Estensione di
ADOBE® FLASH® CS4 PROFESSIONAL

Adobe

Estensione di Flash® per Windows® e Mac OS

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA

Sommario

Capitolo 14: Oggetto Element

Capitolo 15: Oggetto Fill

Capitolo 16: Oggetto Filter

Capitolo 17: Oggetto Flash (fl)

Capitolo 26: Oggetto library

Capitolo 27: Oggetto Math

Capitolo 28: Oggetto Matrix

Capitolo 29: Oggetto outputPanel

Capitolo 30: Oggetto Oval

Capitolo 31: Oggetto Parameter

Capitolo 32: Oggetto Path

Capitolo 33: Oggetto presetItem

Capitolo 34: Oggetto presetPanel

Capitolo 35: Oggetto Rectangle

Capitolo 41: Oggetto swfPanel

Capitolo 42: Oggetto SymbolInstance

Capitolo 43: Oggetto SymbolItem

Capitolo 44: Oggetto Text

Capitolo 46: Oggetto TextRun

Capitolo 47: Oggetto Timeline

Capitolo 48: Oggetto ToolObj

Capitolo 49: Oggetto Tools

Capitolo 50: Oggetto Vertex

Capitolo 51: Oggetto VideoItem

Capitolo 52: Oggetto XMLUI

Capitolo 53: Estensibilità di livello C

Capitolo 1: Introduzione

Gli utenti di Adobe® Flash® CS4 Professional conoscono già il linguaggio Adobe® ActionScript® che permette di creare script da eseguire in fase di runtime in Adobe® Flash® Player. L'API (Application Programming Interface) JavaScript di Flash descritta in questo manuale è uno strumento di programmazione complementare che consente di creare script da eseguire nell'ambiente di creazione.

Questo documento descrive gli oggetti, i metodi e le proprietà disponibili nell'API JavaScript e presuppone che l'utente sappia già usare nell'ambiente di creazione i comandi documentati nel presente manuale. Se sono necessarie informazioni sulle funzioni dei singoli comandi, consultate altri manuali della documentazione di Flash, ad esempio *Uso di Flash*.

Questo documento presuppone anche una certa familiarità con la sintassi JavaScript e ActionScript e con alcuni concetti di programmazione di base, quali le funzioni, i parametri e i tipi di dati.

Operazioni con l'API JavaScript

L'API JavaScript di Flash consente di creare script che eseguono diverse azioni nell'ambiente di creazione di Flash (ovvero mentre un utente lavora nel programma Flash). Questa funzionalità è differente dal linguaggio ActionScript, il quale consente di creare script che eseguono azioni nell'ambiente Flash Player (ovvero durante la riproduzione di un file SWF). Inoltre, questa funzionalità è differente anche dai comandi JavaScript che possono essere impiegati nelle pagine visualizzate nei browser Web.

Utilizzando l'API JavaScript, è possibile creare script di applicazione Flash in grado di semplificare il processo di creazione. Ad esempio, è possibile creare script che consentano di automatizzare le operazioni ripetitive o aggiungere strumenti al pannello Strumenti.

L'API JavaScript di Flash è concepita in modo analogo all'API JavaScript di Adobe® Dreamweaver® e di Adobe® Fireworks®, a loro volta basate sull'API JavaScript di Netscape. L'API JavaScript di Flash si basa su un modello DOM (Document Object Model), che consente di accedere ai documenti Flash mediante gli oggetti JavaScript. L'API JavaScript di Flash comprende tutti gli elementi dell'API JavaScript di Netscape, oltre al modello DOM di Flash. Gli oggetti aggiunti e i metodi e le proprietà corrispondenti sono descritti in questo documento. In uno script di Flash è possibile utilizzare qualunque elemento del linguaggio JavaScript nativo, tuttavia solo gli elementi che hanno un significato nel contesto di un documento Flash avranno effetto.

L'API JavaScript contiene inoltre metodi che consentono di implementare l'estensibilità grazie a una combinazione di codice JavaScript e di codice C personalizzato. Per ulteriori informazioni, consultate “Estensibilità di livello C” a pagina 534.

L'interprete JavaScript di Flash è il motore Mozilla SpiderMonkey versione 1.5, disponibile sul Web all'indirizzo www.mozilla.org/js/spidermonkey/. SpiderMonkey è una delle due implementazioni di riferimento del linguaggio JavaScript sviluppate da Mozilla.org. Si tratta dello stesso motore incluso nel browser Mozilla.

SpiderMonkey implementa il linguaggio JavaScript di base come indicato dalla specifica del linguaggio ECMAScript (ECMA-262) Edition 3, alla quale è completamente conforme. Gli unici elementi non supportati sono gli oggetti host specifici del browser, poiché non fanno parte della specifica ECMA-262. Analogamente, molte guide di riferimento di JavaScript fanno distinzione tra il linguaggio JavaScript di base e client-side (relativo al browser). Per l'interprete JavaScript di Flash è valido solo il linguaggio JavaScript di base.

Creazione di file JSFL

Per creare o modificare i file Flash JavaScript (JSFL), è possibile utilizzare Adobe FlashCS4 Professional o qualsiasi editor di testo. Nel caso di Flash, i file hanno l'estensione .jsfl per impostazione predefinita. Per scrivere uno script, selezionate File > Nuovo > File JavaScript Flash.

Potete anche creare un file JSFL selezionando i comandi nel pannello Cronologia. Poi fate clic sul pulsante Salva nel pannello Cronologia o selezionate Salva con nome nel menu del pannello Il file dei comandi (JSFL) viene salvato nella cartella Commands (consultate "Salvataggio di file JSFL" a pagina 2). A questo punto è possibile aprirlo e modificarlo come qualsiasi altro file di script.

Il pannello Cronologia contiene anche altre opzioni utili. Ad esempio, è possibile copiare i comandi selezionati negli Appunti e visualizzare i comandi JavaScript che vengono generati mentre si lavora in Flash.

Per copiare negli Appunti i comandi del pannello Cronologia:

1. Selezionate uno o più comandi nel pannello Cronologia.
2. Effettuate una delle seguenti operazioni:
   - Fate clic sul pulsante Copia.
   - Selezionate Copia passaggi nel menu del pannello.

Per visualizzare i comandi JavaScript nel pannello Cronologia:

- Selezionate Visualizza > JavaScript nel menu del pannello.

Salvataggio di file JSFL

È possibile salvare gli script JSFL disponibili nell'ambiente di creazione Flash archiviandoli in una delle molte cartelle della cartella Configuration. Per impostazione predefinita, la cartella Configuration si trova nel percorso seguente:

- Windows® Vista™:

  unità di avvio\Utenti*nomeutente*\Impostazioni locali\Dati applicazioni\Adobe\Flash CS4*lingua*\Configuration\

- Windows XP:

  unità di avvio\Documents and Settings*nomeutente*\Impostazioni locali\Dati applicazioni\Adobe\Flash CS4*lingua*\Configuration\

- Mac OS® X:

  Macintosh HD/Users/*nomeutente*/Library/Supporto applicazioni/Adobe/Flash CS4/*lingua*/Configuration/

Per determinare il percorso della cartella Configuration, usate `fl.configDirectory` o `fl.configURI`, come illustrato negli esempi che seguono:

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

All'interno della cartella Configuration, le seguenti cartelle contengono script che potete usare nell'ambiente di creazione: Behaviors (di supporto all'interfaccia utente per i comportamenti); Commands (per gli script visualizzati nel menu Comandi); JavaScript (per gli script usati da Assistente script per compilare i controlli dell'interfaccia utente); Tools (per gli strumenti estensibili del pannello Strumenti) e WindowSWF (per i pannelli visualizzati nel menu Finestra). Questo documento tratta principalmente gli script usati per i comandi e gli strumenti.

