat = fl.getDocumentDOM().selection[0].matrix;  
mat.tx = 0;  
mat.ty = 0;  
fl.getDocumentDOM().selection[0].matrix = mat;
```

matrix.ty

지원 버전

Flash MX 2004

사용법

`matrix.ty`

설명

속성, 심볼의 등록 포인트 또는 모양의 중심에 대한 **y**축 위치를 지정하는 부동 소수점 값입니다. 이 값은 변형을 위한 **y** 이동 값을 정의합니다.

`matrix.tx` 및 `matrix.ty` 속성을 설정하여 객체를 이동할 수 있습니다.

예제

자세한 내용은 [matrix.tx](#) 예제를 참조하십시오.

29장: outputPanel 객체

지원 버전

Flash MX 2004

설명

이 객체는 구문 오류와 같은 문제 해결 정보를 표시하는 [출력] 패널을 나타냅니다. 이 객체에 액세스하려면 `fl.outputPanel`(또는 `flash.outputPanel`)을 사용합니다. [fl.outputPanel](#)을 참조하십시오.

메서드 요약

`outputPanel` 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
outputPanel.clear()	[출력] 패널의 내용을 지웁니다.
outputPanel.save()	[출력] 패널의 내용을 로컬 텍스트 파일에 저장합니다.
outputPanel.trace()	[출력] 패널의 내용에 줄을 추가합니다. 새 줄은 맨 아래에 표시됩니다.

outputPanel.clear()

지원 버전

Flash MX 2004

사용법

```
outputPanel.clear();
```

매개 변수

없음

반환값

없음

설명

메서드, [출력] 패널의 내용을 지웁니다. 이 메서드는 일괄 처리 응용 프로그램에서 오류 목록을 지우는 데 사용하거나 [outputPanel.save\(\)](#)를 통해 오류 목록을 증분적으로 저장하는 데 사용할 수 있습니다.

예제

다음 예제에서는 [출력] 패널의 현재 내용을 지웁니다.

```
fl.outputPanel.clear();
```


outputPanel.save()

지원 버전

Flash MX 2004, Flash 8의 경우 **bUseSystemEncoding** 매개 변수를 추가로 사용할 수 있습니다.

사용법

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

매개 변수

fileURI [출력] 패널의 내용을 포함할 로컬 파일을 지정하는 문자열(file:/// URI로 표시됨)

bAppendToFile 선택적 부울 값. true인 경우 [출력] 패널의 내용을 출력 파일에 추가하고 false인 경우 기존 출력 파일(있을 경우)을 덮어씁니다. 기본값은 false입니다.

bUseSystemEncoding 선택적 부울 값. true인 경우 시스템 인코딩을 사용하여 [출력] 패널 텍스트를 저장하고 false인 경우 텍스트 시작 부분에 BOM(Byte Order Mark) 문자가 있는 UTF-8 인코딩을 사용하여 [출력] 패널 텍스트를 저장합니다. 기본값은 false입니다.

반환값

없음

설명

메서드, 파일을 덮어쓰거나 파일에 추가하여 [출력] 패널의 내용을 로컬 텍스트 파일에 저장합니다.

fileURI가 잘못되었거나 지정되지 않은 경우에는 오류가 보고됩니다.

이 메서드는 일괄 처리 작업에 유용합니다. 예를 들어, 여러 구성 요소를 컴파일하는 JSFL 파일을 만들 수 있습니다. 이때 [출력] 패널에 컴파일 오류가 표시되면 이 메서드를 사용하여 텍스트 파일에 이러한 오류를 저장할 수 있습니다. 이 텍스트 파일은 사용 중인 빌드 시스템을 통해 자동으로 구문 분석할 수 있습니다.

예제

다음 예제에서는 [출력] 패널의 내용을 /tests 폴더의 batch.log 파일에 저장하여 batch.log 파일이 이미 존재하는 경우 이 파일을 덮어씁니다.

```
fl.outputPanel.save("file:///c:/tests/batch.log");
```

outputPanel.trace()

지원 버전

Flash MX 2004

사용법

```
outputPanel.trace(message)
```

매개 변수

message 출력 패널에 추가할 텍스트가 포함된 문자열.

반환값

없음

설명

메서드, 새 행으로 끝나는 텍스트 문자열을 [출력] 패널로 보내고 아직 표시되지 않은 경우 [출력] 패널에 표시합니다. 이 메서드는 `fl.trace()`와 동일하며 `ActionScript`의 `trace()` 명령문과 같은 방식으로 작동합니다.

빈 행을 보내려면 `outputPanel.trace("")` 또는 `outputPanel.trace("\n")`를 사용합니다. 두 번째 명령 인라인을 사용하여 `\n`을 **message** 문자열의 일부로 만들 수 있습니다.

예제

다음 예제에서는 여러 텍스트 행을 [출력] 패널에 표시합니다.

```
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 + "?");
```

30장: Oval 객체

상속 [Element 객체](#) > [Shape 객체](#) > Oval 객체

지원 버전

Flash CS3 Professional

설명

Oval 객체는 [프리미티브 타원형 도구]를 사용하여 그려지는 모양입니다. 항목이 Oval 객체인지 확인하려면 [shape.isOvalObject](#) 를 사용합니다.

속성 요약

[Shape 객체](#) 속성 외에도 Oval 객체에는 다음과 같은 속성을 사용할 수 있습니다. Oval 객체의 속성을 설정하려면 [document.setOvalObjectProperty\(\)](#) 를 사용합니다.

속성	설명
OvalObject.closePath	읽기 전용, 속성 관리자에서 [패스 닫기] 체크 상자가 선택되어 있는지 여부를 지정하는 부울 값
OvalObject.endAngle	읽기 전용, Oval 객체의 종료 각도를 지정하는 부동 소수점 값
OvalObject.innerRadius	읽기 전용, Oval 객체의 내부 반경을 백분율로 지정하는 부동 소수점 값
OvalObject.startAngle	읽기 전용, Oval 객체의 시작 각도를 지정하는 부동 소수점 값

OvalObject.closePath

지원 버전

Flash CS3 Professional

사용법

`OvalObject.closePath`

설명

읽기 전용 속성, 속성 관리자에서 [패스 닫기] 체크 상자가 선택되어 있는지 여부를 지정하는 부울 값입니다. 객체의 시작 각도 및 종료 각도 값이 같은 경우 이 속성을 설정하면 값이 변경될 때까지 아무런 영향을 주지 않습니다.

이 값을 설정하려면 [document.setOvalObjectProperty\(\)](#) 를 사용합니다.

예제

다음 예제에서는 `OvalObject.closePath` 속성을 선택 취소합니다.

```
f1.getDocumentDOM().setOvalObjectProperty("closePath", false);
```

참고 사항

[document.setOvalObjectProperty\(\)](#), [shape.isOvalObject](#)

OvalObject.endAngle

지원 버전

Flash CS3 Professional

사용법

`OvalObject.endAngle`

설명

읽기 전용 속성, Oval 객체의 종료 각도를 지정하는 부동 소수점 값입니다. 사용할 수 있는 값은 0부터 360까지입니다.

이 값을 설정하려면 [document.setOvalObjectProperty\(\)](#)를 사용합니다.

예제

다음 예제에서는 선택된 Oval 객체의 종료 각도를 270으로 설정합니다.

```
f1.getDocumentDOM().setOvalObjectProperty("endAngle", 270);
```

참고 사항

[document.setOvalObjectProperty\(\)](#), [OvalObject.startAngle](#), [shape.isOvalObject](#)

OvalObject.innerRadius

지원 버전

Flash CS3 Professional

사용법

`OvalObject.innerRadius`

설명

읽기 전용 속성, Oval 객체의 내부 반경을 백분율로 지정하는 부동 소수점 값입니다. 사용할 수 있는 값은 0부터 99까지입니다.

이 값을 설정하려면 [document.setOvalObjectProperty\(\)](#)를 사용합니다.

예제

다음 예제에서는 선택된 Oval 객체의 내부 반지름을 50%로 설정합니다.

```
f1.getDocumentDOM().setOvalObjectProperty("innerRadius", 50);
```

참고 사항

[document.setOvalObjectProperty\(\)](#), [shape.isOvalObject](#)

OvalObject.startAngle

지원 버전

Flash CS3 Professional

사용법

`OvalObject.startAngle`

설명

읽기 전용 속성, Oval 객체의 시작 각도를 지정하는 부동 소수점 값입니다. 사용할 수 있는 값은 0부터 360까지입니다. 이 값을 설정하려면 [document.setOvalObjectProperty\(\)](#)를 사용합니다.

예제

다음 예제에서는 선택된 Oval 객체의 시작 각도를 270으로 설정합니다.

```
fl.getDocumentDOM().setOvalObjectProperty("startAngle", 270);
```

참고 사항

[document.setOvalObjectProperty\(\)](#), [OvalObject.endAngle](#), [shape.isOvalObject](#)

31장: Parameter 객체

지원 버전

Flash MX 2004

설명

Parameter 객체 유형은 `componentInstance.parameters` 배열(제작 도구의 구성 요소 속성 관리자에 해당)에서 액세스합니다.

메서드 요약

Parameter 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
<code>parameter.insertItem()</code>	목록, 객체 또는 배열에 항목을 삽입합니다.
<code>parameter.removeItem()</code>	스크린 또는 구성 요소 매개 변수에 대한 목록, 객체 또는 배열 유형의 요소를 제거합니다.

속성 요약

Parameter 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>parameter.category</code>	screen 매개 변수 또는 <code>componentInstance</code> 매개 변수의 <code>category</code> 속성을 지정하는 문자열입니다.
<code>parameter.listIndex</code>	선택한 목록 항목의 값을 지정하는 정수입니다.
<code>parameter.name</code>	읽기 전용, 매개 변수의 이름을 지정하는 문자열입니다.
<code>parameter.value</code>	구성 요소 관리자의 [매개 변수] 탭, 속성 관리자의 [매개 변수] 탭 또는 스크린 속성 관리자에 있는 값 필드에 해당합니다.
<code>parameter.valueType</code>	읽기 전용, 스크린 또는 구성 요소 매개 변수의 유형을 나타내는 문자열입니다.
<code>parameter.verbose</code>	매개 변수가 표시되는 위치를 지정합니다.

parameter.category

지원 버전

Flash MX 2004

사용법

`parameter.category`

설명

속성, screen 매개 변수 또는 `componentInstance` 매개 변수의 `category` 속성을 지정하는 문자열입니다. 이 속성은 매개 변수 목록을 표시하는 다른 방법을 제공합니다. Flash 사용자 인터페이스에서는 이 기능이 제공되지 않습니다.

parameter.insertItem()

지원 버전

Flash MX 2004

사용법

```
parameter.insertItem(index, name, value, type)
```

매개 변수

index 목록, 객체 또는 배열에서 항목이 삽입되는 위치를 지정하는 정수 인덱스(0부터 시작). 인덱스가 0이면 목록의 시작 부분에 항목이 삽입되고 인덱스가 목록 크기보다 크면 배열의 끝에 새 항목이 삽입됩니다.

name 삽입할 항목의 이름을 지정하는 문자열. 객체 매개 변수에 대한 필수 매개 변수입니다.

value 삽입할 항목의 값을 지정하는 문자열

type 삽입할 항목의 유형을 지정하는 문자열

반환값

없음

설명

메서드, 목록, 객체 또는 배열에 항목을 삽입합니다. 매개 변수가 목록, 객체 또는 배열이면 **value** 속성은 배열입니다.

예제

다음 예제에서는 labelPlacement 매개 변수에 New Value의 값을 삽입합니다.

```
// 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

지원 버전

Flash MX 2004

사용법

```
parameter.listIndex
```

설명

속성, 선택한 목록 항목의 값입니다. 이 속성은 parameter.valueType이 "List"인 경우에만 유효합니다.

예제

다음 예제에서는 autoKeyNav 매개 변수인 슬라이드의 첫 번째 매개 변수를 설정합니다. 매개 변수를 사용할 수 있는 값(true, false 또는 inherit) 중 하나로 설정하려면 parameter.listIndex를 목록에 있는 항목의 인덱스로 설정합니다(true의 경우 0, false의 경우 1, inherit의 경우 2).

```
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters;  
parms[0].listIndex = 1;
```

parameter.name

지원 버전
Flash MX 2004

사용법
parameter.name

설명
읽기 전용 속성, 매개 변수의 이름을 지정하는 문자열입니다.

예제
다음 예제에서는 선택한 구성 요소의 다섯 번째 매개 변수 이름을 보여 줍니다.

```
var parms = fl.getDocumentDOM().selection[0].parameters;  
fl.trace("name: " + parms[4].name);
```

다음 예제에서는 지정된 스크린의 다섯 번째 매개 변수 이름을 보여 줍니다.

```
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters; fl.trace("name: " + parms[4].name);
```

parameter.removeItem()

지원 버전
Flash MX 2004

사용법
parameter.removeItem(index)

매개 변수
index 스크린 또는 구성 요소 속성에서 제거할 항목의 0부터 시작하는 정수 인덱스

반환값
없음

설명
메서드, 스크린 또는 구성 요소 매개 변수에 대한 목록, 객체 또는 배열 유형의 요소를 제거합니다.

예제
다음 예제에서는 구성 요소의 labelPlacement 매개 변수에서 인덱스 1에 있는 요소를 제거합니다.


```
// 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);
}
```

다음 예제에서는 스크린의 `autoKeyNav` 매개 변수에서 인덱스 1에 있는 요소를 제거합니다.

```
// 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

지원 버전
Flash MX 2004

사용법
`parameter.value`

설명
속성, 스크린 속성 관리자, 속성 관리자의 [매개 변수] 탭 또는 구성 요소 관리자의 [매개 변수] 탭에 있는 [값] 필드에 해당합니다. `value` 속성의 유형은 매개 변수의 `valueType` 속성에 의해 결정됩니다([parameter.valueType](#) 참조).

parameter.valueType

지원 버전
Flash MX 2004

사용법
`parameter.valueType`

설명

읽기 전용 속성, 스크린 또는 구성 요소 매개 변수의 유형을 나타내는 문자열입니다. 이 유형에 사용할 수 있는 값에는 "Default", "Array", "Object", "List", "String", "Number", "Boolean", "Font Name", "Color", "Collection", "Web Service URL" 또는 "Web Service Operation"이 있습니다.

참고 사항

[parameter.value](#)

parameter.verbose

지원 버전

Flash MX 2004

사용법

`parameter.verbose`

설명

속성, 매개 변수가 표시되는 위치를 지정합니다. 이 속성 값이 0(간략히)이면 매개 변수가 구성 요소 관리자에만 표시됩니다. 이 속성 값이 1(자세히)이면 매개 변수가 구성 요소 관리자 및 속성 관리자의 [매개 변수] 탭에 표시됩니다.

32장: Path 객체

지원 버전

Flash MX 2004

설명

Path 객체는 주로 확장 가능한 도구를 만들 때 사용하는 일련의 선분(직선, 곡선 또는 둘 다)을 정의합니다. 다음 예제에서는 flash 객체에서 반환되는 Path 객체의 인스턴스를 보여 줍니다.

```
path = fl.drawingLayer.newPath();
```

자세한 내용은 [drawingLayer 객체](#)를 참조하십시오.

메서드 요약

Path 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
path.addCubicCurve()	패스에 3차원 베지어 곡선을 추가합니다.
path.addCurve()	패스에 2차원 베지어 곡선을 추가합니다.
path.addPoint()	패스에 점을 추가합니다.
path.clear()	패스에서 모든 점을 제거합니다.
path.close()	패스의 첫 번째 점 위치에 패스를 닫는 점을 추가하고 패스를 이 점까지 확장합니다.
path.makeShape()	현재 획 및 채우기 설정을 사용하여 스테이지에 모양을 만듭니다.
path.newContour()	패스에서 새 윤곽선을 시작합니다.

속성 요약

Path 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
path.nPts	읽기 전용, 패스에 있는 점의 수를 나타내는 정수

path.addCubicCurve()

지원 버전

Flash MX 2004

사용법

```
path.addCubicCurve(xAnchor, yAnchor, x2, y2, x3, y3, x4, y4)
```

매개 변수

xAnchor 첫 번째 제어점의 **x** 위치를 지정하는 부동 소수점 숫자

yAnchor 첫 번째 제어점의 **y** 위치를 지정하는 부동 소수점 숫자

x2 두 번째 제어점의 **x** 위치를 지정하는 부동 소수점 숫자

y2 두 번째 제어점의 y 위치를 지정하는 부동 소수점 숫자
x3 세 번째 제어점의 x 위치를 지정하는 부동 소수점 숫자
y3 세 번째 제어점의 y 위치를 지정하는 부동 소수점 숫자
x4 네 번째 제어점의 x 위치를 지정하는 부동 소수점 숫자
y4 네 번째 제어점의 y 위치를 지정하는 부동 소수점 숫자

반환값

없음

설명

메서드, 패스에 3차원 베지어 곡선을 추가합니다.

예제

다음 예제에서는 새 패스를 만들어 myPath 변수에 저장하고 패스에 곡선을 할당합니다.

```
var myPath = fl.drawingLayer.newPath();  
myPath.addCubicCurve(0, 0, 10, 20, 20, 20, 30, 0);
```

path.addCurve()

지원 버전

Flash MX 2004

사용법

```
path.addCurve(xAnchor, yAnchor, x2, y2, x3, y3)
```

매개 변수

xAnchor 첫 번째 제어점의 x 위치를 지정하는 부동 소수점 숫자
yAnchor 첫 번째 제어점의 y 위치를 지정하는 부동 소수점 숫자
x2 두 번째 제어점의 x 위치를 지정하는 부동 소수점 숫자
y2 두 번째 제어점의 y 위치를 지정하는 부동 소수점 숫자
x3 세 번째 제어점의 x 위치를 지정하는 부동 소수점 숫자
y3 세 번째 제어점의 y 위치를 지정하는 부동 소수점 숫자

반환값

없음

설명

메서드, 패스에 2차원 베지어 곡선을 추가합니다.

예제

다음 예제에서는 새 패스를 만들어 myPath 변수에 저장하고 패스에 곡선을 할당합니다.

```
var myPath = fl.drawingLayer.newPath();  
myPath.addCurve(0, 0, 10, 20, 20, 0);
```

path.addPoint()

지원 버전
Flash MX 2004

사용법
`path.addPoint(x, y)`

매개 변수
x 점의 x 위치를 지정하는 부동 소수점 숫자
y 점의 y 위치를 지정하는 부동 소수점 숫자

반환값
없음

설명
메서드, 패스에 점을 추가합니다.

예제
다음 예제에서는 새 패스를 만들어 myPath 변수에 저장하고 이 패스에 새 점을 할당합니다.

```
var myPath = fl.drawingLayer.newPath();  
myPath.addPoint(10, 100);
```

path.clear()

지원 버전
Flash MX 2004

사용법
`path.clear()`

매개 변수
없음

반환값
없음

설명
메서드, 패스에서 모든 점을 제거합니다.

예제
다음 예제에서는 myPath 변수에 저장된 패스에서 모든 점을 제거합니다.

```
var myPath = fl.drawingLayer.newPath();  
myPath.clear();
```

path.close()

지원 버전
Flash MX 2004

사용법
`path.close()`

매개 변수
없음

반환값
없음

설명
메서드, 패스의 첫 번째 점 위치에 패스를 닫는 점을 추가하고 패스를 이 점까지 확장합니다. 패스에 기존 점이 없으면 새 점이 추가되지 않습니다.

예제
다음 예제에서는 닫힌 패스를 만듭니다.

```
var myPath = fl.drawingLayer.newPath();  
myPath.close();
```

path.makeShape()

지원 버전
Flash MX 2004

사용법
`path.makeShape([bSupressFill [, bSupressStroke]])`

매개 변수
bSupressFill true로 설정된 경우 모양에 채우기가 적용되지 않도록 하는 부울 값. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

bSupressStroke true로 설정된 경우 모양에 획이 적용되지 않도록 하는 부울 값. 기본값은 false입니다. 이 매개 변수는 선택 사항입니다.

반환값
없음

설명
메서드, 현재 획 및 채우기 설정을 사용하여 스테이지에 모양을 만듭니다. 패스는 모양이 만들어진 후에 지워집니다. 이 메서드에는 결과로 만들어지는 **shape** 객체에 채우기와 획이 적용되지 않도록 하는 두 개의 선택적 매개 변수가 있습니다. 이들 매개 변수를 생략하거나 false로 설정하면 채우기 및 획의 현재 값이 사용됩니다.

예제

다음 예제에서는 획 없이 현재 채우기가 적용된 모양을 만듭니다.

```
var myPath = fl.drawingLayer.newPath();  
myPath.makeShape(false, true);
```

path.newContour()

지원 버전

Flash MX 2004

사용법

```
path.newContour()
```

매개 변수

없음

반환값

없음

설명

메서드, 패스에서 새 윤곽선을 시작합니다.

예제

다음 예제에서는 속이 빈 정사각형을 만듭니다.

```
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

지원 버전

Flash MX 2004

사용법

```
path.nPts
```

설명

읽기 전용 속성, 패스에 있는 점의 수를 나타내는 정수입니다. 새 패스에는 점이 없습니다.

예제

다음 예제에서는 myPath 변수에서 참조하는 패스에 있는 점의 수를 [출력] 패널에 표시합니다.

```
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
```


33장: presetItem 객체

지원 버전

Flash CS4 Professional

설명

presetItem 객체는 [모션 프리셋] 패널([윈도우] > [모션 프리셋]) 항목(프리셋 또는 폴더)을 나타냅니다. presetItem 객체의 배열은 presetPanel 객체(presetPanel.items)의 속성입니다.

presetItem 객체의 모든 속성은 읽기 전용입니다. 항목 삭제, 이름 변경 또는 이동과 같은 작업을 수행하려면 presetPanel 객체의 메서드를 사용합니다.

속성 요약

presetItem 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
presetItem.isDefault	항목이 Flash와 함께 설치되는지 여부 또는 항목이 사용자 또는 다른 사용자가 만든 사용자 정의 항목인지 여부를 지정합니다.
presetItem.isFolder	[모션 프리셋] 패널의 항목이 폴더인지 아니면 프리셋인지 여부를 지정합니다.
presetItem.level	[모션 프리셋] 패널 폴더 구조의 항목 레벨입니다.
presetItem.name	프리셋 또는 폴더의 이름(경로 정보는 포함하지 않음)입니다.
presetItem.open	[모션 프리셋] 패널의 폴더가 현재 확장되어 있는지 여부를 지정합니다.
presetItem.path	[모션 프리셋] 패널 폴더 트리에 있는 항목의 경로와 항목 이름입니다.

presetItem.isDefault

지원 버전

Flash CS4 Professional

사용법

```
presetItem.isDefault
```

설명

읽기 전용 속성, 항목이 Flash와 함께 설치되는지(true) 아니면 사용자가 직접 또는 다른 사용자가 만든 사용자 정의 항목인지(false)를 지정하는 부울 값입니다. 이 값이 true이면 해당 항목은 이동 또는 삭제하거나 유사한 작업을 적용할 수 없는 "읽기 전용" 항목으로 간주할 수 있습니다.

예제

다음 예제에서는 [모션 프리셋] 패널의 내용을 표시하고 항목이 Flash와 함께 설치되는지 여부를 나타냅니다.

```
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

지원 버전

Flash CS4 Professional

사용법

```
presetItem.isFolder
```

설명

읽기 전용 속성, [모션 패널] 패널의 항목이 폴더인지(true) 아니면 프리셋인지(false)를 지정하는 부울 값입니다.

예제

다음 예제에서는 [모션 프리셋] 패널의 첫 번째 항목이 폴더이고 두 번째 항목이 프리셋임을 보여 줍니다.

```
var presetItemArray=fl.presetPanel.items;
fl.trace(presetItemArray[0].isFolder);
fl.trace(presetItemArray[1].isFolder);
```

presetItem.level

지원 버전

Flash CS4 Professional

사용법

```
presetItem.level
```

설명

읽기 전용 속성, [모션 프리셋] 패널 폴더 구조에서 항목 레벨을 지정하는 정수입니다. 기본 폴더 및 사용자 정의 프리셋 폴더의 레벨은 0입니다.

예제

다음 예제에서는 [모션 프리셋] 패널의 첫 번째 항목이 레벨 0이고 두 번째 항목이 레벨 1임을 보여 줍니다.

```
var presetItemArray=fl.presetPanel.items;
fl.trace(presetItemArray[0].level);
fl.trace(presetItemArray[1].level);
```

presetItem.name

지원 버전

Flash CS4 Professional

사용법

presetItem.name

설명

읽기 전용 속성, 프리셋 또는 폴더의 이름을 나타내는 문자열입니다(경로 정보는 포함하지 않음).

예제

[presetItem.path](#)를 참조하십시오.

presetItem.open

지원 버전

Flash CS4 Professional

사용법

presetItem.open

설명

읽기 전용 속성, [모션 프리셋] 패널의 폴더가 현재 확장되어 있는지(true) 그렇지 않은지(false)를 지정합니다.

항목이 폴더가 아니면 이 속성은 true입니다. 항목이 폴더 또는 프리셋인지 확인하려면 [presetItem.isFolder](#)를 사용합니다.

예제

다음 예제에서는 [모션 프리셋] 패널의 폴더가 확장되어 있는지 축소되어 있는지에 대한 정보를 표시합니다.

```
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

지원 버전

Flash CS4 Professional

사용법

presetItem.path

설명

읽기 전용 속성, [모션 프리셋] 패널 폴더 트리에 있는 항목의 경로와 항목 이름을 나타내는 문자열입니다.

예제

다음 예제에서는 presetItem.name 값과 presetItem.path 값 간의 차이를 보여 줍니다.

```
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("");
}
```

34장: presetPanel 객체

지원 버전

Flash CS4 Professional

설명

presetPanel 객체는 [모션 프리셋] 패널([윈도우] > [모션 프리셋])을 나타냅니다. flash 객체(fl.presetPanel)의 속성입니다.

메서드 요약

presetPanel 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
presetPanel.addItem()	현재 스테이지에서 단일 모션 트윈이 선택되어 있는 경우 해당 모션을 [모션 프리셋] 패널에 추가합니다.
presetPanel.applyPreset()	스테이지의 현재 선택된 항목에 대해 지정되어 있거나 현재 선택된 프리셋을 적용합니다.
presetPanel.deleteFolder()	[모션 프리셋] 패널의 폴더 트리에서 지정된 폴더와 해당 하위 폴더를 삭제합니다.
presetPanel.deleteItem()	[모션 프리셋] 패널에서 지정된 프리셋을 삭제합니다.
presetPanel.expandFolder()	[모션 프리셋] 패널에서 현재 선택한 폴더를 확장하거나 축소합니다.
presetPanel.exportItem()	현재 선택되어 있거나 지정된 프리셋을 XML 파일로 내보냅니다.
presetPanel.findItemIndex()	[모션 프리셋] 패널 항목의 인덱스 위치를 나타내는 정수를 반환합니다.
presetPanel.getSelectedItems()	[모션 프리셋] 패널에서 현재 선택된 항목에 해당하는 presetItem 객체의 배열을 반환합니다.
presetPanel.importItem()	지정된 XML 파일의 프리셋을 [모션 프리셋] 패널에 추가합니다.
presetPanel.moveToFolder()	지정된 항목을 지정된 폴더로 이동합니다.
presetPanel.newFolder()	[모션 프리셋] 패널의 폴더 트리에 폴더를 만듭니다.
presetPanel.renameItem()	현재 선택한 프리셋 또는 폴더의 이름을 지정된 이름으로 변경합니다.
presetPanel.selectItem()	[모션 프리셋] 패널의 항목을 선택하거나 선택 취소합니다.

속성 요약

presetPanel 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
presetPanel.items	[모션 프리셋] 패널의 presetItem 객체 배열입니다.

presetPanel.addItem()

지원 버전

Flash CS4 Professional

사용법

```
fl.presetPanel.addItem( [namePath] );
```

매개 변수

namePath [모션 프리셋] 패널에 추가할 항목의 경로 및 이름을 지정하는 문자열. 이 매개 변수는 선택 사항입니다.

반환값

항목이 성공적으로 추가되면 **true**를 반환하고 그렇지 않으면 **false**를 반환하는 부울 값

설명

메서드, 현재 스테이지에서 단일 모션 트윈이 선택되어 있는 경우 해당 모션을 지정된 폴더의 [모션 프리셋] 패널에 지정된 이름으로 추가합니다. **namePath**에 지정되어 있는 경로가 패널에 있어야 합니다.

namePath에 상응하는 프리셋이 있으면 이 메서드는 아무런 영향을 주지 않으며 **false**를 반환합니다.

namePath에 대한 값을 전달하지 않으면 항목이 "Custom preset n"이라는 이름으로 사용자 정의 프리셋 폴더에 추가됩니다. 여기서 **n**은 이러한 방식으로 항목을 추가할 때마다 증가합니다.

예제

스테이지에서 단일 모션 트윈을 선택한 경우 다음 코드에서는 튀어오르는 공이라는 프리셋을 사용자 정의 프리셋 폴더에 추가합니다.

```
fl.presetPanel.addItem("Custom Presets/Bouncing Ball");
```

참고 사항

[presetPanel.newFolder\(\)](#)

presetPanel.applyPreset()

지원 버전

Flash CS4 Professional

사용법

```
presetPanel.applyPreset( [presetPath] )
```

매개 변수

presetPath 적용할 프리셋의 [모션 프리셋] 패널에 표시되는 전체 경로 및 이름을 지정하는 문자열. [모션 프리셋] 패널에 프리셋이 나타날 때 적용할 프리셋의 전체 경로 및 이름을 지정하는 문자열.

반환값

프리셋이 성공적으로 적용되면 **true**를 반환하고 그렇지 않으면 **false**를 반환하는 부울 값

설명

메서드, 스테이지의 현재 선택된 항목에 대해 지정되어 있거나 현재 선택된 프리셋을 적용합니다. 이 항목은 모션 트윈, 심볼 또는 심볼로 변환할 수 있는 항목이어야 합니다. 항목이 모션 트윈인 경우 사용자에게 확인을 요청하지 않고 현재 모션이 선택한 프리셋으로 바뀝니다.

이 메서드는 다음과 같은 경우 실패합니다.

- **presetPath**로 지정하는 경로가 없는 경우

- `presetPath`에 대한 값을 전달하지 않았으며 프리셋을 선택하지 않은 경우
- `presetPath`에 대한 값을 전달하지 않았으며 프리셋을 여러 개 선택한 경우
- 스테이지에서 선택한 항목이 심볼이 아니며 심볼로 변환할 수 없는 경우

예제

다음 예제에서는 스테이지에서 현재 선택된 항목에 드리블 프리셋을 적용합니다.

```
var result = fl.presetPanel.applyPreset("Custom Presets/Bounces/aDribble");  
fl.trace(result);
```

presetPanel.deleteFolder()

지원 버전

Flash CS4 Professional

사용법

```
presetPanel.deleteFolder( [folderPath] )
```

매개 변수

`folderPath` [모션 프리셋] 패널에서 삭제할 폴더를 지정하는 문자열. 이 매개 변수는 선택 사항입니다.

반환값

폴더가 성공적으로 삭제되면 `true`를 반환하고 그렇지 않으면 `false`를 반환하는 부울 값

설명

메서드, [모션 프리셋] 패널의 폴더 트리에서 지정된 폴더와 해당 하위 폴더를 삭제합니다. 폴더에 포함된 프리셋도 모두 삭제됩니다. 기본 프리셋 폴더에서는 폴더를 삭제할 수 없습니다.

`folderPath`에 대한 값을 전달하지 않으면 현재 선택되어 있는 폴더가 모두 삭제됩니다.

참고: 폴더는 사용자에게 확인을 요청하지 않고 삭제되며, 이 작업은 실행 취소할 수 없습니다.

예제

다음 코드에서는 사용자 정의 프리셋 폴더 아래의 Bouncing이라는 폴더를 삭제합니다. Bouncing의 하위 폴더도 모두 삭제됩니다.

```
fl.presetPanel.deleteFolder("Custom Presets/Bouncing");
```

참고 사항

[presetPanel.deleteItem\(\)](#)

presetPanel.deleteItem()

지원 버전

Flash CS4 Professional

사용법

```
presetPanel.deleteItem( [namePath] )
```

매개 변수

namePath [모션 프리셋] 패널에서 삭제할 항목의 경로 및 이름을 지정하는 문자열. 이 매개 변수는 선택 사항입니다.

반환값

항목이 성공적으로 삭제되면 **true**를 반환하고 그렇지 않으면 **false**를 반환하는 부울 값

설명

메서드, [모션 프리셋] 패널에서 지정된 프리셋을 삭제합니다. **namePath**에 대한 값을 전달하지 않으면 현재 선택되어 있는 프리셋이 모두 삭제됩니다. 기본 프리셋 폴더에서는 항목을 삭제할 수 없습니다.

참고: 항목은 사용자에게 확인을 요청하지 않고 삭제되며, 이 작업은 실행 취소할 수 없습니다.

예제

다음 코드에서는 사용자 정의 프리셋 폴더에서 드리블이라는 프리셋을 삭제합니다.

```
fl.presetPanel.deleteItem("Custom Presets/aDribble");
```

참고 사항

[presetPanel.deleteFolder\(\)](#)

presetPanel.expandFolder()

지원 버전

Flash CS4 Professional

사용법

```
presetPanel.expandFolder( [bExpand [, bRecurse [, folderPath] ] ] )
```

매개 변수

bExpand 폴더를 확장할지(**true**) 또는 축소할지(**false**)를 지정하는 부울 값. 이 매개 변수는 선택적이며 기본값은 **true**입니다.

bRecurse 폴더의 하위 폴더를 확장/축소할지(**true**) 또는 확장/축소하지 않을지(**false**)를 지정하는 부울 값. 이 매개 변수는 선택적이며 기본값은 **false**입니다.

folderPath 확장 또는 축소할 폴더의 경로를 지정하는 문자열. 이 매개 변수는 선택 사항입니다.

반환값

폴더가 성공적으로 확장 또는 축소되면 **true**를 반환하고 그렇지 않으면 **false**를 반환하는 부울 값

설명

메서드, [모션 프리셋] 패널에서 현재 선택한 폴더를 확장하거나 축소합니다. 현재 선택된 폴더가 아닌 다른 폴더를 확장하거나 축소하려면 **folderPath**에 대한 값을 전달합니다.

예제

다음 예제에서는 사용자 정의 프리셋 폴더는 확장하되 해당 하위 폴더는 확장하지 않습니다.

```
fl.presetPanel.expandFolder(true, false, "Custom Presets");
```


다음 예제에서는 사용자 정의 프리셋 폴더와 해당 하위 폴더를 모두 확장합니다.

```
fl.presetPanel.expandFolder(true, true, "Custom Presets");
```

presetPanel.exportItem()

지원 버전

Flash CS4 Professional

사용법

```
presetPanel.exportItem(fileURI [, namePath] )
```

매개 변수

fileURI 내보낸 파일의 경로와 파일 이름(선택 사항)을 지정하는 문자열(**file:/// URI**로 표시됨). 자세한 내용은 아래의 "설명" 부분을 참조하십시오.

namePath [모션 프리셋] 패널에서 선택할 항목의 경로 및 이름을 지정하는 문자열. 이 매개 변수는 선택 사항입니다.

반환값

프리셋이 성공적으로 내보내지면 **true**를 반환하고 그렇지 않으면 **false**를 반환하는 부울 값

설명

메서드, 현재 선택되어 있거나 지정된 프리셋을 **XML** 파일로 내보냅니다. 프리셋만 내보낼 수 있으며 폴더를 내보내려고 하면 메서드가 실패합니다. 또한 디스크의 파일을 덮어쓰려고 해도 메서드가 실패합니다.

fileURI의 일부분으로 파일 이름을 지정하지 않으면, 즉 **fileURI**의 마지막 문자가 슬래시(/)이면 내보낸 파일이 현재 내보내는 중인 프리셋과 같은 이름으로 저장됩니다. **namePath**에 대한 값을 지정하지 않으면 현재 선택되어 있는 프리셋을 내보냅니다. 다음 예제를 참조하십시오.

예제

다음 예제에서는 이 메서드로 여러 매개 변수를 전달하면 만들어지는 파일을 보여 주며, 지정된 파일이 성공적으로 만들어졌는지를 알려 줍니다. 이 예제를 실행하기 전에 기본 프리셋 폴더에서 왼쪽으로 들어오기 프리셋을 선택하고 디스크에 내 프리셋 폴더를 만듭니다.

```
//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");
```

참고 사항

[presetPanel.importItem\(\)](#)

presetPanel.findItemIndex()

지원 버전

Flash CS4 Professional

사용법

```
presetPanel.findItemIndex([presetName])
```

매개 변수

presetName 인덱스 값이 반환되는 프리셋 이름을 지정하는 문자열. 이 매개 변수는 선택 사항입니다.

반환값

presetPanel.items 배열에서 지정된 프리셋의 인덱스를 나타내는 정수. **presetName**에 대한 값을 전달하지 않으면 현재 지정되어 있는 프리셋의 인덱스가 반환됩니다. 다음과 같은 경우 이 메서드는 -1을 반환합니다.

- **presetName**에 대한 값을 전달하지 않았으며 프리셋을 선택하지 않은 경우
- **presetName**에 대한 값을 전달하지 않았으며 프리셋을 여러 개 선택한 경우
- 패널의 항목에 해당하지 않는 값을 **presetName**에 전달한 경우

설명

메서드, [모션 프리셋] 패널 항목의 인덱스 위치를 나타내는 정수를 반환합니다.