Se modificate uno script nella cartella Commands, il nuovo script è immediatamente disponibile in Flash. Se modificate uno script per uno strumento estensibile, chiudete e riavviate Flash, oppure usate il comando `fl.reloadTools()`. Tuttavia, se si è usato uno script per aggiungere uno strumento estensibile al pannello Strumenti e, in seguito, si modifica lo script, rimuovete lo strumento dal pannello e quindi riaggiungetelo, oppure chiudete e riavviate Flash per rendere disponibile lo strumento modificato.

Per rendere i file di comandi e strumenti accessibili nell'ambiente di creazione, è possibile memorizzarli in due posizioni.

- Per gli script da visualizzare come voci del menu Comandi, salvate il file JSFL nella cartella Commands utilizzando il percorso seguente:

Sistema operativo	Posizione
Windows Vista	*unità di avvio* \Utenti*nomeutente*\Impostazioni locali\Dati applicazioni\Adobe\Flash CS4*lingua*\Configuration\Commands
Windows XP	*unità di avvio*\Documents and Settings*utente*\Impostazioni locali\Dati applicazioni\Adobe\ FlashCS4*lingua*\Configuration\Commands
Mac OS X	Macintosh HD/Users/*nomeUtente*/Library/Supporto Applicazioni/Adobe/FlashCS4/*lingua*/Configuration/Commands

- Per gli script da visualizzare come strumenti estensibili del menu Strumenti, salvate il file JSFL nella cartella Tools utilizzando il percorso seguente:

Sistema operativo	Posizione
Windows Vista	*unità di avvio* \Utenti*nomeutente*\Impostazioni locali\Dati applicazioni\Adobe\Flash CS4*lingua*\Configuration\Tools
Windows XP	*unità di avvio*\Documents and Settings*utente*\Impostazioni locali\Dati applicazioni\Adobe\FlashCS4*lingua*\Configuration\Tools
Mac OS X	Macintosh HD/Users/*nomeUtente*/Library/Supporto Applicazioni/Adobe/FlashCS4/*lingua*/Configuration/Tools

Se a un file JSFL sono associati altri file, ad esempio file XML, memorizzateli nella stessa directory del file JSFL.

Esecuzione degli script

Gli script possono essere eseguiti in diversi modi: i metodi più utilizzati vengono illustrati in questa sezione.

Per eseguire uno script che state visualizzando o modificando:

- Fate clic con il pulsante destro del mouse (Comando+clic in Macintosh) e scegliete Esegui script.
- Fate clic sull'icona Esegui script nella barra degli strumenti della finestra Script.

Questa opzione consente di eseguire uno script prima di averlo salvato. Questa opzione consente inoltre di eseguire uno script anche se non vi sono file FLA aperti.

Per eseguire uno script che si trova nella cartella Commands, effettuate una delle operazioni seguenti:

- Nell'ambiente di creazione, selezionate Comandi > *Esegui comando*, quindi selezionate lo script da eseguire.
- Utilizzate una scelta rapida da tastiera che è stata assegnata allo script. Per assegnare una scelta rapida da tastiera, selezionate Modifica > Scelte rapide da tastiera e selezionate Comandi del menu Disegno dal menu a comparsa Comandi. Espandete il nodo Comandi nella struttura del menu per visualizzare l'elenco degli script disponibili.

Per eseguire lo script di un comando che non si trova nella cartella Commands, effettuate una delle operazioni seguenti:

- Nell'ambiente di creazione, selezionate Comandi > Esegui comando, quindi selezionate lo script da eseguire.
- Dall'interno di uno script, usate il comando `fl.runScript()`.
- Dal file system, fate doppio clic sul file dello script.

Per aggiungere al pannello Strumenti uno strumento implementato in un file JSFL:

1. Copiate nella cartella Tools il file JSFL dello strumento e gli eventuali file associati (consultate "Salvataggio di file JSFL" a pagina 2).
2. Selezionate Modifica > Personalizza pannello Strumenti (Windows) o Flash > Personalizza pannello Strumenti (Macintosh).
3. Aggiungete lo strumento all'elenco degli strumenti disponibili.
4. Fate clic su OK.

Potete aggiungere comandi API JavaScript singoli nei file ActionScript mediante il comando `MMExecute()`, documentato nella *Guida di riferimento del linguaggio e dei componenti ActionScript 3.0*. Tuttavia, la funzione `MMExecute()` ha effetto solo se utilizzata nel contesto di un elemento di interfaccia personalizzato, ad esempio una finestra di ispezione delle proprietà o dei componenti, o di un pannello SWF dell'ambiente di creazione. Anche se vengono chiamati da ActionScript, i comandi API JavaScript non hanno alcun effetto in Flash Player o al di fuori dell'ambiente di creazione.

Per eseguire un comando da uno script di ActionScript:

- Usate la sintassi seguente (è possibile concatenare diversi comandi in una stringa):

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

  È anche possibile eseguire uno script dalla riga di comando.

Per eseguire uno script dalla riga di comando in Windows:

- Adottate la seguente sintassi (aggiungete le informazioni sul percorso secondo necessità):

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

Per eseguire uno script dall'applicazione "Terminal" su un Macintosh:

- Adottate la seguente sintassi (aggiungete le informazioni sul percorso secondo necessità):

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

  Il comando `osascript` può anche eseguire AppleScript in un file. Se si inserisce, ad esempio, il testo seguente in un file denominato myScript:

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

  Per eseguire lo script, è necessario usare questo comando:

```
osascript myScript
```

Novità dell'API JavaScript

In Flash CS4, alcuni oggetti, metodi e proprietà sono stati aggiunti, mentre altri sono stati eliminati. Tali modifiche sono riassunte più avanti.

Se è la prima volta che usate l'API JavaScript, potete ignorare questa sezione e passare direttamente a "Oggetti API JavaScript" a pagina 7.

Nuovi oggetti

Gli oggetti seguenti sono stati introdotti in FlashCS4:

Oggetto presetPanel

Oggetto presetItem

Oggetto swfPanel

Nuovi metodi e proprietà

I seguenti metodi e proprietà per gli oggetti esistenti sono stati introdotti in FlashCS4:

- Oggetto BitmapItem
  - `bitmapItem.exportToFile()`
  - `bitmapItem.fileLastModifiedDate`
  - `bitmapItem.originalCompressionType`
  - `bitmapItem.sourceFileExists`
  - `bitmapItem.sourceFileIsCurrent`
  - `bitmapItem.sourceFilePath`
  - `bitmapItem.useDeblocking`
- Oggetto Contour
  - `contour.fill`
- Oggetto Document
  - `document.addNewPrimitiveOval()`
  - `document.addNewPrimitiveRectangle()`
  - `document.exportPublishProfileString()`
  - `document.externalLibraryPath`
  - `document.importPublishProfileString()`
  - `document.libraryPath`
  - `document.pathURI`
  - `document.rotate3DSelection`
  - `document.setStageVanishingPoint()`
  - `document.setStageViewAngle()`
  - `document.sourcePath`

- document.translate3DCenter()
- document.translate3DSelection()