예제

다음 코드에서는 현재 선택한 프리셋의 인덱스 값과 전체 경로 이름을 표시합니다.

```
// 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()

지원 버전

Flash CS4 Professional

사용법

```
presetPanel.getSelectedItems()
```

매개 변수

없음

반환값

presetItem 객체의 배열입니다.

설명

메서드, [모션 프리셋] 패널에서 현재 선택된 항목에 해당하는 presetItem 객체의 배열을 반환합니다([presetItem 객체 참조](#)). 배열의 각 항목은 폴더 또는 프리셋을 나타냅니다.

예제

다음 코드에서는 [모션 프리셋] 패널에서 현재 선택된 항목의 전체 경로 이름을 표시합니다.

```
var itemArray = fl.presetPanel.getSelectedItems();  
var length = itemArray.length  
for (x=0; x<length; x++) {  
    fl.trace(itemArray[x].path);  
}
```

참고 사항

[presetPanel.items](#)

presetPanel.importItem()

지원 버전

Flash CS4 Professional

사용법

```
presetPanel.importItem(fileURI [,namePath ])
```

매개 변수

fileURI [모션 프리셋] 패널에서 프리셋으로 가져올 XML 파일을 지정하는 문자열(file:/// URI로 표시됨)

namePath 가져온 파일을 저장할 폴더와 해당 파일의 이름을 지정하는 문자열. 이 매개 변수는 선택 사항입니다.

반환값

파일을 성공적으로 가져오면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값

설명

메서드, 지정된 XML 파일의 프리셋을 [모션 프리셋] 패널에 추가합니다. **namePath**에 지정되어 있는 경로가 패널에 있어야 합니다.

가져올 수 있는 XML 파일을 만들려면 [presetPanel.exportItem\(\)](#)을 사용합니다.

namePath에 대한 값을 전달하지 않으면 가져온 프리셋은 사용자 정의 프리셋 폴더에 저장되며 가져온 파일과 같은 이름이 지정됩니다(XML 확장명은 제외됨).

예제

다음 예제에서는 사용자 정의 프리셋/Pulse 폴더로 프리셋을 가져와 이름을 fastPulse로 지정합니다.

```
fl.presetPanel.importItem("file:///C:/My Presets/thePulsePreset.xml", "Custom Presets/Pulse/fastPulse");
```

참고 사항

[presetPanel.exportItem\(\)](#)

presetPanel.items

지원 버전

Flash CS4 Professional

사용법

```
presetPanel.items
```

설명

속성, [모션 프리셋] 패널의 `presetItem` 객체 배열입니다([presetItem 객체 참조](#)). 배열의 각 항목은 폴더 또는 프리셋을 나타냅니다.

예제

다음 코드에서는 [모션 프리셋] 패널 항목의 전체 경로 이름을 표시합니다.

```
var itemArray = fl.presetPanel.items;
var length = itemArray.length
for (x=0; x<length; x++) {
    fl.trace(itemArray[x].path);
}
```

참고 사항

[presetPanel.getSelectedItems\(\)](#)

presetPanel.moveToFolder()

지원 버전

Flash CS4 Professional

사용법

```
presetPanel.moveToFolder(folderPath [, namePath] )
```

매개 변수

folderPath 항목이 이동되는 [모션 프리셋] 패널 폴더의 경로를 지정하는 문자열

namePath 이동할 항목의 경로 및 이름을 지정하는 문자열. 이 매개 변수는 선택 사항입니다.

반환값

항목이 성공적으로 이동되면 `true`를 반환하고 그렇지 않으면 `false`를 반환하는 부울 값

설명

메서드, 지정된 항목을 지정된 폴더로 이동합니다.

folderPath에 빈 문자열("")을 전달하면 항목이 사용자 정의 프리셋 폴더로 이동됩니다. **namePath**에 대한 값을 전달하지 않으면 현재 선택되어 있는 항목이 이동됩니다.

기본 프리셋 폴더의 항목을 이동하거나 해당 폴더로 항목을 이동할 수는 없습니다.

예제

다음 예제에서는 현재 선택된 항목이 사용자 정의 프리셋/Bouncing 폴더로 이동되며 빠르게 튀기 프리셋이 같은 폴더로 이동됩니다.

```
fl.presetPanel.moveToFolder("Custom Presets/Bouncing");
fl.presetPanel.moveToFolder("Custom Presets/Bouncing" , "Custom Presets/Fast Bounce");
```

presetPanel.newFolder()

지원 버전

Flash CS4 Professional

사용법

```
presetPanel.newFolder( [folderPath] )
```

매개 변수

folderPath [모션 프리셋] 패널에 새 폴더를 추가할 위치와 새 폴더의 이름을 지정하는 문자열. 이 매개 변수는 선택 사항입니다.

반환값

폴더가 성공적으로 추가되면 **true**를 반환하고 그렇지 않으면 **false**를 반환하는 부울 값

설명

메서드, [모션 프리셋] 패널의 폴더 트리에 폴더를 만듭니다. 이 메서드를 사용하는 경우 새 폴더 레벨을 하나만 만들 수 있습니다. 즉, **folderPath**에 "사용자 정의 프리셋/My First Folder/My Second Folder"를 전달하는 경우 폴더 트리에 "사용자 정의 프리셋/My First Folder"가 있어야 합니다.

folderPath에 대한 값을 전달하지 않으면 "Untitled folder n"이라는 폴더가 "사용자 정의 프리셋" 아래 첫 번째 레벨에 만들어 집니다. 여기서 **n**은 이러한 방식으로 폴더를 추가할 때마다 증가합니다.

참고: 기본 프리셋 폴더에는 폴더를 추가할 수 없습니다.

예제

다음 예제에서는 사용자 정의 프리셋 폴더 아래에 Bouncing이라는 폴더를 추가합니다.

```
fl.presetPanel.newFolder("Custom Presets/Bouncing");
```

참고 사항

[presetPanel.addItem\(\)](#)

presetPanel.renameItem()

지원 버전

Flash CS4 Professional

사용법

```
presetPanel.renameItem(newName)
```

매개 변수

newName 프리셋 또는 폴더의 새 이름을 지정하는 문자열

반환값

프리셋 또는 폴더의 이름이 성공적으로 변경되면 **true**를 반환하고 그렇지 않으면 **false**를 반환하는 부울 값

설명

메서드, 현재 선택한 프리셋 또는 폴더의 이름을 지정된 이름으로 변경합니다. 이 메서드는 사용자 정의 프리셋 폴더에서 프리셋 또는 폴더를 하나만 선택한 경우에만 성공합니다. 이 메서드는 다음과 같은 경우 실패합니다.

- 항목을 선택하지 않은 경우
- 여러 항목을 선택한 경우
- 선택한 항목이 기본 프리셋 폴더에 있는 경우
- 이름이 **newName**인 항목이 선택한 항목과 같은 위치에 있는 경우

예제

다음 예제에서는 사용자 정의 프리셋 폴더에서 현재 선택되어 있는 프리셋의 이름을 점점 빠르게 튀기기로 변경합니다.

```
var renamed = fl.presetPanel.renameItem("Bounce Faster");  
fl.trace(renamed);
```

presetPanel.selectItem()

지원 버전

Flash CS4 Professional

사용법

```
presetPanel.selectItem(namePath [, bReplaceCurrentSelection [, bSelect] ])
```

매개 변수

namePath [모션 프리셋] 패널에서 선택할 항목의 경로 및 이름을 지정하는 문자열

bReplaceCurrentSelection 지정된 항목이 현재 선택 항목을 대체하는지(**true**) 아니면 현재 선택 항목에 추가되는지(**false**)를 지정하는 부울 값. 이 매개 변수는 선택적이며 기본값은 **true**입니다.

bSelect 항목 선택 여부를 **true** 또는 **false**로 지정하는 부울 값 이 매개 변수는 선택적이며 기본값은 **true**입니다. **bSelect**에 **false**를 전달하면 **bReplaceCurrentSelection** 값이 무시됩니다.

반환값

항목이 성공적으로 선택되거나 선택 취소되면 **true**를 반환하고 그렇지 않으면 **false**를 반환하는 부울 값

설명

메서드, [모션 프리셋] 패널의 항목을 선택하거나 선택 취소하고 선택적으로 현재 선택되어 있는 항목을 대체합니다.

예제

다음 코드에서는 [모션 프리셋] 패널에서 현재 선택된 프리셋(있는 경우)에 앞으로 이동 후 오른쪽에서 흐리기 프리셋을 추가합니다.

```
fl.presetPanel.selectItem("Default Presets/fly-in-blur-right", false);
```

35장: Rectangle 객체

상속 [Element 객체](#) > [Shape 객체](#) > Rectangle 객체

지원 버전

Flash CS3 Professional

설명

Rectangle 객체는 [프리미티브 사각형 도구]를 사용하여 그려지는 모양입니다. 항목이 Rectangle 객체인지 확인하려면 [shape.isRectangleObject](#)를 사용합니다.

속성 요약

Shape 객체 속성 외에도 Rectangle 객체에는 다음과 같은 속성을 사용할 수 있습니다. Rectangle 객체의 속성을 설정하려면 [document.setRectangleObjectProperty\(\)](#)를 사용합니다.

속성	설명
RectangleObject.bottomLeftRadius	읽기 전용, Rectangle 객체의 왼쪽 아래 모서리의 반경을 설정하는 부동 소수점 값
RectangleObject.bottomRightRadius	읽기 전용, Rectangle 객체의 오른쪽 아래 모서리의 반경을 설정하는 부동 소수점 값
RectangleObject.lockFlag	읽기 전용, 사각형의 각 모서리의 반경 값이 서로 다를 수 있는지 여부를 지정하는 부울 값
RectangleObject.topLeftRadius	읽기 전용, 사각형의 모든 모서리의 반경을 설정하거나 Rectangle 객체의 왼쪽 위 모서리의 반경만을 설정하는 부동 소수점 값
RectangleObject.topRightRadius	읽기 전용, Rectangle 객체의 오른쪽 위 모서리의 반경을 설정하는 부동 소수점 값

RectangleObject.bottomLeftRadius

지원 버전

Flash CS3 Professional

사용법

`RectangleObject.bottomLeftRadius`

설명

읽기 전용 속성, Rectangle 객체의 왼쪽 아래 모서리의 반경을 설정하는 부동 소수점 값입니다. [RectangleObject.lockFlag](#)가 true로 설정된 경우 이 값을 설정해도 아무런 영향을 주지 않습니다.

이 값을 설정하려면 [document.setRectangleObjectProperty\(\)](#)를 사용합니다.

참고 사항

[document.setRectangleObjectProperty\(\)](#), [RectangleObject.bottomRightRadius](#), [RectangleObject.lockFlag](#), [RectangleObject.topLeftRadius](#), [RectangleObject.topRightRadius](#)

RectangleObject.bottomRightRadius

지원 버전

Flash CS3 Professional

사용법

`RectangleObject.bottomRightRadius`

설명

읽기 전용, `Rectangle` 객체의 오른쪽 아래 모서리의 반경을 설정하는 부동 소수점 값입니다. `RectangleObject.lockFlag`가 `true`로 설정된 경우 이 값을 설정해도 아무런 영향을 주지 않습니다.

이 값을 설정하려면 `document.setRectangleObjectProperty()`를 사용합니다.

참고 사항

[document.setRectangleObjectProperty\(\)](#), [RectangleObject.bottomLeftRadius](#), [RectangleObject.lockFlag](#), [RectangleObject.topLeftRadius](#), [RectangleObject.topRightRadius](#)

RectangleObject.lockFlag

지원 버전

Flash CS3 Professional

사용법

`RectangleObject.lockFlag`

설명

읽기 전용 속성, 사각형의 각 모서리의 반경 값이 서로 다를 수 있는지 여부를 지정하는 부울 값입니다. 이 값이 `true`로 설정된 경우 모든 모서리의 반경은 `RectangleObject.topLeftRadius`에 지정된 값으로 설정됩니다. `false`로 설정된 경우 각 모서리의 반경을 개별적으로 설정할 수 있습니다.

이 값을 설정하려면 `document.setRectangleObjectProperty()`를 사용합니다.

참고 사항

[document.setRectangleObjectProperty\(\)](#), [RectangleObject.bottomLeftRadius](#), [RectangleObject.bottomRightRadius](#), [RectangleObject.topLeftRadius](#), [RectangleObject.topRightRadius](#)

RectangleObject.topLeftRadius

지원 버전

Flash CS3 Professional

사용법

`RectangleObject.topLeftRadius`

설명

읽기 전용 속성, 사각형의 모든 모서리의 반경을 설정하거나(`RectangleObject.lockFlag`가 `true`로 설정된 경우) 왼쪽 위 모서리의 반경만을 설정하는(`RectangleObject.lockFlag`가 `false`로 설정된 경우) 부동 소수점 값입니다.

이 값을 설정하려면 `document.setRectangleObjectProperty()`를 사용합니다.

참고 사항

`document.setRectangleObjectProperty()`, `RectangleObject.bottomLeftRadius`, `RectangleObject.bottomRightRadius`, `RectangleObject.lockFlag`, `RectangleObject.topRightRadius`

RectangleObject.topRightRadius

지원 버전

Flash CS3 Professional

사용법

`RectangleObject.topRightRadius`

설명

읽기 전용 속성, `Rectangle` 객체의 오른쪽 위 모서리의 반경을 설정하는 부동 소수점 값입니다. `RectangleObject.lockFlag`가 `true`로 설정된 경우 이 값을 설정해도 아무런 영향을 주지 않습니다.

이 값을 설정하려면 `document.setRectangleObjectProperty()`를 사용합니다.

참고 사항

`document.setRectangleObjectProperty()`, `RectangleObject.bottomLeftRadius`, `RectangleObject.bottomRightRadius`, `RectangleObject.lockFlag`, `RectangleObject.topLeftRadius`

36장: Shape 객체

상속 [Element 객체](#) > Shape 객체

지원 버전

Flash MX 2004

설명

Shape 객체는 Element 객체의 하위 클래스입니다. Shape 객체는 스테이지에서 기하 도형을 만들거나 조작할 때 드로잉 API 보다 더 정확한 컨트롤을 제공합니다. 이 컨트롤을 사용하면 스크립트를 통해 유용한 효과와 기타 드로잉 명령을 만들 수 있습니다([Element 객체](#) 참조).

모양과 모양에 포함된 모든 하위 부분을 변경하는 모든 Shape 메서드와 속성은 함수에 대한 [shape.beginEdit\(\)](#) 및 [shape.endEdit\(\)](#) 호출 사이에 정확하게 배치되어야 합니다.

메서드 요약

Element 객체 메서드 외에도 Shape 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
shape.getCubicSegmentPoints()	3차원 곡선을 정의하는 점 배열을 반환합니다.
shape.beginEdit()	편집 세션의 시작을 정의합니다.
shape.deleteEdge()	지정된 가장자리를 삭제합니다.
shape.endEdit()	모양 편집 세션의 끝을 정의합니다.

속성 요약

Element 객체 속성 외에도 Shape 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
shape.contours	읽기 전용, 모양에 대한 Contour 객체의 배열입니다(Contour 객체 참조).
shape.edges	읽기 전용, Edge 객체의 배열입니다(Edge 객체 참조).
shape.isDrawingObject	읽기 전용, true인 경우 모양이 Drawing 객체입니다.
shape.isGroup	읽기 전용, true인 경우 모양이 그룹입니다.
shape.isOvalObject	읽기 전용, true인 경우 모양은 [타원형 도구]를 사용하여 만들어진 프리미티브 Oval 객체입니다.
shape.isRectangleObject	읽기 전용, true인 경우 모양은 [사각형 도구]를 사용하여 만들어진 프리미티브 Rectangle 객체입니다.
shape.members	현재 선택한 그룹의 객체 배열입니다.
shape.numCubicSegments	읽기 전용, 모양의 3차원 선분 수입니다.
shape.vertices	읽기 전용, Vertex 객체의 배열입니다(Vertex 객체 참조).

shape.beginEdit()

지원 버전
Flash MX 2004

사용법
`shape.beginEdit()`

매개 변수
없음

반환값
없음

설명
메서드, 편집 세션의 시작을 정의합니다. 이 메서드는 Shape 객체와 Shape 객체에 포함된 모든 하위 부분을 변경하는 명령을 실행하기 전에 사용해야 합니다.

예제
다음 예제에서는 현재 선택한 모양을 가져와서 가장자리 배열에 있는 첫 번째 가장자리를 제거합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
shape.beginEdit();  
shape.deleteEdge(0);  
shape.endEdit();
```

shape.contours

지원 버전
Flash MX 2004

사용법
`shape.contours`

설명
읽기 전용 속성, 모양에 대한 Contour 객체의 배열([Contour 객체](#) 참조)입니다.

예제
다음 예제에서는 윤곽선 배열의 첫 번째 윤곽선을 `c` 변수에 저장한 다음 해당 윤곽선의 [HalfEdge 객체](#)를 `he` 변수에 저장합니다.

```
var c = fl.getDocumentDOM().selection[0].contours[0];  
var he = c.getHalfEdge();
```

shape.deleteEdge()

지원 버전
Flash MX 2004

사용법

`shape.deleteEdge(index)`

매개 변수

index `shape.edges` 배열에서 삭제할 가장자리를 지정하는 인덱스(0부터 시작) 이 메서드는 `shape.edges` 배열의 길이를 변경합니다.

반환값

없음

설명

메서드, 지정된 가장자리를 삭제합니다. 이 메서드를 사용하려면 먼저 `shape.beginEdit()`을 호출해야 합니다.

예제

다음 예제에서는 현재 선택한 모양을 가져와서 가장자리 배열에 있는 첫 번째 가장자리를 제거합니다.

```
var shape = fl.getDocumentDOM().selection[0];  
shape.beginEdit();  
shape.deleteEdge(0);  
shape.endEdit();
```

shape.edges

지원 버전

Flash MX 2004

사용법

`shape.edges`

설명

읽기 전용 속성, Edge 객체의 배열([Edge 객체 참조](#))입니다.

shape.endEdit()

지원 버전

Flash MX 2004

사용법

`shape.endEdit()`

매개 변수

없음

반환값

없음

설명

메서드, 모양 편집 세션의 끝을 정의합니다. Shape 객체 또는 Shape 객체에 포함된 모든 하위 부분의 모든 변경 내용은 모양에 적용됩니다. 이 메서드는 Shape 객체 또는 Shape 객체에 포함된 모든 하위 부분을 변경하는 명령을 실행한 후에 사용해야 합니다.

예제

다음 예제에서는 현재 선택한 모양을 가져와서 가장자리 배열에 있는 첫 번째 가장자리를 제거합니다.

```
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit();
shape.deleteEdge(0);
shape.endEdit();
```

shape.getCubicSegmentPoints()

지원 버전

Flash CS4 Professional

사용법

```
shape.getCubicSegmentPoints(cubicSegmentIndex)
```

매개 변수

cubicSegmentIndex 점이 반환되는 3차원 선분을 지정하는 정수

반환값

지정된 cubicSegmentIndex에 해당하는 Edge 객체에 대해 3차원 곡선을 정의하는 점 배열([edge.cubicSegmentIndex](#) 참조)

설명

메서드, 3차원 곡선을 정의하는 점 배열을 반환합니다.

예제

다음 예제에서는 선택 항목 첫 번째 가장자리의 3차원 곡선에 있는 각 점에 대해 x 및 y 값을 표시합니다.

```
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

지원 버전

Flash 8

사용법

```
shape.isDrawingObject
```

설명

읽기 전용 속성, true인 경우 모양이 Drawing 객체입니다.

예제

다음 예제에서는 선택한 첫 번째 객체를 sel 변수에 저장한 다음 `element.elementType` 및 `shape.isDrawingObject` 속성을 사용하여 선택한 항목이 Drawing 객체인지 확인합니다.

```
var sel = fl.getDocumentDOM().selection[0];  
var shapeDrawingObject = (sel.elementType == "shape") && sel.isDrawingObject;  
fl.trace(shapeDrawingObject);
```

참고 사항

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

shape.isGroup

지원 버전

Flash MX 2004

사용법

`shape.isGroup`

설명

읽기 전용 속성, true인 경우 모양이 그룹입니다. 그룹은 텍스트 요소와 심볼 등의 다양한 요소 유형을 포함할 수 있습니다. 그러나 그룹 자체는 모양으로 간주되므로 그룹에 포함된 요소의 유형에 관계없이 `shape.isGroup` 속성을 사용할 수 있습니다.

예제

다음 예제에서는 선택한 첫 번째 객체를 sel 변수에 저장한 다음 `element.elementType` 및 `shape.isGroup` 속성을 사용하여 선택한 항목이 그룹인지 확인합니다.

```
var sel = fl.getDocumentDOM().selection[0];  
var shapeGroup = (sel.elementType == "shape") && sel.isGroup;  
fl.trace(shapeGroup);
```

참고 사항

[shape.isDrawingObject](#)

shape.isOvalObject

지원 버전

Flash CS3 Professional

사용법

`shape.isOvalObject`

설명

읽기 전용 속성, true로 설정된 경우 모양은 [프리미티브 타원형 도구]를 사용하여 만들어진 프리미티브 Oval 객체입니다.

예제

다음 예제에서는 첫 번째 선택한 항목이 프리미티브 Oval 객체인 경우 "true"를 표시하고 그렇지 않으면 "false"를 표시합니다.

```
var sel = fl.getDocumentDOM().selection[0];  
fl.trace(sel.isOvalObject);
```

참고 사항

[shape.isRectangleObject](#)

shape.isRectangleObject

지원 버전

Flash CS3 Professional

사용법

`shape.isRectangleObject`

설명

읽기 전용 속성, true로 설정된 경우 모양은 [프리미티브 사각형 도구]를 사용하여 만들어진 프리미티브 Rectangle 객체입니다.

예제

다음 예제에서는 첫 번째 선택한 항목이 프리미티브 Rectangle 객체인 경우 "true"를 표시하고 그렇지 않으면 "false"를 표시합니다.

```
var sel = fl.getDocumentDOM().selection[0];  
fl.trace(sel.isRectangleObject);
```

참고 사항

[shape.isOvalObject](#)

shape.members

지원 버전

Flash CS4 Professional

사용법

`shape.members`

설명

읽기 전용 속성, 현재 선택한 그룹의 객체 배열입니다. 이 속성은 `shape.isGroup` 값이 true일 경우에만 사용할 수 있습니다. 이 그룹의 원시 모양은 `shape.members` 배열에 포함되지 않습니다.

예를 들어 그룹에 드로잉 객체와 원시 모양이 각각 3개씩 있으면 `shape.members` 배열에는 각 드로잉 객체에 대해 하나씩 3개의 항목이 포함됩니다. 그룹에 원시 모양만 있으면 배열은 비어 있게 됩니다.

예제

다음 코드에서는 현재 선택한 그룹에 포함된 각 드로잉 객체의 3차원 선분 수를 표시합니다.

```
var shapesArray = fl.getDocumentDOM().selection[0].members;  
for (i=0; i<shapesArray.length; i++) {  
    fl.trace(shapesArray[i].numCubicSegments);  
}
```

참고 사항

[shape.isGroup](#)

shape.numCubicSegments

지원 버전

Flash CS4 Professional

사용법

`shape.numCubicSegments`

설명

읽기 전용 속성, 모양의 3차원 선분 수입니다.

예제

다음 코드에서는 정사각형 또는 직사각형 모양을 선택했다고 가정할 때 [출력] 패널에 "4"를 표시합니다.

```
var theShape = fl.getDocumentDOM().selection[0];  
fl.trace(theShape.numCubicSegments);
```

shape.vertices

지원 버전

Flash MX 2004

사용법

`shape.vertices`

설명

읽기 전용 속성, Vertex 객체의 배열([Vertex 객체](#) 참조)입니다.

예제

다음 예제에서는 선택한 첫 번째 객체를 `someShape` 변수에 저장한 다음 해당 객체의 꼭지점 수를 [출력] 패널에 표시합니다.

```
var someShape = fl.getDocumentDOM().selection[0];  
fl.trace("The shape has " + someShape.vertices.length + " vertices.");
```


37장: SoundItem 객체

상속 [Item 객체](#) > [SoundItem 객체](#)

지원 버전
Flash MX 2004

설명

SoundItem 객체는 Item 객체의 하위 클래스입니다. 사운드를 만드는 데 사용되는 라이브러리 항목을 나타냅니다. 자세한 내용은 [frame.soundLibraryItem](#) 및 [Item 객체](#)를 참조하십시오.

메서드 요약

Item 객체 메서드 외에도 SoundItem 객체에는 다음과 같은 메서드가 있습니다.

속성	설명
soundItem.exportToFile()	지정된 항목을 Macintosh의 경우 QuickTime 파일로, Windows의 경우 WAV 또는 QT 파일로 내보냅니다.

속성 요약

Item 객체 속성 외에도 SoundItem 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
soundItem.bitRate	라이브러리에 있는 사운드의 비트율을 지정하는 문자열. MP3 압축 유형에만 사용할 수 있습니다.
soundItem.bits	라이브러리에 있는 ADPCM 압축 유형 사운드의 비트 값을 지정하는 문자열
soundItem.compressionType	라이브러리에 있는 사운드의 압축 유형을 지정하는 문자열
soundItem.convertStereoToMono	MP3 및 Raw 압축 유형에만 사용할 수 있는 부울 값
soundItem.fileLastModifiedDate	읽기 전용, 1970년 1월 1일과 파일을 라이브러리로 가져왔을 때 디스크에 있었던 원본 파일의 수정 날짜 사이에 경과된 시간(초)을 나타내는 16진수를 포함하는 문자열
soundItem.originalCompressionType	읽기 전용, 지정된 항목을 MP3 파일로 가져왔는지를 지정하는 문자열
soundItem.quality	라이브러리에 있는 사운드의 재생 품질을 지정하는 문자열. MP3 압축 유형에만 사용할 수 있습니다.
soundItem.sampleRate	오디오 클립의 샘플 빈도를 지정하는 문자열
soundItem.sourceFileExists	읽기 전용, 라이브러리로 가져온 파일이 아직 가져온 위치에 있는지 여부를 지정하는 부울 값
soundItem.sourceFilesCurrent	읽기 전용, 라이브러리 항목의 파일 수정 날짜가 가져온 파일을 디스크에서 수정한 날짜와 같은지 여부를 지정하는 부울 값
soundItem.sourceFilePath	읽기 전용, 라이브러리로 가져온 파일의 경로와 이름을 나타내는 문자열(file:/// URI로 표시됨)
soundItem.useImportedMP3Quality	부울 값, true인 경우 다른 모든 속성이 무시되고 가져온 MP3 품질이 사용됩니다.

soundItem.bitRate

지원 버전

Flash MX 2004

사용법

```
soundItem.bitRate
```

설명

속성, 라이브러리에 있는 사운드의 비트율을 지정하는 문자열입니다. 이 속성은 MP3 압축 유형에만 사용할 수 있습니다. 사용할 수 있는 값에는 "8 kbps", "16kbps", "20kbps", "24kbps", "32kbps", "48kbps", "56kbps", "64kbps", "80kbps", "112kbps", "128kbps" 및 "160kbps"가 있습니다. 8Kbps 또는 16Kbps로 내보낸 스테레오 사운드는 모노로 변환됩니다. 다른 압축 유형의 속성은 undefined로 지정됩니다.

이 속성에 대한 값을 지정하려면 [soundItem.useImportedMP3Quality](#)를 false로 설정합니다.

예제

다음 예제에서는 라이브러리에 지정되어 있는 항목이 MP3 압축 유형인 경우에 bitRate 값을 [출력] 패널에 표시합니다.

```
alert(fl.getDocumentDOM().library.items[0].bitRate);
```

참고 사항

[soundItem.compressionType](#), [soundItem.convertStereoToMono](#)

soundItem.bits

지원 버전

Flash MX 2004

사용법

```
soundItem.bits
```

설명

속성, 라이브러리에 있는 ADPCM 압축 유형 사운드의 비트 값을 지정하는 문자열입니다. 사용할 수 있는 값에는 "2bit", "3bit", "4bit" 및 "5bit"가 있습니다.

이 속성에 대한 값을 지정하려면 [soundItem.useImportedMP3Quality](#)를 false로 설정합니다.

예제

다음 예제에서는 라이브러리에 현재 선택되어 있는 항목이 ADPCM 압축 유형인 경우에 비트 값을 [출력] 패널에 표시합니다.

```
alert(fl.getDocumentDOM().library.items[0].bits);
```

참고 사항

[soundItem.compressionType](#)

soundItem.compressionType

지원 버전

Flash MX 2004

사용법

```
soundItem.compressionType
```

설명

속성, 라이브러리에 있는 사운드의 압축 유형을 지정하는 문자열입니다. 사용할 수 있는 값에는 "Default", "ADPCM", "MP3", "Raw" 및 "음성"이 있습니다.

이 속성에 대한 값을 지정하려면 [soundItem.useImportedMP3Quality](#)를 false로 설정합니다.

예제

다음 예제에서는 라이브러리 항목의 압축 유형을 Raw로 변경합니다.

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

다음 예제에서는 선택한 라이브러리 항목의 압축 유형을 음성으로 변경합니다.

```
fl.getDocumentDOM().library.getSelectedItems().compressionType = "Speech";
```

참고 사항

[soundItem.originalCompressionType](#)

soundItem.convertStereoToMono

지원 버전

Flash MX 2004

사용법

```
soundItem.convertStereoToMono
```

설명

속성, MP3 및 Raw 압축 유형에만 사용할 수 있는 부울 값입니다. 이 값을 true로 설정하면 스테레오 사운드가 모노로 변환되고 false로 설정하면 스테레오로 유지됩니다. MP3 압축 유형의 경우 [soundItem.bitRate](#)가 20Kbps 미만이면 이 속성이 무시되고 무조건 true로 설정됩니다([soundItem.bitRate](#) 참조).

이 속성에 대한 값을 지정하려면 [soundItem.useImportedMP3Quality](#)를 false로 설정합니다.

예제

다음 예제에서는 MP3 또는 Raw 압축 유형일 경우에 한해서 라이브러리 항목을 모노로 변환합니다.

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

참고 사항

[soundItem.compressionType](#)

soundItem.exportToFile()

지원 버전

Flash CS4 Professional

사용법

```
soundItem.exportToFile(fileURI)
```

매개 변수

fileURI 내보낸 파일의 경로와 이름을 지정하는 문자열(file:/// URI로 표시됨)

반환값

파일을 성공적으로 내보냈으면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값

설명

메서드, 지정된 항목을 Macintosh의 경우 QuickTime 파일로, Windows의 경우 WAV 또는 QT 파일로 내보냅니다. 내보낸 QuickTime 또는 QT 파일에는 오디오만 포함되며 비디오는 포함되지 않습니다. 내보내기 설정은 내보내는 항목을 기준으로 합니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 사운드 항목이라고 가정할 경우 WAV 파일로 내보냅니다.

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

soundItem.fileLastModifiedDate

지원 버전

Flash CS4 Professional

사용법

```
soundItem.fileLastModifiedDate
```

설명

읽기 전용 속성, 1970년 1월 1일과 파일을 라이브러리로 가져왔을 때 디스크에 있었던 원본 파일의 수정 날짜 사이에 경과된 시간(초)을 나타내는 16진수를 포함하는 문자열. 파일이 없으면 이 값은 "00000000"입니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 사운드 항목으로 가정할 경우 위에서 설명한 대로 16진수를 표시합니다.

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

참고 사항

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

soundItem.originalCompressionType

지원 버전

Flash CS4 Professional

사용법

```
soundItem.originalCompressionType
```

설명

읽기 전용 속성, 지정된 항목을 mp3 파일로 가져왔는지를 지정하는 문자열입니다. 이 속성에 사용할 수 있는 값은 "RAW" 및 "MP3"입니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 사운드 항목으로 가정할 경우 해당 파일을 MP3 파일로 라이브러리에 가져온 경우 "MP3"를 표시하고 그렇지 않은 경우 "RAW"를 표시합니다.

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

참고 사항

[soundItem.compressionType](#)

soundItem.quality

지원 버전

Flash MX 2004

사용법

```
soundItem.quality
```

설명

속성, 라이브러리에 있는 사운드의 재생 품질을 지정하는 문자열입니다. 이 속성은 MP3 압축 유형에만 사용할 수 있습니다. 사용할 수 있는 값에는 "Fast", "Medium" 및 "Best"가 있습니다.

이 속성에 대한 값을 지정하려면 [soundItem.useImportedMP3Quality](#)를 false로 설정합니다.

예제

다음 예제에서는 MP3 압축 유형일 경우에 한해서 라이브러리 항목의 재생 품질을 Best로 설정합니다.

```
fl.getDocumentDOM().library.items[0].quality = "Best";
```

참고 사항

[soundItem.compressionType](#)

soundItem.sampleRate

지원 버전

Flash MX 2004

사용법

```
soundItem.sampleRate
```

설명

속성, 오디오 클립의 샘플 빈도를 지정하는 문자열입니다. 이 속성은 ADPCM, Raw 및 음성 압축 유형에만 사용할 수 있습니다. 사용할 수 있는 값에는 "5kHz", "11kHz", "22kHz" 및 "44kHz"가 있습니다.

이 속성에 대한 값을 지정하려면 [soundItem.useImportedMP3Quality](#)를 false로 설정합니다.

예제

다음 예제에서는 ADPCM, Raw 또는 음성 압축 유형일 경우에 한해서 라이브러리 항목의 샘플 빈도를 5kHz로 설정합니다.

```
fl.getDocumentDOM().library.items[0].sampleRate = "5 kHz";
```

참고 사항

[soundItem.compressionType](#)

soundItem.sourceFileExists

지원 버전

Flash CS4 Professional

사용법

```
soundItem.sourceFileExists
```

설명

읽기 전용 속성, 라이브러리로 가져온 파일이 아직 가져온 위치에 있으면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 사운드 항목이라고 가정할 경우 라이브러리로 가져온 파일이 아직 있으면 "true"를 표시합니다.

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

참고 사항

[soundItem.sourceFileIsCurrent](#), [soundItem.sourceFilePath](#)

soundItem.sourceFileIsCurrent

지원 버전

Flash CS4 Professional

사용법

`soundItem.sourceFileIsCurrent`

설명

읽기 전용 속성, 라이브러리 항목의 파일 수정 날짜가 가져온 파일을 디스크에서 수정한 날짜와 같으면 `true`를 반환하고 그렇지 않으면 `false`를 반환하는 부울 값입니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 사운드 항목으로 간주하여 가져온 파일을 가져온 이후 디스크에서 수정하지 않은 경우 `"true"`를 표시합니다.

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

참고 사항

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

soundItem.sourceFilePath

지원 버전

Flash CS4 Professional

사용법

`soundItem.sourceFilePath`

설명

읽기 전용 속성, 라이브러리로 가져온 파일의 경로와 이름을 나타내는 문자열(`file:///` URI로 표시됨)

예제

다음 예제에서는 유형이 `"sound"`인 라이브러리에 있는 모든 항목의 이름과 소스 파일 경로를 표시합니다.

```
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);  
  }  
}
```

참고 사항

[soundItem.sourceFileExists](#)

soundItem.useImportedMP3Quality

지원 버전

Flash MX 2004

사용법

`soundItem.useImportedMP3Quality`

설명

속성, 부울 값입니다. true인 경우 다른 모든 속성이 무시되고 가져온 MP3 품질이 사용됩니다.

예제

다음 예제에서는 라이브러리 항목을 가져온 MP3 품질을 사용하도록 설정합니다.

```
fl.getDocumentDOM().library.items[0].useImportedMP3Quality = true;
```

참고 사항

[soundItem.compressionType](#)

38장: Stroke 객체

지원 버전

Flash MX 2004

설명

Stroke 객체에는 사용자 정의 설정을 포함하여 획에 대한 모든 설정이 포함됩니다. 이 객체는 속성 관리자에 포함되어 있는 정보를 나타냅니다. Stroke 객체를 `document.setCustomStroke()` 메서드와 함께 사용하면 [도구] 패널, 속성 관리자 및 현재 선택한 항목에 대한 획 설정을 변경할 수 있습니다. 또한 `document.getCustomStroke()` 메서드를 사용하면 [도구] 패널 및 속성 관리자 또는 현재 선택한 항목의 획 설정을 가져올 수도 있습니다.

이 객체에는 항상 `style`, `thickness`, `color` 및 `breakAtCorners`의 네 가지 속성이 설정되어 있습니다. (Flash CS3에서는 `breakAtCorners` 속성 대신 `stroke.joinType`이 사용됩니다.) 다른 속성은 `stroke.style` 속성의 값에 따라 설정될 수 있습니다.

속성 요약

Stroke 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>stroke.breakAtCorners</code>	부울 값, [획 스타일 사용자 정의] 대화 상자의 [선명한 모서리] 설정과 같습니다.
<code>stroke.capType</code>	획 끝의 유형을 지정하는 문자열
<code>stroke.color</code>	획 색상을 나타내는 문자열, 16진수 값 또는 정수
<code>stroke.curve</code>	획의 해치 유형을 지정하는 문자열
<code>stroke.dash1</code>	파선의 실선 부분 길이를 지정하는 정수
<code>stroke.dash2</code>	파선의 공백 부분 길이를 지정하는 정수
<code>stroke.density</code>	점모 선의 밀도를 지정하는 문자열
<code>stroke.dotSize</code>	점모 선의 도트 크기를 지정하는 문자열
<code>stroke.dotSpace</code>	점선의 점 간격을 지정하는 정수
<code>stroke.hatchThickness</code>	해치 선의 두께를 지정하는 문자열
<code>stroke.jiggle</code>	해치 선의 움직임 속성을 지정하는 문자열
<code>stroke.joinType</code>	획의 연결 유형을 지정하는 문자열
<code>stroke.length</code>	해치 선의 길이를 지정하는 문자열
<code>stroke.miterLimit</code>	이음의 끝을 선분으로 자를 각도를 지정하는 부동 소수점 값
<code>stroke.pattern</code>	불규칙 선의 패턴을 지정하는 문자열
<code>stroke.rotate</code>	해치 선의 회전을 지정하는 문자열
<code>stroke.scaleType</code>	획에 적용할 크기 조절 유형을 지정하는 문자열
<code>stroke.shapeFill</code>	획의 채우기 설정을 나타내는 Fill 객체
<code>stroke.space</code>	해치 선의 간격을 지정하는 문자열
<code>stroke.strokeHinting</code>	획에 획 힌트를 설정할 것인지를 지정하는 부울 값
<code>stroke.style</code>	획 스타일을 설명하는 문자열

속성	설명
stroke.thickness	획 크기를 지정하는 정수
stroke.variation	점묘 선의 변형을 지정하는 문자열
stroke.waveHeight	불규칙 선의 웨이브 높이를 지정하는 문자열
stroke.waveLength	불규칙 선의 웨이브 길이를 지정하는 문자열

stroke.breakAtCorners

지원 버전

Flash MX 2004 Flash CS3에서는 이 속성 대신 [stroke.joinType](#)이 사용되고 있습니다.

사용법

```
stroke.breakAtCorners
```

설명

속성, 부울 값입니다. 이 속성은 [획 스타일 사용자 정의] 대화 상자의 [선명한 모서리] 설정과 같습니다.

예제

다음 예제에서는 `breakAtCorners` 속성을 `true`로 설정합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.breakAtCorners = true;  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.capType

지원 버전

Flash 8

사용법

```
stroke.capType
```

설명

속성, 획 끝의 유형을 지정하는 문자열입니다. 사용할 수 있는 값에는 "none", "round" 및 "square"가 있습니다.

예제

다음 예제에서는 획 끝의 유형을 `round`로 설정합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.capType = "round";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.color

지원 버전

Flash MX 2004 Flash 8 이상에서는 이 속성 대신 `stroke.shapeFill.color`가 사용되고 있습니다.

사용법

`stroke.color`

설명

속성, 형식이 다음 중 하나인 획의 색상입니다.

- "#RRGGBB" 또는 "#RRGGBBAA" 형식의 문자열
- 0xRRGGBB 형식의 16진수
- 16진수 숫자와 같은 십진수를 나타내는 정수

예제

다음 예제에서는 획 색상을 설정합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.color = "#000000";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

참고 사항

[stroke.shapeFill](#)

stroke.curve

지원 버전

Flash MX 2004

사용법

`stroke.curve`

설명

속성, 획의 해치 유형을 지정하는 문자열입니다. 이 속성은 `stroke.style` 속성이 "hatched"인 경우에만 설정할 수 있습니다 ([stroke.style](#) 참조). 사용할 수 있는 값에는 "straight", "slight curve", "medium curve" 및 "very curved"가 있습니다.

예제

다음 예제에서는 hatched 스타일 획의 `curve` 속성을 다른 속성과 함께 설정합니다.

```
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

지원 버전

Flash MX 2004

사용법

`stroke.dash1`

설명

속성, 파선의 실선 부분 길이를 지정하는 정수입니다. 이 속성은 `stroke.style` 속성이 `dashed`로 설정되어 있는 경우에만 사용할 수 있습니다([stroke.style](#) 참조).

예제

다음 예제에서는 `dashed` 스타일 획의 `dash1` 및 `dash2` 속성을 설정합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "dashed";  
myStroke.dash1 = 1;  
myStroke.dash2 = 2;  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.dash2

지원 버전

Flash MX 2004

사용법

`stroke.dash2`

설명

속성, 파선의 공백 부분 길이를 지정하는 정수입니다. 이 속성은 `stroke.style` 속성이 `dashed`로 설정되어 있는 경우에만 사용할 수 있습니다([stroke.style](#) 참조).

예제

[stroke.dash1](#)을 참조하십시오.

stroke.density

지원 버전

Flash MX 2004

사용법

`stroke.density`

설명

속성, 점묘 선의 밀도를 지정하는 문자열입니다. 이 속성은 `stroke.style` 속성이 `stipple`로 설정되어 있는 경우에만 사용할 수 있습니다([stroke.style](#) 참조). 사용할 수 있는 값에는 "very dense", "dense", "sparse" 및 "very sparse"가 있습니다.

예제

다음 예제에서는 `stipple` 스타일 획의 밀도 속성을 `sparse`로 설정합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "stipple";
myStroke.dotSpace= 3;
myStroke.variation = "random sizes";
myStroke.density = "sparse";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.dotSize

지원 버전

Flash MX 2004

사용법

`stroke.dotSize`

설명

속성, 점묘 선의 도트 크기를 지정하는 문자열입니다. 이 속성은 `stroke.style` 속성이 `stipple`로 설정되어 있는 경우에만 사용할 수 있습니다([stroke.style](#) 참조). 사용할 수 있는 값에는 "tiny", "small", "medium" 및 "large"가 있습니다.

다음 예제에서는 `stipple` 스타일 획의 `dotSize` 속성을 `tiny`로 설정합니다.

```
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

지원 버전

Flash MX 2004

사용법

`stroke.dotSpace`

설명

속성, 점선의 점 간격을 지정하는 정수입니다. 이 속성은 `stroke.style` 속성이 `dotted`로 설정되어 있는 경우에만 사용할 수 있습니다. [stroke.style](#)을 참조하십시오.

예제

다음 예제에서는 `dotted` 스타일 획의 `dotSpace` 속성을 3으로 설정합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "dotted";  
myStroke.dotSpace= 3;  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.hatchThickness

지원 버전

Flash MX 2004

사용법

stroke.hatchThickness

설명

속성, 해치 선의 두께를 지정하는 문자열입니다. 이 속성은 stroke.style 속성이 hatched로 설정되어 있는 경우에만 사용할 수 있습니다([stroke.style](#) 참조). 사용할 수 있는 값에는 "hairline", "thin", "medium" 및 "thick"이 있습니다.

예제

다음 예제에서는 hatched 스타일 획의 hatchThickness 속성을 thin으로 설정합니다.

```
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

지원 버전

Flash MX 2004

사용법

stroke.jiggle

설명

속성, 해치 선의 움직임 속성을 지정하는 문자열입니다. 이 속성은 stroke.style 속성이 hatched로 설정되어 있는 경우에만 사용할 수 있습니다([stroke.style](#) 참조). 사용할 수 있는 값에는 "none", "bounce", "loose" 및 "wild"가 있습니다.

예제

다음 예제에서는 hatched 스타일 획의 jiggle 속성을 wild로 설정합니다.

```
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

지원 버전
Flash 8

사용법

stroke.joinType

설명

속성, 획의 연결 유형을 지정하는 문자열입니다. 사용할 수 있는 값에는 "miter", "round" 및 "bevel"이 있습니다.

참고 사항

[stroke.capType](#)

stroke.length

지원 버전
Flash MX 2004

사용법

stroke.length

설명

속성, 해치 선의 길이를 지정하는 문자열입니다. 이 속성은 stroke.style 속성이 hatched로 설정되어 있는 경우에만 사용할 수 있습니다([stroke.style](#) 참조). 허용 가능한 값은 "equal", "slight variation", "medium variation" 및 "random"(값 "random"은 "medium variation"에 매핑됨)입니다.

예제

다음 예제에서는 hatched 스타일 획의 length 속성을 slight로 설정합니다.

```
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

지원 버전

Flash 8

사용법

```
stroke.miterLimit
```

설명

속성, 이음의 끝을 선분으로 자를 각도를 지정하는 부동 소수점 값입니다. 즉, 이음 각도가 miterLimit의 값보다 큰 경우에만 이음이 잘립니다.

예제

다음 예제에서는 획의 이음 제한 설정을 3으로 변경합니다. 이음 각도가 3보다 크면 이음이 잘립니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.miterLimit = 3;  
var myStroke = fl.getDocumentDOM().setCustomStroke();
```

stroke.pattern

지원 버전

Flash MX 2004

사용법

```
stroke.pattern
```

설명

속성, 불규칙 선의 패턴을 지정하는 문자열입니다. 이 속성은 stroke.style 속성이 ragged로 설정되어 있는 경우에만 사용할 수 있습니다(stroke.style 참조). 사용할 수 있는 값에는 "solid", "simple", "random", "dotted", "random dotted", "triple dotted" 및 "random triple dotted"가 있습니다.

예제

다음 예제에서는 ragged 스타일 획의 pattern 속성을 random으로 설정합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "ragged";  
myStroke.pattern = "random";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.rotate

지원 버전

Flash MX 2004

사용법

```
stroke.rotate
```


설명

속성, 해치 선의 회전을 지정하는 문자열입니다. 이 속성은 `stroke.style` 속성이 `hatched`로 설정되어 있는 경우에만 사용할 수 있습니다([stroke.style](#) 참조). 사용할 수 있는 값에는 "none", "slight", "medium" 및 "free"가 있습니다.

예제

다음 예제에서는 `hatched` 스타일 획의 `rotate` 속성을 `free`로 설정합니다.

```
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

지원 버전

Flash 8

사용법

```
stroke.scaleType
```

설명

속성, 획에 적용할 크기 조절 유형을 지정하는 문자열입니다. 사용할 수 있는 값에는 "normal", "horizontal", "vertical" 및 "none"이 있습니다.

예제

다음 예제에서는 획의 크기 조절 유형을 `horizontal`로 설정합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.scaleType = "horizontal";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.shapeFill

지원 버전

Flash 8

사용법

```
stroke.shapeFill
```

설명

속성, 획의 채우기 설정을 나타내는 [Fill](#) 객체입니다.

예제

다음 예제에서는 채우기 설정을 지정한 다음 이 설정을 획에 적용합니다.

```
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

지원 버전
Flash MX 2004

사용법
stroke.space

설명
속성, 헤치 선의 간격을 지정하는 문자열입니다. 이 속성은 stroke.style 속성이 hatched로 설정되어 있는 경우에만 사용할 수 있습니다([stroke.style](#) 참조). 사용할 수 있는 값에는 "very close", "close", "distant" 및 "very distant"가 있습니다.

예제
다음 예제에서는 hatched 스타일 획의 space 속성을 close로 설정합니다.

```
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

지원 버전
Flash 8

사용법
stroke.strokeHinting

설명
속성, 획에 획 힌트를 설정할 것인지를 지정하는 부울 값입니다.

예제
다음 예제에서는 획에 대한 획 힌트를 활성화합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.strokeHinting = true;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.style

지원 버전

Flash MX 2004

사용법

`stroke.style`

설명

속성, 획 스타일을 설명하는 문자열입니다. 사용할 수 있는 값에는 "noStroke", "solid", "dashed", "dotted", "ragged", "stipple" 및 "hatched"가 있습니다. 이러한 값 중 일부는 다음 목록과 같이 Stroke 객체의 추가 속성을 설정해야 사용할 수 있습니다.

- 값이 "solid" 또는 "noStroke"이면 설정할 추가 속성이 없습니다.
- 값이 "dashed"이면 dash1과 dash2라는 두 개의 추가 속성을 설정해야 합니다.
- 값이 "dotted"이면 dotSpace라는 하나의 추가 속성을 설정해야 합니다.
- "ragged"이면 pattern, waveHeight 및 waveLength라는 세 개의 추가 속성을 설정해야 합니다.
- 값이 "stipple"이면 dotSize, variation 및 density라는 세 개의 추가 속성을 설정해야 합니다.
- 값이 "hatched"이면 hatchThickness, space, jiggle, rotate, curve 및 length라는 여섯 개의 추가 속성을 설정해야 합니다.

예제

다음 예제에서는 획 스타일을 ragged로 설정합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "ragged";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.thickness

지원 버전

Flash MX 2004

사용법

`stroke.thickness`

설명

속성, 획 크기를 지정하는 정수입니다.

예제

다음 예제에서는 획의 thickness 속성 값을 2로 설정합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.thickness = 2;  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.variation

지원 버전

Flash MX 2004

사용법

stroke.variation

설명

속성, 점묘 선의 변형을 지정하는 문자열입니다. 이 속성은 stroke.style 속성이 stipple로 설정되어 있는 경우에만 사용할 수 있습니다([stroke.style](#) 참조). 사용할 수 있는 값에는 "one size", "small variation", "varied sizes" 및 "random sizes"가 있습니다.

예제

다음 예제에서는 stipple 스타일 획의 변형 속성을 random sizes로 설정합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "stipple";
myStroke.dotSpace = 3;
myStroke.variation = "random sizes";
myStroke.density = "sparse";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.waveHeight

지원 버전

Flash MX 2004

사용법

stroke.waveHeight

설명

속성, 불규칙 선의 웨이브 높이를 지정하는 문자열입니다. 이 속성은 stroke.style 속성이 ragged로 설정되어 있는 경우에만 사용할 수 있습니다([stroke.style](#) 참조). 사용할 수 있는 값에는 "flat", "wavy", "very wavy" 및 "wild"가 있습니다.

예제

다음 예제에서는 ragged 스타일 획의 waveHeight 속성을 flat으로 설정합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "ragged";
myStroke.pattern = "random";
myStroke.waveHeight = "flat";
myStroke.waveLength = "short";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.waveLength

지원 버전

Flash MX 2004

사용법

`stroke.waveLength`

설명

속성, 불규칙 선의 웨이브 길이를 지정하는 문자열입니다. 이 속성은 `stroke.style` 속성이 `ragged`로 설정되어 있는 경우에만 사용할 수 있습니다([stroke.style](#) 참조). 사용할 수 있는 값에는 "very short", "short", "medium" 및 "long"이 있습니다.

예제

다음 예제에서는 `ragged`: 스타일 획의 `waveLength` 속성을 `short`로 설정합니다.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "ragged";  
myStroke.pattern = "random";  
myStroke.waveHeight = "flat";  
myStroke.waveLength = "short";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

39장: swfPanel 객체

지원 버전

Flash CS4 Professional

설명

swfPanel 객체는 [윈도우 SWF] 패널을 나타냅니다. [윈도우 SWF] 패널은 Flash 제작 환경에서 실행할 수 있는 응용 프로그램을 구현하는 SWF 파일이며 [윈도우] > [기타 패널] 메뉴에서 사용할 수 있습니다. 기본적으로 [윈도우 SWF] 패널은 Configuration 폴더의 하위 폴더에 저장됩니다(2페이지의 “[JSFL 파일 저장](#)” 참조). 예를 들어 Windows XP에서는 이 폴더가 부트 드라이브\Documents and Settings\사용자\Local Settings\Application Data\Adobe\Flash CS4\언어\Configuration\WindowSWF입니다. [윈도우 SWF] 패널 샘플을 사용할 수 있습니다(12페이지의 “[비트맵 추적 패널 샘플](#)” 참조). 등록된 [윈도우 SWF] 패널의 배열은 [fl.swfPanels](#) 속성에 저장되어 있습니다.

메서드 요약

swfPanel 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
swfPanel.call()	ActionScript ExternalInterface.addCallback() 및 MMExecute() 메서드와 함께 작동하여 제작 환경에서 [SWF] 패널과 통신합니다.
393 페이지의 "swfPanel.setFocus()"	키보드 포커스를 지정된 SWF 패널에 설정합니다.

속성 요약

swfPanel 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
swfPanel.name	읽기 전용, 지정된 [윈도우 SWF] 패널의 이름을 나타내는 문자열
swfPanel.path	읽기 전용, 지정된 [윈도우 SWF] 패널에 사용되는 SWF 파일 경로를 나타내는 문자열

swfPanel.call()

지원 버전

Flash CS4 Professional

사용법

```
swfPanel.call(request)
```

매개 변수

request 함수로 전달할 매개 변수입니다(아래의 "설명" 및 "예제" 참조).

반환값

함수 호출을 통해 반환되는 null 또는 문자열. 이 함수를 호출할 때 빈 문자열이 반환될 수 있습니다.

설명

메서드, ActionScript ExternalInterface.addCallback() 및 MMExecute() 메서드와 함께 작동하여 제작 환경에서 [SWF] 패널과 통신합니다.

예제

다음 예제에서는 ActionScript 및 JavaScript 코드를 사용하여 [윈도우 SWF] 패널을 만들고 제작 환경에서 [윈도우 SWF] 패널과 통신하는 방법을 보여 줍니다.

- 1 ActionScript 3.0 FLA 파일을 만들어 파일 색상은 중간 회색으로, 파일 크기는 400픽셀(폭) x 250픽셀(높이)로 설정합니다.
- 2 스테이지 중심에 동적 텍스트 상자를 배치하고 텍스트 상자의 인스턴스 이름을 myTextField로 설정한 다음 텍스트 상자에 "Status"라는 단어를 입력합니다.
- 3 기타 텍스트 상자 속성을 다음과 유사하게 설정합니다.
 - 가운데 정렬
 - 355픽셀(폭) x 46픽셀(높이)
 - Times New Roman 글꼴, 28 포인트, 빨강

- 4 다음 ActionScript 코드를 추가합니다.

```
// 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 파일을 fileStatus fla로 저장하고 기본 [제작 설정]으로 SWF 파일을 제작합니다.
- 6 Flash를 닫습니다.
- 7 Configuration 폴더의 하위 디렉토리인 WindowSWF 폴더에 fileStatus.swf 파일을 복사합니다(2페이지의 “[JSFL 파일 저장](#)” 참조). 예를 들어 Windows XP에서는 이 폴더가 부트 드라이브\Documents and Settings\사용자\Local Settings\Application Data\Adobe\Flash CS4\언어\Configuration\WindowSWF입니다.
- 8 Flash를 시작합니다.
- 9 다음 코드를 사용하여 JSFL 파일을 만듭니다.

```
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 SWF 파일과 동일한 디렉토리에 fileOp.jsfl이라는 JSFL 파일을 저장합니다.

11 [윈도우] > [기타 패널] > [fileStatus]를 선택합니다.

FLA 파일을 만들고 열고 닫으면 [fileStatus] 패널에 수행한 액션을 나타내는 메시지가 표시됩니다.

swfPanel.name

지원 버전

Flash CS4 Professional

사용법

swfPanel.name

설명

읽기 전용 속성, 지정된 [윈도우 SWF] 패널의 이름을 나타내는 문자열

예제

다음 코드에서는 등록된 첫 번째 [윈도우 SWF] 패널의 이름을 [출력] 패널에 표시합니다.

```
fl.trace(fl.swfPanels[0].name);
```

참고 사항

[swfPanel.path](#), [fl.swfPanels](#)

swfPanel.path

지원 버전

Flash CS4 Professional

사용법

```
swfPanel.path
```

설명

읽기 전용 속성, 지정된 [윈도우 SWF] 패널에 사용되는 SWF 파일 경로를 나타내는 문자열

예제

다음 코드에서는 등록된 첫 번째 [윈도우 SWF] 패널에 사용되는 SWF 파일의 경로를 [출력] 패널에 표시합니다.

```
fl.trace(fl.swfPanels[0].path);
```

참고 사항

[swfPanel.name](#), [fl.swfPanels](#)

swfPanel.setFocus()

지원 버전

Flash CS5.5 Professional

사용법

```
swfPanel.setFocus()
```

설명

메서드: 키보드 포커스를 지정된 SWF 패널에 설정합니다.

예제

다음 코드에서는 “프로젝트”라는 SWF 패널로 포커스를 설정합니다.

이 명령을 실행하기 전에 다음 단계를 수행하십시오.

- 1 [프로젝트] 패널은 부동 패널이므로 도킹을 해제합니다.
- 2 [프로젝트] 패널에서 [파일 만들기] 대화 상자를 연 다음 스테이지를 클릭합니다.
- 3 Tab 키를 몇 번 눌러 [프로젝트] 패널에 포커스가 없는지 확인합니다.
- 4 [명령] 메뉴에서 아래 스크립트를 실행합니다(user/config/Commands 디렉토리에 아래 코드가 포함된 JSFL 파일을 넣음).
- 5 Tab 키를 누릅니다. [파일 만들기] 대화 상자의 텍스트 필드 중 하나에 삽입 커서가 나타납니다.

```
flash.getSwfPanel("Project").setFocus();
```

참고 사항

[swfPanel.name](#), [fl.swfPanels](#)

40장: SymbolInstance 객체

상속 Element 객체 > Instance 객체 > SymbolInstance 객체

지원 버전
Flash MX 2004

설명
SymbolInstance는 Instance 객체의 하위 클래스이며 프레임의 심볼을 나타냅니다([Instance 객체 참조](#)).

속성 요약
Instance 객체 속성 외에도 SymbolInstance 객체에는 다음과 같은 속성이 있습니다.

속성	설명
<code>symbolInstance.accName</code>	[액세스 가능성] 패널의 [이름] 필드에 해당하는 문자열
<code>symbolInstance.actionScript</code>	심볼에 할당된 액션을 지정하는 문자열
<code>symbolInstance.backgroundColor</code>	불투명이 선택되었을 때 매트 색상을 지정하는 문자열입니다.
<code>symbolInstance.bitmapRenderMode</code>	심볼 인스턴스의 표시 유형을 지정하는 문자열입니다.
<code>symbolInstance.blendMode</code>	무비 클립 심볼에 적용할 혼합 모드를 지정하는 문자열
<code>symbolInstance.buttonTracking</code>	버튼 심볼에 대해서만 속성 관리자의 [버튼으로 추적] 또는 [메뉴 항목으로 추적]에 대한 팝업 메뉴와 동일한 속성을 설정하는 문자열
<code>symbolInstance.cacheAsBitmap</code>	런타임 비트맵 캐싱의 활성 여부를 지정하는 부울 값
<code>symbolInstance.colorAlphaAmount</code>	[고급 효과 알파] 설정을 통해 인스턴스에 대한 색상 변환의 일부인 정수. 속성 관리자의 [색상] > [고급] 설정을 사용하여 대화 상자의 오른쪽에 있는 컨트롤을 조정하는 것과 같습니다.
<code>symbolInstance.colorAlphaPercent</code>	인스턴스에 대한 색상 변환의 일부를 지정하는 정수. 인스턴스 속성 관리자의 [색상] > [고급] 설정(대화 상자의 왼쪽에 있는 백분율 컨트롤)을 사용하는 것과 같습니다.
<code>symbolInstance.colorBlueAmount</code>	인스턴스에 대한 색상 변환의 일부인 정수. 인스턴스 속성 관리자의 [색상] > [고급] 설정을 사용하는 것과 같습니다.
<code>symbolInstance.colorBluePercent</code>	인스턴스에 대한 색상 변환의 일부인 정수. 인스턴스 속성 관리자의 [색상] > [고급] 설정(대화 상자의 왼쪽에 있는 백분율 컨트롤)을 사용하는 것과 같습니다.
<code>symbolInstance.colorGreenAmount</code>	인스턴스에 대한 색상 변환의 일부인 정수. 인스턴스 속성 관리자의 [색상] > [고급] 설정을 사용하는 것과 같습니다. 사용할 수 있는 값은 -255부터 255까지의 정수입니다.
<code>symbolInstance.colorGreenPercent</code>	인스턴스에 대한 색상 변환의 일부. 인스턴스 속성 관리자의 [색상] > [고급] 설정(대화 상자의 왼쪽에 있는 백분율 컨트롤)을 사용하는 것과 같습니다.
<code>symbolInstance.colorMode</code>	심볼 속성 관리자의 [색상] 팝업 메뉴에 표시된 대로 색상 모드를 지정하는 문자열
<code>symbolInstance.colorRedAmount</code>	인스턴스에 대한 색상 변환의 일부인 정수. 인스턴스 속성 관리자의 [색상] > [고급] 설정을 사용하는 것과 같습니다.
<code>symbolInstance.colorRedPercent</code>	인스턴스에 대한 색상 변환의 일부. 인스턴스 속성 관리자의 [색상] > [고급] 설정(대화 상자의 왼쪽에 있는 백분율 컨트롤)을 사용하는 것과 같습니다.
<code>symbolInstance.description</code>	[액세스 가능성] 패널의 [설명] 필드에 해당하는 문자열

속성	설명
<code>symbolInstance.filters</code>	Filter 객체의 배열(Filter 객체 참조)
<code>symbolInstance.firstFrame</code>	그래픽의 타임라인에 표시할 첫 번째 프레임을 지정하는 정수(0부터 시작)
<code>symbolInstance.forceSimple</code>	객체의 자식에 대한 액세스 가능 여부를 지정하는 부울 값으로, [액세스 가능성] 패널에 있는 [자식 객체 액세스 가능] 설정의 역논리에 해당합니다.
<code>symbolInstance.loop</code>	그래픽 심볼에 대해 속성 관리자의 [루프] 팝업 메뉴와 동일한 속성을 설정하는 문자열
<code>symbolInstance.shortcut</code>	심볼과 연관된 단축키에 해당하는 문자열로, [액세스 가능성] 패널의 [단축키] 필드에 해당합니다.
<code>symbolInstance.silent</code>	객체에 대한 액세스 가능 여부를 지정하는 부울 값으로, [액세스 가능성] 패널에 있는 [객체를 액세스 가능하게 만들기] 설정의 역논리에 해당합니다.
<code>symbolInstance.symbolType</code>	심볼의 유형을 지정하는 문자열. [새 심볼 생성] 및 [심볼로 변환] 대화 상자의 [비헤이비어] 값에 해당합니다.
<code>symbolInstance.tabIndex</code>	[액세스 가능성] 패널의 [탭 인덱스] 필드에 해당하는 정수
405페이지의 <code>"symbolInstance.usesBackgroundColor"</code>	표시 포맷을 지정하는 부울 값입니다.
406페이지의 <code>"symbolInstance.visible"</code>	인스턴스가 표시되는지 여부를 지정하는 부울 값입니다.

symbolInstance.accName

지원 버전
Flash MX 2004

사용법
`symbolInstance.accName`

설명
속성, [액세스 가능성] 패널의 [이름] 필드에 해당하는 문자열입니다. 화면 판독기에서는 이름을 읽어 객체를 식별합니다. 그래픽 심볼에는 이 속성을 사용할 수 없습니다.

예제
다음 예제에서는 객체의 [액세스 가능성] 패널 이름 값을 `theName` 변수에 저장합니다.

```
var theName = fl.getDocumentDOM().selection[0].accName;
```

다음 예제에서는 [객체의 액세스 가능성] 패널 이름 값을 Home Button으로 설정합니다.

```
fl.getDocumentDOM().selection[0].accName = "Home Button";
```

symbolInstance.actionScript

지원 버전
Flash MX 2004

사용법

`symbolInstance.actionScript`

설명

속성, 심볼에 할당된 액션을 지정하는 문자열입니다. 이 속성은 무비 클립과 버튼 인스턴스에만 적용됩니다. 그래픽 심볼 인스턴스의 경우 `undefined` 값이 반환됩니다.

예제

다음 예제에서는 타임라인에 있는 첫 번째 레이어의 첫 번째 프레임에 있는 첫 번째 항목에 `onClipEvent` 액션을 할당합니다.

```
f1.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].actionScript  
= "onClipEvent(enterFrame) {trace('movie clip enterFrame');}";
```

symbolInstance.backgroundColor

지원 버전

Flash CS5.5 Professional

사용법

`symbolInstance.backgroundColor`

설명

속성, 인스턴스에 대해 24비트 모드가 선택되었을 때 매트 색상을 지정하는 문자열입니다. 16진수 `#rrggbb` 포맷으로 되어 있는 문자열이거나 값을 포함하는 정수입니다.

예제

다음 예제는 심볼 인스턴스에 검정 배경색을 할당합니다.

```
var bitmapInstance = f1.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];  
bitmapInstance.backgroundColor = "#000000";
```

symbolInstance.bitmapRenderMode

지원 버전

Flash CS5.5 Professional

사용법

`symbolInstance.bitmapRenderMode`

설명

속성, 심볼의 표시 유형을 설정하는 문자열입니다.

허용되는 값은 다음과 같습니다.

- “none”
- “cache” - 런타임에 Flash Player에 의해 비트맵으로 캐시되도록 심볼을 설정합니다.
- “export” - SWF가 컴파일될 때 비트맵으로 내보내도록 심볼을 설정합니다.

이전의 398페이지의 "`symbolInstance.cacheAsBitmap`" 속성이 이 속성과 유사하지만 부울이기 때문에 선택의 폭이 좁습니다. 앞으로는 `cacheAsBitmap` 속성을 사용할 수 없으므로 새로운 속성을 사용해야 합니다. 부울 `cacheAsBitmap` 속성의 `true/false` 옵션은 이 새로운 속성의 "`cache`" / "`none`" 값과 동일합니다.

예제

다음 예제는 심볼의 `bitmapRenderMode`를 "export"로 할당합니다.

```
var symbol = fl.getDocumentDOM().selection[0];  
fl.trace(symbol.bitmapRenderMode);  
symbol.bitmapRenderMode = "export";
```

symbolInstance.blendMode

지원 버전

Flash 8

사용법

`symbolInstance.blendMode`

설명

속성, 무비 클립 심볼에 적용할 혼합 모드를 지정하는 문자열입니다. 사용할 수 있는 값에는 "normal", "layer", "multiply", "screen", "overlay", "hardlight", "lighten", "darken", "difference", "add", "subtract", "invert", "alpha" 및 "erase"가 있습니다.

예제

다음 예제에서는 첫 번째 수준에 있는 첫 번째 프레임의 첫 번째 무비 클립 심볼에 대한 혼합 모드를 `add`로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].blendMode = "add";
```

참고 사항

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

symbolInstance.buttonTracking

지원 버전

Flash MX 2004

사용법

`symbolInstance.buttonTracking`

설명

속성, 버튼 심볼에 대해서만 속성 관리자의 [버튼으로 추적] 또는 [메뉴 항목으로 추적]에 대한 팝업 메뉴와 동일한 속성을 설정하는 문자열입니다. 다른 유형의 심볼에는 이 속성을 사용할 수 없습니다. 사용할 수 있는 값에는 "button" 또는 "menu"가 있습니다.

예제

다음 예제에서는 심볼이 버튼인 경우에 한해서 타임라인에 있는 첫 번째 레이어의 첫 번째 프레임에 있는 첫 번째 심볼을 [메뉴 항목으로 추적]으로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].buttonTracking = "menu";
```

symbolInstance.cacheAsBitmap

지원 버전

Flash 8

사용법

```
symbolInstance.cacheAsBitmap
```

설명

속성, 런타임 비트맵 캐싱의 활성화 여부를 지정하는 부울 값입니다.

참고: Flash Professional CS5.5 이상부터는 이 속성 대신 396페이지의 "[symbolInstance.bitmapRenderMode](#)" 속성을 사용해야 합니다.

예제

다음 예제에서는 첫 번째 레이어에 있는 첫 번째 프레임의 첫 번째 요소에 대해 런타임 비트맵 캐싱을 활성화합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].cacheAsBitmap = true;
```

symbolInstance.colorAlphaAmount

지원 버전

Flash MX 2004

사용법

```
symbolInstance.colorAlphaAmount
```

설명

속성, 인스턴스 색상 변환의 일부로 [고급 효과 알파] 설정을 지정하는 지정하는 정수입니다. 이 속성은 속성 관리자의 [색상] > [고급] 설정을 사용하여 대화 상자의 오른쪽에 있는 컨트롤을 조정하는 것과 같습니다. 이 값은 농도 및 알파 값을 일정한 양만큼 감소 또는 증가시킵니다. 이 값은 현재 값에 더해집니다. 이 속성은 [symbolInstance.colorAlphaPercent](#)와 함께 사용하면 가장 유용합니다. 사용할 수 있는 값은 -255부터 255까지의 정수입니다.

예제

다음 예제에서는 선택한 심볼 인스턴스의 알파 설정값에서 100을 뺍니다.

```
fl.getDocumentDOM().selection[0].colorAlphaAmount = -100;
```

symbolInstance.colorAlphaPercent

지원 버전

Flash MX 2004

사용법

```
symbolInstance.colorAlphaPercent
```

설명

속성, 인스턴스에 대한 색상 변환의 일부를 지정하는 정수입니다. 이 속성은 인스턴스 속성 관리자의 [색상] > [고급] 설정(대화 상자의 왼쪽에 있는 백분율 컨트롤)을 사용하는 것과 같습니다. 이 값은 농도 및 알파 값을 지정된 백분율로 변경합니다. 사용할 수 있는 값은 -100부터 100까지의 정수입니다. 자세한 내용은 [symbolInstance.colorAlphaAmount](#)를 참조하십시오.

예제

다음 예제에서는 선택한 심볼 인스턴스의 colorAlphaPercent를 80으로 설정합니다.

```
fl.getDocumentDOM().selection[0].colorAlphaPercent = 80;
```

symbolInstance.colorBlueAmount

지원 버전

Flash MX 2004

사용법

```
symbolInstance.colorBlueAmount
```

설명

속성, 인스턴스에 대한 색상 변환의 일부인 정수입니다. 이 속성은 인스턴스 속성 관리자의 [색상] > [고급] 설정을 사용하는 것과 같습니다. 사용할 수 있는 값은 -255부터 255까지의 정수입니다.

symbolInstance.colorBluePercent

지원 버전

Flash MX 2004

사용법

```
symbolInstance.colorBluePercent
```

설명

속성, 인스턴스에 대한 색상 변환의 일부인 정수입니다. 이 속성은 인스턴스 속성 관리자의 [색상] > [고급] 설정(대화 상자의 왼쪽에 있는 백분율 컨트롤)을 사용하는 것과 같습니다. 이 값은 파란색 값을 지정된 백분율로 설정합니다. 사용할 수 있는 값은 -100부터 100까지의 정수입니다.

예제

다음 예제에서는 선택한 심볼 인스턴스의 colorBluePercent를 80으로 설정합니다.

```
fl.getDocumentDOM().selection[0].colorBluePercent = 80;
```

symbolInstance.colorGreenAmount

지원 버전

Flash MX 2004

사용법

`symbolInstance.colorGreenAmount`

설명

속성, 인스턴스에 대한 색상 변환의 일부인 정수입니다. 이 속성은 인스턴스 속성 관리자의 [색상] > [고급] 설정을 사용하는 것과 같습니다. 사용할 수 있는 값은 -255부터 255까지의 정수입니다.

symbolInstance.colorGreenPercent

지원 버전

Flash MX 2004

사용법

`symbolInstance.colorGreenPercent`

설명

속성, 인스턴스에 대한 색상 변환의 일부입니다. 이 속성은 인스턴스 속성 관리자의 [색상] > [고급] 설정(대화 상자의 왼쪽에 있는 백분율 컨트롤)을 사용하는 것과 같습니다. 이 값은 녹색 값을 지정된 백분율로 설정합니다. 사용할 수 있는 값은 -100부터 100까지의 정수입니다.

예제

다음 예제에서는 선택한 심볼 인스턴스의 `colorGreenPercent`를 70으로 설정합니다.

```
fl.getDocumentDOM().selection[0].colorGreenPercent = 70;
```

symbolInstance.colorMode

지원 버전

Flash MX 2004

사용법

`symbolInstance.colorMode`

설명

속성, 심볼 속성 관리자의 [색상] 팝업 메뉴에 표시된 대로 색상 모드를 지정하는 문자열입니다. 사용할 수 있는 값에는 "none", "brightness", "tint", "alpha" 및 "advanced"가 있습니다.

예제

다음 예제에서는 타임라인에 있는 첫 번째 레이어의 첫 번째 프레임에 있는 첫 번째 요소의 `colorMode` 속성을 `alpha`로 변경합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].colorMode = "alpha";
```


symbolInstance.colorRedAmount

지원 버전

Flash MX 2004

사용법

```
symbolInstance.colorRedAmount
```

설명

속성, 인스턴스에 대한 색상 변환의 일부인 정수입니다. 이 속성은 인스턴스 속성 관리자의 [색상] > [고급] 설정을 사용하는 것과 같습니다. 사용할 수 있는 값은 -255부터 255까지의 정수입니다.

예제

다음 예제에서는 선택한 심볼 인스턴스의 colorRedAmount를 255로 설정합니다.

```
fl.getDocumentDOM().selection[0].colorRedAmount = 255;
```

symbolInstance.colorRedPercent

지원 버전

Flash MX 2004

사용법

```
symbolInstance.colorRedPercent
```

설명

속성, 인스턴스에 대한 색상 변환의 일부입니다. 이 속성은 인스턴스 속성 관리자의 [색상] > [고급] 설정(대화 상자의 왼쪽에 있는 백분율 컨트롤)을 사용하는 것과 같습니다. 이 값은 빨간색 값을 지정된 백분율로 설정합니다. 사용할 수 있는 값은 -100부터 100까지의 정수입니다.

예제

다음 예제에서는 선택한 심볼 인스턴스의 colorRedPercent를 10으로 설정합니다.

```
fl.getDocumentDOM().selection[0].colorRedPercent = 10;
```

symbolInstance.description

지원 버전

Flash MX 2004

사용법

```
symbolInstance.description
```

설명

속성, [액세스 가능성] 패널의 [설명] 필드에 해당하는 문자열입니다. 이 설명은 화면 판독기에서 읽습니다. 그래픽 심볼에는 이 속성을 사용할 수 없습니다.