- Oggetto Edge
  - edge.cubicSegmentIndex
  - edge.stroke
- Oggetto Fill
  - fill.bitmapIsClipped
  - fill.bitmapPath
- Oggetto Flash (fl)
  - fl.externalLibraryPath
  - fl.flexSDKPath
  - fl.isFontInstalled()
  - fl.libraryPath
  - fl.presetPanel
  - fl.sourcePath
  - fl.swfPanels
- Oggetto FLfile
  - FLfile.platformPathToURI()
  - FLfile.uriToPlatformPath()
- Oggetto fontItem
  - fontItem.bitmap
  - fontItem.bold
  - fontItem.embedVariantGlyphs
  - fontItem.font
  - fontItem.isDefineFont4Symbol
  - fontItem.italic
  - fontItem.size
- Oggetto Shape
  - shape.getCubicSegmentPoints()
  - shape.members
  - shape.numCubicSegments
- Oggetto SoundItem
  - soundItem.exportToFile()
  - soundItem.fileLastModifiedDate
  - soundItem.originalCompressionType
  - soundItem.sourceFileExists
  - soundItem.sourceFileIsCurrent

    - `soundItem.sourceFilePath`
- Oggetto Text
    - `text.embedVariantGlyphs`
- Oggetto Timeline
    - `timeline.getGuidelines()`
    - `timeline.setGuidelines()`
- Oggetto VideoItem
    - `videoItem.exportToFLV()`
    - `videoItem.fileLastModifiedDate`
    - `videoItem.sourceFileExists`
    - `videoItem.sourceFileIsCurrent`

Altre modifiche

La proprietà seguente supporta un nuovo valore in Flash CS4:

- `fill.style`

Gli oggetti, i metodi e le proprietà seguenti non sono più disponibili in Flash CS4:

- Oggetto Project
- Oggetto ProjectItem
- `fl.openProject()`
- `fl.closeProject()`
- `fl.createProject()`
- `fl.getProject()`
- Oggetto Effect
- `configureEffect()`
- `executeEffect()`
- `removeEffect()`
- `fl.activeEffect`
- `fl.effects`
- `fl.enableImmediateUpdates()`
- `fl.reloadEffects()`

Oggetti API JavaScript

In questa sezione viene fornito un riepilogo degli oggetti disponibili nell'API Flash e come iniziare a utilizzarli. Tutti i comuni comandi JavaScript sono disponibili anche quando utilizzate l'API JavaScript.

Nella seguente tabella sono brevemente descritti i singoli oggetti nell'API JavaScript. Gli oggetti sono elencati in ordine alfabetico.

Oggetto	**Descrizione**
Oggetto actionsPanel	L'oggetto actionsPanel rappresenta il pannello Azioni attualmente visualizzato.
Oggetto BitmapInstance	L'oggetto BitmapInstance è una sottoclasse dell'oggetto Instance e rappresenta una bitmap in un fotogramma.
Oggetto BitmapItem	Un oggetto BitmapItem fa riferimento a una bitmap presente nella libreria di un documento ed è una sottoclasse dell'oggetto Item.
Oggetto CompiledClipInstance	L'oggetto CompiledClipInstance è una sottoclasse dell'oggetto Instance.
Oggetto compilerErrors	L'oggetto compilerErrors rappresenta il pannello Errori del compilatore. È una proprietà dell'oggetto flash (fl.compilerErrors).
Oggetto ComponentInstance	L'oggetto ComponentInstance è una sottoclasse dell'oggetto SymbolInstance e rappresenta un componente in un fotogramma.
Oggetto componentsPanel	L'oggetto componentsPanel, che rappresenta il pannello Componenti, è una proprietà dell'oggetto flash (`fl.componentsPanel`).
Oggetto Contour	Un oggetto Contour rappresenta un percorso chiuso di mezzi bordi sul contorno di una forma.
Oggetto Document	L'oggetto Document rappresenta lo stage.
Oggetto drawingLayer	L'oggetto drawingLayer è accessibile da JavaScript come elemento secondario dell'oggetto Flash.
Oggetto Edge	L'oggetto Edge rappresenta un bordo di una forma sullo stage.
Oggetto Element	Tutti gli elementi che compaiono sullo stage appartengono al tipo Element.
Oggetto Fill	L'oggetto Fill contiene tutte le proprietà dell'impostazione Colore di riempimento del pannello Strumenti o di una forma selezionata.
Oggetto Filter	L'oggetto Filter contiene tutte le proprietà per tutti i filtri.
Oggetto Flash (fl)	L'oggetto flash rappresenta l'applicazione Flash.
Oggetto FLfile	L'oggetto FLfile consente di scrivere estensioni di Flash che accedano, modifichino e rimuovano i file e le cartelle nel file system locale.
Oggetto folderItem	L'oggetto folderItem è una sottoclasse dell'oggetto Item.
Oggetto fontItem	L'oggetto fontItem è una sottoclasse dell'oggetto Item.
Oggetto Frame	L'oggetto Frame rappresenta i fotogrammi del livello.
Oggetto HalfEdge	Lato indicato del bordo di un oggetto Shape.
Oggetto Instance	L'oggetto Instance è una sottoclasse dell'oggetto Element.
Oggetto Item	L'oggetto Item è una classe base astratta.
Oggetto Layer	L'oggetto Layer rappresenta un livello nella linea temporale.
Oggetto library	L'oggetto library rappresenta il pannello Libreria.
Oggetto Math	L'oggetto Math è disponibile come proprietà di sola lettura dell'oggetto flash (`fl.Math`).
Oggetto Matrix	L'oggetto Matrix rappresenta una matrice di trasformazione.
Oggetto outputPanel	L'oggetto outputPanel rappresenta il pannello Output, che visualizza informazioni utili per la risoluzione dei problemi (ad esempio, gli errori di sintassi). È una proprietà dell'oggetto flash (`fl.outputPanel`).
Oggetto Oval	L'oggetto Oval è una forma disegnata utilizzando lo strumento Ovale. Per determinare se un elemento è un oggetto Oval, usate `shape.isOvalObject`.