예제

다음 예제에서는 객체의 [액세스 가능성] 패널 설명 값을 theDescription 변수에 저장합니다.

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

다음 예제에서는 [액세스 가능성] 패널 설명 값을 Click the home button to go to home으로 설정합니다.

```
fl.getDocumentDOM().selection[0].description= "Click the home button to go to home";
```

symbolInstance.filters

지원 버전

Flash 8

사용법

symbolInstance.filters

설명

속성, Filter 객체의 배열(Filter 객체 참조)입니다. 필터 속성을 수정하기 위해 이 배열에 직접 쓰는 대신 배열을 검색하고 개별 속성을 설정한 다음 새 속성을 반영하도록 배열을 설정합니다.

예제

다음 예제에서는 인덱스 0에서 필터의 이름을 추적합니다. 광선 필터인 경우 해당 blurX 속성이 100으로 설정되고 새 값이 필터 배열에 기록됩니다.

```
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

지원 버전

Flash MX 2004

사용법

symbolInstance.firstFrame

설명

속성, 그래픽의 타임라인에 표시할 첫 번째 프레임을 지정하는 정수(0부터 시작)입니다. 이 속성은 그래픽 심볼에만 적용되며 속성 관리자의 [첫 번째] 필드와 동일한 속성을 설정합니다. 다른 유형의 심볼의 경우 이 속성은 undefined입니다.

예제

다음 예제에서는 프레임 10을 지정된 요소의 타임라인에 표시할 첫 번째 프레임으로 지정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].firstFrame = 10;
```

symbolInstance.forceSimple

지원 버전

Flash MX 2004

사용법

```
symbolInstance.forceSimple
```

설명

속성, 객체의 자식에 대한 액세스 가능 여부를 지정하는 부울 값입니다. 이 속성은 [액세스 가능성] 패널에 있는 [자식 객체 액세스 가능] 설정의 역논리에 해당합니다. 예를 들어, forceSimple이 true이면 [자식 객체 액세스 가능] 옵션이 선택되어 있지 않은 것과 같고 forceSimple이 false이면 [자식 객체 액세스 가능] 옵션이 선택되어 있는 것과 같습니다.

이 속성은 MovieClip 객체에만 사용할 수 있습니다.

예제

다음 예제에서는 객체의 자식에 액세스할 수 있는지 여부를 확인합니다. false 반환값은 자식에 액세스할 수 있음을 의미합니다.

```
var areChildrenAccessible = fl.getDocumentDOM().selection[0].forceSimple;
```

다음 예제에서는 객체의 자식에 액세스할 수 있도록 설정합니다.

```
fl.getDocumentDOM().selection[0].forceSimple = false;
```

symbolInstance.loop

지원 버전

Flash MX 2004

사용법

```
symbolInstance.loop
```

설명

속성, 그래픽 심볼에 대해 속성 관리자의 [루프] 팝업 메뉴와 동일한 속성을 설정하는 문자열입니다. 다른 유형의 심볼의 경우 이 속성은 undefined입니다. 사용할 수 있는 값에는 "loop", "play once" 및 "single frame"이 있습니다. 이 값에 따라 그래픽의 애니메이션이 설정됩니다.

예제

다음 예제에서는 심볼이 그래픽인 경우에 한해서 타임라인에 있는 첫 번째 레이어의 첫 번째 프레임에 있는 첫 번째 심볼을 단일 프레임(그래픽 타임라인에 있는 특정 프레임 하나를 표시함)으로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].loop = 'single frame';
```

symbolInstance.shortcut

지원 버전

Flash MX 2004

사용법

`symbolInstance.shortcut`

설명

속성, 심볼과 연관된 단축키에 해당하는 문자열로, [액세스 가능성] 패널의 [단축키] 필드에 해당합니다. 이 단축키는 화면 판독기에서 읽습니다. 그래픽 심볼에는 이 속성을 사용할 수 없습니다.

예제

다음 예제에서는 객체의 단축키 값을 `theShortcut` 변수에 저장합니다.

```
var theShortcut = fl.getDocumentDOM().selection[0].shortcut;
```

다음 예제에서는 객체의 단축키를 Ctrl+i로 설정합니다.

```
fl.getDocumentDOM().selection[0].shortcut = "Ctrl+i";
```

symbolInstance.silent

지원 버전

Flash MX 2004

사용법

`symbolInstance.silent`

설명

속성, 객체에 대한 액세스 가능 여부를 지정하는 부울 값입니다. 이 속성은 [액세스 가능성] 패널에 있는 [객체를 액세스 가능하게 만들기] 설정의 역논리에 해당합니다. 예를 들어, `silent`가 `true`이면 [객체를 액세스 가능하게 만들기] 옵션이 선택되어 있지 않은 것과 같고 `silent`가 `false`이면 [객체를 액세스 가능하게 만들기] 옵션이 선택되어 있는 것과 같습니다.

그래픽 객체에는 이 속성을 사용할 수 없습니다.

예제

다음 예제에서는 객체에 액세스할 수 있는지 여부를 확인합니다. `false` 반환값은 객체에 액세스할 수 있음을 의미합니다.

```
var isSilent = fl.getDocumentDOM().selection[0].silent;
```

다음 예제에서는 객체에 액세스할 수 있도록 설정합니다.

```
fl.getDocumentDOM().selection[0].silent = false;
```

symbolInstance.symbolType

지원 버전

Flash MX 2004

사용법

`symbolInstance.symbolType`

설명

속성, 심볼의 유형을 지정하는 문자열입니다. 이 속성은 [새 심볼 생성] 및 [심볼로 변환] 대화 상자의 [비헤이비어] 값에 해당합니다. 사용할 수 있는 값에는 "button", "movie clip" 및 "graphic"이 있습니다.

예제

다음 예제에서는 현재 문서의 타임라인에 있는 첫 번째 레이어의 첫 번째 프레임에 있는 첫 번째 심볼을 그래픽 심볼처럼 동작하도록 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].symbolType = "graphic";
```

symbolInstance.tabIndex

지원 버전

Flash MX 2004

사용법

```
symbolInstance.tabIndex
```

설명

속성, [액세스 가능성] 패널의 [탭 인덱스] 필드에 해당하는 정수입니다. 사용자가 Tab 키를 누를 때 객체가 액세스되는 순서를 지정하는 탭 순서를 만듭니다. 그래픽 심볼에는 이 속성을 사용할 수 없습니다.

예제

다음 예제에서는 mySymbol 객체의 tabIndex 속성을 3으로 설정한 다음 이 값을 [출력] 패널에 표시합니다.

```
var mySymbol = fl.getDocumentDOM().selection[0];  
mySymbol.tabIndex = 3;  
fl.trace(mySymbol.tabIndex);
```

symbolInstance.usesBackgroundColor

지원 버전

Flash CS5.5 Professional

사용법

```
symbolInstance.usesBackgroundColor
```

설명

속성, 인스턴스에 대해 24비트 모드(true)와 알파가 포함된 32비트 모드(false) 중 무엇을 사용해야 하는지를 나타내는 부울 값입니다. true이면 인스턴스에 대해 지정된 backgroundColor가 사용됩니다.

예제

다음 예제는 인스턴스의 usesBackgroundColor 속성을 true로 설정합니다.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].useTransparentBackground = true;
```

symbolInstance.visible

지원 버전

Flash CS5.5 Professional

사용법

`symbolInstance.visible`

설명

속성, 객체의 **Visible** 속성을 **on(true)** 또는 **off(false)**로 설정하는 부울 값입니다.

예제

다음 예제는 첫 레이어의 첫 프레임에 있는 첫 항목의 가시성을 **false**로 설정합니다.

```
f1.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].visible = false;
```

41장: SymbolItem 객체

상속 [Item 객체](#) > SymbolItem 객체

지원 버전
Flash MX 2004

설명
SymbolItem 객체는 [Item 객체](#)의 하위 클래스입니다.

메서드 요약
Item object 메서드 외에도 SymbolItem 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
symbolItem.convertToCompiledClip()	라이브러리의 심볼 항목을 컴파일된 무비 클립으로 변환합니다.
symbolItem.exportSWC()	심볼 항목을 SWC 파일로 내보냅니다.
symbolItem.exportSWF()	심볼 항목을 SWF 파일로 내보냅니다.

속성 요약
Item 객체 속성 외에도 SymbolItem 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
symbolItem.scalingGrid	항목에 대한 9-슬라이스 크기 조절의 활성화 여부를 지정하는 부울 값
symbolItem.scalingGridRect	4개의 9-슬라이스 안내선의 위치를 지정하는 Rectangle 객체
symbolItem.sourceAutoUpdate	FLA 파일이 제작될 때 항목이 업데이트되는지 여부를 지정하는 부울 값
symbolItem.sourceFilePath	소스 FLA 파일의 경로를 file:/// URI로 지정하는 문자열
symbolItem.sourceLibraryName	소스 파일 라이브러리에 있는 항목의 이름을 지정하는 문자열
symbolItem.symbolType	심볼의 유형을 지정하는 문자열
symbolItem.timeline	읽기 전용, Timeline 객체

symbolItem.convertToCompiledClip()

지원 버전
Flash MX 2004

사용법
`symbolItem.convertToCompiledClip()`

매개 변수
없음

반환값

없음

설명

메서드, 라이브러리의 심볼 항목을 컴파일된 무비 클립으로 변환합니다.

예제

다음 예제에서는 라이브러리 항목을 컴파일된 무비 클립으로 변환합니다.

```
fl.getDocumentDOM().library.items[3].convertToCompiledClip();
```

symbolItem.exportSWC()

지원 버전

Flash MX 2004

사용법

```
symbolItem.exportSWC(outputURI)
```

매개 변수

outputURI 심볼을 내보낼 SWC 파일을 지정하는 문자열(file:/// URI로 표시됨). outputURI는 로컬 파일을 참조해야 합니다. outputURI가 없으면 폴더가 만들어지지 않습니다.

반환값

없음

설명

메서드, 심볼 항목을 SWC 파일로 내보냅니다.

예제

다음 예제에서는 라이브러리 항목을 tests 폴더에 있는 mySymbol.swc라는 SWC 파일로 내보냅니다.

```
fl.getDocumentDOM.library.selectItem("mySymbol");  
var currentSelection = fl.getDocumentDOM().library.getSelectedItems();  
currentSelection[0].exportSWC("file:///Macintosh HD/SWCDirectory/mySymbol.swc");
```

symbolItem.exportSWF()

지원 버전

Flash MX 2004

사용법

```
symbolItem.exportSWF(outputURI)
```

매개 변수

outputURI 심볼을 내보낼 SWF 파일을 지정하는 문자열(file:/// URI로 표시됨). outputURI는 로컬 파일을 참조해야 합니다. outputURI가 없으면 폴더가 만들어지지 않습니다.

반환값

없음

설명

메서드, 심볼 항목을 SWF 파일로 내보냅니다.

예제

다음 예제에서는 라이브러리 항목을 tests 폴더에 있는 my.swf 파일로 내보냅니다.

```
fl.getDocumentDOM().library.items[0].exportSWF("file:///c:/tests/my.swf");
```

symbolItem.scalingGrid

지원 버전

Flash 8

사용법

```
symbolItem.scalingGrid
```

설명

속성, 항목에 대한 9-슬라이스 크기 조절을 활성화할지 여부를 지정하는 부울 값입니다.

예제

다음 예제에서는 라이브러리에 있는 항목에 대해 9-슬라이스 크기 조절을 활성화합니다.

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

참고 사항

[symbolItem.scalingGridRect](#)

symbolItem.scalingGridRect

지원 버전

Flash 8

사용법

```
symbolItem.scalingGridRect
```

설명

속성, 4개의 9-슬라이스 안내선의 위치를 지정하는 Rectangle 객체입니다. 사각형 형식에 대한 자세한 내용은 [document.addNewRectangle\(\)](#)을 참조하십시오.

예제

다음 예제에서는 9 슬라이스 안내선의 위치를 지정합니다.

```
fl.getDocumentDOM().library.items[0].scalingGridRect = {left:338, top:237, right:3859, bottom:713};
```

참고 사항

[symbolItem.scalingGrid](#)

symbolItem.sourceAutoUpdate

지원 버전

Flash MX 2004

사용법

`symbolItem.sourceAutoUpdate`

설명

속성, FLA 파일이 제작될 때 항목이 업데이트되는지 여부를 지정하는 부울 값입니다. 기본값은 false입니다. 이 속성은 공유 라이브러리 심볼에 사용됩니다.

예제

다음 예제에서는 라이브러리 항목의 `sourceAutoUpdate` 속성을 설정합니다.

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

symbolItem.sourceFilePath

지원 버전

Flash MX 2004

사용법

`symbolItem.sourceFilePath`

설명

속성, 소스 FLA 파일의 경로를 `file:///` URI로 지정하는 문자열입니다. 경로는 상대 경로가 아닌 절대 경로여야 합니다. 이 속성은 공유 라이브러리 심볼에 사용됩니다.

예제

다음 예제에서는 `sourceFilePath` 속성의 값을 [출력] 패널에 표시합니다.

```
fl.trace(fl.getDocumentDOM().library.items[0].sourceFilePath);
```

symbolItem.sourceLibraryName

지원 버전

Flash MX 2004

사용법

`symbolItem.sourceLibraryName`

설명

속성, 소스 파일 라이브러리에 있는 항목의 이름을 지정하는 문자열입니다. 이 속성은 공유 라이브러리 심볼에 사용됩니다.

예제

다음 예제에서는 sourceLibraryName 속성 값을 [출력] 패널에 표시합니다.

```
f1.trace(f1.getDocumentDOM().library.items[0].sourceLibraryName);
```

symbolItem.symbolType

지원 버전

Flash MX 2004

사용법

```
symbolItem.symbolType
```

설명

속성, 심볼의 유형을 지정하는 문자열입니다. 사용할 수 있는 값에는 "movie clip", "button" 및 "graphic"이 있습니다.

예제

다음 예제에서는 symbolType 속성의 현재 값을 보여 주고 이 값을 button으로 변경한 다음 변경된 값을 다시 보여 줍니다.

```
alert(f1.getDocumentDOM().library.items[0].symbolType);  
f1.getDocumentDOM().library.items[0].symbolType = "button";  
alert(f1.getDocumentDOM().library.items[0].symbolType);
```

symbolItem.timeline

지원 버전

Flash MX 2004

사용법

```
symbolItem.timeline
```

설명

읽기 전용 속성, [Timeline 객체](#)입니다.

예제

다음 예제에서는 라이브러리에 있는 선택된 무비 클립에 포함된 레이어 수를 보여 줍니다.

```
var t1 = f1.getDocumentDOM().library.getSelectedItems()[0].timeline;  
alert(t1.layerCount);
```

42장: Text 객체

상속 [Element 객체](#) > Text 객체

지원 버전

Flash MX 2004

설명

Text 객체는 문서 내의 단일 텍스트 항목을 나타냅니다. 모든 텍스트 속성은 전체 텍스트 블록에 영향을 줍니다.

텍스트 필드 내의 텍스트 흐름 속성을 설정하려면 [TextAttrs 객체](#)의 속성 요약을 참조하십시오. 텍스트 필드 내의 선택 영역 속성을 변경하려면 [document.setTextAttr\(\)](#)를 사용하여 텍스트 범위를 지정하거나 현재 선택 영역을 사용할 수 있습니다.

선택한 텍스트 필드의 일반 속성을 설정하려면 [document.setProperty\(\)](#)를 사용합니다. 다음 예제에서는 선택한 텍스트 필드 등록 포인트의 x 값을 50으로 설정합니다.

```
fl.getDocumentDOM().setProperty("x", 50);
```

메서드 요약

Element 객체 메서드 외에도 Text 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
text.getTextAttr()	선택적 <i>startIndex</i> 및 <i>endIndex</i> 매개 변수에 의해 식별된 텍스트의 특정 특성을 가져옵니다.
text.getTextString()	지정된 텍스트의 범위를 가져옵니다.
text.setTextAttr()	<i>startIndex</i> 및 <i>endIndex</i> 에 의해 식별된 텍스트와 연관된 특정 특성을 설정합니다.
text.setTextString()	이 Text 객체 내에서 텍스트 문자열을 변경합니다.

속성 요약

Element 객체 속성 외에도 Text 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
text.accName	[액세스 가능성] 패널의 [이름] 필드에 해당하는 문자열
text.antiAliasSharpness	텍스트의 안티앨리어싱 선명도를 지정하는 부동 소수점 값
text.antiAliasThickness	텍스트의 안티앨리어싱 두께를 지정하는 부동 소수점 값
text.autoExpand	정적 텍스트 필드의 경계 폭 확장 또는 동적/입력 텍스트 필드의 경계 폭 및 높이 확장을 제어하는 부울 값
text.border	동적 또는 입력 텍스트 주위에 테두리를 표시할 것인지(true) 아니면 표시하지 않을 것인지(false)를 제어하는 부울 값
text.description	[액세스 가능성] 패널의 [설명] 필드에 해당하는 문자열
text.embeddedCharacters	포함할 문자를 지정하는 문자열. 이 속성은 [문자 포함] 대화 상자에 텍스트를 입력하는 것과 같습니다.
text.embedRanges	[문자 포함] 대화 상자에서 선택할 수 있는 항목에 해당하는 일련의 정수(구분 기호로 분리됨)로 구성된 문자열
text.embedVariantGlyphs	변형 글리프 포함 가능 여부를 지정하는 부울 값
text.fontRenderingMode	텍스트에 대해 렌더링 모드를 지정하는 문자열

속성	설명
<code>text.length</code>	읽기 전용, Text 객체의 문자 수를 나타내는 정수
<code>text.lineType</code>	행 유형을 "single line", "multiline", "multiline no wrap" 또는 "password"로 설정하는 문자열
<code>text.maxCharacters</code>	이 Text 객체에 입력할 수 있는 최대 문자 수를 지정하는 정수
<code>text.orientation</code>	텍스트 필드의 방향을 지정하는 문자열
<code>text.renderAsHTML</code>	텍스트를 HTML로 그리고 포함된 HTML 태그를 해석할 것인지 여부를 제어하는 부울 값
<code>text.scrollable</code>	텍스트를 스크롤할 수 있게 할 것인지(true) 아니면 스크롤할 수 없게 할 것인지(false)를 제어하는 부울 값
<code>text.selectable</code>	텍스트를 선택할 수 있게 할 것인지(true) 아니면 선택할 수 없게 할 것인지(false)를 제어하는 부울 값. 입력 텍스트는 항상 선택할 수 있습니다.
<code>text.selectionEnd</code>	텍스트 세부 선택 끝 위치의 오프셋을 지정하는 정수(0부터 시작)
<code>text.selectionStart</code>	텍스트 세부 선택 시작 위치의 오프셋을 지정하는 정수(0부터 시작)
<code>text.shortcut</code>	[액세스 가능성] 패널의 [단축키] 필드에 해당하는 문자열
<code>text.silent</code>	객체에 액세스할 수 있는지 여부를 지정하는 부울 값
<code>text.tabIndex</code>	[액세스 가능성] 패널의 [탭 인덱스] 필드에 해당하는 정수
<code>text.textRuns</code>	읽기 전용, TextRun 객체의 배열
<code>text.textType</code>	텍스트 필드의 유형을 지정하는 문자열. 사용할 수 있는 값에는 "static", "dynamic" 및 "input"이 있습니다.
<code>text.useDeviceFonts</code>	부울 값. true로 설정하면 텍스트를 그리는 데 장치 글꼴이 사용됩니다.
<code>text.variableName</code>	Text 객체의 내용이 포함된 문자열

text.accName

지원 버전
Flash MX 2004

사용법
`text.accName`

설명

속성, [액세스 가능성] 패널의 [이름] 필드에 해당하는 문자열입니다. 화면 판독기에서는 이름을 읽어 객체를 식별합니다. 동적 텍스트에는 이 속성을 사용할 수 없습니다.

예제

다음 예제에서는 객체의 이름을 가져옵니다.

```
var doc = fl.getDocumentDOM();
var theName = doc.selection[0].accName;
```

다음 예제에서는 현재 선택한 객체의 이름을 설정합니다.

```
fl.getDocumentDOM().selection[0].accName = "Home Button";
```

text.antiAliasSharpness

지원 버전

Flash 8

사용법

`text.antiAliasSharpness`

설명

속성이며, 텍스트의 엔티앨리어싱 선명도를 지정하는 부동 소수점 값입니다. 이 속성은 텍스트의 선명도를 제어합니다. 값이 커질수록 텍스트가 더 선명해집니다. 0 값은 일반적인 선명도를 지정합니다. 이 속성은 `text.fontRenderingMode`가 `customThicknessSharpness`로 설정되어 있는 경우에만 사용할 수 있습니다.

예제

[text.fontRenderingMode](#)를 참조하십시오.

참고 사항

[text.antiAliasThickness](#), [text.fontRenderingMode](#)

text.antiAliasThickness

지원 버전

Flash 8

사용법

`text.antiAliasThickness`

설명

속성, 텍스트의 엔티앨리어싱 두께를 지정하는 부동 소수점 값입니다. 이 속성은 텍스트를 두껍게 그리는 방법을 제어합니다. 값이 커질수록 텍스트가 보다 두꺼워집니다. 0 값은 일반적인 두께를 지정합니다. 이 속성은 `text.fontRenderingMode`가 `customThicknessSharpness`로 설정되어 있는 경우에만 사용할 수 있습니다.

예제

[text.fontRenderingMode](#)를 참조하십시오.

참고 사항

[text.antiAliasSharpness](#), [text.fontRenderingMode](#)

text.autoExpand

지원 버전

Flash MX 2004

사용법

```
text.autoExpand
```

설명

속성, 부울 값입니다. 정적 텍스트 필드의 경우 **true**로 설정하면 모든 텍스트를 표시할 수 있도록 경계 폭이 확장됩니다. 동적 또는 입력 텍스트 필드의 경우 **true**로 설정하면 모든 텍스트를 표시할 수 있도록 경계 폭 및 높이가 확장됩니다.

예제

다음 예제에서는 autoExpand 속성 값을 true로 설정합니다.

```
fl.getDocumentDOM().selection[0].autoExpand = true;
```

text.border

지원 버전

Flash MX 2004

사용법

```
text.border
```

설명

속성, 부울 값입니다. true로 설정하면 텍스트 주위에 테두리가 표시됩니다.

예제

다음 예제에서는 border 속성 값을 true로 설정합니다.

```
fl.getDocumentDOM().selection[0].border = true;
```

text.description

지원 버전

Flash MX 2004

사용법

```
text.description
```

설명

속성, [액세스 가능성] 패널의 [설명] 필드에 해당하는 문자열입니다. 이 설명은 화면 판독기에서 읽습니다.

예제

다음 예제에서는 객체에 대한 설명을 가져옵니다.

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

다음 예제에서는 객체에 대한 설명을 설정합니다.

```
var doc = fl.getDocumentDOM();  
doc.selection[0].description= "Enter your name here";
```

text.embeddedCharacters

지원 버전

Flash MX 2004

사용법

`text.embeddedCharacters`

설명

속성, 포함할 문자를 지정하는 문자열입니다. 이 속성은 [문자 포함] 대화 상자에 텍스트를 입력하는 것과 같습니다.

이 속성은 동적 또는 입력 텍스트에만 사용할 수 있으며 다른 텍스트 유형에 사용하면 경고가 표시됩니다.

참고: Flash Professional CS5에서 시작하는 경우 글꼴 포함은 텍스트 객체 수준이 아닌 문서 수준에서 제어됩니다. `text.embeddedCharacters` 속성 대신 266페이지의 “[fontItem.embeddedCharacters](#)” 속성을 사용하십시오.

예제

다음 예제에서는 현재 선택 영역의 첫 번째 또는 유일한 항목을 클래식 텍스트 객체로 간주하고 `embeddedCharacters` 속성을 abc로 설정합니다.

```
fl.getDocumentDOM().selection[0].embeddedCharacters = "abc";
```

text.embedRanges

지원 버전

Flash MX 2004

사용법

`text.embedRanges`

설명

속성, [문자 포함] 대화 상자에서 선택할 수 있는 항목에 해당하는 일련의 정수(구분 기호로 분리됨)로 구성된 문자열입니다. 이 속성은 동적 또는 입력 텍스트에만 사용할 수 있으며 정적 텍스트에 사용하면 무시됩니다.

이 속성은 Configuration/Font Embedding 폴더에 있는 XML 파일에 해당합니다.

참고: Flash Professional CS5에서 시작하는 경우 글꼴 포함은 텍스트 객체 수준이 아닌 문서 수준에서 제어됩니다. `text.embedRanges` 속성 대신 266페이지의 “[fontItem.embedRanges](#)” 속성을 사용하십시오.

예제

다음 예제에서는 현재 선택 영역의 첫 번째 또는 유일한 항목을 클래식 텍스트로 간주하고 `embedRanges` 속성을 "1|3|7"로 설정합니다.

```
var doc = fl.getDocumentDOM();  
doc.selection[0].embedRanges = "1|3|7";
```

다음 예제에서는 속성을 재설정합니다.

```
var doc = fl.getDocumentDOM();  
doc.selection[0].embedRanges = "";
```


text.embedVariantGlyphs

지원 버전

Flash CS4 Professional

사용법

```
text.embedVariantGlyphs
```

설명

속성, 변형 글리프를 포함할 수 있는지(true) 포함할 수 없는지(false)를 나타내는 부울 값입니다. 이 속성은 동적 또는 입력 텍스트에만 사용할 수 있으며 정적 텍스트에 사용하면 무시됩니다. 기본값은 false입니다.

참고: Flash Professional CS5에서 시작하는 경우 글꼴 포함은 텍스트 객체 수준이 아닌 문서 수준에서 제어됩니다. text.embedVariantGlyphs 속성 대신 267페이지의 “fontItem.embedVariantGlyphs” 속성을 사용하십시오. Flash Professional CS5에서는 text.embedVariantGlyphs 속성이 아무런 영향도 미치지 않습니다. Flash에서 항상 TLF 텍스트에 변형 글리프를 포함하고 클래식 텍스트에는 변형 글리프를 포함하지 않기 때문입니다.

예제

다음 예제는 선택한 Text 객체에 변형 글리프가 포함될 수 있도록 합니다.

```
fl.getDocumentDOM().selection[0].embedVariantGlyphs = true;
```

참고 사항

[fontItem.embedVariantGlyphs](#)

text.fontRenderingMode

지원 버전

Flash 8

사용법

```
text.fontRenderingMode
```

설명

속성, 텍스트에 대해 렌더링 모드를 지정하는 문자열입니다. 이 속성은 스테이지 및 Flash Player에 텍스트가 표시되는 방법에 영향을 줍니다. 사용할 수 있는 값은 다음 표에 설명되어 있습니다.

속성 값	텍스트 렌더링 방법
device	장치 글꼴로 텍스트를 렌더링합니다.
bitmap	앨리어싱된 텍스트를 비트맵 또는 픽셀 글꼴로 렌더링합니다.
standard	Flash MX 2004에서 사용되는 표준 앨리어싱 방법을 사용하여 텍스트를 렌더링합니다. 애니메이션 텍스트, 매우 큰 텍스트 또는 기울어진 텍스트에 사용하기 가장 적합한 설정입니다.
advanced	특히 작은 텍스트에 대해 보다 나은 앨리어싱을 제공하고 가독성을 향상시키는 Flash 8에서 구현되는 고급 앨리어싱 글꼴 렌더링 기술을 사용하여 텍스트를 렌더링합니다.
customThicknessSharpness	Flash 8에서 구현되는 고급 앨리어싱 글꼴 렌더링 기술을 사용하여 텍스트의 선명도 및 두께에 대한 사용자 정의 설정을 지정할 수 있습니다.

예제

다음 예제에서는 `customThicknessSharpness` 값을 사용하여 텍스트의 선명도 및 두께를 지정하는 방법을 보여 줍니다.

```
fl.getDocumentDOM().setElementProperty("fontRenderingMode", "customThicknessSharpness");  
fl.getDocumentDOM().setElementProperty("antiAliasSharpness", 400);  
fl.getDocumentDOM().setElementProperty("antiAliasThickness", -200);
```

참고 사항

[text.antiAliasSharpness](#), [text.antiAliasThickness](#)

text.getTextAttr()

지원 버전

Flash MX 2004

사용법

```
text.getTextAttr(attrName [, startIndex [, endIndex]])
```

매개 변수

attrName 반환할 `TextAttrs` 객체 속성의 이름을 지정하는 문자열. **attrName**에 사용 가능한 값 목록을 보려면 [TextAttrs 객체](#)의 속성 요약을 참조하십시오.

startIndex 첫 번째 문자의 인덱스를 나타내는 정수. 이 매개 변수는 선택 사항입니다.

endIndex **startIndex**부터 **endIndex**(포함 안 됨) 사이의 텍스트 범위 끝을 지정하는 정수. 이 매개 변수는 선택 사항입니다.

반환값

attrName 매개 변수에 지정된 특성 값

설명

메서드, 선택적 매개 변수 **startIndex** 및 **endIndex**에 의해 식별된 텍스트의 **attrName** 매개 변수에 지정된 특성을 가져옵니다. 특성이 지정된 범위와 일치하지 않으면 `undefined`가 반환됩니다. 선택적 매개 변수 **startIndex**와 **endIndex**를 생략하면 전체 텍스트 범위가 사용됩니다. **startIndex**만 지정하면 해당 위치에 있는 문자 하나만 사용됩니다. **startIndex**와 **endIndex**를 둘 다 지정하면 **startIndex**부터 **endIndex**(포함 안 됨) 사이의 문자가 사용됩니다.

예제

다음 예제에서는 현재 선택한 텍스트 필드의 글꼴 크기를 가져와서 보여 줍니다.

```
var TheTextSize = fl.getDocumentDOM().selection[0].getTextAttr("size");  
fl.trace(TheTextSize);
```

다음 예제에서는 선택한 텍스트 필드의 텍스트 채움 색상을 가져옵니다.

```
var TheFill = fl.getDocumentDOM().selection[0].getTextAttr("fillColor");  
fl.trace(TheFill);
```

다음 예제에서는 세 번째 문자의 크기를 가져옵니다.

```
var Char3 = fl.getDocumentDOM().selection[0].getTextAttr("size", 2);  
fl.trace(Char3);
```

다음 예제에서는 선택한 텍스트 필드에서 세 번째에서 여덟 번째 사이에 있는 문자의 색상을 가져옵니다.

```
fl.getDocumentDOM().selection[0].getTextAttr("fillColor", 2, 8);
```

text.getTextString()

지원 버전

Flash MX 2004

사용법

```
text.getTextString([startIndex [, endIndex]])
```

매개 변수

startIndex 첫 번째 문자의 0부터 시작하는 인덱스를 지정하는 정수. 이 매개 변수는 선택 사항입니다.

endIndex **startIndex**부터 **endIndex**(포함 안 됨) 사이의 텍스트 범위 끝을 지정하는 정수. 이 매개 변수는 선택 사항입니다.

반환값

지정된 범위의 텍스트 문자열

설명

메서드, 지정된 텍스트 범위를 가져옵니다. 선택적 매개 변수 **startIndex**와 **endIndex**를 생략하면 전체 텍스트 문자열이 반환됩니다. **startIndex**만 지정하면 이 인덱스 위치부터 필드 끝 사이의 문자열이 반환됩니다. **startIndex**와 **endIndex**를 둘 다 지정하면 **startIndex**부터 **endIndex**(포함 안 됨) 사이의 문자열이 반환됩니다.

예제

다음 예제에서는 선택한 텍스트 필드에서 5번째 이후의 모든 문자를 가져옵니다.

```
var myText = fl.getDocumentDOM().selection[0].getTextString(4);  
fl.trace(myText);
```

다음 예제에서는 선택한 텍스트 필드에서 네 번째에서 아홉 번째 사이에 있는 문자를 가져옵니다.

```
var myText = fl.getDocumentDOM().selection[0].getTextString(3, 9);  
fl.trace(myText);
```

text.length

지원 버전

Flash MX 2004

사용법

```
text.length
```

설명

읽기 전용 속성, Text 객체의 문자 수를 나타내는 정수입니다.

예제

다음 예제에서는 선택한 텍스트의 문자 수를 반환합니다.

```
var textLength = fl.getDocumentDOM().selection[0].length;
```

text.lineType

지원 버전

Flash MX 2004

사용법

`text.lineType`

설명

속성, 행 유형을 설정하는 문자열입니다. 사용할 수 있는 값에는 "single line", "multiline", "multiline no wrap" 및 "password"가 있습니다.

이 속성은 동적 또는 입력 텍스트에만 사용할 수 있으며 정적 텍스트에 사용하면 경고가 표시됩니다. "password" 값은 입력 텍스트에만 사용할 수 있습니다.

예제

다음 예제에서는 lineType 속성 값을 multiline no wrap으로 설정합니다.

```
fl.getDocumentDOM().selection[0].lineType = "multiline no wrap";
```

text.maxCharacters

지원 버전

Flash MX 2004

사용법

`text.maxCharacters`

설명

속성, 이 Text 객체에 입력할 수 있는 최대 문자 수를 지정하는 정수입니다.

이 속성은 입력 텍스트에만 사용할 수 있으며 다른 텍스트 유형에 사용하면 경고가 표시됩니다.

예제

다음 예제에서는 maxCharacters 속성 값을 30으로 설정합니다.

```
fl.getDocumentDOM().selection[0].maxCharacters = 30;
```

text.orientation

지원 버전

Flash MX 2004

사용법

`text.orientation`

설명

속성, 텍스트 필드의 방향을 지정하는 문자열입니다. 사용할 수 있는 값에는 "horizontal", "vertical left to right" 및 "vertical right to left"가 있습니다.

이 속성은 정적 텍스트에만 사용할 수 있으며 다른 텍스트 유형에 사용하면 경고가 표시됩니다.

예제

다음 예제에서는 방향 속성을 vertical right to left로 설정합니다.

```
fl.getDocumentDOM().selection[0].orientation = "vertical right to left";
```

text.renderAsHTML

지원 버전

Flash MX 2004

사용법

```
text.renderAsHTML
```

설명

속성, 부울 값입니다. true로 설정하면 텍스트가 HTML로 그려지고 포함된 HTML 태그가 해석됩니다.

이 속성은 동적 또는 입력 텍스트에만 사용할 수 있으며 다른 텍스트 유형에 사용하면 경고가 표시됩니다.

예제

다음 예제에서는 renderAsHTML 속성을 true로 설정합니다.

```
fl.getDocumentDOM().selection[0].renderAsHTML = true;
```

text.scrollable

지원 버전

Flash MX 2004

사용법

```
text.scrollable
```

설명

속성, 부울 값입니다. 이 값이 true이면 텍스트를 스크롤할 수 있습니다.

이 속성은 동적 또는 입력 텍스트에만 사용할 수 있으며 정적 텍스트에 사용하면 경고가 표시됩니다.

예제

다음 예제에서는 scrollable 속성을 false로 설정합니다.

```
fl.getDocumentDOM().selection[0].scrollable = false;
```

text.selectable

지원 버전

Flash MX 2004

사용법

```
text.selectable
```

설명

속성, 부울 값입니다. 이 값이 true이면 텍스트를 선택할 수 있습니다.

입력 텍스트는 항상 선택할 수 있습니다. 이 속성을 false로 설정하고 입력 텍스트에 사용하면 경고가 표시됩니다.

예제

다음 예제에서는 selectable 속성을 true로 설정합니다.

```
fl.getDocumentDOM().selection[0].selectable = true;
```

text.selectionEnd

지원 버전

Flash MX 2004

사용법

```
text.selectionEnd
```

설명

속성, 텍스트 세부 선택의 끝 위치를 지정하는 정수(0부터 시작)입니다. 자세한 내용은 [text.selectionStart](#)를 참조하십시오.

text.selectionStart

지원 버전

Flash MX 2004

사용법

```
text.selectionStart
```

설명

속성, 텍스트 세부 선택의 시작 위치를 지정하는 정수(0부터 시작)입니다. 이 속성을 [text.selectionEnd](#)와 함께 사용하면 문자 범위를 선택할 수 있습니다. [text.selectionEnd](#) 앞까지 문자가 선택됩니다. [text.selectionEnd](#)를 참조하십시오.

- 삽입점이 있거나 선택한 영역이 없으면 [text.selectionEnd](#)와 [text.selectionStart](#)가 같습니다.
- [text.selectionStart](#)가 [text.selectionEnd](#)보다 큰 값으로 설정되면 [text.selectionEnd](#)가 [text.selectionStart](#)로 설정되고 텍스트가 선택되지 않습니다.

예제

다음 예제에서는 텍스트 세부 선택의 시작 위치를 여섯 번째 문자로 설정합니다.

```
fl.getDocumentDOM().selection[0].selectionStart = 5;
```

다음 예제에서는 My name is Barbara라는 텍스트가 포함된 텍스트 필드에서 Barbara라는 문자를 선택하고 서식을 굵게 녹색으로 적용합니다.