Oggetto	**Descrizione**
Oggetto Parameter	Il tipo di oggetto Parameter è accessibile dall'array `screen.parameters`, che corrisponde alla finestra di ispezione Proprietà delle schermate nel programma di creazione Flash, o dall'array `componentInstance.parameters`, che corrisponde alla finestra di ispezione Proprietà dei componenti nel programma di creazione Flash.
Oggetto Path	L'oggetto Path definisce una sequenza di segmenti di linea (retta, curva o entrambe), utilizzata generalmente quando create strumenti estensibili.
Oggetto presetItem	L'oggetto presetItem rappresenta un elemento (preimpostazione o cartella) nel pannello Preimpostazioni di movimento.
Oggetto presetPanel	L'oggetto presetPanel rappresenta il pannello Preimpostazioni di movimento (Finestra > Preimpostazioni di movimento). È una proprietà dell'oggetto flash (`fl.presetPanel`).
Oggetto Rectangle	L'oggetto Rectangle è una forma disegnata utilizzando lo strumento Rettangolo. Per determinare se un elemento è un oggetto Rectangle, utilizzate `shape.isRectangleObject`.
Oggetto Screen	L'oggetto Screen rappresenta una schermata singola in una diapositiva o in un form.
Oggetto ScreenOutline	L'oggetto ScreenOutline rappresenta il gruppo di schermate in una diapositiva o in un form.
Oggetto Shape	L'oggetto Shape è una sottoclasse dell'oggetto Element. L'oggetto Shape fornisce un controllo più preciso rispetto alle API di disegno per la manipolazione o la creazione di forme geometriche sullo stage.
Oggetto SoundItem	L'oggetto SoundItem è una sottoclasse dell'oggetto Item. Rappresenta un elemento della libreria utilizzato per creare un suono.
Oggetto Stroke	L'oggetto Stroke contiene tutte le impostazioni relative a un tratto, comprese quelle personalizzate.
Oggetto swfPanel	L'oggetto swfPanel rappresenta un pannello Finestra SWF. I pannelli Finestra SWF sono file SWF che implementano le applicazioni che potete eseguire dall'ambiente di creazione Flash. L'array di oggetti swfPanel è una proprietà dell'oggetto flash (`fl.swfPanels`).
Oggetto SymbolInstance	L'oggetto SymbolInstance è una sottoclasse dell'oggetto Instance e rappresenta un simbolo in un fotogramma.
Oggetto SymbolItem	L'oggetto SymbolItem è una sottoclasse dell'oggetto Item.
Oggetto Text	L'oggetto Text rappresenta un elemento di testo singolo in un documento.
Oggetto TextAttrs	L'oggetto TextAttrs contiene tutte le proprietà di testo che è possibile applicare a una sottoselezione. Questo oggetto è una sottoclasse dell'oggetto Text.
Oggetto TextRun	L'oggetto TextRun rappresenta una sequenza di caratteri che dispongono di attributi che corrispondono a tutte le proprietà dell'oggetto TextAttrs.
Oggetto Timeline	L'oggetto Timeline rappresenta la linea temporale di Flash, a cui potete accedere mediante `fl.getDocumentDOM().getTimeline()`.
Oggetto ToolObj	L'oggetto ToolObj rappresenta un singolo strumento nel pannello Strumenti.
Oggetto Tools	L'oggetto Tools è accessibile dall'oggetto Flash (`fl.tools`).
Oggetto Vertex	L'oggetto Vertex è la parte della struttura di dati di una forma che contiene i dati relativi alle coordinate.
Oggetto VideoItem	L'oggetto VideoItem è una sottoclasse dell'oggetto Item.
Oggetto XMLUI	L'oggetto XMLUI consente di ottenere e impostare le proprietà di una finestra di dialogo XMLUI e di uscire dalla finestra facendo clic su OK o su Annulla.

DOM (Document Object Model) di Flash

Il DOM dell'API JavaScript di Flash è composto da una serie di funzioni di primo livello (consultate la sezione su "Funzioni e metodi di primo livello" a pagina 15) e da due oggetti di primo livello: FLfile e Flash (fl). Ogni oggetto è sempre disponibile per uno script poiché è sempre presente quando è aperto l'ambiente di creazione Flash. Per ulteriori informazioni, consultate Oggetto FLfile e Oggetto Flash (fl).

Quando si fa riferimento all'oggetto Flash, è possibile utilizzare `flash` o `fl`. Ad esempio, per chiudere tutti i file FLA aperti, è possibile utilizzare una qualsiasi delle seguenti istruzioni:

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

L'oggetto Flash contiene i seguenti oggetti *secondari*:

Oggetto	**Metodo di accesso**
Oggetto actionsPanel	Utilizzate `fl.actionsPanel` per accedere all'oggetto actionsPanel. Questo oggetto corrisponde al pannello Azioni dell'ambiente di creazione Flash.
Oggetto compilerErrors	Utilizzate `fl.compilerErrors` per accedere all'oggetto compilerErrors. Questo oggetto corrisponde al pannello Errori del compilatore dell'ambiente di creazione Flash.
Oggetto componentsPanel	Utilizzate `fl.componentsPanel` per accedere all'oggetto componentsPanel. Questo oggetto corrisponde al pannello Componenti dell'ambiente di creazione Flash.
Oggetto Document	Utilizzate `fl.documents` per recuperare un array di tutti i documenti aperti; utilizzate `fl.documents[index]` per accedere a un documento specifico; utilizzate `fl.getDocumentDOM()` per accedere al documento corrente, ovvero quello attivo.
Oggetto drawingLayer	Utilizzate `fl.drawingLayer` per accedere all'oggetto drawingLayer.
Oggetto Math	Utilizzate `fl.Math` per accedere all'oggetto Math.
Oggetto outputPanel	Utilizzate `fl.outputPanel` per accedere all'oggetto outputPanel. Questo oggetto corrisponde al pannello Output dell'ambiente di creazione Flash.
Oggetto presetPanel	Utilizzate `fl.presetPanel` per accedere all'oggetto presetPanel. Questo oggetto corrisponde al pannello Preimpostazioni movimento (Finestra > Preimpostazioni movimento).
Oggetto swfPanel	Utilizzate `fl.swfPanels` per accedere a un array di oggetti swPanel. Questi oggetti corrispondono ai pannelli Finestra SWF.
Oggetto Tools	Utilizzate `fl.tools` per accedere a un array di oggetti Tools.
Oggetto XMLUI	Utilizzate `fl.xmlui` per accedere a un oggetto XMLUI (XML User Interface). L'oggetto XMLUI consente di ottenere e impostare le proprietà di una finestra di dialogo XMLUI.

Oggetto Document

Una proprietà importante dell'oggetto Flash di primo livello è la proprietà `fl.documents` contenente un array di oggetti Document, ognuno dei quali rappresenta uno dei file FLA aperti nell'ambiente di creazione. Le proprietà di ogni oggetto Document rappresentano la maggior parte degli elementi che possono essere contenuti in un file FLA. Pertanto, una parte rilevante del DOM è composta da oggetti e proprietà secondarie dell'oggetto Document. Per ulteriori informazioni, consultate Oggetto Document.

Ad esempio, per fare riferimento al primo documento aperto, utilizzate l'istruzione `flash.documents[0]` o `fl.documents[0]`. Il primo documento è il primo documento Flash che è stato aperto nell'ambiente di creazione durante la sessione corrente. Quando il primo documento aperto viene chiuso, gli indici degli altri documenti aperti vengono diminuiti.

Per trovare l'indice di un documento specifico, utilizzate `flash.findDocumentIndex(`*nameOfDocument*`)` o `fl.findDocumentIndex(`*nameOfDocument*`)`. Consultate `fl.findDocumentIndex()`.

Per accedere al documento attivo, utilizzate l'istruzione `flash.getDocumentDOM()` o `fl.getDocumentDOM()`. Consultate `fl.getDocumentDOM()`. Quest'ultima è la sintassi utilizzata nella maggior parte degli esempi contenuti in questo documento.

Per trovare un documento specifico nell'array di `fl.documents`, eseguite le iterazioni sull'array e verificate la proprietà `document.name` di ogni documento. Consultate `fl.documents` e `document.name`.

Dall'oggetto Document è possibile accedere a tutti gli oggetti del DOM non riportati nella tabella precedente (consultate "DOM (Document Object Model) di Flash" a pagina 10). Ad esempio, per accedere alla libreria di un documento, utilizzate la proprietà `document.library` che recupera un oggetto Library:

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

Per accedere all'array di elementi della libreria, utilizzate la proprietà `library.items`; ogni elemento dell'array è un oggetto Item:

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

Per accedere a un elemento specifico della libreria, specificate un membro dell'array di `library.items`:

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

In altre parole, l'oggetto Library è un elemento secondario dell'oggetto Document, mentre l'oggetto Item è un elemento secondario dell'oggetto Library. Per ulteriori informazioni, consultate `document.library`, Oggetto library, library.items`library.items` e Oggetto Item.

Indicazione della destinazione di un'azione

A meno che non sia specificato diversamente, i metodi influiscono sull'elemento o la selezione attiva. Ad esempio, lo script seguente raddoppia le dimensioni della selezione corrente poiché non è specificato alcun oggetto specifico:

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

In alcuni casi, può essere necessario che un'azione venga eseguita specificamente su un elemento selezionato nel documento Flash. A questo scopo, utilizzate l'array contenuto nella proprietà `document.selection` (consultate `document.selection`). Il primo elemento dell'array rappresenta la selezione corrente, come indicato nell'esempio seguente:

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

Lo script seguente raddoppia le dimensioni del primo elemento sullo stage che viene memorizzato nell'array di elementi, anziché la selezione corrente:

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

È anche possibile eseguire un ciclo tra tutti gli elementi presenti sullo stage e aumentare la larghezza e l'altezza in base a una quantità specifica, come illustrato nell'esempio seguente:

```
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;
    }
```

Riepilogo della struttura del DOM

L'elenco seguente mostra la struttura del DOM. I numeri all'inizio di ogni riga rappresentano il livello di un oggetto. Ad esempio, un oggetto preceduto da "03" è un elemento secondario dell'oggetto "02" di livello immediatamente superiore, il quale a propria volta è un elemento secondario dell'oggetto "01" di livello immediatamente superiore.

In alcuni casi, un oggetto è disponibile se si specifica una proprietà del suo oggetto superiore. Ad esempio, la proprietà `document.timelines` contiene un array di oggetti Timeline. Queste proprietà sono indicate nella struttura riportata di seguito.

Alcuni oggetti sono sottoclassi, e non elementi secondari, di altri oggetti. Un oggetto che è una sottoclasse di un altro oggetto dispone di metodi e/o proprietà proprie, oltre ai metodi e alle proprietà dell'oggetto principale (la superclasse). Le sottoclassi condividono lo stesso livello gerarchico della rispettiva superclasse. L'oggetto Item, ad esempio, è una superclasse dell'oggetto BitmapItem. Queste relazioni sono illustrate nella struttura riportata di seguito:

```
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
  03 ScreenOutline object
    04 Screen object (screenOutline.screens array)
      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
```

Esempi di implementazione

Sono disponibili molte implementazioni JSFL per Adobe Flash CS4 Professional. È possibile esaminare e installare questi file per familiarizzare con le API JavaScript. Gli esempi si trovano nella cartella Samples/ExtendingFlash all'interno del file Samples.zip disponibile all'indirizzo www.adobe.com/go/learn_fl_samples_it.

Comando Shape di esempio

Nella cartella ExtendingFlash/Shape è disponibile uno script di esempio dell'API JavaScript denominato Shape.jsfl (consultate la precedente sezione “Esempi di implementazione”). Questo script visualizza informazioni sui contorni della forma nel pannello Output.

Per installare ed eseguire lo script Shape:

1 Copiate il file Shape.jsfl nella cartella Configuration/Commands (consultate “Salvataggio di file JSFL” a pagina 2).

2 In un documento Flash (file FLA), selezionate un oggetto Shape.

3 Selezionate Comandi > Shape per eseguire lo script.

Comando di esempio per ottenere e impostare filtri

Nella cartella ExtendingFlash/filtersGetSet è disponibile uno script di esempio dell'API JavaScript denominato filtersGetSet.jsfl (consultate la precedente sezione “Esempi di implementazione”). Questo script aggiunge filtri a un oggetto selezionato e visualizza le informazioni sui filtri che vengono aggiunti nel pannello Output.

Per installare ed eseguire lo script filtersGetSet:

1 Copiate il file filtersGetSet.jsfl nella cartella Configuration/Commands (consultate "Salvataggio di file JSFL" a pagina 2).

2 In un documento Flash (file FLA), selezionate un testo, un clip filmato o un oggetto Button.

3 Selezionate Comandi > filtersGetSet per eseguire lo script.

Strumento di esempio PolyStar

Nella cartella ExtendingFlash/PolyStar è disponibile uno script di esempio dell'API JavaScript denominato PolyStar.jsfl (consultate la precedente sezione "Esempi di implementazione").

Lo script PolyStar.jsfl replica lo strumento PolyStar presente nel pannello Strumenti di Flash e illustra come creare lo strumento PolyStar mediante l'API JavaScript. Questo script contiene inoltre commenti dettagliati che descrivono le operazioni eseguite dal codice. La lettura di questo file consente di approfondire le possibilità di applicazione dell'API JavaScript. Si consiglia inoltre di leggere il file PolyStar.xml nella directory Tools per informazioni su come creare uno strumento personalizzato.

Pannello Ricalco bitmap di esempio

Nella cartella ExtendingFlash/TraceBitmapPanel è disponibile una serie di file denominati TraceBitmap.fla e TraceBitmap.swf (consultate la precedente sezione "Esempi di implementazione"). Questi file illustrano come progettare e creare un pannello per controllare le funzioni di Flash. Inoltre, indicano come usare la funzione `MMExecute()` per chiamare i comandi JavaScript da uno script di ActionScript.

Per eseguire l'esempio TraceBitmap:

1 Se Flash è in esecuzione, chiudete l'applicazione.

2 Copiate il file TraceBitmap.swf nella cartella WindowSWF, una sottodirectory della cartella Configuration (consultate "Salvataggio di file JSFL" a pagina 2). Ad esempio, in Windows XP la cartella si trova in *unità di avvio*\Documents and Settings*utente*\Impostazioni locali\Dati applicazioni\Adobe\FlashCS4*lingua*\Configuration\WindowSWF.

3 Avviate Flash.

4 Create o aprite un documento Flash (file FLA) e importate un'immagine bitmap o JPEG nel file.

È possibile utilizzare il file flower.jpg fornito nella cartella TraceBitmapPanel o un'altra immagine a scelta.

5 Con l'immagine importata selezionata, selezionate Finestra > Altri pannelli > Ricalca bitmap.

6 Fate clic su Invia.

L'immagine viene convertita in un gruppo di forme.

DLL di esempio

Nella cartella ExtendingFlash/dllSampleComputeSum è disponibile un esempio di implementazione della DLL (consultate la precedente sezione "Esempi di implementazione"). Per ulteriori informazioni sulla creazione di DLL, consultate "Estensibilità di livello C" a pagina 534.

Capitolo 2: Funzioni e metodi di primo livello

Informazioni sulla sezione

Questa sezione descrive le funzioni e i metodi di primo livello disponibili quando utilizzate l'API (Application Programming Interface) JavaScript di Adobe Flash. Per informazioni su dove memorizzare i file dell'API JavaScript, consultate "Salvataggio di file JSFL" a pagina 2.

Metodi globali

I metodi seguenti possono essere chiamati da qualunque script API JavaScript.

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

Strumenti estensibili

Le funzioni seguenti sono disponibili negli script che creano strumenti estensibili:

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

activate()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

Nessuno.

Restituisce

Nulla.

Descrizione
Funzione; chiamata quando lo strumento estensibile diventa attivo (ovvero quando viene selezionato nel pannello Strumenti). Utilizzate questa funzione per eseguire le operazioni di inizializzazione richieste dallo strumento.

Esempio
L'esempio seguente imposta il valore di `tools.activeTool` quando nel pannello Strumenti è selezionato lo strumento estensibile:

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

Consultate anche
`tools.activeTool`

alert()

Disponibilità
Flash MX 2004.