```
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()

지원 버전

Flash MX 2004

사용법

```
text.setTextAttr(attrName, attrValue [, startIndex [, endIndex]])
```

매개 변수

attrName 변경할 TextAttrs 객체 속성의 이름을 지정하는 문자열

attrValue TextAttrs 객체 속성의 값

attrName 및 **attrValue**에 사용 가능한 값 목록을 보려면 [TextAttrs 객체](#)의 속성 요약을 참조하십시오.

startIndex 배열에 있는 첫 번째 문자의 0부터 시작하는 인덱스인 정수. 이 매개 변수는 선택 사항입니다.

endIndex **startIndex**부터 **endIndex**(포함 안 됨) 사이의 선택된 텍스트 문자열에서 끝점의 인덱스를 지정하는 정수. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, **startIndex** 및 **endIndex**에 의해 식별된 텍스트와 연관된 **attrName** 매개 변수에 지정된 특정 특성을 **attrValue**에 지정된 값으로 설정합니다. 이 메서드는 TextRun 요소([TextRun 객체](#) 참조)를 포함할 수 있거나 기존 TextRun 요소에 포함된 특성을 변경하는 데 사용할 수 있습니다. 이 메서드를 사용하면 이 객체의 text.textRuns 배열 내에서 TextRun 요소의 위치와 개수를 변경할 수 있습니다([text.textRuns](#) 참조).

선택적 매개 변수를 생략하면 전체 Text 객체의 문자 범위가 사용됩니다. **startIndex**만 지정하면 해당 위치에 있는 문자 하나만 사용됩니다. **startIndex**와 **endIndex**를 둘 다 지정하면 **startIndex**부터 **endIndex**(포함 안 됨) 사이의 문자가 사용됩니다.

예제

다음 예제에서는 선택한 텍스트 필드를 기울임체로 설정합니다.

```
fl.getDocumentDOM().selection[0].setTextAttr("italic", true);
```

다음 예제에서는 세 번째 문자의 크기를 10으로 설정합니다.

```
fl.getDocumentDOM().selection[0].setTextAttr("size", 10, 2);
```

다음 예제에서는 선택한 텍스트에서 세 번째에서 여덟 번째 사이에 있는 문자의 색상을 빨간색으로 설정합니다.

```
fl.getDocumentDOM().selection[0].setTextAttr("fillColor", 0xff0000, 2, 8);
```

text.setTextString()

지원 버전
Flash MX 2004

사용법

```
text.setTextString(text [, startIndex [, endIndex]])
```

매개 변수

text 이 Text 객체에 삽입할 문자로 구성된 문자열

startIndex 텍스트를 삽입할 문자열에 있는 문자열의 0부터 시작하는 인덱스를 지정하는 정수. 이 매개 변수는 선택 사항입니다.

endIndex 선택한 텍스트 문자열의 끝 위치에 대한 인덱스를 지정하는 정수. 새 텍스트는 **startIndex**부터 **endIndex**(포함 안 됨) 사이의 텍스트를 덮어씁니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

속성, 이 Text 객체 내의 텍스트 문자열을 변경합니다. 선택적 매개 변수를 생략하면 전체 Text 객체가 바뀝니다. **startIndex**만 지정하면 **startIndex** 위치에 지정된 문자열이 삽입됩니다. **startIndex**와 **endIndex**를 둘 다 지정하면 **startIndex**부터 **endIndex**(포함 안 됨) 사이의 텍스트 세그먼트가 지정된 문자열로 바뀝니다.

예제

다음 예제에서는 선택한 텍스트 필드에 **this is a string**이라는 문자열을 할당합니다.

```
fl.getDocumentDOM().selection[0].setTextString("this is a string");
```

다음 예제에서는 선택한 텍스트 필드에서 다섯 번째 문자 위치에 **abc**라는 문자열을 삽입합니다.

```
fl.getDocumentDOM().selection[0].setTextString("01234567890");  
fl.getDocumentDOM().selection[0].setTextString("abc", 4);  
// text field is now "0123abc4567890"
```

다음 예제에서는 선택한 텍스트 문자열에서 세 번째에서 여덟 번째 사이에 있는 문자를 **abcdefghij**라는 문자열로 바꿉니다. 즉, **startIndex**와 **endIndex** 사이의 문자를 덮어씁니다. **endIndex** 뒤에 있는 문자는 삽입된 문자열 뒤에 옵니다.

```
fl.getDocumentDOM().selection[0].setTextString("01234567890");  
fl.getDocumentDOM().selection[0].setTextString("abcdefghij", 2, 8);  
// text field is now "01abcdefghij890"
```

text.shortcut

지원 버전
Flash MX 2004

사용법

```
text.shortcut
```


설명

속성, [액세스 가능성] 패널의 [단축키] 필드에 해당하는 문자열입니다. 이 단축키는 화면 판독기에서 읽습니다. 동적 텍스트에는 이 속성을 사용할 수 없습니다.

예제

다음 예제에서는 선택한 객체의 단축키를 가져와서 해당 값을 보여 줍니다.

```
var theShortcut = fl.getDocumentDOM().selection[0].shortcut;  
fl.trace(theShortcut);
```

다음 예제에서는 현재 선택한 객체의 단축키를 설정합니다.

```
fl.getDocumentDOM().selection[0].shortcut = "Ctrl+i";
```

text.silent

지원 버전

Flash MX 2004

사용법

`text.silent`

설명

속성, 객체에 액세스할 수 있는지 여부를 지정하는 부울 값입니다. 이 속성은 [액세스 가능성] 패널에 있는 [객체를 액세스 가능하게 만들기] 설정의 역논리에 해당합니다. 즉, `silent`가 `true`이면 [객체를 액세스 가능하게 만들기]가 선택 취소되고 `false`이면 [객체를 액세스 가능하게 만들기]가 선택됩니다.

예제

다음 예제에서는 객체에 액세스할 수 있는지 여부를 확인합니다. `false` 값은 객체에 액세스할 수 있음을 의미합니다.

```
var isSilent = fl.getDocumentDOM().selection[0].silent;
```

다음 예제에서는 객체에 액세스할 수 있도록 설정합니다.

```
fl.getDocumentDOM().selection[0].silent = false;
```

text.tabIndex

지원 버전

Flash MX 2004

사용법

`text.tabIndex`

설명

속성, [액세스 가능성] 패널의 [탭 인덱스] 필드에 해당하는 정수입니다. 이 값을 사용하면 사용자가 Tab 키를 누를 때 객체가 액세스되는 순서를 지정하는 탭 순서를 확인할 수 있습니다.

예제

다음 예제에서는 현재 선택한 객체의 `tabIndex`를 가져옵니다.

```
var theTabIndex = fl.getDocumentDOM().selection[0].tabIndex;
```

다음 예제에서는 현재 선택한 객체의 `tabIndex`를 설정합니다.

```
fl.getDocumentDOM().selection[0].tabIndex = 1;
```

text.textRuns

지원 버전
Flash MX 2004

사용법
`text.textRuns`

설명
읽기 전용 속성, `TextRun` 객체의 배열([TextRun 객체](#) 참조)입니다.

예제
다음 예제에서는 `textRuns` 속성의 값을 `myTextRuns` 변수에 저장합니다.

```
var myTextRuns = fl.getDocumentDOM().selection[0].textRuns;
```

text.textType

지원 버전
Flash MX 2004

사용법
`text.textType`

설명
속성, 텍스트 필드의 유형을 지정하는 문자열입니다. 사용할 수 있는 값에는 "static", "dynamic" 및 "input"이 있습니다.

예제
다음 예제에서는 `textType` 속성을 `input`으로 설정합니다.

```
fl.getDocumentDOM().selection[0].textType = "input";
```

text.useDeviceFonts

지원 버전
Flash MX 2004

사용법
`text.useDeviceFonts`

설명

속성, 부울 값입니다. true로 설정하면 텍스트를 그리는 데 장치 글꼴이 사용됩니다.

예제

다음 예제에서는 Flash에서 텍스트를 그릴 때 장치 글꼴을 사용하도록 합니다.

```
fl.getDocumentDOM().selection[0].useDeviceFonts = true;
```

text.variableName

지원 버전

Flash MX 2004

사용법

```
text.variableName
```

설명

속성, Text 객체와 연관된 변수의 이름이 포함된 문자열입니다. 이 속성은 동적 또는 입력 텍스트에만 사용할 수 있으며 다른 텍스트 유형에 사용하면 경고가 표시됩니다.

이 속성은 ActionScript 1.0 및 ActionScript 2.0에서만 지원됩니다.

예제

다음 예제에서는 선택한 텍스트 상자의 변수 이름을 firstName으로 설정합니다.

```
fl.getDocumentDOM().selection[0].variableName = "firstName";
```

43장: TextAttrs 객체

지원 버전

Flash MX 2004

설명

TextAttrs 객체에는 세부 선택에 적용할 수 있는 텍스트의 모든 속성이 포함됩니다. 이 객체는 TextRun 객체(`textRun.textAttrs`)의 속성입니다.

속성 요약

TextAttrs 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>textAttrs.aliasText</code>	작은 크기의 텍스트를 읽기 쉽게 하기 위해 최적화된 메서드를 사용하여 텍스트를 그릴지 여부를 지정하는 부울 값
<code>textAttrs.alignment</code>	단락 정렬을 지정하는 문자열. 사용할 수 있는 값에는 "left", "center", "right" 및 "justify"가 있습니다.
<code>textAttrs.autoKern</code>	글꼴에 내장된 문자쌍 자간 정보를 사용하여 텍스트의 자간을 조절할 것인지 여부(true 또는 false)를 결정하는 부울 값
<code>textAttrs.bold</code>	부울 값. true로 설정하면 텍스트가 굵은체로 표시됩니다.
<code>textAttrs.characterPosition</code>	텍스트의 기준선을 결정하는 문자열
<code>textAttrs.characterSpacing</code>	이제는 이 속성 대신 <code>textAttrs.letterSpacing</code> 이 사용되고 있습니다. 문자 사이의 여백을 나타내는 정수
<code>textAttrs.face</code>	"Arial"과 같은 글꼴의 이름을 나타내는 문자열
<code>textAttrs.fillColor</code>	채움 색상을 나타내는 문자열, 16진수 값 또는 정수
<code>textAttrs.indent</code>	단락 들여쓰기를 지정하는 정수
<code>textAttrs.italic</code>	부울 값. true로 설정하면 텍스트가 기울임체로 표시됩니다.
<code>textAttrs.leftMargin</code>	단락의 왼쪽 여백을 지정하는 정수
<code>textAttrs.letterSpacing</code>	문자 사이의 여백을 나타내는 정수
<code>textAttrs.lineSpacing</code>	단락의 행 간격(leading)을 지정하는 정수
<code>textAttrs.rightMargin</code>	단락의 오른쪽 여백을 지정하는 정수
<code>textAttrs.rotation</code>	부울 값. true로 설정하면 문자가 90° 회전합니다. 기본값은 false입니다.
<code>textAttrs.size</code>	글꼴의 크기를 지정하는 정수
<code>textAttrs.target</code>	텍스트 필드의 target 속성을 나타내는 문자열
<code>textAttrs.url</code>	텍스트 필드의 URL 속성을 나타내는 문자열

textAttrs.aliasText

지원 버전
Flash MX 2004

사용법
`textAttrs.aliasText`

설명
속성, 작은 크기의 텍스트를 읽기 쉽게 하기 위해 최적화된 메서드를 사용하여 텍스트를 그릴지 여부를 지정하는 부울 값입니다.

예제
다음 예제에서는 현재 선택한 텍스트 필드에 있는 모든 텍스트의 `aliasText` 속성을 `true`로 설정합니다.

```
fl.getDocumentDOM().setElementTextAttr('aliasText', true);
```

textAttrs.alignment

지원 버전
Flash MX 2004

사용법
`textAttrs.alignment`

설명
속성, 단락 정렬을 지정하는 문자열입니다. 사용할 수 있는 값에는 "left", "center", "right" 및 "justify"가 있습니다.

예제
다음 예제에서는 인덱스 0부터 인덱스 3(포함 안 됨) 사이의 문자가 포함된 단락을 양쪽으로 정렬하도록 설정합니다. 이 설정은 지정된 범위에 들어 있지 않지만 같은 단락에 있는 다른 문자에도 적용됩니다.

```
fl.getDocumentDOM().setTextSelection(0, 3);  
fl.getDocumentDOM().setElementTextAttr("alignment", "justify");
```

textAttrs.autoKern

지원 버전
Flash MX 2004

사용법
`textAttrs.autoKern`

설명
속성, 텍스트의 자간을 조절할 때 글꼴에 내장된 문자쌍 자간 정보를 사용할 것인지 여부(`true` 또는 `false`)를 지정하는 부울 값입니다.

예제

다음 예제에서는 인덱스 2부터 인덱스 6(포함 안 됨) 사이의 문자를 선택하고 `autoKern` 속성을 `true`로 설정합니다.

```
f1.getDocumentDOM().setTextSelection(3, 6);  
f1.getDocumentDOM().setElementTextAttr('autoKern', true);
```

textAttrs.bold

지원 버전

Flash MX 2004

사용법

`textAttrs.bold`

설명

속성, 부울 값입니다. `true`로 설정하면 텍스트가 굵은체로 표시됩니다.

예제

다음 예제에서는 선택한 Text 객체의 첫 번째 문자를 선택하고 `bold` 속성을 `true`로 설정합니다.

```
f1.getDocumentDOM().setTextSelection(0, 1);  
f1.getDocumentDOM().setElementTextAttr('bold', true);
```

textAttrs.characterPosition

지원 버전

Flash MX 2004

사용법

`textAttrs.characterPosition`

설명

속성, 텍스트의 기준선을 결정하는 문자열입니다. 사용할 수 있는 값에는 "normal", "subscript" 및 "superscript"가 있습니다. 이 속성은 정적 텍스트에만 적용됩니다.

예제

다음 예제에서는 선택한 텍스트 필드에서 인덱스 2부터 인덱스 6(포함 안 됨) 사이에 있는 문자를 선택하고 `characterPosition` 속성을 `subscript`로 설정합니다.

```
f1.getDocumentDOM().setTextSelection(2, 6);  
f1.getDocumentDOM().setElementTextAttr("characterPosition", "subscript");
```

textAttrs.characterSpacing

지원 버전

Flash MX 2004 Flash 8에서는 이 속성 대신 `textAttrs.letterSpacing`이 사용되고 있습니다.

사용법

`textAttrs.characterSpacing`

설명

속성, 문자 사이의 간격을 나타내는 정수입니다. 사용할 수 있는 값은 -60부터 60까지입니다.

이 속성은 정적 텍스트에만 적용되며 다른 텍스트 유형에 사용하면 경고가 표시됩니다.

예제

다음 예제에서는 선택한 텍스트 필드의 문자 간격을 10으로 설정합니다.

```
f1.getDocumentDOM().setElementTextAttr("characterSpacing", 10);
```

textAttrs.face

지원 버전

Flash MX 2004

사용법

`textAttrs.face`

설명

속성, "Arial"과 같은 글꼴의 이름을 나타내는 문자열입니다.

예제

다음 예제에서는 선택한 텍스트 필드에서 인덱스 2부터 인덱스 8(포함 안 됨) 사이에 있는 문자의 글꼴을 Arial로 설정합니다.

```
f1.getDocumentDOM().selection[0].setTextAttr("face", "Arial", 2, 8);
```

textAttrs.fillColor

지원 버전

Flash MX 2004

사용법

`textAttrs.fillColor`

설명

속성, 형식이 다음 중 하나인 채우기 색상입니다.

- "#RRGGBB" 또는 "#RRGGBBAA" 형식의 문자열
- 0xRRGGBB 형식의 16진수
- 16진수 숫자와 같은 십진수를 나타내는 정수

예제

다음 예제에서는 선택한 텍스트 필드에서 인덱스 2부터 인덱스 8(포함 안 됨) 사이에 있는 문자의 색상을 빨간색으로 설정합니다.

```
fl.getDocumentDOM().selection[0].setTextAttr("fillColor", 0xff0000, 2, 8);
```

textAttrs.indent

지원 버전
Flash MX 2004

사용법
`textAttrs.indent`

설명
속성, 단락 들여쓰기를 지정하는 정수입니다. 사용할 수 있는 값은 -720부터 720까지입니다.

예제
다음 예제에서는 선택한 텍스트 필드에서 인덱스 2부터 인덱스 8(포함 안 됨) 사이에 있는 문자의 들여쓰기를 100으로 설정합니다. 이 설정은 지정된 범위에 들어 있지 않지만 같은 단락에 있는 다른 문자에도 적용됩니다.

```
fl.getDocumentDOM().selection[0].setTextAttr("indent", 100, 2, 8);
```

textAttrs.italic

지원 버전
Flash MX 2004

사용법
`textAttrs.italic`

설명
속성, 부울 값입니다. `true`로 설정하면 텍스트가 기울임체로 표시됩니다.

예제
다음 예제에서는 선택한 텍스트 필드를 기울임체로 설정합니다.

```
fl.getDocumentDOM().selection[0].setTextAttr("italic", true);
```

textAttrs.leftMargin

지원 버전
Flash MX 2004

사용법
`textAttrs.leftMargin`

설명
속성, 단락의 왼쪽 여백을 지정하는 정수입니다. 사용할 수 있는 값은 0부터 720까지입니다.

예제

다음 예제에서는 선택한 텍스트 필드에서 인덱스 2부터 인덱스 8(포함 안 됨) 사이에 있는 문자의 `leftMargin` 속성을 100으로 설정합니다. 이 설정은 지정된 범위에 들어 있지 않지만 같은 단락에 있는 다른 문자에도 적용됩니다.

```
fl.getDocumentDOM().selection[0].setTextAttr("leftMargin", 100, 2, 8);
```

textAttrs.letterSpacing

지원 버전

Flash 8

사용법

`textAttrs.letterSpacing`

설명

속성, 문자 사이의 간격을 나타내는 정수입니다. 사용할 수 있는 값은 -60부터 60까지입니다.

이 속성은 정적 텍스트에만 적용되며 다른 텍스트 유형에 사용하면 경고가 표시됩니다.

예제

다음 코드에서는 인덱스 0부터 인덱스 10(포함 안 됨) 사이의 문자를 선택하고 문자 간격을 60으로 설정합니다.

```
fl.getDocumentDOM().setTextSelection(0, 10);  
fl.getDocumentDOM().setElementTextAttr("letterSpacing", 60);
```

textAttrs.lineSpacing

지원 버전

Flash MX 2004

사용법

`textAttrs.lineSpacing`

설명

속성, 단락의 행 간격(행간)을 지정하는 정수입니다. 사용할 수 있는 값은 -360부터 720까지입니다.

예제

다음 예제에서는 선택한 텍스트 필드의 `lineSpacing` 속성을 100으로 설정합니다.

```
fl.getDocumentDOM().selection[0].setTextAttr("lineSpacing", 100);
```

textAttrs.rightMargin

지원 버전

Flash MX 2004

사용법

`textAttrs.rightMargin`

설명

속성, 단락의 오른쪽 여백을 지정하는 정수입니다. 사용할 수 있는 값은 0부터 720까지입니다.

예제

다음 예제에서는 선택한 텍스트 필드에서 인덱스 2부터 인덱스 8(포함 안 됨) 사이에 있는 문자의 `rightMargin` 속성을 100으로 설정합니다. 이 설정은 지정된 범위에 들어 있지 않지만 같은 단락에 있는 다른 문자에도 적용됩니다.

```
f1.getDocumentDOM().selection[0].setTextAttr("rightMargin", 100, 2, 8);
```

textAttrs.rotation

지원 버전

Flash MX 2004

사용법

`textAttrs.rotation`

설명

속성, 부울 값입니다. `true`로 설정하면 문자가 90° 회전합니다. 기본값은 `false`입니다. 이 속성은 수직 방향의 정적 텍스트에만 적용되며 다른 텍스트 유형에 사용하면 경고가 표시됩니다.

예제

다음 예제에서는 선택한 텍스트 필드의 회전을 `true`로 설정합니다.

```
f1.getDocumentDOM().setElementTextAttr("rotation", true);
```

textAttrs.size

지원 버전

Flash MX 2004

사용법

`textAttrs.size`

설명

속성, 글꼴의 크기를 지정하는 정수입니다.

예제

다음 예제에서는 인덱스 2에 있는 문자의 크기를 가져와서 [출력] 패널에 표시합니다.

```
f1.outputPanel.trace(f1.getDocumentDOM().selection[0].getTextAttr("size", 2));
```

textAttrs.target

지원 버전

Flash MX 2004

사용법

```
textAttrs.target
```

설명

속성, 텍스트 필드의 **target** 속성을 나타내는 문자열입니다. 이 속성은 정적 텍스트에만 적용됩니다.

예제

다음 예제에서는 현재 장면에서 있는 최상위 레이어의 첫 번째 프레임에 있는 텍스트 필드의 **target** 속성을 가져와서 [출력] 패널에 표시합니다.

```
fl.outputPanel.trace(fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].getTextAttr("target"));
```

textAttrs.url

지원 버전

Flash MX 2004

사용법

```
textAttrs.url
```

설명

속성, 텍스트 필드의 URL 속성을 나타내는 문자열입니다. 이 속성은 정적 텍스트에만 적용됩니다.

예제

다음 예제에서는 선택한 텍스트 필드의 URL을 <http://www.adobe.com>으로 설정합니다.

```
fl.getDocumentDOM().setElementTextAttr("url", "http://www.adobe.com");
```

44장: TextRun 객체

지원 버전

Flash MX 2004

설명

TextRun 객체는 [TextAttrs](#) 객체의 모든 속성과 일치하는 특성을 가진 문자의 흐름을 나타냅니다. 이 객체는 [Text](#) 객체 ([text.textRuns](#))의 속성입니다.

속성 요약

Text 객체 속성 외에도 TextRun 객체는 다음 속성을 제공합니다.

속성	설명
textRun.characters	TextRun 객체에 포함되어 있는 텍스트를 나타내는 문자열
textRun.textAttrs	텍스트의 흐름 특성이 포함된 TextAttrs 객체

textRun.textAttrs

지원 버전

Flash MX 2004

사용법

```
textRun.textAttrs
```

설명

속성, 텍스트의 흐름 특성이 포함된 [TextAttrs](#) 객체입니다.

예제

다음 예제에서는 선택한 텍스트 필드에 있는 문자의 첫 번째 흐름 속성을 [출력] 패널에 표시합니다.

```
var curTextAttrs = fl.getDocumentDOM().selection[0].textRuns[0].textAttrs;
for (var prop in curTextAttrs) {
    fl.trace(prop + " = " + curTextAttrs[prop]);
}
```

textRun.characters

지원 버전

Flash MX 2004

사용법

```
textRun.characters
```

설명

속성, TextRun 객체에 포함되어 있는 텍스트입니다.

예제

다음 예제에서는 선택한 텍스트 필드에 있는 문자의 첫 번째 흐름을 구성하는 문자를 [출력] 패널에 표시합니다.

```
f1.trace(f1.getDocumentDOM().selection[0].textRuns[0].characters);
```

45장: Timeline 객체

지원 버전

Flash MX 2004

설명

Timeline 객체는 `fl.getDocumentDOM().getTimeline()`을 사용하여 액세스할 수 있는 현재 문서의 Flash 타임라인을 나타냅니다. 이 메서드는 편집 중인 현재 장면이나 심볼의 타임라인을 반환합니다.

장면에 대해 작업할 때 각 장면의 타임라인에는 인덱스 값이 지정되어 있기 때문에 `fl.getDocumentDOM().timelines[i]`를 통해 현재 문서의 타임라인에 액세스할 수 있습니다. 여기서 `i`는 타임라인 값의 인덱스입니다.

Timeline 객체의 메서드와 속성을 사용하여 프레임에 대해 작업할 때는 프레임 값이 타임라인에 있는 프레임 시퀀스의 실제 프레임 번호가 아닌 0부터 시작하는 인덱스라는 점에 유의해야 합니다. 즉, 첫 번째 프레임의 프레임 인덱스가 0입니다.

메서드 요약

Timeline 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
<code>timeline.addMotionGuide()</code>	현재 레이어 위에 모션 안내선 레이어를 추가하고 현재 레이어를 새로 추가한 안내선 레이어에 첨부합니다.
<code>timeline.addNewLayer()</code>	새 레이어를 문서에 추가하고 현재 레이어로 만듭니다.
<code>timeline.clearFrames()</code>	현재 레이어에 있는 하나 또는 여러 프레임의 내용을 모두 삭제합니다.
<code>timeline.clearKeyframes()</code>	현재 레이어에서 키프레임을 일반 프레임으로 변환하고 해당 내용을 삭제합니다.
<code>timeline.convertToBlankKeyframes()</code>	현재 레이어에서 프레임을 빈 키프레임으로 변환합니다.
<code>timeline.convertToKeyframes()</code>	현재 레이어에서 여러 프레임을 키프레임으로 변환하거나 지정된 프레임이 없는 경우 선택한 영역을 키프레임으로 변환합니다.
<code>timeline.copyFrames()</code>	현재 레이어에 있는 여러 프레임을 클립보드에 복사합니다.
<code>timeline.copyLayers()</code>	타임라인 레이어 범위를 클립보드에 복사합니다.
<code>timeline.copyMotion()</code>	다른 프레임에 적용할 수 있도록 모션 트윈 또는 프레임별 애니메이션에서 선택한 프레임의 모션을 복사합니다.
<code>timeline.copyMotionAsAS3()</code>	모션 트윈 또는 프레임별 애니메이션에서 선택한 프레임의 모션을 클립보드에 ActionScript 3.0 코드로 복사합니다.
<code>timeline.createMotionObject()</code>	지정된 시작 및 종료 프레임에서 새 모션 객체를 만듭니다.
<code>timeline.createMotionTween()</code>	현재 레이어에 선택되어 있는 각 키프레임의 <code>frame.tweenType</code> 속성을 <code>motion</code> 으로 설정하고 필요한 경우 각 프레임의 내용을 단일 심볼 인스턴스로 변환합니다.
<code>timeline.cutFrames()</code>	타임라인의 현재 레이어에서 여러 프레임을 잘라내어 클립보드에 저장합니다.
<code>timeline.cutLayers()</code>	타임라인 레이어 범위를 잘라내어 클립보드에 저장합니다.
<code>timeline.deleteLayer()</code>	레이어를 삭제합니다.
<code>timeline.duplicateLayers()</code>	선택된 레이어 또는 지정된 레이어를 복제합니다.

메서드	설명
<code>timeline.expandFolder()</code>	지정된 폴더를 확장하거나 축소합니다.
<code>timeline.findLayerIndex()</code>	주어진 이름으로 레이어 인덱스의 배열을 찾습니다.
<code>timeline.getFrameProperty()</code>	선택한 프레임의 특정 속성 값을 가져옵니다.
<code>timeline.getGuidelines()</code>	타임라인 가로 및 세로 안내선의 현재 위치를 나타내는 XML 문자열을 반환합니다([보기] > [안내선] > [안내선 표시]).
<code>timeline.getLayerProperty()</code>	선택한 레이어의 특정 속성 값을 가져옵니다.
<code>timeline.getSelectedFrames()</code>	배열에서 현재 선택한 프레임을 가져옵니다.
<code>timeline.getSelectedLayers()</code>	현재 선택한 레이어의 0부터 시작하는 인덱스 값을 가져옵니다.
<code>timeline.insertBlankKeyframe()</code>	지정된 프레임 인덱스에 빈 키프레임을 삽입합니다. 인덱스가 지정되어 있지 않은 경우 재생 헤드/선택한 위치를 사용하여 빈 키프레임을 삽입합니다.
<code>timeline.insertFrames()</code>	주어진 프레임 번호에 지정된 수의 프레임을 삽입합니다.
<code>timeline.insertKeyframe()</code>	지정된 프레임에 키프레임을 삽입합니다.
<code>timeline.pasteFrames()</code>	클립보드에 있는 여러 프레임을 지정된 프레임에 붙여넣습니다.
<code>timeline.pasteLayers()</code>	복사한 레이어를 지정된 레이어 인덱스 위의 타임라인에 붙여넣습니다.
<code>timeline.pasteMotion()</code>	<code>timeline.copyMotion()</code> 으로 가져온 여러 모션 프레임을 Timeline에 붙여넣습니다.
<code>timeline.removeFrames()</code>	프레임을 삭제합니다.
<code>timeline.removeMotionObject()</code>	<code>timeline.createMotionObject()</code> 로 만든 모션 객체를 제거하고 프레임을 정적 프레임으로 변환합니다.
<code>timeline.reorderLayer()</code>	첫 번째로 지정된 레이어를 두 번째로 지정된 레이어의 앞 또는 뒤로 이동합니다.
<code>timeline.reverseFrames()</code>	여러 프레임을 반전시킵니다.
<code>timeline.selectAllFrames()</code>	현재 타임라인에 있는 모든 프레임을 선택합니다.
<code>timeline setFrameProperty()</code>	선택한 프레임의 Frame 객체 속성을 설정합니다.
<code>timeline.setGuidelines()</code>	타임라인의 안내선을 지정된 정보로 대체합니다.
<code>timeline.setLayerProperty()</code>	선택한 모든 레이어의 특정 속성을 지정된 값으로 설정합니다.
<code>timeline.setSelectedFrames()</code>	현재 레이어에서 여러 프레임을 선택하거나 선택한 프레임을 이 메서드에 전달된 선택 영역 배열로 설정합니다.
<code>timeline.setSelectedLayers()</code>	레이어를 선택된 상태로 설정하고 지정된 레이어를 현재 레이어로 만듭니다.
<code>timeline.showLayerMasking()</code>	마스크 및 마스크 처리된 레이어를 잠가 제작하는 동안 레이어 마스크를 표시합니다.
<code>timeline.startPlayback()</code>	현재 재생 중이 아니면 타임라인의 자동 재생을 시작합니다.
<code>timeline.stopPlayback()</code>	현재 재생 중이면 타임라인의 자동 재생을 중지합니다.

속성 요약

Timeline 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>timeline.currentFrame</code>	현재 재생 헤드 위치에 있는 프레임의 0부터 시작하는 인덱스
<code>timeline.currentLayer</code>	현재 활성화된 레이어의 0부터 시작하는 인덱스
<code>timeline.frameCount</code>	읽기 전용, 이 타임라인에 있는 가장 긴 레이어의 프레임 수를 나타내는 정수
<code>timeline.layerCount</code>	읽기 전용, 지정된 타임라인에 있는 레이어 수를 나타내는 정수
<code>timeline.layers</code>	읽기 전용, 레이어 객체의 배열
<code>timeline.libraryItem</code>	읽기 전용 속성, 타임라인이 장면에 속해 있는지 여부를 나타냄
<code>timeline.name</code>	현재 타임라인의 이름을 나타내는 문자열

timeline.addMotionGuide()

지원 버전
Flash MX 2004

사용법

```
timeline.addMotionGuide()
```

매개 변수

없음

반환값

새로 추가한 안내선 레이어의 0부터 시작하는 인덱스를 나타내는 정수. 현재 레이어의 유형이 "보통"이 아니면 -1이 반환됩니다.

설명

메서드, 현재 레이어 위에 모션 안내선 레이어를 추가하고, 현재 레이어의 유형을 "안내 처리"로 변환하여 현재 레이어를 새로 추가한 안내선 레이어에 첨부합니다.

이 메서드는 "보통" 유형의 레이어에서만 작동하고 "폴터", "마스크", "마스크 처리", "안내선" 또는 "안내 처리" 유형의 레이어에는 효과가 없습니다.

예제

다음 예제에서는 현재 레이어 위에 모션 안내선 레이어를 추가하고 현재 레이어를 Guided로 변환합니다.

```
fl.getDocumentDOM().getTimeline().addMotionGuide();
```

timeline.addNewLayer()

지원 버전
Flash MX 2004

사용법

```
timeline.addNewLayer([name] [, layerType [, bAddAbove]])
```


매개 변수

name 새 레이어의 이름을 지정하는 문자열입니다. 이 매개 변수를 생략하면 새 레이어에 기본 이름인 "레이어 n"이 지정됩니다. 여기서 **n**은 전체 레이어 수입니다. 이 매개 변수는 선택 사항입니다.

layerType 추가할 레이어의 유형을 지정하는 문자열입니다. 이 매개 변수를 생략하면 "보통" 유형의 레이어가 만들어집니다. 이 매개 변수는 선택 사항입니다. 사용할 수 있는 값에는 "normal", "guide", "guided", "mask", "masked" 및 "folder"가 있습니다.

bAddAbove true(기본값)로 설정하면 현재 레이어 위에 새 레이어가 추가되고 **false**로 설정하면 현재 레이어 아래에 레이어가 추가되는 부울 값. 이 매개 변수는 선택 사항입니다.

반환값

새로 추가한 레이어의 0부터 시작하는 인덱스의 정수 값

설명

메서드, 새 레이어를 문서에 추가하고 현재 레이어로 만듭니다.

예제

다음 예제에서는 기본 이름이 지정된 새 레이어를 타임라인에 추가합니다.

```
fl.getDocumentDOM().getTimeline().addNewLayer();
```

다음 예제에서는 현재 레이어의 맨 위에 새 폴더 레이어를 추가하고 이름을 Folder1로 지정합니다.

```
fl.getDocumentDOM().getTimeline().addNewLayer("Folder1", "folder", true);
```

timeline.clearFrames()

지원 버전

Flash MX 2004

사용법

```
timeline.clearFrames([startFrameIndex [, endFrameIndex]])
```

매개 변수

startFrameIndex 지울 프레임 범위 중 첫 번째 프레임을 지정하는 인덱스(0부터 시작). **startFrameIndex**를 생략하면 현재 선택한 프레임이 사용됩니다. 이 매개 변수는 선택 사항입니다.

endFrameIndex 지울 프레임 범위 중 마지막 프레임을 지정하는 인덱스(0부터 시작). **endFrameIndex** 앞까지 프레임 범위가 설정됩니다. **startFrameIndex**만 지정하면 기본적으로 **endFrameIndex**는 **startFrameIndex**의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 현재 레이어에 있는 하나 또는 여러 프레임의 내용을 모두 삭제합니다.

예제

다음 예제에서는 프레임 6부터 프레임 11(포함 안 됨) 사이의 프레임을 지웁니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
fl.getDocumentDOM().getTimeline().clearFrames(5, 10);
```

다음 예제에서는 프레임 15를 지웁니다.

```
fl.getDocumentDOM().getTimeline().clearFrames(14);
```

timeline.clearKeyframes()

지원 버전

Flash MX 2004

사용법

```
timeline.clearKeyframes([startFrameIndex [, endFrameIndex]])
```

매개 변수

startFrameIndex 지울 프레임 범위 중 첫 번째 프레임을 지정하는 인덱스(0부터 시작). **startFrameIndex**를 생략하면 현재 선택한 프레임이 사용됩니다. 이 매개 변수는 선택 사항입니다.

endFrameIndex 지울 프레임 범위 중 마지막 프레임을 지정하는 인덱스(0부터 시작). **endFrameIndex** 앞까지 프레임 범위가 설정됩니다. **startFrameIndex**만 지정하면 기본적으로 **endFrameIndex**는 **startFrameIndex**의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 키프레임을 일반 프레임으로 변환하고 현재 레이어에서 해당 내용을 삭제합니다.

예제

다음 예제에서는 프레임 5부터 프레임 10(포함 안 됨) 사이의 키프레임을 지웁니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
fl.getDocumentDOM().getTimeline().clearKeyframes(4, 9);
```

다음 예제에서는 프레임 15 위치에 있는 키프레임을 지우고 일반 프레임으로 변환합니다.

```
fl.getDocumentDOM().getTimeline().clearKeyframes(14);
```

timeline.convertToBlankKeyframes()

지원 버전

Flash MX 2004

사용법

```
timeline.convertToBlankKeyframes([startFrameIndex [, endFrameIndex]])
```

매개 변수

startFrameIndex 키프레임으로 변환할 첫 번째 프레임을 지정하는 인덱스(0부터 시작). **startFrameIndex**를 생략하면 현재 선택한 프레임이 변환됩니다. 이 매개 변수는 선택 사항입니다.

endFrameIndex 키프레임으로 변환할 마지막 프레임을 지정하는 인덱스(0부터 시작). **endFrameIndex** 앞까지 변환할 프레임 범위가 설정됩니다. **startFrameIndex**만 지정하면 기본적으로 **endFrameIndex**는 **startFrameIndex**의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 현재 레이어에서 프레임을 빈 키프레임으로 변환합니다.

예제

다음 예제에서는 프레임 2부터 프레임 10(포함 안 됨) 사이의 프레임을 빈 키프레임으로 변환합니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
fl.getDocumentDOM().getTimeline().convertToBlankKeyframes(1, 9);
```

다음 예제에서는 프레임 5를 빈 키프레임으로 변환합니다.

```
fl.getDocumentDOM().getTimeline().convertToBlankKeyframes(4);
```

timeline.convertToKeyframes()

지원 버전

Flash MX 2004

사용법

```
timeline.convertToKeyframes([startFrameIndex [, endFrameIndex]])
```

매개 변수

startFrameIndex 키프레임으로 변환할 첫 번째 프레임을 지정하는 인덱스(0부터 시작). **startFrameIndex**를 생략하면 현재 선택한 프레임이 변환됩니다. 이 매개 변수는 선택 사항입니다.

endFrameIndex 키프레임으로 변환할 마지막 프레임을 지정하는 인덱스(0부터 시작). **endFrameIndex** 앞까지 변환할 프레임 범위가 설정됩니다. **startFrameIndex**만 지정하면 기본적으로 **endFrameIndex**는 **startFrameIndex**의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 현재 레이어에 있는 여러 프레임을 키프레임으로 변환하거나 지정된 프레임이 없는 경우 선택한 영역을 키프레임으로 변환합니다.

예제

다음 예제에서는 선택한 프레임을 키프레임으로 변환합니다.