Uso
```
alert ( alertText )
```

Parametri
`alertText` Una stringa che specifica il messaggio che desiderate visualizzare nella finestra di avviso.

Restituisce
Nulla.

Descrizione
Metodo; visualizza una stringa in una finestra di avviso a scelta obbligatoria, insieme a un pulsante OK.

Esempio
L'esempio seguente visualizza il messaggio "Process Complete" (Processo completato) in una finestra di avviso.

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

Consultate anche
`confirm()`, `prompt()`

configureTool()

Disponibilità
Flash MX 2004.

Uso

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

Parametri

Nessuno.

Restituisce

Nulla.

Descrizione

Funzione chiamata all'apertura di Flash, quando lo strumento estensibile viene caricato nel pannello Strumenti. Utilizzatela per impostare tutte le informazioni eventualmente richieste da Flash per lo strumento.

Esempio

Gli esempi seguenti mostrano due possibili implementazioni della funzione:

```
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()

Disponibilità

Flash 8.

Uso

```
confirm ( strAlert )
```

Parametri

`strAlert` Una stringa che specifica il messaggio che desiderate visualizzare nella finestra di avviso.

Restituisce

Il valore booleano `true` o `false` se l'utente fa clic rispettivamente su OK o Annulla.

Descrizione
Metodo; visualizza una stringa in una finestra di avviso a scelta obbligatoria, insieme ai pulsanti OK e Annulla.

Nota: *se non sono presenti documenti (file FLA) aperti, questo metodo dà esito negativo e genera una condizione di errore.*

Esempio
L'esempio seguente mostra il messaggio "Sort data?" (Ordinare i dati?) in una finestra di avviso:

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

Consultate anche
`alert()`, `prompt()`

deactivate()

Disponibilità
Flash MX 2004.

Uso

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

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Funzione chiamata quando lo strumento estensibile diventa inattivo, vale a dire quando diventa attivo un altro strumento. Utilizzate questa funzione per eseguire l'eventuale ottimizzazione richiesta dallo strumento.

Esempio
Nell'esempio seguente viene visualizzato un messaggio nel pannello Output quando lo strumento diventa inattivo:

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

keyDown()

Disponibilità
Flash MX 2004.

Uso

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

Parametri

Nessuno.

Restituisce

Nulla.

Descrizione

Funzione chiamata quando lo strumento estensibile è attivo e l'utente preme un tasto. Lo script deve chiamare `tools.getKeyDown()` per determinare quale tasto sia stato premuto.

Esempio

L'esempio seguente visualizza informazioni sul tasto che è stato premuto quando lo strumento estensibile è attivo e l'utente preme un tasto.

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

Consultate anche

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

keyUp()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

Nessuno.

Restituisce

Nulla.

Descrizione

Funzione chiamata quando lo strumento estensibile è attivo e viene rilasciato un tasto.

Esempio

L'esempio seguente visualizza un messaggio nel pannello Output quando lo strumento estensibile è attivo e viene rilasciato un tasto.

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

Consultate anche

keyDown()

mouseDoubleClick()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

Nessuno.

Restituisce

Nulla.

Descrizione

Funzione chiamata quando lo strumento estensibile è attivo e fate doppio clic sullo stage.

Esempio

L'esempio seguente visualizza un messaggio nel pannello Output quando lo strumento estensibile è attivo e fate doppio clic con il pulsante del mouse.

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

mouseDown()

Disponibilità

Flash MX 2004.

Uso

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

Parametri
pt Un punto che specifica la posizione del mouse quando viene premuto il pulsante. Viene passato alla funzione quando l'utente preme il pulsante del mouse. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Funzione chiamata quando lo strumento estensibile è attivo e l'utente preme il pulsante del mouse mentre il puntatore si trova sullo stage.

Esempio
Gli esempi seguenti illustrano come è possibile utilizzare questa funzione quando lo strumento estensibile è attivo. Il primo esempio visualizza un messaggio nel pannello Output che avvisa che il pulsante del mouse è stato premuto. Il secondo esempio visualizza le coordinate *x* e *y* della posizione in cui è stato premuto il pulsante del mouse.

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

mouseMove()

Disponibilità
Flash MX 2004.

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

Parametri
pt Un punto che specifica la posizione attuale del mouse. Poiché viene passato alla funzione ogni volta che l'utente sposta il mouse, tiene traccia della posizione del mouse. Se lo stage è in modalità Modifica o Modifica in posizione, le coordinate del punto sono relative all'oggetto che state modificando. In caso contrario, sono relative allo stage. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Funzione chiamata quando lo strumento estensibile è attivo e l'utente sposta il mouse in un punto specifico dello stage. Il pulsante del mouse può essere premuto o rilasciato.

Esempio

Gli esempi seguenti illustrano come è possibile utilizzare questa funzione. Il primo esempio visualizza un messaggio nel pannello Output che avvisa che il mouse è in movimento. Il secondo messaggio visualizza le coordinate *x* e *y* della posizione del mouse mentre si sposta.

```
function mouseMove() {
    fl.trace("moving");
}

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

mouseUp()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

Nessuno.

Restituisce

Nulla.

Descrizione

Funzione chiamata quando lo strumento estensibile è attivo e l'utente rilascia il pulsante del mouse dopo averlo premuto sullo stage.

Esempio

L'esempio seguente visualizza un messaggio nel pannello Output quando lo strumento estensibile è attivo e il pulsante del mouse viene rilasciato.

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

notifySettingsChanged()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

Nessuno.

Restituisce

Nulla.

Descrizione

Funzione chiamata quando lo strumento estensibile è attivo e l'utente ne modifica le opzioni nella finestra di ispezione Proprietà. È possibile utilizzare la proprietà `tools.activeTool` per interrogare i valori correnti delle opzioni (consultate `tools.activeTool`).

Esempio

L'esempio seguente visualizza un messaggio nel pannello Output quando lo strumento estensibile è attivo e l'utente ne modifica le opzioni nella finestra di ispezione Proprietà.

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

prompt()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

`promptMsg` Un stringa da visualizzare nella finestra di dialogo di richiesta (in Mac OS X la lunghezza massima consentita è di 256 caratteri).

`text` Una stringa opzionale da visualizzare come valore predefinito per il campo di testo.

Restituisce

La stringa specificata dall'utente se quest'ultimo fa clic su OK; `null` se l'utente fa clic su Annulla.

Descrizione

Metodo; visualizza una richiesta e un testo facoltativo in una finestra di avviso a scelta obbligatoria, insieme ai pulsanti OK e Annulla.

Esempio

L'esempio seguente richiede all'utente di immettere un nome utente. Se l'utente digita un nome e fa clic su OK, il nome viene visualizzato nel pannello Output.

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

Consultate anche

`alert()`, `confirm()`

setCursor()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

Nessuno.

Restituisce

Nulla.

Descrizione

Funzione chiamata quando lo strumento estensibile è attivo e viene spostato il mouse, al fine di consentire allo script di impostare puntatori personalizzati. Lo script deve chiamare `tools.setCursor()` per specificare il puntatore da utilizzare. Per consultare un elenco delle corrispondenze tra puntatori e valori interi, consultate `tools.setCursor()`.

Esempio

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

Capitolo 3: Oggetto actionsPanel

Disponibilità
Flash CS3 Professional.

Descrizione
L'oggetto actionsPanel, che rappresenta il pannello Azioni attualmente visualizzato, è una proprietà dell'oggetto flash (consultate fl.actionsPanel).