```
fl.getDocumentDOM().getTimeline().convertToKeyframes();
```

다음 예제에서는 프레임 2부터 프레임 10(포함 안 됨) 사이의 프레임을 키프레임으로 변환합니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
fl.getDocumentDOM().getTimeline().convertToKeyframes(1, 9);
```

다음 예제에서는 프레임 5를 키프레임으로 변환합니다.

```
fl.getDocumentDOM().getTimeline().convertToKeyframes(4);
```

timeline.copyFrames()

지원 버전

Flash MX 2004

사용법

```
timeline.copyFrames([startFrameIndex [, endFrameIndex]])
```

매개 변수

startFrameIndex 복사할 프레임 범위 중 첫 번째 프레임을 지정하는 인덱스(0부터 시작). **startFrameIndex**를 생략하면 현재 선택한 프레임이 사용됩니다. 이 매개 변수는 선택 사항입니다.

endFrameIndex 복사할 마지막 프레임을 지정하는 인덱스(0부터 시작). **endFrameIndex** 앞까지 복사할 프레임 범위가 설정됩니다. **startFrameIndex**만 지정하면 기본적으로 **endFrameIndex**는 **startFrameIndex**의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 현재 레이어의 여러 프레임을 클립보드에 복사합니다.

예제

다음 예제에서는 선택한 프레임을 클립보드에 복사합니다.

```
fl.getDocumentDOM().getTimeline().copyFrames();
```

다음 예제에서는 프레임 2부터 프레임 10(포함 안 됨) 사이의 프레임을 클립보드에 복사합니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
fl.getDocumentDOM().getTimeline().copyFrames(1, 9);
```

다음 예제에서는 프레임 5를 클립보드에 복사합니다.

```
fl.getDocumentDOM().getTimeline().copyFrames(4);
```

timeline.copyLayers()

지원 버전

Flash CS5.5 Professional

사용법

```
timeline.copyLayers([startLayerIndex [, endLayerIndex]])
```

매개 변수

startLayerIndex 선택 사항입니다. 복사할 레이어 범위 중 첫 번째 레이어를 지정하는 인덱스(0부터 시작). **startLayerIndex**를 생략하면 현재 선택한 레이어가 사용됩니다.

`endLayerIndex` 선택 사항입니다. 복사할 마지막 레이어를 지정하는 인덱스(0부터 시작). 복사할 레이어 범위는 `endLayerIndex`(포함)까지입니다. `startLayerIndex`만 지정하는 경우 `endLayerIndex`의 기본값은 `startLayerIndex`의 값으로 지정됩니다.

반환값

없음

설명

메서드, **Timeline**에 현재 선택되어 있는 레이어 또는 지정된 범위에 있는 레이어를 복사합니다. 복사할 레이어 또는 레이어 범위를 지정하도록 선택적 인수를 제공할 수 있습니다.

예제

다음 예제는 **Timeline**에서 인덱스 2의 레이어를 인덱스 7에 복사합니다.

```
fl.getDocumentDOM().getTimeline().copyLayers(2, 7);
```

참고 사항

[timeline.cutLayers\(\)](#), [timeline.pasteLayers\(\)](#), [timeline.duplicateLayers\(\)](#)

timeline.copyMotion()

지원 버전

Flash CS3 Professional

사용법

```
timeline.copyMotion()
```

매개 변수

없음

반환값

없음

설명

메서드, 모션 트윈 또는 프레임별 애니메이션에서 선택한 프레임의 모션을 복사합니다. 그런 다음 [timeline.pasteMotion\(\)](#)을 사용하여 다른 프레임에 모션을 적용할 수 있습니다.

모션을 스크립트에 붙여 넣을 수 있는 텍스트 코드로 복사하려면 [timeline.copyMotionAsAS3\(\)](#)을 참조하십시오.

예제

다음 예제에서는 선택한 하나 이상의 프레임에서 모션을 복사합니다.

```
fl.getDocumentDOM().getTimeline().copyMotion();
```

참고 사항

[timeline.copyMotionAsAS3\(\)](#), [timeline.pasteMotion\(\)](#)

timeline.copyMotionAsAS3()

지원 버전

Flash CS3 Professional

사용법

```
timeline.copyMotionAsAS3 ()
```

매개 변수

없음

반환값

없음

설명

메서드, 모션 트윈 또는 프레임별 애니메이션에서 선택한 프레임의 모션을 클립보드에 ActionScript 3.0 코드로 복사합니다. 그런 다음 스크립트에 이 코드를 붙여 넣을 수 있습니다.

다른 프레임에 적용할 수 있는 형식으로 모션을 복사하려면 [timeline.copyMotion\(\)](#)을 참조하십시오.

예제

다음 예제에서는 하나 이상의 선택된 프레임의 모션을 클립보드에 ActionScript3.0 코드로 복사합니다.

```
f1.getDocumentDOM().getTimeline().copyMotionAsAS3 ();
```

참고 사항

[timeline.copyMotion\(\)](#)

timeline.createMotionObject()

지원 버전

Flash Professional CS5

사용법

```
timeline.createMotionObject([startFrame [,endFrame])
```

매개 변수

startFrame 모션 객체를 만들 첫 번째 프레임을 지정합니다. **startFrame**을 생략하면 현재 선택한 프레임이 사용되고 선택한 프레임이 없으면 모든 레이어의 현재 재생 헤드에 있는 모든 프레임이 제거됩니다. 이 매개 변수는 선택 사항입니다.

endFrame 모션 객체 만들기를 중지할 프레임을 지정합니다. **endFrame** 앞까지 프레임 범위가 설정됩니다. **startFrame**만 지정하면 기본적으로 **endFrame**은 **startFrame**의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 새 모션 객체를 만듭니다. 매개 변수는 선택 사항입니다. 매개 변수가 지정된 경우 모션 객체를 만들기 전에 타임라인 선택 영역을 지정된 프레임으로 설정합니다.

예제

다음 예제에서는 현재 재생 위치의 최상위 레이어에서 모션 객체를 만듭니다.

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().createMotionObject();
```

다음 예제에서는 현재 장면의 최상위 레이어에서 **Frame 5**부터 시작하여 **Frame 15** 앞에서 끝나는 모션 객체를 만듭니다.

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().createMotionObject(5, 15);
```

timeline.createMotionTween()

지원 버전

Flash MX 2004

사용법

```
timeline.createMotionTween([startFrameIndex [, endFrameIndex]])
```

매개 변수

startFrameIndex 모션 트윈을 시작할 프레임을 지정하는 인덱스(0부터 시작). **startFrameIndex**를 생략하면 현재 선택한 프레임이 사용됩니다. 이 매개 변수는 선택 사항입니다.

endFrameIndex 모션 트윈을 중지할 프레임을 지정하는 인덱스(0부터 시작). **endFrameIndex** 앞까지 프레임 범위가 설정됩니다. **startFrameIndex**만 지정하면 기본적으로 **endFrameIndex**는 **startFrameIndex**의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 현재 레이어에 선택되어 있는 각 키프레임의 **frame.tweenType** 속성을 **motion**으로 설정하고 필요한 경우 각 프레임의 내용을 단일 심볼 인스턴스로 변환합니다. 이 속성은 Flash 제작 도구의 [모션 트윈 생성] 메뉴 항목에 해당합니다.

예제

다음 예제에서는 프레임 1부터 프레임 10(포함 안 함) 사이의 프레임 모양을 그래픽 심볼 인스턴스로 변환하고 **frame.tweenType**을 **motion**으로 설정합니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
fl.getDocumentDOM().getTimeline().createMotionTween(0, 9);
```

timeline.currentFrame

지원 버전

Flash MX 2004

사용법

```
timeline.currentFrame
```

설명

속성, 현재 재생 헤드 위치에 있는 프레임의 0부터 시작하는 인덱스입니다.

예제

다음 예제에서는 현재 타임라인의 재생 헤드를 프레임 10으로 설정합니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
fl.getDocumentDOM().getTimeline().currentFrame = 9;
```

다음 예제에서는 현재 재생 헤드 위치의 값을 `curFrame` 변수에 저장합니다.

```
var curFrame = fl.getDocumentDOM().getTimeline().currentFrame;
```

timeline.currentLayer

지원 버전

Flash MX 2004

사용법

```
timeline.currentLayer
```

설명

속성, 현재 활성화된 레이어의 0부터 시작하는 인덱스입니다. 레이어의 인덱스는 값이 0이면 최상위 레이어로 지정되고 값이 1이면 그 다음 레이어로 지정되는 방식으로 지정됩니다.

예제

다음 예제에서는 최상위 레이어를 활성화합니다.

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
```

다음 예제에서는 현재 활성화된 레이어의 인덱스를 `curLayer` 변수에 저장합니다.

```
var curLayer = fl.getDocumentDOM().getTimeline().currentLayer;
```

timeline.cutFrames()

지원 버전

Flash MX 2004

사용법

```
timeline.cutFrames([startFrameIndex [, endFrameIndex]])
```

매개 변수

`startFrameIndex` 잘라낼 프레임 범위 중 첫 번째 프레임을 지정하는 인덱스(0부터 시작). `startFrameIndex`를 생략하면 현재 선택한 프레임이 사용됩니다. 이 매개 변수는 선택 사항입니다.

endFrameIndex 잘라낼 마지막 프레임을 지정하는 인덱스(0부터 시작). **endFrameIndex** 앞까지 프레임 범위가 설정됩니다. **startFrameIndex**만 지정하면 기본적으로 **endFrameIndex**는 **startFrameIndex**의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 타임라인에 있는 현재 레이어의 여러 프레임을 잘라내어 클립보드에 저장합니다.

예제

다음 예제에서는 타임라인에서 선택한 프레임을 잘라내어 클립보드에 저장합니다.

```
fl.getDocumentDOM().getTimeline().cutFrames();
```

다음 예제에서는 타임라인에서 프레임 2부터 프레임 10(포함 안 됨) 사이의 프레임을 잘라내어 클립보드에 저장합니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
fl.getDocumentDOM().getTimeline().cutFrames(1, 9);
```

다음 예제에서는 타임라인에서 프레임 5를 잘라내어 클립보드에 저장합니다.

```
fl.getDocumentDOM().getTimeline().cutFrames(4);
```

timeline.cutLayers()

지원 버전

Flash CS5.5 Professional

사용법

```
timeline.cutLayers([startLayerIndex [, endLayerIndex]])
```

매개 변수

startLayerIndex 선택 사항입니다. 잘라낼 레이어 범위 중 첫 번째 레이어를 지정하는 인덱스(0부터 시작)입니다.

startLayerIndex를 생략하면 현재 선택한 레이어가 사용됩니다.

endLayerIndex 선택 사항입니다. 잘라낼 마지막 레이어를 지정하는 인덱스(0부터 시작)입니다. 잘라낼 레이어 범위는 **endLayerIndex**(포함)까지입니다. **startLayerIndex**만 지정하는 경우 **endLayerIndex**의 기본값은 **startLayerIndex**의 값으로 지정됩니다.

반환값

없음

설명

메서드, **Timeline**에 현재 선택되어 있는 레이어 또는 지정된 범위에 있는 레이어를 잘라냅니다. 잘라낼 레이어 또는 레이어 범위를 지정하도록 선택적 인수를 제공할 수 있습니다.

예제

다음 예제는 **Timeline**에서 인덱스 2의 레이어를 인덱스 7에 잘라냅니다.

```
fl.getDocumentDOM().getTimeline().cutLayers(2, 7);
```

참고 사항

[timeline.copyLayers\(\)](#), [timeline.pasteLayers\(\)](#), [timeline.duplicateLayers\(\)](#)

timeline.deleteLayer()

지원 버전

Flash MX 2004

사용법

```
timeline.deleteLayer([index])
```

매개 변수

index 삭제할 레이어를 지정하는 인덱스(0부터 시작). 타임라인에 레이어가 하나만 있으면 이 메서드는 아무런 영향을 주지 않습니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 레이어를 삭제합니다. 레이어가 폴더이면 폴더 내의 모든 레이어가 삭제됩니다. 레이어 인덱스를 지정하지 않으면 현재 선택한 레이어가 삭제됩니다.

예제

다음 예제에서는 맨 위에서부터 두 번째에 있는 레이어를 삭제합니다.

```
fl.getDocumentDOM().getTimeline().deleteLayer(1);
```

다음 예제에서는 현재 선택한 레이어를 삭제합니다.

```
fl.getDocumentDOM().getTimeline().deleteLayer();
```

timeline.duplicateLayers()

지원 버전

Flash CS5.5 Professional

사용법

```
timeline.duplicateLayers([startLayerIndex [, endLayerIndex]])
```

매개 변수

startLayerIndex 선택 사항입니다. 복사할 레이어 범위 중 첫 번째 레이어를 지정하는 인덱스(0부터 시작). 또한 클립보드의 레이어를 붙여넣을 레이어를 지정합니다. **startLayerIndex**를 생략하면 현재 선택한 레이어가 사용됩니다.

endLayerIndex 선택 사항입니다. 복사할 마지막 레이어를 지정하는 인덱스(0부터 시작). 복사할 레이어 범위는 **endLayerIndex**(포함)까지입니다. **startLayerIndex**만 지정하는 경우 **endLayerIndex**의 기본값은 **startLayerIndex**의 값으로 지정됩니다.

반환값

없음

설명

메서드, **Timeline**에 현재 선택되어 있는 레이어 또는 지정된 범위에 있는 레이어를 복제합니다. 복제할 레이어 또는 레이어 범위를 지정하도록 선택적 인수를 제공할 수 있습니다.

예제

다음 예제는 **Timeline**에서 현재 선택되어 있는 레이어를 복제합니다.

```
fl.getDocumentDOM().getTimeline().duplicateLayers();
```

다음 예제는 인덱스 2에서 인덱스 7까지의 레이어를 레이어 인덱스 2 위에 복제합니다.

```
fl.getDocumentDOM().getTimeline().duplicatedLayers(2,7);
```

참고 사항

[timeline.copyLayers\(\)](#), [timeline.cutLayers\(\)](#), [timeline.pasteLayers\(\)](#)

timeline.expandFolder()

지원 버전

Flash MX 2004

사용법

```
timeline.expandFolder(bExpand [, bRecurseNestedParents [, index]])
```

매개 변수

bExpand true로 설정하면 폴더가 확장되고 false로 설정하면 폴더가 축소되는 부울 값

bRecurseNestedParents true로 설정하면 **bExpand** 매개 변수에 따라 지정된 폴더 내의 모든 레이어가 열리거나 닫히는 부울 값
. 이 매개 변수는 선택 사항입니다.

index 확장하거나 축소할 폴더의 0부터 시작하는 인덱스. 모든 레이어에 적용하려면 -1을 사용하고 **bRecurseNestedParents**를 true로 설정해야 합니다. 이 속성은 Flash 제작 도구의 [모두 확장]/[모두 축소] 메뉴 항목에 해당합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 지정된 폴더를 확장하거나 축소합니다. 레이어를 지정하지 않으면 현재 레이어가 사용됩니다.

예제

다음 예제에서는 이 폴더 구조를 사용합니다.

```
Folder 1 ***  
--layer 7  
--Folder 2 ****  
---Layer 5
```

다음 예제에서는 폴더 1만 확장합니다.

```
fl.getDocumentDOM().getTimeline().currentLayer = 1;  
fl.getDocumentDOM().getTimeline().expandFolder(true);
```

다음 예제에서는 폴더 1만 확장합니다. 이때 폴더 1이 마지막으로 축소될 때 폴더 2도 축소된 것으로 간주합니다. 그렇지 않으면 폴더 2가 확장된 것처럼 보입니다.

```
f1.getDocumentDOM().getTimeline().expandFolder(true, false, 0);
```

다음 예제에서는 현재 타임라인에 있는 모든 폴더를 축소합니다.

```
f1.getDocumentDOM().getTimeline().expandFolder(false, true, -1);
```

timeline.findLayerIndex()

지원 버전

Flash MX 2004

사용법

```
timeline.findLayerIndex(name)
```

매개 변수

name 찾을 레이어의 이름을 지정하는 문자열

반환값

지정된 레이어의 인덱스 값 배열. 지정된 레이어를 찾을 수 없으면 `undefined`가 반환됩니다.

설명

메서드, 주어진 이름으로 레이어 인덱스의 배열을 찾습니다. 레이어 인덱스는 평면적이기 때문에 폴더는 기본 인덱스의 일부로 간주됩니다.

예제

다음 예제에서는 Layer 7이라는 모든 레이어의 인덱스 값을 [출력] 패널에 표시합니다.

```
var layerIndex = f1.getDocumentDOM().getTimeline().findLayerIndex("Layer 7");  
f1.trace(layerIndex);
```

다음 예제에서는 이 메서드에서 반환된 값을 `timeline.setSelectedLayers()`로 다시 전달하는 방법을 보여 줍니다.

```
var layerIndex = f1.getDocumentDOM().getTimeline().findLayerIndex("Layer 1");  
f1.getDocumentDOM().getTimeline().setSelectedLayers(layerIndex[0], true);
```

timeline.frameCount

지원 버전

Flash MX 2004

사용법

```
timeline.frameCount
```

설명

읽기 전용 속성, 이 타임라인에 있는 가장 긴 레이어의 프레임 수를 나타내는 정수입니다.

예제

다음 예제에서는 현재 문서에 있는 가장 긴 레이어의 프레임 수를 `countNum` 변수에 저장합니다.

```
var countNum = fl.getDocumentDOM().getTimeline().frameCount;
```

timeline.getFrameProperty()

지원 버전

Flash MX 2004

사용법

```
timeline.getFrameProperty(property [, startFrameIndex [, endFrameIndex]])
```

매개 변수

`property` 값을 가져올 속성의 이름을 지정하는 문자열입니다. 전체 속성 목록을 보려면 [Frame 객체](#)의 속성 요약을 참조하십시오.

`startFrameIndex` 값을 가져올 첫 번째 프레임 번호를 지정하는 인덱스(0부터 시작). `startFrameIndex`를 생략하면 현재 선택한 프레임이 사용됩니다. 이 매개 변수는 선택 사항입니다.

`endFrameIndex` 마지막 프레임을 지정하는 인덱스(0부터 시작). `endFrameIndex` 앞까지 프레임 범위가 설정됩니다.

`startFrameIndex`만 지정하면 기본적으로 `endFrameIndex`는 `startFrameIndex`의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

지정된 속성의 값을 반환하거나 선택한 모든 프레임에 동일한 속성 값이 없으면 `undefined`를 반환합니다.

설명

메서드, 선택한 프레임에 대해 지정된 속성 값을 가져옵니다.

예제

다음 예제에서는 현재 문서의 최상위 레이어에 있는 첫 번째 프레임의 이름을 가져와서 [출력] 패널에 표시합니다.

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().setSelectedFrames(0, 0, true);  
var frameName = fl.getDocumentDOM().getTimeline().getFrameProperty("name");  
fl.trace(frameName);
```

timeline.getGuidelines()

지원 버전

Flash CS4 Professional

사용법

```
timeline.getGuidelines()
```

매개 변수

없음

반환값

XML 문자열

설명

메서드, 타임라인 가로 및 세로 안내선의 현재 위치를 나타내는 XML 문자열을 반환합니다([보기] > [안내선] > [안내선 표시]). 타임라인에 이러한 안내선을 적용하려면 `timeline.setGuidelines()`를 사용합니다.

예제

다음 예제에서는 첫 번째 타임라인에 일부 안내선이 있다고 가정할 때, 해당 안내선을 [출력] 패널에 XML 문자열로 표시합니다.

```
var currentTimeline = fl.getDocumentDOM().timelines[0];  
fl.trace(currentTimeline.getGuidelines());
```

timeline.getLayerProperty()

지원 버전

Flash MX 2004

사용법

```
timeline.getLayerProperty(property)
```

매개 변수

`property` 값을 가져올 속성의 이름을 지정하는 문자열입니다. 속성 목록을 보려면 [Frame 객체](#)의 속성 요약을 참조하십시오.

반환값

지정된 속성의 값. Flash에서는 레이어의 속성을 찾아서 해당 유형을 확인합니다. 지정된 모든 레이어에 동일한 속성 값이 없으면 `undefined`가 반환됩니다.

설명

메서드, 선택한 레이어의 특정 속성 값을 가져옵니다.

예제

다음 예제에서는 현재 문서에 있는 최상위 레이어의 이름을 가져와서 [출력] 패널에 표시합니다.

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
var layerName = fl.getDocumentDOM().getTimeline().getLayerProperty("name");  
fl.trace(layerName);
```

timeline.getSelectedFrames()

지원 버전

Flash MX 2004

매개 변수

없음

반환값

3n 정수가 포함된 배열. 여기서 n은 선택한 영역의 수입니다. 각 그룹의 첫 번째 정수는 레이어 인덱스이고 두 번째와 세 번째 정수는 각각 선택한 영역의 첫 번째 프레임과 마지막 프레임을 지정합니다. 마지막 프레임은 선택한 영역에 포함되지 않습니다.

설명

메서드, 배열에서 현재 선택한 프레임을 가져옵니다.

예제

다음 예제에서는 최상위 레이어가 현재 레이어일 경우 0,5,10,0,20,25를 [출력] 패널에 표시합니다.

```
var timeline = fl.getDocumentDOM().getTimeline();
timeline.setSelectedFrames(5,10);
timeline.setSelectedFrames(20,25,false);
var theSelectedFrames = timeline.getSelectedFrames();
fl.trace(theSelectedFrames);
```

참고 사항

[timeline.setSelectedFrames\(\)](#)

timeline.getSelectedLayers()

지원 버전

Flash MX 2004

매개 변수

없음

반환값

선택한 레이어의 0부터 시작하는 인덱스 값 배열.

설명

메서드, 현재 선택한 레이어의 0부터 시작하는 인덱스 값을 가져옵니다.

예제

다음 예제에서는 1,0을 [출력] 패널에 표시합니다.

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(0);
fl.getDocumentDOM().getTimeline().setSelectedLayers(1, false);
var layerArray = fl.getDocumentDOM().getTimeline().getSelectedLayers();
fl.trace(layerArray);
```

참고 사항

[timeline.setSelectedLayers\(\)](#)

timeline.insertBlankKeyframe()

지원 버전

Flash MX 2004

사용법

```
timeline.insertBlankKeyframe ([frameNumIndex])
```

매개 변수

frameNumIndex 키프레임을 삽입할 프레임에 지정하는 인덱스(0부터 시작). **frameNumIndex**를 생략하면 현재 재생 헤드 프레임 번호가 사용됩니다. 이 매개 변수는 선택 사항입니다.

지정되었거나 선택한 프레임이 일반 프레임이면 해당 프레임 위치에 키프레임이 삽입됩니다. 예를 들어, 1번부터 10번까지 10개의 프레임으로 구성된 프레임 범위에서 프레임 5를 선택하면 프레임 5는 빈 키프레임이 되고 프레임 범위의 길이는 계속 프레임 10개를 유지합니다. 선택한 프레임 5가 키프레임이고 그 다음 프레임이 일반 프레임이면 프레임 6 위치에 빈 키프레임이 삽입됩니다. 선택한 프레임 5가 키프레임이고 그 다음 프레임도 키프레임이면 키프레임이 삽입되지 않고 재생 헤드가 프레임 6 위치로 이동합니다.

반환값

없음

설명

메서드, 지정된 프레임 인덱스에 빈 키프레임을 삽입합니다. 인덱스가 지정되어 있지 않은 경우 재생 헤드/선택한 위치를 사용하여 빈 키프레임을 삽입합니다. 자세한 내용은 [timeline.insertKeyframe\(\)](#)을 참조하십시오.

예제

다음 예제에서는 프레임 20 위치에 빈 키프레임을 삽입합니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
fl.getDocumentDOM().getTimeline().insertBlankKeyframe(19);
```

다음 예제에서는 현재 선택한 프레임 위치(또는 선택한 프레임이 없는 경우 재생 헤드 위치)에 빈 키프레임을 삽입합니다.

```
fl.getDocumentDOM().getTimeline().insertBlankKeyframe();
```

timeline.insertFrames()

지원 버전

Flash MX 2004

사용법

```
timeline.insertFrames([numFrames [, bAllLayers [, frameNumIndex]])
```

매개 변수

numFrames 삽입할 프레임 수를 지정하는 정수. 이 매개 변수를 생략하면 현재 레이어의 현재 선택한 위치에 프레임이 삽입됩니다. 이 매개 변수는 선택 사항입니다.

bAllLayers true(기본값)로 설정하면 모든 레이어에 **numFrames** 매개 변수에 지정된 프레임 수가 삽입되고 false로 설정하면 현재 레이어에 프레임이 삽입됩니다. 이 매개 변수는 선택 사항입니다.

frameNumIndex 새 프레임을 삽입할 프레임에 지정하는 인덱스(0부터 시작). 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 지정된 인덱스에 지정된 수의 프레임을 삽입합니다.

지정된 매개 변수가 없으면 이 메서드는 다음과 같이 작동합니다.

- 하나 이상의 프레임을 선택한 경우 현재 레이어에서 첫 번째로 선택한 프레임 위치에 선택한 수의 프레임을 삽입합니다. 즉, 6번부터 10번까지 모두 다섯 개의 프레임을 선택한 경우 선택한 프레임이 포함된 레이어의 프레임 6 위치에 이 다섯 개의 프레임 추가합니다.
- 선택한 프레임이 없는 경우 모든 레이어의 현재 위치에 하나의 프레임을 삽입합니다.

매개 변수가 지정되어 있으면 이 메서드는 다음과 같이 작동합니다.

- **numFrames**가 지정된 경우에만 현재 레이어의 현재 프레임 위치에 지정된 수의 프레임을 삽입합니다.
- **numFrames**가 지정되어 있고 **bAllLayers**가 true이면 모든 레이어의 현재 프레임 위치에 지정된 수의 프레임을 삽입합니다.
- 세 개의 매개 변수가 모두 지정된 경우 지정된 인덱스(**frameIndex**) 위치에 지정된 수의 프레임을 삽입합니다. **bAllLayers**에 대해 전달되는 값은 프레임이 현재 레이어에만 추가되었는지 아니면 모든 레이어에 추가되었는지 여부를 확인합니다.

지정되었거나 선택한 프레임이 일반 프레임이면 해당 프레임 위치에 프레임이 삽입됩니다. 예를 들어, 1번부터 10번까지 10개의 프레임으로 구성된 프레임 범위에서 프레임 5를 선택하면(또는 **frameIndex**에 값을 4로 전달하면) 프레임 5 위치에 프레임이 추가되고 프레임 범위의 길이는 프레임 11개가 됩니다. 선택한 프레임 5가 키프레임이면 그 다음 프레임이 키프레임인지 여부와 상관없이 프레임 6 위치에 프레임이 삽입됩니다.

예제

다음 예제에서는 현재 레이어의 현재 선택한 위치에 선택한 내용에 따라 하나 또는 여러 프레임을 삽입합니다.

```
fl.getDocumentDOM().getTimeline().insertFrames();
```

다음 예제에서는 모든 레이어의 현재 프레임에 다섯 개의 프레임을 삽입합니다.

```
fl.getDocumentDOM().getTimeline().insertFrames(5);
```

참고: 프레임에 여러 개의 레이어를 사용하고 있고 이전 명령을 사용할 때 한 레이어에서 프레임을 선택한 경우 선택한 레이어에만 프레임이 삽입됩니다. 여러 개의 레이어를 사용하고 있지만 레이어에 선택된 프레임이 없는 경우 모든 레이어에 프레임이 삽입됩니다.

다음 예제에서는 현재 레이어에만 세 개의 프레임을 삽입합니다.

```
fl.getDocumentDOM().getTimeline().insertFrames(3, false);
```

다음 예제에서는 모든 레이어에 첫 번째 프레임부터 네 개의 프레임을 삽입합니다.

```
fl.getDocumentDOM().getTimeline().insertFrames(4, true, 0);
```

timeline.insertKeyframe()

지원 버전

Flash MX 2004

사용법

```
timeline.insertKeyframe([frameNumIndex])
```

매개 변수

frameNumIndex 현재 레이어에서 키프레임을 삽입할 프레임 인덱스를 지정하는 인덱스(0부터 시작). **frameNumIndex**를 생략하면 현재 재생 헤드 또는 현재 선택한 프레임의 프레임 번호가 사용됩니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 지정된 프레임에 키프레임을 삽입합니다. 매개 변수를 생략하면 재생 헤드/선택한 위치를 사용하여 키프레임을 삽입합니다.

이 메서드는 삽입된 키프레임에 변환된(즉, 비어 있지 않은) 프레임의 내용이 포함되어 있다는 점만 제외하고 `timeline.insertBlankKeyframe()`과 같은 방식으로 작동합니다.

예제

다음 예제에서는 재생 헤드 또는 선택한 위치에 키프레임을 삽입합니다.

```
f1.getDocumentDOM().getTimeline().insertKeyframe();
```

다음 예제에서는 두 번째 레이어의 프레임 10 위치에 키프레임을 삽입합니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
f1.getDocumentDOM().getTimeline().currentLayer = 1;  
f1.getDocumentDOM().getTimeline().insertKeyframe(9);
```

timeline.layerCount

지원 버전

Flash MX 2004

사용법

```
timeline.layerCount
```

설명

읽기 전용 속성, 지정된 타임라인에 있는 레이어 수를 나타내는 정수입니다.

예제

다음 예제에서는 현재 장면의 레이어 수를 NumLayer 변수에 저장합니다.

```
var NumLayer = f1.getDocumentDOM().getTimeline().layerCount;
```

timeline.layers

지원 버전

Flash MX 2004

사용법

```
timeline.layers
```

설명

읽기 전용 속성, 레이어 객체의 배열입니다.

예제

다음 예제에서는 현재 문서의 레이어 객체 배열을 `currentLayers` 변수에 저장합니다.

```
var currentLayers = fl.getDocumentDOM().getTimeline().layers;
```

timeline.libraryItem

지원 버전

Flash Professional CS5

사용법

`timeline.libraryItem`

설명

읽기 전용 속성, 타임라인의 `libraryItem` 속성이 `null`이면 해당 타임라인은 장면에 속합니다. 반대로 `null`이 아니면 해당 타임라인을 `LibraryItem` 객체로 취급할 수 있습니다.

예제

다음 예제는 `libraryItem`이 `null`이 아닌 경우 `libraryItem`의 이름을 출력하고, `libraryItem`이 `null`인 경우 장면의 이름을 출력합니다.

```
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

지원 버전

Flash MX 2004

사용법

`timeline.name`

설명

속성, 현재 타임라인의 이름을 지정하는 문자열입니다. 이 이름은 편집 중인 현재 장면, 스크린(슬라이드 또는 양식) 또는 심볼의 이름입니다.

예제

다음 예제에서는 첫 번째 장면 이름을 가져옵니다.

```
var sceneName = fl.getDocumentDOM().timelines[0].name;
```

다음 예제에서는 첫 번째 장면 이름을 `FirstScene`으로 설정합니다.

```
fl.getDocumentDOM().timelines[0].name = "FirstScene";
```

timeline.pasteFrames()

지원 버전

Flash MX 2004

사용법

```
timeline.pasteFrames([startFrameIndex [, endFrameIndex]])
```

매개 변수

startFrameIndex 붙여넣을 프레임 범위 중 첫 번째 프레임을 지정하는 인덱스(0부터 시작). **startFrameIndex**를 생략하면 현재 선택한 프레임이 사용됩니다. 이 매개 변수는 선택 사항입니다.

endFrameIndex 붙여넣을 마지막 프레임을 지정하는 인덱스(0부터 시작). 이 메서드는 **endFrameIndex** 앞까지 붙여넣습니다. **startFrameIndex**만 지정하면 기본적으로 **endFrameIndex**는 **startFrameIndex**의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 클립보드에 있는 여러 프레임을 지정된 프레임에 붙여넣습니다.

예제

다음 예제에서는 클립보드에 있는 프레임을 현재 선택한 프레임 또는 재생 헤드 위치에 붙여넣습니다.

```
fl.getDocumentDOM().getTimeline().pasteFrames();
```

다음 예제에서는 클립보드에 있는 프레임을 프레임 2부터 프레임 10(포함 안 됨) 사이에 붙여넣습니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
fl.getDocumentDOM().getTimeline().pasteFrames(1, 9);
```

다음 예제에서는 클립보드에 있는 프레임을 프레임 5부터 붙여넣습니다.

```
fl.getDocumentDOM().getTimeline().pasteFrames(4);
```

timeline.pasteLayers()

지원 버전

Flash CS5.5 Professional

사용법

```
timeline.pasteLayers([layerIndex])
```

매개 변수

layerIndex 선택 사항입니다. 클립보드의 레이어를 붙여넣는 레이어를 지정하는 인덱스(0부터 시작). **layerIndex**를 생략하면 현재 선택한 레이어가 사용됩니다.

반환값

붙여넣은 레이어의 가장 낮은 레이어 인덱스를 나타내는 정수입니다.

설명

메서드, 이전에 잘라내거나 복사한 레이어를 현재 선택된 레이어 위에 또는 지정된 레이어 인덱스 위에 붙여넣습니다. 지정된 레이어가 폴더 레이어인 경우에는 레이어를 폴더에 붙여넣습니다. 붙여넣은 레이어의 가장 낮은 레이어 인덱스를 반환합니다. 이 작업은 시스템 클립보드에 영향을 주지 않습니다.

예제

다음 예제는 레이어 클립보드에 있는 레이어를 **Timeline**에서 현재 선택되어 있는 레이어 위에 붙여넣습니다.

```
fl.getDocumentDOM().getTimeline().pasteLayers();
```

다음 예제는 레이어 클립보드의 레이어를 레이어 인덱스 2 위에 붙여넣습니다.

```
fl.getDocumentDOM().getTimeline().pasteLayers(2);
```

참고 사항

[timeline.cutLayers\(\)](#), [timeline.copyLayers\(\)](#), [timeline.duplicateLayers\(\)](#)

timeline.pasteMotion()

지원 버전

Flash CS3 Professional

사용법

```
timeline.pasteMotion()
```

매개 변수

없음

반환값

없음

설명

메서드, [timeline.copyMotion\(\)](#)으로 검색된 모션 프레임의 범위를 타임라인에 붙여넣습니다. 필요한 경우, 붙여넣을 프레임의 공간을 마련하기 위해 기존 프레임의 위치가 오른쪽으로 옮겨집니다.

예제

다음 예제에서는 클립보드의 모션을 현재 선택한 프레임이나 재생 헤드 위치에 붙여넣고, 해당 프레임의 위치를 붙여넣은 프레임의 오른쪽으로 변경합니다.

```
fl.getDocumentDOM().getTimeline().pasteMotion();
```

참고 사항

[timeline.copyMotion\(\)](#)

timeline.removeFrames()

지원 버전

Flash MX 2004

사용법

```
timeline.removeFrames([startFrameIndex [,endFrameIndex]])
```

매개 변수

startFrameIndex 제거할 첫 번째 프레임을 지정하는 인덱스(0부터 시작). **startFrameIndex**를 생략하면 현재 선택한 프레임이 사용되고 선택한 프레임이 없으면 모든 레이어의 현재 재생 헤드에 있는 모든 프레임이 제거됩니다. 이 매개 변수는 선택 사항입니다.

endFrameIndex 제거할 마지막 프레임을 지정하는 인덱스(0부터 시작). **endFrameIndex** 앞까지 프레임 범위가 설정됩니다. **startFrameIndex**만 지정하면 기본적으로 **endFrameIndex**는 **startFrameIndex**의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 프레임을 삭제합니다.

예제

다음 예제에서는 현재 장면의 최상위 레이어에서 프레임 5부터 프레임 10(포함 안 됨) 사이의 프레임을 삭제합니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().removeFrames(4, 9);
```

다음 예제에서는 현재 장면의 최상위 레이어에서 프레임 8을 삭제합니다.

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().removeFrames(7);
```

timeline.removeMotionObject()

지원 버전

Flash Professional CS5

사용법

```
timeline.removeMotionObject([startFrame [,endFrame])
```

매개 변수

startFrame 모션 객체의 제거를 시작할 첫 번째 프레임을 지정합니다. **startFrame**을 생략하면 현재 선택한 프레임이 사용되고 선택한 프레임이 없으면 모든 레이어의 현재 재생 헤드에 있는 모든 프레임이 제거됩니다. 이 매개 변수는 선택 사항입니다.

endFrame 모션 객체의 제거를 중지할 프레임을 지정합니다. **endFrame** 앞까지 프레임 범위가 설정됩니다. **startFrame**만 지정하면 기본적으로 **endFrame**은 **startFrame**의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 모션 객체를 제거하고 프레임을 정적 프레임으로 다시 변환합니다. 매개 변수는 선택 사항입니다. 매개 변수가 지정된 경우 모션 객체를 제거하기 전에 타임라인 선택 영역을 지정된 프레임으로 설정합니다.

예제

다음 예제에서는 모든 모션 객체를 삭제하고 현재 재생 위치의 최상위 레이어에서 프레임을 다시 정적 프레임으로 변환합니다.

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().removeMotionObject();
```

다음 예제에서는 현재 장면의 최상위 레이어에서 **Frame 5**부터 시작하여 **Frame 15** 앞에서 끝나는 모션 객체를 삭제합니다.

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().removeMotionObject(5, 15);
```

참고 사항

446페이지의 "[timeline.createMotionObject\(\)](#)"

timeline.reorderLayer()

지원 버전

Flash MX 2004

사용법

```
timeline.reorderLayer(layerToMove, layerToPutItBy [, bAddBefore])
```

매개 변수

layerToMove 이동할 레이어를 지정하는 인덱스(0부터 시작)

layerToPutItBy 이동할 레이어의 다음 레이어를 지정하는 인덱스(0부터 시작). 예를 들어, **layerToMove**에 1을 지정하고 **layerToPutItBy**에 0을 지정하면 두 번째 레이어가 첫 번째 레이어 뒤로 이동합니다.

bAddBefore 레이어를 **layerToPutItBy**의 앞과 뒤 중 어디로 이동할지 지정합니다. **false**를 지정하면 레이어가 **layerToPutItBy** 뒤로 이동합니다. 기본값은 **true**입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 첫 번째로 지정된 레이어를 두 번째로 지정된 레이어의 앞 또는 뒤로 이동합니다.