Riepilogo dei metodi
Con l'oggetto actionsPanel è possibile utilizzare i metodi seguenti:

Metodo	Descrizione
`actionsPanel.getClassForObject()`	Restituisce la classe della variabile specificata.
`actionsPanel.getScriptAssistMode()`	Specifica se è abilitata la modalità Assistente script.
`actionsPanel.getSelectedText()`	Restituisce il testo attualmente selezionato nel pannello Azioni.
`actionsPanel.getText()`	Restituisce il testo nel pannello Azioni.
`actionsPanel.hasSelection()`	Specifica se attualmente nel pannello Azioni è presente testo selezionato.
`actionsPanel.replaceSelectedText()`	Sostituisce il testo attualmente selezionato con il testo specificato.
`actionsPanel.setScriptAssistMode()`	Abilita o disabilita la modalità Assistente script.
`actionsPanel.setSelection()`	Seleziona un set di caratteri specificato nel pannello Azioni.
`actionsPanel.setText()`	Cancella il testo nel pannello Azioni e vi aggiunge il testo specificato.

actionsPanel.getClassForObject()

Disponibilità
Flash CS3 Professional.

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

Parametri
`ASvariableName` Una stringa che rappresenta il nome di una variabile di ActionScript.

Restituisce
Una stringa che rappresenta la classe di cui *ASvariableName* è membro.

Descrizione
Metodo; restituisce la classe della variabile specificata, la quale deve essere definita nel pannello Azioni attualmente visualizzato. Inoltre, il cursore o il testo selezionato nel pannello Azioni devono essere posizionati dopo la definizione della variabile.

Esempio
Se il cursore è posizionato dopo l'istruzione varmyVar:ActivityEvent; nel pannello Azioni, l'esempio seguente mostra la classe assegnata alla variabile myVar.

```
// 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()

Disponibilità
Flash CS3 Professional.

Uso
`actionsPanel.getScriptAssistMode()`

Parametri
Nessuno.

Restituisce
Un valore booleano che specifica se la modalità Assistente script è abilitata (`true`) o meno (`false`).

Descrizione
Metodo; specifica se la modalità Assistente script è abilitata.

Esempio
Se la modalità Assistente script non è abilitata, l'esempio seguente visualizza un messaggio.

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

Consultate anche
`actionsPanel.setScriptAssistMode()`

actionsPanel.getSelectedText()

Disponibilità
Flash CS3 Professional.

Uso
```
actionsPanel.getSelectedText()
```

Parametri
Nessuno.

Restituisce
Una stringa che contiene il testo attualmente selezionato nel pannello Azioni.

Descrizione
Metodo; restituisce il testo attualmente selezionato nel pannello Azioni.

Esempio
Nell'esempio seguente viene visualizzato il testo attualmente selezionato nel pannello Azioni.

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

Consultate anche
`actionsPanel.getText()`, `actionsPanel.hasSelection()`, `actionsPanel.replaceSelectedText()`, `actionsPanel.setSelection()`

actionsPanel.getText()

Disponibilità
Flash CS3 Professional.

Uso
```
actionsPanel.getText()
```

Parametri
Nessuno.

Restituisce
Una stringa che contiene tutto il testo presente nel pannello Azioni.

Descrizione
Metodo; restituisce il testo nel pannello Azioni.

Esempio
Nell'esempio seguente viene visualizzato il testo presente nel pannello Azioni.

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

Consultate anche

actionsPanel.getSelectedText(), actionsPanel.setText()

actionsPanel.hasSelection()

Disponibilità

Flash CS3 Professional.

Uso

```
actionsPanel.hasSelection()
```

Parametri

Nessuno.

Restituisce

Un valore booleano che specifica se nel pannello Azioni è selezionato del testo (`true`) o meno (`false`).

Descrizione

Metodo; specifica se attualmente nel pannello Azioni è presente testo selezionato.

Esempio

Nell'esempio seguente viene visualizzato il testo attualmente selezionato nel pannello Azioni. Se non è selezionato alcun testo, viene visualizzato tutto il testo presente.

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

Consultate anche

actionsPanel.getSelectedText(), actionsPanel.getText(), actionsPanel.replaceSelectedText(), actionsPanel.setSelection()

actionsPanel.replaceSelectedText()

Disponibilità

Flash CS3 Professional.

Uso

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

Parametri

`replacementText` Una stringa che rappresenta il testo che deve sostituire il testo selezionato nel pannello Azioni.

Restituisce

Il valore booleano `true` se il pannello Azioni è presente; `false` in caso contrario.

Descrizione

Metodo; sostituisce il testo attualmente selezionato con il testo specificato in *replacementText*. Se *replacementText* contiene un numero di caratteri superiore a quello del testo selezionato, i caratteri alla fine del testo selezionato saranno inseriti dopo *replacementText*, pertanto non saranno sovrascritti.

Esempio

L'esempio seguente sostituisce il testo attualmente selezionato nel pannello Azioni.

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

Consultate anche

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

actionsPanel.setScriptAssistMode()

Disponibilità

Flash CS3 Professional.

Uso

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

Parametri

`bScriptAssist` Un valore booleano che specifica se abilitare o disabilitare la modalità Assistente script.

Restituisce

Un valore booleano che specifica se la modalità Assistente script è stata abilitata o disabilitata correttamente.

Descrizione

Metodo; abilita o disabilita la modalità Assistente script.

Esempio

L'esempio seguente cambia lo stato della modalità Assistente script.

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

Consultate anche

`actionsPanel.getScriptAssistMode()`

actionsPanel.setSelection()

Disponibilità

Flash CS3 Professional.

Uso

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

Parametri

`startIndex` Un numero intero a base zero che specifica il primo carattere da selezionare.

`numberOfChars` Un numero intero che specifica il numero di caratteri da selezionare.

Restituisce

Un valore booleano che specifica se i caratteri richiesti possono essere selezionati (`true`) o meno (`false`).

Descrizione

Metodo; seleziona un set di caratteri specificato nel pannello Azioni.

Esempio

L'esempio seguente sostituisce i caratteri “2006” nel pannello Azioni con il testo specificato.