예제

다음 예제에서는 인덱스 2에 있는 레이어를 맨 위(인덱스 0에 있는 레이어 위)로 이동합니다.

```
fl.getDocumentDOM().getTimeline().reorderLayer(2, 0);
```

다음 예제에서는 인덱스 3에 있는 레이어를 인덱스 5에 있는 레이어 뒤로 이동합니다.

```
fl.getDocumentDOM().getTimeline().reorderLayer(3, 5, false);
```

timeline.reverseFrames()

지원 버전

Flash MX 2004

사용법

```
timeline.reverseFrames([startFrameIndex [, endFrameIndex]])
```

매개 변수

startFrameIndex 반전시킬 첫 번째 프레임을 지정하는 인덱스(0부터 시작). **startFrameIndex**를 생략하면 현재 선택한 프레임이 사용됩니다. 이 매개 변수는 선택 사항입니다.

endFrameIndex 반전시킬 마지막 프레임을 지정하는 인덱스(0부터 시작). **endFrameIndex** 앞까지 프레임 범위가 설정됩니다. **startFrameIndex**만 지정하면 기본적으로 **endFrameIndex**는 **startFrameIndex**의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 여러 프레임을 반전시킵니다.

예제

다음 예제에서는 현재 선택한 프레임의 위치를 반전시킵니다.

```
fl.getDocumentDOM().getTimeline().reverseFrames();
```

다음 예제에서는 프레임 10부터 프레임 15(포함 안 됨) 사이의 프레임을 반전시킵니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
fl.getDocumentDOM().getTimeline().reverseFrames(9, 14);
```

timeline.selectAllFrames()

지원 버전

Flash MX 2004

사용법

```
timeline.selectAllFrames()
```

매개 변수

없음

반환값

없음

설명

메서드, 현재 타임라인에 있는 모든 프레임을 선택합니다.

예제

다음 예제에서는 현재 타임라인에 있는 모든 프레임을 선택합니다.

```
fl.getDocumentDOM().getTimeline().selectAllFrames();
```


timeline.setFrameProperty()

지원 버전

Flash MX 2004

사용법

```
timeline.setFrameProperty(property, value [, startFrameIndex [, endFrameIndex]])
```

매개 변수

property 수정할 속성의 이름을 지정하는 문자열입니다. 전체 속성 및 값 목록을 보려면 **Frame 객체**의 속성 요약을 참조하십시오.

이 메서드는 **frame.duration** 및 **frame.elements** 같은 읽기 전용 속성의 값을 설정하는 데 사용할 수 없습니다.

value 속성에 설정할 값을 지정합니다. 적절한 값 및 유형을 확인하려면 **Frame 객체**의 속성 요약을 참조하십시오.

startFrameIndex 수정할 첫 번째 프레임을 지정하는 인덱스(0부터 시작). **startFrameIndex**를 생략하면 현재 선택한 프레임이 사용됩니다. 이 매개 변수는 선택 사항입니다.

endFrameIndex 중지할 첫 번째 프레임을 지정하는 인덱스(0부터 시작). **endFrameIndex** 앞까지 프레임 범위가 설정됩니다. **startFrameIndex**만 지정하고 **endFrameIndex**를 지정하지 않으면 기본적으로 **endFrameIndex**는 **startFrameIndex**의 값을 사용합니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 선택한 프레임의 **Frame** 객체 속성을 설정합니다.

예제

다음 예제에서는 **ActionScript stop()** 명령을 현재 문서에 있는 최상위 레이어의 첫 번째 프레임에 할당합니다.

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().setSelectedFrames(0,0,true);  
fl.getDocumentDOM().getTimeline().setFrameProperty("actionScript", "stop();");
```

다음 예제에서는 현재 레이어의 프레임 2부터 프레임 5(포함 안 됨) 사이에 모션 트윈을 설정합니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
var doc = fl.getDocumentDOM();  
doc.getTimeline().setFrameProperty("tweenType", "motion", 1,4);
```

timeline.setGuidelines()

지원 버전

Flash CS4 Professional

사용법

```
timeline.setGuidelines(xmlString)
```

매개 변수

xmlString 적용할 안내선 정보가 포함된 XML 문자열

반환값

안내선이 성공적으로 적용되면 `true`를 반환하고 그렇지 않으면 `false`를 반환하는 부울 값

설명

메서드, 타임라인의 안내선([보기] > [안내선] > [안내선 표시])을 `xmlString`에 지정된 정보로 대체합니다. 이 메서드로 전달할 수 있는 XML 문자열을 검색하려면 `timeline.getGuidelines()`를 사용합니다.

새로 설정된 안내선을 보려면 안내선을 숨겼다가 다시 표시하면 됩니다.

예제

다음 예제에서는 특정 FLA 파일의 안내선을 다른 FLA 파일에 적용합니다.

```
var doc0 = fl.documents[0];  
var guides0 = doc0.timelines[0].getGuidelines();  
var doc1 = fl.documents[1];  
doc1.timelines[0].setGuidelines(guides0);
```

timeline.setLayerProperty()

지원 버전

Flash MX 2004

사용법

```
timeline.setLayerProperty(property, value [, layersToChange])
```

매개 변수

`property` 설정할 속성을 지정하는 문자열입니다. 속성 목록은 304페이지의 “[Layer 객체](#)”를 참조하십시오.

`value` 속성에 설정할 값. Layer 객체에서 속성을 설정할 때 사용하는 것과 동일한 값 유형을 사용합니다.

`layersToChange` 수정할 레이어를 나타내는 문자열입니다. 사용할 수 있는 값에는 "selected", "all" 및 "others"가 있습니다. 이 매개 변수를 생략하면 기본값은 "selected"입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 선택한 모든 레이어의 특정 속성을 지정된 값으로 설정합니다.

예제

다음 예제에서는 선택한 레이어를 숨깁니다.

```
fl.getDocumentDOM().getTimeline().setLayerProperty("visible", false);
```

다음 예제에서는 선택한 레이어의 이름을 `selLayer`로 설정합니다.

```
fl.getDocumentDOM().getTimeline().setLayerProperty("name", "selLayer");
```

timeline.setSelectedFrames()

지원 버전

Flash MX 2004

사용법

```
timeline.setSelectedFrames(startFrameIndex, endFrameIndex [, bReplaceCurrentSelection])  
timeline.setSelectedFrames(selectionList [, bReplaceCurrentSelection])
```

매개 변수

startFrameIndex 설정할 첫 번째 프레임을 지정하는 인덱스(0부터 시작)

endFrameIndex 선택한 영역의 끝을 지정하는 인덱스(0부터 시작). **endFrameIndex**는 선택할 범위의 마지막 프레임 뒤에 있는 프레임입니다.

bReplaceCurrentSelection true로 설정하면 지정된 프레임을 선택하기 전에 현재 선택한 프레임을 선택 취소하는 부울 값. 기본 값은 true입니다.

selectionList timeline.getSelectedFrames()에 의해 반환된 세 정수의 배열

반환값

없음

설명

메서드, 현재 레이어에서 여러 프레임을 선택하거나 선택한 프레임을 이 메서드에 전달된 선택 영역 배열로 설정합니다.

예제

다음 예제에서는 최상위 레이어에서 프레임 1부터 프레임 10(포함 안 됨) 사이의 프레임을 선택한 다음 동일한 레이어에서 프레임 12부터 프레임 15(포함 안 됨) 사이의 프레임을 현재 선택한 프레임에 추가하는 두 가지 방법을 보여 줍니다. 이때 인덱스 값이 프레임 번호 값과 다르다는 점에 유의해야 합니다.

```
f1.getDocumentDOM().getTimeline().setSelectedFrames(0, 9);  
f1.getDocumentDOM().getTimeline().setSelectedFrames(11, 14, false);  
f1.getDocumentDOM().getTimeline().setSelectedFrames([0, 0, 9]);  
f1.getDocumentDOM().getTimeline().setSelectedFrames([0, 11, 14], false);
```

다음 예제에서는 먼저 선택한 프레임의 배열을 **savedSelectionList** 변수에 저장한 다음 나중에 명령이나 사용자 인터페이스에 의해 선택한 프레임이 변경되면 코드에서 이 배열을 사용하여 이러한 프레임을 다시 선택합니다.

```
var savedSelectionList = f1.getDocumentDOM().getTimeline().getSelectedFrames();  
// Do something that changes the selection.  
f1.getDocumentDOM().getTimeline().setSelectedFrames(savedSelectionList);
```

참고 사항

[timeline.getSelectedFrames\(\)](#)

timeline.setSelectedLayers()

지원 버전

Flash MX 2004

사용법

```
timeline.setSelectedLayers(index [, bReplaceCurrentSelection])
```

매개 변수

index 선택할 레이어의 0부터 시작하는 인덱스

bReplaceCurrentSelection true로 설정하면 현재 선택한 영역을 바꾸고 false로 설정하면 현재 선택한 영역을 확장하는 부울 값. 기본값은 true입니다. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 레이어를 선택된 상태로 설정하고 지정된 레이어를 현재 레이어로 만듭니다. 레이어를 선택하면 레이어에 포함된 모든 프레임도 함께 선택됩니다.

예제

다음 예제에서는 최상위 레이어를 선택합니다.

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(0);
```

다음 예제에서는 선택한 영역에 다음 레이어를 추가합니다.

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(1, false);
```

참고 사항

[timeline.getSelectedLayers\(\)](#)

timeline.showLayerMasking()

지원 버전

Flash MX 2004

사용법

```
timeline.showLayerMasking([layer])
```

매개 변수

레이어 제작하는 동안 마스크를 보여 주기 위한 마스크 또는 마스크 처리된 레이어의 0부터 시작하는 인덱스. 이 매개 변수는 선택 사항입니다.

반환값

없음

설명

메서드, 마스크 및 마스크 처리된 레이어를 잠가 제작하는 동안 레이어 마스크를 표시합니다. 이 메서드는 지정된 레이어가 없는 경우 현재 레이어를 사용합니다. 마스크 또는 마스크 처리 유형이 아닌 레이어에서 이 메서드를 사용하면 [출력] 패널에 오류가 표시됩니다.

예제

다음 예제에서는 제작하는 동안 첫 번째 레이어의 레이어 마스크가 표시되도록 지정합니다.

```
fl.getDocumentDOM().getTimeline().showLayerMasking(0);
```

timeline.startPlayback()

지원 버전

Flash Professional CS5

사용법

```
timeline.startPlayback();
```

반환값

없음

설명

메서드, 현재 재생 중이면 타임라인의 자동 재생을 시작합니다. 이 메서드는 SWF 패널과 함께 사용하여 제작 환경에서 타임라인의 재생을 제어할 수 있습니다.

예제

다음 예제는 타임라인의 재생을 시작합니다.

```
fl.getDocumentDOM().getTimeline().startPlayback();
```

timeline.stopPlayback()

지원 버전

Flash Professional CS5

사용법

```
timeline.stopPlayback();
```

반환값

없음

설명

메서드, 현재 재생 중이면 타임라인의 자동 재생을 중지합니다. 이 메서드는 SWF 패널과 함께 사용하여 제작 환경에서 타임라인의 재생을 제어할 수 있습니다.

예제

다음 예제는 타임라인의 재생을 중지합니다.

```
fl.getDocumentDOM().getTimeline().stopPlayback();
```

46장: ToolObj 객체

지원 버전

Flash MX 2004

설명

ToolObj 객체는 [도구] 패널 내의 개별 도구를 나타냅니다. ToolObj 객체에 액세스하려면 **Tools** 객체의 속성, 즉 `tools.toolObjs` 배열 또는 `tools.activeTool`을 사용합니다.

메서드 요약

ToolObj 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

참고: 다음 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

메서드	설명
<code>toolObj.enablePIControl()</code>	속성 관리자에서 지정된 컨트롤을 활성화하거나 비활성화합니다. 확장 가능한 도구를 만들 때만 사용됩니다.
<code>toolObj.setIcn()</code>	Flash [도구] 패널에서 도구 아이콘으로 사용할 PNG 파일을 식별합니다.
<code>toolObj.setMenuString()</code>	팝업 메뉴에 도구 이름으로 표시할 문자열을 설정합니다.
<code>toolObj.setOptionsFile()</code>	XML 파일을 도구와 연결합니다.
<code>toolObj.setPI()</code>	도구를 활성화할 때 사용할 특정 속성 관리자를 설정합니다.
<code>toolObj.setToolName()</code>	[도구] 패널의 구성을 위한 도구에 이름을 지정합니다.
<code>toolObj.setToolTip()</code>	마우스가 도구 아이콘 위에 있을 때 표시되는 도구 설명을 설정합니다.
<code>toolObj.showPIControl()</code>	속성 관리자에서 컨트롤을 표시하거나 숨깁니다.
<code>toolObj.showTransformHandles()</code>	도구가 활성화되어 있을 때 자유 변형 핸들이 표시되어야 함을 나타내기 위해 확장 가능한 도구 JavaScript 파일의 <code>configureTool()</code> 메서드에서 호출됩니다.

속성 요약

ToolObj 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>toolObj.depth</code>	[도구] 패널의 팝업 메뉴에서 도구의 깊이를 지정하는 정수
<code>toolObj.iconID</code>	도구의 리소스 ID를 지정하는 정수
<code>toolObj.position</code>	읽기 전용, [도구] 패널의 도구 위치를 지정하는 정수

toolObj.depth

지원 버전

Flash MX 2004

사용법

`toolObj.depth`

설명

읽기 전용 속성, [도구] 패널의 팝업 메뉴에서 도구의 깊이를 지정하는 정수입니다. 이 속성은 확장 가능한 도구를 만들 때만 사용됩니다.

예제

다음 예제에서는 도구의 깊이를 1로 지정합니다. 이는 [도구] 패널에서 도구의 한 레벨 아래로 지정하는 것을 의미합니다.

```
fl.tools.activeTool.depth = 1;
```

toolObj.enablePIControl()

지원 버전

Flash MX 2004

사용법

`toolObj.enablePIControl(control, bEnable)`

매개 변수

control 활성화하거나 비활성화할 도구의 이름을 지정하는 문자열입니다. 사용할 수 있는 값은 이 도구에서 호출하는 속성 관리자에 따라 달라집니다([toolObj.setPI\(\)](#) 참조).

모양 속성 관리자에는 다음과 같은 컨트롤이 있습니다.

stroke	fill
--------	------

텍스트 속성 관리자에는 다음과 같은 컨트롤이 있습니다.

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

무비 속성 관리자에는 다음과 같은 컨트롤이 있습니다.

size	publish	background
framerate	player	profile

bEnable 컨트롤을 활성화할지 여부(true 또는 false)를 지정하는 부울 값

반환값

없음

설명

메서드, 속성 관리자에서 지정된 컨트롤을 활성화하거나 비활성화합니다. 확장 가능한 도구를 만들 때만 사용됩니다.

예제

확장 가능한 도구 JavaScript 파일에 있는 다음 명령은 해당 도구의 속성 관리자에 획 옵션이 표시되지 않도록 설정합니다.

```
theTool.enablePIControl("stroke",false);
```

toolObj.iconID

지원 버전

Flash MX 2004

사용법

toolObj.iconID

설명

읽기 전용 속성, 값이 -1인 정수입니다. 이 속성은 확장 가능한 도구를 만들 때만 사용됩니다. iconID 값이 -1이면 Flash가 도구 아이콘을 찾으려 시도하지 않습니다. 대신 도구 스크립트에서 [도구] 패널에 표시할 아이콘을 지정해야 합니다. 자세한 내용은 [toolObj.setIcon\(\)](#)을 참조하십시오.

예제

다음 예제에서는 toolIconID 변수에 값 -1(현재 도구의 아이콘 ID)을 할당합니다.

```
var toolIconID = fl.tools.activeTool.iconID
```

toolObj.position

지원 버전

Flash MX 2004

사용법

toolObj.position

설명

읽기 전용 속성, [도구] 패널의 도구 위치를 지정하는 정수입니다. 이 속성은 확장 가능한 도구를 만들 때만 사용됩니다.

예제

도구 JavaScript 파일의 mouseDown() 메서드에 있는 다음 명령은 [도구] 패널의 도구 위치를 [출력] 패널에 정수로 표시합니다.

```
myToolPos = fl.tools.activeTool.position;  
fl.trace(myToolPos);
```


toolObj.setIcon()

지원 버전

Flash MX 2004

사용법

```
toolObj.setIcon(file)
```

매개 변수

file 아이콘으로 사용할 PNG 파일의 이름을 지정하는 문자열입니다. PNG 파일은 JSFL 파일과 같은 폴더에 배치해야 합니다.

반환값

없음

설명

메서드, [도구] 패널에서 도구 아이콘으로 사용할 PNG 파일을 식별합니다. 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

예제

다음 예제에서는 PolyStar.png 파일에 있는 이미지가 PolyStar라는 도구의 아이콘으로 사용되도록 지정합니다. 이 코드는 샘플 PolyStar.jsfl 파일에서 가져온 것입니다(12페이지의 “다각형 별 도구 샘플” 참조).

```
theTool = fl.tools.activeTool;  
theTool.setIcon("PolyStar.png");
```

toolObj.setMenuString()

지원 버전

Flash MX 2004

사용법

```
toolObj.setMenuString(menuStr)
```

매개 변수

menuStr 팝업 메뉴에 도구의 이름으로 표시할 이름을 지정하는 문자열

반환값

없음

설명

메서드, 팝업 메뉴에 도구의 이름으로 표시할 문자열을 설정합니다. 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

예제

다음 예제에서는 theTool이라는 도구가 자신의 팝업 메뉴에 "PolyStar Tool"이라는 이름을 표시하도록 지정합니다. 이 코드는 샘플 PolyStar.jsfl 파일에서 가져온 것입니다(12페이지의 “다각형 별 도구 샘플” 참조).

```
theTool = fl.tools.activeTool;  
theTool.setMenuString("PolyStar Tool");
```

toolObj.setOptionsFile()

지원 버전

Flash MX 2004

사용법

```
toolObj.setOptionsFile(xmlFile)
```

매개 변수

xmlFile 도구의 옵션에 대해 설명하는 XML 파일의 이름을 지정하는 문자열입니다. XML 파일은 JSFL 파일과 같은 폴더에 배치해야 합니다.

반환값

없음

설명

메서드, XML 파일을 도구와 연결합니다. 파일은 속성 관리자의 [옵션] 버튼에 의해 호출되는 모달 패널에 옵션이 나타나도록 지정합니다. 보통 JSFL 파일 내의 configureTool() 함수에서 이 메서드를 사용합니다. [configureTool\(\)](#)을 참조하십시오.

예를 들어, PolyStar.xml 파일은 [다각형] 도구와 연관된 세 가지 옵션을 지정합니다.

```
<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>
```

예제

다음 예제에서는 PolyStar.xml이라는 파일을 현재 활성화된 도구와 연결합니다. 이 코드는 샘플 PolyStar.jsfl 파일에서 가져온 것입니다(12페이지의 “다각형 별 도구 샘플” 참조).

```
theTool = fl.tools.activeTool;
theTool.setOptionsFile("PolyStar.xml");
```

toolObj.setPI()

지원 버전
Flash MX 2004

사용법
`toolObj.setPI(pi)`

매개 변수
`pi` 이 도구를 호출할 속성 관리자를 지정하는 문자열

반환값
없음

설명
메서드, 도구를 활성화할 때 사용할 속성 관리자를 지정합니다. 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다. 사용할 수 있는 값에는 "shape"(기본값), "text" 및 "movie"가 있습니다.

예제
다음 예제에서는 도구를 활성화할 때 모양 속성 관리자를 사용하도록 지정합니다. 이 코드는 샘플 `PolyStar.jsfl` 파일에서 가져온 것입니다(12페이지의 “[다각형 별 도구 샘플](#)” 참조).

```
theTool = fl.tools.activeTool;  
theTool.setPI("shape");
```

toolObj.setToolName()

지원 버전
Flash MX 2004

사용법
`toolObj.setToolName(name)`

매개 변수
`name` 도구의 이름을 지정하는 문자열

반환값
없음

설명
메서드, [도구] 패널의 구성을 위한 도구에 이름을 지정합니다. 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다. 이 이름은 Flash에서 [도구] 패널을 구성하기 위해 읽는 XML 레이아웃 파일에만 사용되고 Flash 사용자 인터페이스에는 나타나지 않습니다.

예제
다음 예제에서는 `theTool`이라는 도구에 `polystar`라는 이름을 지정합니다. 이 코드는 샘플 `PolyStar.jsfl` 파일에서 가져온 것입니다(12페이지의 “[다각형 별 도구 샘플](#)” 참조).

```
theTool = fl.tools.activeTool;  
theTool.setToolName("polystar");
```

toolObj.setToolTip()

지원 버전
Flash MX 2004

사용법
toolObj.setToolTip(toolTip)

매개 변수
toolTip 도구에 사용할 도구 설명을 지정하는 문자열

반환값
없음

설명
메서드, 마우스가 도구 아이콘 위에 있을 때 표시되는 도구 설명을 설정합니다. 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

예제
다음 예제에서는 도구에 대한 도구 설명을 PolyStar Tool로 지정합니다. 이 코드는 샘플 PolyStar.jsfl 파일에서 가져온 것입니다 (12페이지의 “[다각형 별 도구 샘플](#)” 참조).

```
theTool = fl.tools.activeTool;  
theTool.setToolTip("PolyStar Tool");
```

toolObj.showPControl()

지원 버전
Flash MX 2004

사용법
toolObj.showPControl(control, bShow)

매개 변수
control 표시하거나 숨길 컨트롤의 이름을 지정하는 문자열입니다. 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다. 사용할 수 있는 값은 이 도구에서 호출하는 속성 관리자에 따라 달라집니다([toolObj.setPI\(\)](#) 참조).

모양 속성 관리자에게는 다음과 같은 컨트롤이 있습니다.

stroke	fill
--------	------

텍스트 속성 관리자에게는 다음과 같은 컨트롤이 있습니다.

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

무비 속성 관리자에는 다음과 같은 컨트롤이 있습니다.

size	publish	background
framerate	player	profile

bShow 지정된 컨트롤을 표시할 것인지(true) 아니면 숨길 것인지(false)를 결정하는 부울 값

반환값

없음

설명

메서드, 속성 관리자에서 컨트롤을 표시하거나 숨깁니다. 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

예제

확장 가능한 도구 JavaScript 파일에 있는 다음 명령은 해당 도구의 속성 관리자에 채우기 옵션이 표시되지 않도록 설정합니다.

```
fl.tools.activeTool.showPIControl("fill", false);
```

toolObj.showTransformHandles()

지원 버전

Flash MX 2004

사용법

```
toolObj.showTransformHandles(bShow)
```

매개 변수

bShow 현재 도구에 대한 자유 변형 핸들을 표시할 것인지(true) 아니면 숨길 것인지(false)를 결정하는 부울 값

반환값

없음

설명

메서드, 도구가 활성화되어 있을 때 자유 변형 핸들이 표시되어야 함을 나타내기 위해 확장 가능한 도구 JavaScript 파일의 `configureTool()` 메서드에서 호출됩니다. 이 메서드는 확장 가능한 도구를 만들 때만 사용됩니다.

예제

`configureTool()`을 참조하십시오.

47장: Tools 객체

지원 버전

Flash MX 2004

설명

Tools 객체는 Flash 객체(`fl.tools`)에서 액세스할 수 있습니다. `tools.toolObjs` 속성은 ToolObj 객체의 배열을 포함하고 `tools.activeTool` 속성은 현재 활성화된 도구의 ToolObj 객체를 반환합니다. (자세한 내용은 [ToolObj 객체](#) 및 14페이지의 “[최상위 함수 및 메서드](#)”의 확장 가능한 도구 목록을 참조하십시오.)

참고: 다음은 확장 가능한 도구를 만들 때만 사용되는 메서드와 속성입니다.

메서드 요약

Tools 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
<code>tools.constrainPoint()</code>	두 점을 가져와 새로 조절되었거나 제한된 점을 반환합니다.
<code>tools.getKeyDown()</code>	가장 최근에 누른 키를 반환합니다.
483페이지의 " <code>tools.setCreatingBbox()</code> "	PLACEHOLDER
<code>tools.setCursor()</code>	포인터를 지정된 모양으로 설정합니다.
<code>tools.snapPoint()</code>	점을 입력으로 가져오고 조절할 수 있거나 가장 가까이 있는 기하학적 객체에 물릴 수 있는 점을 반환합니다.

속성 요약

Tools 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
<code>tools.activeTool</code>	읽기 전용, 현재 활성화된 도구에 대한 ToolObj 객체 를 반환합니다.
<code>tools.altIsDown</code>	읽기 전용, Alt 키가 눌러져 있는지 여부를 나타내는 부울 값
<code>tools.ctrlIsDown</code>	읽기 전용, Ctrl 키가 눌러져 있는지 여부를 나타내는 부울 값
<code>tools.mouselsDown</code>	읽기 전용, 왼쪽 마우스 버튼이 현재 눌러져 있는지 여부를 나타내는 부울 값
<code>tools.penDownLoc</code>	읽기 전용, 스테이지에서 마지막으로 수행된 마우스 누름 이벤트의 위치를 나타내는 점
<code>tools.penLoc</code>	읽기 전용, 마우스의 현재 위치를 나타내는 점
<code>tools.shiftIsDown</code>	읽기 전용, Shift 키가 눌러져 있는지 여부를 나타내는 부울 값
<code>tools.toolObjs</code>	읽기 전용, ToolObj 객체의 배열

tools.activeTool

지원 버전

Flash MX 2004

사용법

`tools.activeTool`

설명

읽기 전용 속성, 현재 활성화된 도구에 대한 **ToolObj** 객체를 반환합니다.

예제

다음 예제에서는 현재 활성화된 도구를 나타내는 객체를 `theTool` 변수에 저장합니다.

```
var theTool = fl.tools.activeTool;
```

tools.altIsDown

지원 버전

Flash MX 2004

사용법

`tools.altIsDown`

설명

읽기 전용 속성, Alt 키가 눌러져 있는지 여부를 나타내는 부울 값입니다. Alt 키가 눌러져 있으면 `true`이고 그렇지 않으면 `false`입니다.

예제

다음 예제에서는 Alt 키가 눌러져 있는지 여부를 결정합니다.

```
var isAltDown = fl.tools.altIsDown;
```

tools.constrainPoint()

지원 버전

Flash MX 2004

사용법

`tools.constrainPoint(pt1, pt2)`

매개 변수

`pt1, pt2` 시작 클릭 점과 드래그 점을 지정하는 점

반환값

새로 조절되었거나 제한된 점입니다.

설명

메서드, 두 점을 취하고 새로 조절되었거나 제한된 점을 반환합니다. 명령을 실행할 때 **Shift** 키가 눌러져 있으면 반환된 점은 45° 제한(화살촉이 있는 선 등에 유용함)을 지키거나 [사각형] 도구로 정사각형을 잡아당겨 늘릴 때처럼 객체의 종횡비를 유지하도록 제한됩니다.

예제

다음 예제에서는 제한된 점을 반환합니다.

```
pt2 = fl.tools.constrainPoint(pt1, tempPt);
```

tools.ctrlIsDown

지원 버전

Flash MX 2004

사용법

```
tools.ctrlIsDown
```

설명

읽기 전용 속성, Ctrl 키가 눌러져 있으면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값입니다.

예제

다음 예제에서는 Ctrl 키가 눌러져 있는지 여부를 결정합니다.

```
var isCtrlDown = fl.tools.ctrlIsDown;
```

tools.getKeyDown()

지원 버전

Flash MX 2004

사용법

```
tools.getKeyDown()
```

매개 변수

없음

반환값

키의 정수 값

설명

메서드, 가장 최근에 누른 키를 반환합니다.

예제

다음 예제에서는 가장 최근에 누른 키의 정수 값을 표시합니다.

```
var theKey = fl.tools.getKeyDown();  
fl.trace(theKey);
```

tools.mouseIsDown

지원 버전
Flash MX 2004

사용법
`tools.mouseIsDown`

설명
읽기 전용 속성, 현재 왼쪽 마우스 버튼이 눌러져 있으면 `true`를 반환하고 그렇지 않은 경우 `false`를 반환하는 부울 값입니다.

예제
다음 예제에서는 왼쪽 마우스 버튼이 눌러져 있는지 여부를 결정합니다.

```
var isMouseDown = fl.tools.mouseIsDown;
```

tools.penDownLoc

지원 버전
Flash MX 2004

사용법
`tools.penDownLoc`

설명
읽기 전용 속성, 스테이지에서 마지막으로 수행된 마우스 누름 이벤트의 위치를 나타내는 점입니다. `tools.penDownLoc` 속성은 마우스 포인터의 `x,y` 위치에 해당하는 `x` 및 `y`의 두 가지 속성으로 구성됩니다.

예제
다음 예제에서는 스테이지에서 마지막으로 수행된 마우스 누름 이벤트의 위치를 결정하고 [출력] 패널에 `x` 및 `y` 값을 표시합니다.

```
var pt1 = fl.tools.penDownLoc;  
fl.trace("x,y location of last mouseDown event was " + pt1.x + ", " + pt1.y)
```

참고 사항
[tools.penLoc](#)

tools.penLoc

지원 버전
Flash MX 2004

사용법
`tools.penLoc`

설명

읽기 전용 속성, 마우스 포인터의 현재 위치를 나타내는 점입니다. `tools.penLoc` 속성은 마우스 포인터의 `x,y` 위치에 해당하는 `x` 및 `y`의 두 가지 속성으로 구성됩니다.

예제

다음 예제에서는 마우스의 현재 위치를 결정합니다.

```
var tempPt = fl.tools.penLoc;
```

참고 사항

[tools.penDownLoc](#)

tools.setCreatingBbox()

지원 버전

Flash 11

사용법

```
tools.setCreatingBbox()
```

매개 변수

자리 표시자 다음 목록에 설명된 대로 포인터의 모양을 정의하는 정수

반환값

없음

설명

메서드, 포인터를 지정된 모양으로 설정합니다.

예제

다음 예제에서는 포인터를 검정색 화살표로 설정합니다.

```
fl.tools.setCursor(1);
```

tools.setCursor()

지원 버전

Flash MX 2004

사용법

```
tools.setCursor(cursor)
```

매개 변수

커서 다음 목록에 설명된 대로 포인터의 모양을 정의하는 정수

- 0 = 플러스 커서(+)
- 1 = 검정색 화살표

- 2 = 흰색 화살표
- 3 = 4방향 화살표
- 4 = 양방향 수평 화살표
- 5 = 양방향 수직 화살표
- 6 = X
- 7 = 손 모양 커서

반환값

없음

설명

메서드, 포인터를 지정된 모양으로 설정합니다.

예제

다음 예제에서는 포인터를 검정색 화살표로 설정합니다.

```
fl.tools.setCursor(1);
```

tools.shiftIsDown

지원 버전

Flash MX 2004

사용법

```
tools.shiftIsDown
```

설명

읽기 전용 속성, Shift 키가 눌러져 있으면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값입니다.

예제

다음 예제에서는 Shift 키가 눌러져 있는지 여부를 결정합니다.

```
var isShiftDown = fl.tools.shiftIsDown;
```

tools.snapPoint()

지원 버전

Flash MX 2004

사용법

```
tools.snapPoint(pt)
```

매개 변수

pt 물리기 점을 반환할 점의 위치를 지정합니다.

반환값

가장 가까이 있는 기하학적 객체로 조정하거나 물릴 수 있는 새 점입니다.

설명

메서드, 점을 입력으로 가져오고 조절할 수 있거나 가장 가까이 있는 기하학적 객체에 물릴 수 있는 새 점을 반환합니다. Flash 사용자 인터페이스의 [보기] 메뉴에서 물리기 기능이 비활성화되어 있으면 원래 점이 반환됩니다.

예제

다음 예제에서는 가장 가까이 있는 기하학적 객체에 물릴 수 있는 새 점을 반환합니다.

```
var theSnapPoint = fl.tools.snapPoint(pt1);
```

tools.toolObjs

지원 버전

Flash MX 2004

사용법

```
tools.toolObjs
```

설명

읽기 전용 속성, ToolObj 객체의 배열([ToolObj 객체 참조](#))입니다.

48장: Vertex 객체

지원 버전

Flash MX 2004

설명

Vertex 객체는 좌표 데이터가 포함된 모양 데이터 구조의 일부입니다.

메서드 요약

Vertex 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
vertex.getHalfEdge()	이 꼭지점을 공유하는 HalfEdge 객체를 가져옵니다.
vertex.setLocation()	꼭지점의 위치를 설정합니다.

속성 요약

Vertex 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
vertex.x	읽기 전용, 픽셀 단위로 지정된 꼭지점의 x 위치
vertex.y	읽기 전용, 픽셀 단위로 지정된 꼭지점의 y 위치

vertex.getHalfEdge()

지원 버전

Flash MX 2004

사용법

```
vertex.getHalfEdge()
```

매개 변수

없음

반환값

[HalfEdge](#) 객체

설명

메서드, 이 꼭지점을 공유하는 [HalfEdge](#) 객체를 가져옵니다.

예제

다음 예제에서는 같은 꼭지점을 공유하는 다른 반쪽 가장자리를 가져오는 방법을 보여 줍니다.

```
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()

지원 버전

Flash MX 2004

사용법

```
vertex.setLocation(x, y)
```

매개 변수

x 꼭지점을 놓을 x 좌표를 픽셀 단위로 지정하는 부동 소수점 값

y 꼭지점을 놓을 y 좌표를 픽셀 단위로 지정하는 부동 소수점 값

반환값

없음

설명

메서드, 꼭지점의 위치를 설정합니다. 이 메서드를 사용하려면 먼저 [shape.beginEdit\(\)](#)을 호출해야 합니다.

예제

다음 예제에서는 꼭지점을 원점으로 설정합니다.

```
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

지원 버전

Flash MX 2004

사용법

```
vertex.x
```

설명

읽기 전용 속성, 픽셀 단위로 지정된 꼭지점의 x 위치입니다.

예제

다음 예제에서는 꼭지점의 **x** 및 **y** 값 위치를 [출력] 패널에 표시합니다.

```
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

지원 버전

Flash MX 2004

사용법

vertex.y

설명

읽기 전용 속성, 픽셀 단위로 된 꼭지점의 **y** 위치입니다.

예제

[vertex.x](#)를 참조하십시오.

49장: VideoItem 객체

상속 [Item 객체](#) > VideoItem 객체

지원 버전
Flash MX 2004

설명
VideoItem 객체는 [Item 객체](#)의 하위 클래스입니다.

메서드 요약
Item 객체 메서드 외에도 VideoItem 객체에는 다음과 같은 메서드가 있습니다.

속성	설명
videoItem.exportToFLV()	지정된 항목을 FLV 파일로 내보냅니다.

속성 요약
Item 객체 속성 외에도 VideoItem 객체에는 다음과 같은 속성을 사용할 수 있습니다.

속성	설명
videoItem.fileLastModifiedDate	읽기 전용, 1970년 1월 1일과 파일을 라이브러리로 가져왔을 때 디스크에 있었던 원본 파일의 수정 날짜 사이에 경과된 시간(초)을 나타내는 16진수를 포함하는 문자열
videoItem.sourceFileExists	읽기 전용, 라이브러리로 가져온 파일이 아직 가져온 위치에 있는지 여부를 지정하는 부울 값
videoItem.sourceFilesCurrent	읽기 전용, 라이브러리 항목의 파일 수정 날짜가 가져온 파일을 디스크에서 수정한 날짜와 같은지 여부를 지정하는 부울 값
videoItem.sourceFilePath	읽기 전용, 비디오 항목에 대한 경로를 지정하는 문자열
videoItem.videoType	읽기 전용, 항목이 나타내는 비디오 유형을 지정하는 문자열

videoItem.exportToFLV()

지원 버전
Flash CS4 Professional

사용법
`videoItem.exportToFLV(fileURI)`

매개 변수
fileURI 내보낸 파일의 경로와 이름을 지정하는 문자열(file:/// URI로 표시됨)

반환값
파일이 성공적으로 내보내지면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값

설명

메서드, 지정된 항목을 FLV 파일로 내보냅니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 비디오 항목이라고 가정할 경우 FLV 파일로 내보냅니다.

```
var videoFileURL = "file:///C:/out.flv";  
var libItem = fl.getDocumentDOM().library.items[0];  
libItem.exportToFLV(videoFileURL);
```

videoItem.fileLastModifiedDate

지원 버전

Flash CS4 Professional

사용법

videoItem.fileLastModifiedDate

설명

읽기 전용 속성, 1970년 1월 1일과 파일을 라이브러리로 가져왔을 때 디스크에 있었던 원본 파일의 수정 날짜 사이에 경과된 시간(초)을 나타내는 16진수를 포함하는 문자열입니다. 파일이 없으면 이 값은 "00000000"입니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 비디오 항목으로 가정할 경우 위에서 설명한 대로 16진수를 표시합니다.

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

참고 사항

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

videoItem.sourceFileExists

지원 버전

Flash CS4 Professional

사용법

videoItem.sourceFileExists

설명

읽기 전용 속성, 라이브러리로 가져온 파일이 아직 가져온 위치에 있으면 true를 반환하고 그렇지 않으면 false를 반환하는 부울 값

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 비디오 항목이라고 가정할 경우 라이브러리로 가져온 파일이 아직 있으면 "true"를 표시합니다.

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

참고 사항

[videoItem.sourceFilesCurrent](#), [videoItem.sourceFilePath](#)

videoItem.sourceFilesCurrent

지원 버전

Flash CS4 Professional

사용법

`videoItem.sourceFileIsCurrent`

설명

읽기 전용 속성, 라이브러리 항목의 파일 수정 날짜가 가져온 파일을 디스크에서 수정한 날짜와 같으면 `true`를 반환하고 그렇지 않으면 `false`를 반환하는 부울 값입니다.

예제

다음 코드에서는 라이브러리의 첫 번째 항목을 비디오 항목이라고 가정할 경우 가져온 파일을 가져온 이후 디스크에서 수정하지 않은 경우 `"true"`를 표시합니다.

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

참고 사항

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

videoItem.sourceFilePath

지원 버전

Flash 8

사용법

`videoItem.sourceFilePath`

설명

읽기 전용 속성, 비디오 항목에 대한 경로를 지정하는 문자열(`file:///URI`로 표시됨)입니다.

예제

다음 예제에서는 유형이 `video`인 라이브러리에 있는 모든 항목의 이름과 소스 파일 경로를 표시합니다.

```
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);  
  }  
}
```

참고 사항

[videoItem.sourceFileExists](#)

videoItem.videoType

지원 버전

Flash 8

사용법

videoItem.videoType

설명

읽기 전용 속성, 항목이 나타내는 비디오 유형을 지정하는 문자열입니다. 사용 가능한 값에는 "embedded video", "linked video" 및 "video"가 있습니다.

예제

다음 예제에서는 유형이 video인 라이브러리에 있는 모든 항목의 이름과 유형을 표시합니다.

```
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);  
  }  
}
```

50장: XMLUI 객체

지원 버전

Flash MX 2004

설명

Flash 8에서는 XUL(XML 사용자 인터페이스 언어)의 하위 집합으로 작성된 사용자 정의 대화 상자를 지원합니다. XMLUI(XML 사용자 인터페이스) 대화 상자는 확장성을 사용하여 사용자가 만든 기능에 대한 사용자 인터페이스를 제공하기 위해 명령과 비헤이비어 같은 다양한 Flash 기능에서 사용할 수 있습니다. XMLUI 객체에서는 XMLUI 대화 상자의 속성을 가져와서 설정하고, 수락하거나 취소함으로써 해당 대화 상자에서 벗어나도록 할 수 있습니다. XMLUI 메서드는 버튼의 oncommand 핸들러 같은 콜백에 사용됩니다.

`document.xmlPanel()` 메서드를 사용하면 `dialog.xml` 파일을 작성하고 JavaScript API에서 이 파일을 호출할 수 있습니다. 현재 XMLUI 대화 상자를 나타내는 객체를 가져오려면 `fl.xmlui`를 사용합니다.

메서드 요약

XMLUI 객체에는 다음과 같은 메서드를 사용할 수 있습니다.

메서드	설명
<code>xmlui.accept()</code>	수락 상태일 경우 현재 XMLUI 대화 상자를 닫습니다.
<code>xmlui.cancel()</code>	취소 상태일 경우 현재 XMLUI 대화 상자를 닫습니다.
<code>xmlui.get()</code>	현재 XMLUI 대화 상자의 특정 속성 값을 가져옵니다.
<code>xmlui.getControlItemElement()</code>	특정 컨트롤에 대한 현재 컨트롤 항목을 반환합니다.
<code>xmlui.setEnabled()</code>	컨트롤의 활성화 또는 비활성화(희미하게 표시됨) 여부를 지정하는 부울 값을 반환합니다.
<code>xmlui.setVisible()</code>	컨트롤의 표시 또는 숨김 여부를 지정하는 부울 값을 반환합니다.
<code>xmlui.set()</code>	현재 XMLUI 대화 상자의 특정 속성 값을 수정합니다.
<code>xmlui.setControlItemElement()</code>	현재 항목에 대한 레이블 및 값을 설정합니다.
<code>xmlui.setControlItemElements()</code>	현재 항목의 레이블, 값 쌍을 설정합니다.
<code>xmlui.setEnabled()</code>	컨트롤을 활성화 또는 비활성화(희미하게 표시)합니다.
<code>xmlui.setVisible()</code>	컨트롤을 표시하거나 숨깁니다.

xmlui.accept()

지원 버전

Flash MX 2004

사용법

```
xmlui.accept()
```

매개 변수

없음

반환값

없음

설명

메서드, 수락 상태일 경우 현재 XMLUI 대화 상자를 닫습니다. 이는 사용자가 [확인] 버튼을 클릭하는 것과 같습니다.

참고 사항

[fl.xmlui](#), [document.xmlPanel\(\)](#), [xmlui.cancel\(\)](#)

xmlui.cancel()

지원 버전

Flash MX 2004

사용법

```
xmlui.cancel()
```

매개 변수

없음

반환값

없음

설명

메서드, 취소 상태일 경우 현재 XMLUI 대화 상자를 닫습니다. 이는 사용자가 [취소] 버튼을 클릭하는 것과 같습니다.

참고 사항

[fl.xmlui](#), [document.xmlPanel\(\)](#), [xmlui.accept\(\)](#)

xmlui.get()

지원 버전

Flash MX 2004

사용법

```
xmlui.get(controlPropertyName)
```

매개 변수

`controlPropertyName` 값을 가져올 XMLUI 속성의 이름을 지정하는 문자열

반환값

특정 속성의 값을 나타내는 문자열입니다. `true` 또는 `false`의 부울 값이 필요한 경우 `"true"` 또는 `"false"` 문자열이 반환됩니다.

설명

메서드, 현재 XMLUI 대화 상자의 특정 속성 값을 가져옵니다.

예제

다음 예제에서는 URL이라는 속성의 값을 반환합니다.

```
fl.xmlui.get("URL");
```

참고 사항

[fl.xmlui](#), [document.xmlPanel\(\)](#), [xmlui.getControlItemElement\(\)](#), [xmlui.set\(\)](#)

xmlui.getControlItemElement()

지원 버전

Flash 8

사용법

```
xmlui.getControlItemElement(controlPropertyName)
```

매개 변수

controlPropertyName 컨트롤 항목 요소를 가져올 속성을 지정하는 문자열

반환값

controlPropertyName에 의해 지정된 컨트롤의 현재 컨트롤 항목을 나타내는 객체

설명

메서드, **controlPropertyName**에서 지정된 컨트롤에 대해 ListBox 또는 ComboBox 컨트롤에서 선택된 행의 레이블 및 값을 반환합니다.

예제

다음 예제에서는 myListBox 컨트롤에 대해 현재 선택된 행의 레이블 및 값을 반환합니다.

```
var elem = new Object();  
elem = fl.xmlui.getControlItemElement("myListBox");  
fl.trace("label = " + elem.label + " value = " + elem.value);
```

참고 사항

[fl.xmlui](#), [document.xmlPanel\(\)](#), [xmlui.get\(\)](#), [xmlui.setControlItemElement\(\)](#), [xmlui.setControlItemElements\(\)](#)

xmlui.getEnabled()

지원 버전

Flash 8

사용법

```
xmlui.getEnabled(controlID)
```

매개 변수

controlID 상태를 검색할 컨트롤의 ID 특성을 지정하는 문자열

반환값

컨트롤이 활성화된 경우 true를 반환하고 그렇지 않은 경우 false를 반환하는 부울 값

설명

메서드, 컨트롤의 활성화 또는 비활성화(희미하게 표시됨) 여부를 지정하는 부울 값을 반환합니다.

예제

다음 예제에서는 ID 특성이 myListBox인 컨트롤의 활성화 여부를 나타내는 값을 반환합니다.

```
var isEnabled = fl.xmlui.setEnabled("myListBox");  
fl.trace(isEnabled);
```

참고 사항

[fl.xmlui](#), [document.xmlPanel\(\)](#), [xmlui.setEnabled\(\)](#)

xmlui.getVisible()

지원 버전

Flash 8

사용법

```
xmlui.getVisible(controlID)
```

매개 변수

controlID 가시성 상태를 검색할 컨트롤의 ID 특성을 지정하는 문자열

반환값

컨트롤을 표시하려면 true로 지정하고 컨트롤을 표시하지 않으려면(숨김) false로 지정하는 부울 값.

설명

메서드, 컨트롤을 표시하거나 숨길지 여부를 지정하는 부울 값을 반환합니다.

예제

다음 예제에서는 ID 특성이 myListBox인 컨트롤의 표시 여부를 나타내는 값을 반환합니다.

```
var isVisible = fl.xmlui.getVisible("myListBox");  
fl.trace(isVisible);
```

참고 사항

[xmlui.setVisible\(\)](#)

xmlui.set()

지원 버전

Flash MX 2004

사용법

```
xmlui.set(controlPropertyName, value)
```

매개 변수

controlPropertyName 수정할 XMLUI 속성의 이름을 지정하는 문자열

value XMLUI 속성에 설정할 값을 지정하는 문자열

반환값

없음

설명

메서드, 현재 XMLUI 대화 상자의 특정 속성 값을 수정합니다.

예제

다음 예제에서는 URL이라는 속성의 값을 www.adobe.com으로 설정합니다.

```
fl.xmlui.set("URL", "www.adobe.com");
```

참고 사항

[fl.xmlui](#), [document.xmlPanel\(\)](#), [xmlui.get\(\)](#), [xmlui.setControlItemElement\(\)](#), [xmlui.setControlItemElements\(\)](#)

xmlui.setControlItemElement()

지원 버전

Flash 8

사용법

```
xmlui.setControlItemElement(controlPropertyName, elementItem)
```

매개 변수

controlPropertyName 설정할 컨트롤 항목 요소를 지정하는 문자열

elementItem label이라는 문자열 속성과 value라는 선택적 문자열 속성이 있는 JavaScript 객체. value 속성이 없으면 자동으로 만들어지고 label과 동일한 값이 할당됩니다.

반환값

없음

설명

메서드, **controlPropertyName**에서 지정된 ListBox 또는 ComboBox 컨트롤에서 현재 선택된 행의 레이블 및 값을 설정합니다.

예제

다음 예제에서는 PhoneNumber라는 컨트롤 속성의 현재 항목에 대한 레이블 및 값을 설정합니다.

```
var elem = new Object();  
elem.label = "Fax";  
elem.value = "707-555-5555";  
fl.xmlui.setControlItemElement("PhoneNumber",elem);
```

참고 사항

[fl.xmlui](#), [document.xmlPanel\(\)](#), [xmlui.getControlItemElement\(\)](#), [xmlui.set\(\)](#), [xmlui.setControlItemElements\(\)](#)

xmlui.setControlItemElements()

지원 버전

Flash 8

사용법

```
xmlui.setControlItemElements(controlID, elementItemArray)
```

매개 변수

controlID 설정할 컨트롤의 ID 특성을 지정하는 문자열

elementItemArray label이라는 문자열 속성과 value라는 선택적 문자열 속성이 있는 각 JavaScript 객체의 배열. value 속성이 없으면 자동으로 만들어지고 label과 동일한 값이 할당됩니다.

반환값

없음

설명

메서드, **controlID**에서 지정된 ListBox 또는 ComboBox 컨트롤의 값을 지우고 목록이나 메뉴 항목을 **elementItemArray**에서 지정된 label, value 쌍으로 바꿉니다.

예제

다음 예제에서는 ID 특성이 myControlID인 컨트롤의 항목 레이블 및 값을 지정된 label,value 쌍으로 설정합니다.

```
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);
```

참고 사항

[xmlui.getControlItemElement\(\)](#), [xmlui.set\(\)](#), [xmlui.setControlItemElement\(\)](#)

xmlui.setEnabled()

지원 버전

Flash 8

사용법

```
xmlui.setEnabled(controlID, enable)
```

매개 변수

controlID 활성화하거나 비활성화하려는 컨트롤의 ID 특성을 지정하는 문자열입니다.

활성화 컨트롤을 활성화하려면 **true**로 지정하고 컨트롤을 비활성화(회미하게 표시)하려면 **false**로 지정하는 부울 값.

반환값

없음

설명

메서드, 컨트롤을 활성화 또는 비활성화(회미하게 표시)합니다.

예제

다음 예제에서는 ID 특성이 myControl인 컨트롤을 회미하게 표시합니다.

```
f1.xmlui.setEnabled("myControl", false);
```

참고 사항

[xmlui.setEnabled\(\)](#)

xmlui.setVisible()

지원 버전

Flash 8

사용법

```
xmlui.setVisible(controlID, visible)
```

매개 변수

controlID 표시하거나 숨길 컨트롤의 ID 특성을 지정하는 문자열

visible 컨트롤을 표시하려면 **true**로 지정하고 컨트롤을 숨기려면 **false**로 지정하는 부울 값.

반환값

없음

설명

메서드, 컨트롤을 표시하거나 숨깁니다.

예제

다음 예제에서는 ID 특성이 myControl인 컨트롤을 숨깁니다.

```
f1.xmlui.setVisible("myControl", false);
```

참고 사항

[xmlui.setVisible\(\)](#)

51장: C 레벨 확장성

이 장에서는 C 레벨 확장성 메커니즘에 대해 설명합니다. 이 메커니즘을 사용하면 JavaScript와 사용자 정의 C 코드의 조합을 통해 Adobe Flash CS4 Professional 확장성 파일을 구현할 수 있습니다. 이번 Flash 릴리스에서는 이 메커니즘이 변경되지 않았습니다.

확장성

확장을 구현하려면 C 언어를 사용하여 함수를 정의하고, 정의된 함수를 DLL(동적 연결 라이브러리)이나 공유 라이브러리에 하나로 묶고, 적절한 디렉토리에 이 라이브러리를 저장한 다음 Adobe Flash JavaScript API를 사용하여 JavaScript에서 함수를 호출합니다.

예를 들어, JavaScript보다 더 효율적으로 복잡한 계산을 수행하는 함수를 정의하여 성능을 향상시키거나 더욱 향상된 도구나 효과를 만들 수 있습니다.

이 확장성 메커니즘은 Adobe Dreamweaver CS3 API의 하위 집합입니다. Dreamweaver API에 익숙한 사용자는 C 레벨 확장성 메커니즘 API의 함수도 이해할 수 있습니다. 그러나 이 API와 Dreamweaver API 사이에는 다음과 같은 차이점이 있습니다.

- Dreamweaver API의 일부 명령어 이 API에는 없습니다.
- Dreamweaver API에서의 `wchar_t` 및 `char` 유형에 대한 모든 선언이 이 API에서는 문자열 전달 시 유니코드를 지원하기 위해 `unsigned short` 선언으로 구현됩니다.
- 이 API의 `JSVal JS_BytesToValue()` 함수가 Dreamweaver API에는 없습니다.
- DLL 또는 공유 라이브러리 파일의 저장 위치가 다릅니다(501페이지의 “C 함수 통합” 참조).

C 함수 통합

C 레벨 확장성 메커니즘을 사용하면 JavaScript와 C 코드의 조합을 통해 Flash 확장성 파일을 구현할 수 있습니다. 다음 단계는 이 기능을 구현하는 과정을 간략하게 보여 줍니다.

- 1 C 또는 C++ 언어를 사용하여 함수를 정의합니다.
- 2 정의된 함수를 DLL 파일(Windows)이나 공유 라이브러리(Macintosh)에 하나로 묶습니다.
- 3 다음과 같은 적절한 위치에 DLL 파일이나 라이브러리를 저장합니다.
 - Windows 7:
부트 드라이브\Users\사용자 이름\AppData\Adobe\Flash CS5 또는 CS5.5\언어\Configuration\External Libraries
 - Windows Vista:
부트 드라이브\Users\사용자 이름\Local Settings\Application Data\Adobe\Flash CS5 또는 CS5.5\언어\Configuration\External Libraries
 - Windows XP:
부트 드라이브\Documents and Settings\사용자 언어\Local Settings\Application Data\Adobe\Flash CS5 또는 CS5.5\언어\Configuration\External Libraries
 - Mac OS X:

Macintosh HD/Users/사용자 이름/Library/Application Support/Adobe/Flash CS5 또는 CS5.5/언어
/Configuration/External Libraries

4 함수를 호출하는 JSFL 파일을 만듭니다.

5 Flash 제작 환경에서 [명령] 메뉴를 사용하여 JSFL 파일을 실행합니다.

자세한 내용은 505페이지의 “DLL 구현 샘플”을 참조하십시오.

C 레벨 확장성 및 JavaScript 인터프리터

DLL 또는 공유 라이브러리의 C 코드는 다음과 같은 세 경우에 Flash JavaScript API와 상호 작용합니다.

- 시작 시 라이브러리 함수를 등록할 때
- JavaScript에서 C로 전달되는 인수의 압축을 풀기 위해 C 함수를 호출할 때
- C 함수가 값을 반환하기 전에, 반환값을 패키지화하기 위해

이러한 작업을 수행하려면 인터프리터에서 여러 가지 데이터 유형을 정의하고 API를 공개해야 합니다. 이 단원에서 설명하는 데이터 유형과 함수에 대한 정의는 mm_jsapi.h 파일에 나타납니다. 라이브러리가 제대로 작동하도록 하려면 다음 코드를 사용하여 라이브러리의 각 파일 위에 mm_jsapi.h 파일을 추가해야 합니다.

```
#include "mm_jsapi.h"
```

mm_jsapi.h 파일을 추가하면 MM_Environment 구조를 정의하는 mm_jsapi_environment.h 파일도 함께 추가됩니다.

mm_jsapi.h 파일의 사본을 가져오려면 샘플 ZIP 또는 SIT 파일에서 해당 사본을 추출하거나(505페이지의 “DLL 구현 샘플” 참조) mm_jsapi.h로 이름을 지정한 파일에 다음 코드를 복사합니다.

```
#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 구현 샘플

이 단원에서는 간단한 DLL 구현을 구성하는 방법을 보여 줍니다. 실제로 직접 DLL을 구성하지 않고도 이 과정이 어떻게 진행되는지 보려면 ExtendingFlash/dllSampleComputeSum 폴더에 있는 Samples.zip 파일에 제공된 DLL 샘플 파일을 설치하면 됩니다. (Samples.zip 파일 다운로드에 대한 자세한 내용은 12페이지의 “샘플 구현”을 참조하십시오.)

dllSampleComputeSum.dmg 또는 dllSampleComputeSum.zip 파일에서 샘플 파일을 추출하고 다음을 수행합니다.

- Configuration/Commands 디렉토리에 Sample.jsfl 파일을 저장합니다(2페이지의 “JSFL 파일 저장” 참조).
- Configuration/External Libraries 디렉토리에 Sample.dll 파일을 저장합니다(501페이지의 “C 함수 통합” 참조).
- Flash 제작 환경에서 명령 > 샘플을 선택합니다. JSFL 파일의 trace 명령문은 Sample.dll에 정의된 함수의 결과를 [출력] 패널에 보냅니다.

이 단원의 나머지 부분에서는 샘플의 개발에 대해 설명합니다. 이 경우 DLL에는 두 개의 숫자를 추가하는 하나의 함수만 포함됩니다. 다음 예제는 C 코드를 보여 줍니다.

```
// 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;
}
```

이 코드를 쓴 후 DLL 파일이나 공유 라이브러리를 구성하고 해당 Configuration/External Libraries 디렉토리에 저장합니다(501페이지의 “C 함수 통합” 참조). 그런 후, 다음 코드를 사용하여 JSFL 파일을 만들고 Configuration/Commands 디렉토리에 저장합니다(2페이지의 “JSFL 파일 저장” 참조).

```
// 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 );
```

DLL에 정의된 함수를 실행하려면 Flash 제작 환경에서 명령 > 샘플을 선택합니다.

데이터 유형

JavaScript 인터프리터는 이 단원에서 설명하는 데이터 유형을 정의합니다.

typedef struct JSContext JSContext

이 불투명 데이터 유형에 대한 포인터는 C 레벨 함수에 전달됩니다. API의 일부 함수는 이 포인터를 해당 인수 중 하나로 받아 들입니다.

typedef struct JSObject JSObject

이 불투명 데이터 유형에 대한 포인터는 C 레벨 함수에 전달됩니다. 이 데이터 유형은 객체(Array 객체 또는 다른 객체)를 나타 냅니다.

typedef struct jsval jsval

부동 소수점, 문자열 또는 객체에 대한 포인터 또는 정수를 포함할 수 있는 불투명 데이터 구조입니다. API의 함수 중에는 jsval 구조의 내용을 읽는 방식으로 함수 인수의 값을 읽을 수 있는 함수도 있고 jsval 구조를 작성하는 방식으로 함수의 반환값을 작성할 수 있는 함수도 있습니다.

typedef enum { JS_FALSE = 0, JS_TRUE = 1 } JSBool

부울 값을 저장하는 단순한 데이터 유형

C 레벨 API

C 레벨 확장성 API는 JSBool(*JSNative) 함수 서명과 다음 함수로 구성됩니다.

- 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)

설명

메서드, 다음과 같은 경우의 JavaScript 함수의 C 레벨 구현을 설명합니다.

- **cx** 포인터는 JavaScript API의 일부 함수에 전달해야 하는 불투명 JSContext 구조에 대한 포인터입니다. 이 변수에는 인터프리터의 실행 컨텍스트가 포함됩니다.

- **obj** 포인터는 해당 컨텍스트에서 스크립트가 실행되는 객체에 대한 포인터입니다. 스크립트가 실행되는 동안에는 **this** 키워드가 이 객체를 나타냅니다.
- **argc** 정수는 함수에 전달되는 인수의 수입입니다.
- **argv** 포인터는 **jsval** 구조의 배열에 대한 포인터입니다. 배열은 길이의 **argc** 요소입니다.
- **rval** 포인터는 단일 **jsval** 구조에 대한 포인터입니다. 함수의 반환값은 ***rval**에 기록되어야 합니다.

성공적으로 실행되면 **JS_TRUE**가 반환되고 그렇지 않으면 **JS_FALSE**가 반환됩니다. **JS_FALSE**가 반환되면 현재 스크립트 실행이 중지되고 오류 메시지가 나타납니다.

JSBool JS_DefineFunction()

사용법

```
JSBool JS_DefineFunction(unsigned short *name, JSNative call, unsigned int nargs)
```

설명

메서드, Flash의 JavaScript 인터프리터를 사용하여 C 레벨 함수를 등록합니다. **JS_DefineFunction()** 함수가 **call** 인수에 지정된 C 레벨 함수를 등록하고 나면 이 함수를 **name** 인수에 지정된 이름으로 참조하여 JavaScript 스크립트에서 호출할 수 있습니다. **name** 인수는 대/소문자를 구분합니다.

일반적으로 이 함수는 Flash가 시작할 때 호출하는 **MM_Init()** 함수에서 호출됩니다.

인수

unsigned short ***name**, JSNative**call**, unsigned int **nargs**

- **name** 인수는 JavaScript에 공개되는 함수의 이름입니다.
- **call** 인수는 C 레벨 함수에 대한 포인터입니다. 이 함수는 성공 또는 실패를 나타내는 JSBool을 반환해야 합니다.
- **nargs** 인수는 함수에 전달되는 인수의 수입입니다.

반환값

부울 값, **JS_TRUE**는 성공을 나타내고 **JS_FALSE**는 실패를 나타냅니다.

unsigned short *JS_ValueToString()

사용법

```
unsigned short *JS_ValueToString(JSContext *cx, jsval v, unsigned int *pLength)
```

설명

메서드, **jsval** 구조에서 함수 인수를 추출하고 가능한 경우 문자열로 변환한 다음 변환된 값을 호출자에게 다시 전달합니다.

참고: JavaScript 인터프리터의 데이터 구조가 손상될 수 있으므로 반환된 버퍼 포인터는 수정하지 마십시오. 문자열을 변경하려면 문자를 다른 버퍼에 복사하고 새 JavaScript 문자열을 만들어야 합니다.

인수

JSContext ***cx**, jsval **v**, unsigned int ***pLength**

- **cx** 인수는 JavaScript 함수에 전달되는 불투명 JSContext 포인터입니다.
- **v** 인수는 문자열을 추출해 내는 **jsval** 구조입니다.
- **pLength** 인수는 부호 없는 정수에 대한 포인터입니다. 이 함수는 ***pLength**를 문자열의 길이(바이트 단위)와 함께 설정합니다.

반환값

성공적으로 실행되면 포인터가 null 종료 문자열을 가리키고 실패하면 null 값을 가리킵니다. 호출 루틴은 종료될 때 이 문자열을 해제하면 안 됩니다.

JSBool JS_ValueToInteger()

사용법

```
JSBool JS_ValueToInteger(JSContext *cx, jsval v, long *lp);
```

설명

메서드, jsval 구조에서 함수 인수를 추출하고 가능한 경우 정수로 변환한 다음 변환된 값을 호출자에게 다시 전달합니다.

인수

JSContext *cx, jsval v, long *lp

- cx 인수는 JavaScript 함수에 전달되는 불투명 JSContext 포인터입니다.
- v 인수는 정수를 추출해 내는 jsval 구조입니다.
- lp 인수는 4바이트 정수에 대한 포인터입니다. 이 함수는 변환된 값을 *lp에 저장합니다.

반환값

부울 값, JS_TRUE는 성공을 나타내고 JS_FALSE는 실패를 나타냅니다.

JSBool JS_ValueToDouble()

사용법

```
JSBool JS_ValueToDouble(JSContext *cx, jsval v, double *dp);
```

설명

메서드, jsval 구조에서 함수 인수를 추출하고 가능한 경우 double로 변환한 다음 변환된 값을 호출자에게 다시 전달합니다.

인수

JSContext *cx, jsval v, double *dp

- cx 인수는 JavaScript 함수에 전달되는 불투명 JSContext 포인터입니다.
- v 인수는 double을 추출해 내는 jsval 구조입니다.
- dp 인수는 8바이트 double에 대한 포인터입니다. 이 함수는 변환된 값을 *dp에 저장합니다.

반환값

부울 값, JS_TRUE는 성공을 나타내고 JS_FALSE는 실패를 나타냅니다.

JSBool JS_ValueToBoolean()

사용법

```
JSBool JS_ValueToBoolean(JSContext *cx, jsval v, JSBool *bp);
```

설명

메서드, jsval 구조에서 함수 인수를 추출하고 가능한 경우 부울 값으로 변환한 다음 변환된 값을 호출자에게 다시 전달합니다.

인수

JSContext *cx, jsval v, JSBool *bp

- cx 인수는 JavaScript 함수에 전달되는 불투명 JSContext 포인터입니다.
- v 인수는 부울 값을 추출해 내는 jsval 구조입니다.
- bp 인수는 JSBool 부울 값에 대한 포인터입니다. 이 함수는 변환된 값을 *bp에 저장합니다.

반환값

부울 값, JS_TRUE는 성공을 나타내고 JS_FALSE는 실패를 나타냅니다.

JSBool JS_ValueToObject()

사용법

```
JSBool JS_ValueToObject(JSContext *cx, jsval v, JSObject **op);
```

설명

메서드, jsval 구조에서 함수 인수를 추출하고 가능한 경우 객체로 변환한 다음 변환된 값을 호출자에게 다시 전달합니다. 객체가 배열일 경우 JS_GetArrayLength() 및 JS_GetElement()를 사용하여 해당 내용을 읽습니다.

인수

JSContext *cx, jsval v, JSObject **op

- cx 인수는 JavaScript 함수에 전달되는 불투명 JSContext 포인터입니다.
- v 인수는 객체를 추출해 내는 jsval 구조입니다.
- op 인수는 JSObject 포인터에 대한 포인터입니다. 이 함수는 변환된 값을 *op에 저장합니다.

반환값

부울 값, JS_TRUE는 성공을 나타내고 JS_FALSE는 실패를 나타냅니다.

JSBool JS_StringToValue()

사용법

```
JSBool JS_StringToValue(JSContext *cx, unsigned short *bytes, uint sz, jsval *vp);
```

설명

메서드, jsval 구조에 문자열 반환값을 저장하며 새 JavaScript 문자열 객체를 할당합니다.

인수

JSContext *cx, unsigned short *bytes, size_t sz, jsval *vp

- cx 인수는 JavaScript 함수에 전달되는 불투명 JSContext 포인터입니다.
- bytes 인수는 jsval 구조에 저장할 문자열입니다. 문자열 데이터는 복사되기 때문에 호출자는 필요 없는 문자열을 해제해야 합니다. 문자열 크기가 지정되지 않은 경우(sz 인수 참조) 문자열은 null로 종료되어야 합니다.
- sz 인수는 문자열의 크기(바이트 단위)입니다. sz가 0이면 null 종료 문자열의 길이가 자동으로 계산됩니다.

- **vp** 인수는 문자열의 내용을 복사해야 하는 `jsval` 구조에 대한 포인터입니다.

반환값

부울 값, `JS_TRUE`는 성공을 나타내고 `JS_FALSE`는 실패를 나타냅니다.

JSBool JS_DoubleToValue()

사용법

```
JSBool JS_DoubleToValue(JSContext *cx, double dv, jsval *vp);
```

설명

메서드, `jsval` 구조에 부동 소수점 숫자 반환값을 저장합니다.

인수

`JSContext *cx`, `double dv`, `jsval *vp`

- **cx** 인수는 JavaScript 함수에 전달되는 불투명 `JSContext` 포인터입니다.
- **dv** 인수는 8바이트 부동 소수점 숫자입니다.
- **vp** 인수는 `double`의 내용을 복사해야 하는 `jsval` 구조에 대한 포인터입니다.

반환값

부울 값, `JS_TRUE`는 성공을 나타내고 `JS_FALSE`는 실패를 나타냅니다.

JSVal JS_BooleanToValue()

사용법

```
jsval JS_BooleanToValue(JSBool bv);
```

설명

메서드, `jsval` 구조에 부울 반환값을 저장합니다.

인수

`JSBool bv`

- **bv** 인수는 부울 값입니다. `JS_TRUE`는 성공을 나타내고 `JS_FALSE`는 실패를 나타냅니다.

반환값

함수에 인수로 전달되는 부울 값이 포함된 `JSVal` 구조

JSVal JS_BytesToValue()

사용법

```
JSBool JS_BytesToValue(JSContext *cx, unsigned short *bytes, uint sz, jsval *vp);
```

설명

메서드, 바이트를 JavaScript 값으로 변환합니다.

인수

`JContext *cx, unsignedshort*bytes, uintsz, jsval *vp`

- **cx** 인수는 JavaScript 컨텍스트입니다.
- **bytes** 인수는 JavaScript 객체로 변환할 바이트의 문자열입니다.
- **sz** 인수는 변환할 바이트의 수입니다.
- **vp** 인수는 JavaScript 값입니다.

반환값

부울 값, JS_TRUE는 성공을 나타내고 JS_FALSE는 실패를 나타냅니다.

JSVal JS_IntegerToValue()

사용법

```
jsval JS_IntegerToValue(long lv);
```

설명

메서드, 정수(Long) 값을 JSVal 구조로 변환합니다.

인수

lv

lv 인수는 jsval 구조로 변환할 정수(Long) 값입니다.

반환값

함수에 인수로 전달되는 정수가 포함된 JSVal 구조

JSVal JS_ObjectToValue()

사용법

```
jsval JS_ObjectToValue(JSObject *obj);
```

설명

메서드, JSVal에 객체 반환값을 저장합니다. 배열 객체를 만들려면 JS_NewArrayObject()를 사용하고 해당 내용을 정의하려면 JS_SetElement()를 사용합니다.

인수

`JSObject *obj`

obj 인수는 JSVal 구조로 변환할 JSObject 객체에 대한 포인터입니다.

반환값

함수에 인수로 전달되는 객체가 포함된 JSVal 구조

unsigned short *JS_ObjectType()

사용법

```
unsigned short *JS_ObjectType(JSObject *obj);
```

설명

메서드, 객체 참조가 지정된 경우 객체의 클래스 이름을 반환합니다. 예를 들어, 객체가 DOM 객체이면 "Document"를 반환하고 객체가 문서의 노드이면 "Element"를 반환합니다. 배열 객체일 경우 "Array"를 반환합니다.

참고: JavaScript 인터프리터의 데이터 구조가 손상될 수 있으므로 반환된 버퍼 포인터는 수정하지 마십시오.

인수

JSObject *obj

일반적으로 이 인수는 JS_ValueToObject() 함수를 통해 전달되고 변환됩니다.

반환값

null 종료 문자열에 대한 포인터. 호출자는 종료될 때 이 문자열을 해제하면 안 됩니다.

JSObject *JS_NewArrayObject()

사용법

```
JSObject *JS_NewArrayObject(JSContext *cx, unsigned int length [, jsval *v])
```

설명

메서드, JSVal의 배열이 포함된 새 객체를 만듭니다.

인수

JSContext *cx, unsigned int length, jsval *v

- cx 인수는 JavaScript 함수에 전달되는 불투명 JSContext 포인터입니다.
- length 인수는 배열에 포함할 수 있는 요소의 수입니다.
- v 인수는 배열에 저장할 jsval에 대한 선택적 포인터입니다. 반환값이 null이 아닐 경우 v는 length 요소가 포함된 배열입니다. 반환값이 null일 경우 배열 객체의 초기 내용은 정의되지 않고 JS_SetElement() 함수를 사용하여 설정할 수 있습니다.

반환값

새 배열 객체에 대한 포인터를 반환하거나 실패할 경우 null 값을 반환합니다.

long JS_GetArrayLength()

사용법

```
long JS_GetArrayLength(JSContext *cx, JSObject *obj)
```

설명

메서드, 배열 객체에 대한 포인터가 지정된 경우 배열 요소의 수를 가져옵니다.

인수

`JContext *cx, JSObject *obj`

- **cx** 인수는 JavaScript 함수에 전달되는 불투명 JContext 포인터입니다.
- **obj** 인수는 배열 객체에 대한 포인터입니다.

반환값

배열 요소의 수를 반환하거나 실패할 경우 -1을 반환합니다.

JSBool JS_GetElement()

사용법

```
JSBool JS_GetElement(JContext *cx, JSObject *obj, jsint idx, jsval *vp)
```

설명

메서드, 배열 객체의 단일 요소를 읽습니다.

인수

`JContext *cx, JSObject *obj, jsint idx, jsval *vp`

- **cx** 인수는 JavaScript 함수에 전달되는 불투명 JContext 포인터입니다.
- **obj** 인수는 배열 객체에 대한 포인터입니다.
- **idx** 인수는 배열의 정수 인덱스입니다. 첫 번째 요소는 인덱스 0이고 마지막 요소는 인덱스 (length-1)입니다.
- **vp** 인수는 배열 내의 jsval 구조 내용을 복사해야 하는 jsval에 대한 포인터입니다.

반환값

부울 값, JS_TRUE는 성공을 나타내고 JS_FALSE는 실패를 나타냅니다.

JSBool JS_SetElement()

사용법

```
JSBool JS_SetElement(JContext *cx, JSObject *obj, jsint idx, jsval *vp)
```

설명

메서드, 배열 객체의 단일 요소를 작성합니다.

인수

`JContext *cx, JSObject *obj, jsint idx, jsval *vp`

- **cx** 인수는 JavaScript 함수에 전달되는 불투명 JContext 포인터입니다.
- **obj** 인수는 배열 객체에 대한 포인터입니다.
- **idx** 인수는 배열의 정수 인덱스입니다. 첫 번째 요소는 인덱스 0이고 마지막 요소는 인덱스 (length-1)입니다.
- **vp** 인수는 배열 내의 jsval에 내용을 복사해야 하는 jsval 구조에 대한 포인터입니다.

반환값

부울 값, JS_TRUE는 성공을 나타내고 JS_FALSE는 실패를 나타냅니다.

JSBool JS_ExecuteScript()

사용법

```
JS_ExecuteScript (JSContext *cx, JSObject *obj, unsigned short *script, unsigned int sz, jsval *rval)
```

설명

메서드, JavaScript 문자열을 컴파일하고 실행합니다. 스크립트에서 반환값을 생성하면 *rval에 반환됩니다.

인수

JSContext ***cx**, JSObject ***obj**, unsigned short ***script**, unsigned int**sz**, jsval ***rval**

- **cx** 인수는 JavaScript 함수에 전달되는 불투명 JSContext 포인터입니다.
- **obj** 인수는 해당 컨텍스트에서 스크립트가 실행되는 객체에 대한 포인터입니다. 스크립트가 실행되는 동안에는 this 키워드가 이 객체를 나타냅니다. 일반적으로 이것은 JavaScript 함수에 전달되는 JSObject 포인터입니다.
- **script** 인수는 JavaScript 코드가 포함된 문자열입니다. 문자열 크기가 지정되지 않은 경우(**sz** 인수 참조) 문자열은 null로 종료되어야 합니다.
- **sz** 인수는 문자열의 크기(바이트 단위)입니다. **sz**가 0이면 null 종료 문자열의 길이가 자동으로 계산됩니다.
- **rval** 인수는 단일 jsval 구조에 대한 포인터입니다. 함수의 반환값은 *rval에 저장됩니다.

반환값

부울 값, JS_TRUE는 성공을 나타내고 JS_FALSE는 실패를 나타냅니다.