```
// 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");
```

Consultate anche

`actionsPanel.getSelectedText()`, `actionsPanel.hasSelection()`,
`actionsPanel.replaceSelectedText()`

actionsPanel.setText()

Disponibilità
Flash CS3 Professional.

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

Parametri
`replacementText` Una stringa che rappresenta il testo da inserire nel pannello Azioni.

Restituisce
Il valore booleano `true` se il testo specificato è stato inserito nel pannello Azioni; `false` in caso contrario.

Descrizione
Metodo; cancella il testo eventualmente presente nel pannello Azioni e vi aggiunge il testo specificato in *replacementText*.

Esempio
L'esempio seguente sostituisce il testo presente nel pannello Azioni con il testo specificato.

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

Consultate anche
`actionsPanel.getText()`, `actionsPanel.replaceSelectedText()`

Capitolo 4: Oggetto BitmapInstance

Ereditarietà Oggetto Element > Oggetto Instance > Oggetto BitmapInstance

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto BitmapInstance è una sottoclasse dell'oggetto Instance e rappresenta una bitmap in un fotogramma (consultate Oggetto Instance).

Riepilogo dei metodi
Oltre a quelli dell'Oggetto Instance, con l'oggetto BitmapInstance è possibile utilizzare i seguenti metodi:

Metodo	Descrizione
`bitmapInstance.getBits()`	Consente di creare effetti bitmap estraendo i bit dalla bitmap, manipolandoli e infine riportandoli in Flash.
`bitmapInstance.setBits()`	Imposta i bit di un elemento bitmap esistente.

Riepilogo delle proprietà
Oltre a quelle dell'Oggetto Instance, con l'oggetto BitmapInstance è possibile utilizzare le seguenti proprietà:

Proprietà	Descrizione
`bitmapInstance.hPixels`	Di sola lettura; un numero intero che rappresenta la larghezza della bitmap, espressa in pixel.
`bitmapInstance.vPixels`	Di sola lettura; un numero intero che rappresenta l'altezza della bitmap, espressa in pixel.

bitmapInstance.getBits()

Disponibilità
Flash MX 2004.

Uso
```
bitmapInstance.getBits()
```

Parametri
Nessuno.

Restituisce
Un oggetto che contiene le proprietà `width`, `height`, `depth`, `bits` e, se la bitmap ha una tavola colori, `cTab`. L'elemento `bits` è un array di byte. L'elemento `cTab` è un array di valori di colore espressi nel formato `"#RRGGBB"`. La lunghezza dell'array corrisponde alla lunghezza della tavola colori.

L'array dei byte è significativo solo quando vi si fa riferimento da una DLL o da una libreria condivisa e generalmente viene utilizzato solo nella creazione di oggetti estensibili o effetti. Per informazioni sulla creazione di DLL da usare con JavaScript di Flash, consultate "Estensibilità di livello C" a pagina 534.

Descrizione

Metodo; consente di creare effetti bitmap estraendo i bit dalla bitmap, manipolandoli e infine riportandoli in Flash.

Esempio

Il codice seguente crea un riferimento all'oggetto selezionato, verifica se l'oggetto è una bitmap e traccia l'altezza, la larghezza e la profondità di bit della bitmap:

```
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);
}
```

Consultate anche

`bitmapInstance.setBits()`

bitmapInstance.hPixels

Disponibilità

Flash MX 2004.

Uso

```
bitmapInstance.hPixels
```

Descrizione

Proprietà di sola lettura; un numero intero che rappresenta la larghezza della bitmap, vale a dire il numero dei pixel sulla dimensione orizzontale.

Esempio

Il codice seguente ottiene la larghezza della bitmap espressa in pixel:

```
// 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;
}
```

Consultate anche

`bitmapInstance.vPixels`

bitmapInstance.setBits()

Disponibilità
Flash MX 2004.

Uso
`bitmapInstance.setBits(bitmap)`

Parametri
bitmap Un oggetto che contiene le proprietà `height`, `width`, `depth`, `bits` e `cTab`. Le proprietà `height`, `width` e `depth` sono numeri interi. La proprietà `bits` è un array di byte. La proprietà `cTab` è richiesta solo per le bitmap con una profondità di bit non superiore a 8 ed è una stringa che rappresenta un valore colore espresso nel formato `"#RRGGBB"`.

Nota: *l'array dei byte è significativo solo quando vi si fa riferimento da una libreria esterna e generalmente viene utilizzato solo nella creazione di oggetti estensibili o effetti.*

Restituisce
Nulla.

Descrizione
Metodo; imposta i bit di un elemento bitmap esistente. Consente di creare effetti bitmap estraendo i bit dalla bitmap, manipolandoli e infine riportando la bitmap in Flash.

Esempio
Il codice seguente verifica se la selezione corrente è una bitmap, quindi imposta l'altezza della bitmap su 150 pixel:

```
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);
}
```

Consultate anche
`bitmapInstance.getBits()`

bitmapInstance.vPixels

Disponibilità
Flash MX 2004.

Uso
`bitmapInstance.vPixels`

Descrizione
Di sola lettura; un numero intero che rappresenta l'altezza della bitmap, vale a dire il numero dei pixel sulla dimensione verticale.

Esempio

Il codice seguente ottiene l'altezza della bitmap espressa in pixel:

```
// 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;
}
```

Consultate anche

bitmapInstance.hPixels

Capitolo 5: Oggetto BitmapItem

Ereditarietà Oggetto Item > Oggetto SymbolItem

Disponibilità
Flash MX 2004.

Descrizione
Un oggetto BitmapItem fa riferimento a una bitmap presente nella libreria di un documento ed è una sottoclasse dell'oggetto Item (consultate Oggetto Item).

Riepilogo delle proprietà
Oltre a quelle dell'Oggetto Item, l'oggetto BitmapItem è dotato delle seguenti proprietà:

Proprietà	Descrizione
`bitmapItem.allowSmoothing`	Un valore booleano che specifica se è consentita la smussatura di una bitmap.
`bitmapItem.compressionType`	Una stringa che determina il tipo di compressione immagini applicata alla bitmap.
`bitmapItem.fileLastModifiedDate`	Il numero di secondi trascorsi tra il 1° gennaio 1970 e la data della modifica del file originale.
`bitmapItem.originalCompressionType`	Specifica se l'elemento è stato importato come file JPEG.
`bitmapItem.sourceFileExists`	Specifica se il file importato nella libreria esiste ancora nel percorso da cui è stato importato.
`bitmapItem.sourceFileIsCurrent`	Specifica se la data di modifica del file dell'elemento della libreria corrisponde alla data di modifica su disco del file importato.
`bitmapItem.sourceFilePath`	Il percorso e il nome del file importato nella libreria.
`bitmapItem.useDeblocking`	Specifica se il deblocking è attivato.
`bitmapItem.useImportedJPEGQuality`	Un valore booleano che specifica se deve essere utilizzata la qualità JPEG importata predefinita.

Riepilogo dei metodi
Oltre a quelli dell'Oggetto Item, l'oggetto BitmapItem è dotato dei seguenti metodi:

Metodo	Descrizione
`bitmapItem.exportToFile()`	Esporta l'elemento specificato in un file in formato PNG o JPG.

bitmapItem.allowSmoothing

Disponibilità
Flash MX 2004.

Uso
`bitmapItem.allowSmoothing`

Descrizione
Proprietà; un valore booleano che specifica se viene consentita la smussatura di una bitmap (`true`) o meno (`false`).

Esempio
Il codice seguente imposta su `true` la proprietà `allowSmoothing` del primo elemento nella libreria del documento corrente:

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

bitmapItem.compressionType

Disponibilità
Flash MX 2004.

Uso
`bitmapItem.compressionType`

Descrizione
Proprietà; una stringa che determina il tipo di compressione delle immagini applicata alla bitmap. I valori accettabili sono `"photo"` e `"lossless"`. Se il valore di `bitmapItem.useImportedJPEGQuality` è `false`, `"photo"` corrisponde a un file JPEG con una qualità compresa tra 0 e100; se `bitmapItem.useImportedJPEGQuality` è `true`, `"photo"` corrisponde a un file JPEG con il valore della qualità predefinita del documento. Il valore `"lossless"` corrisponde al formato GIF o PNG (consultate bitmapItem.useImportedJPEGQuality).

Esempio
Il codice seguente imposta su `"photo"` la proprietà `compressionType` del primo elemento nella libreria del documento corrente:

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

bitmapItem.exportToFile()

Disponibilità
Flash CS4 Professional.

Uso
`bitmapItem.exportToFile(fileURI)`

Parametri
`fileURI` Una stringa, espressa nel formato file:///URI, che specifica il nome e il percorso del file esportato.

Restituisce
Un valore booleano: `true` se il file è stato esportato con successo; `false` in caso contrario.

Descrizione
Metodo; esporta l'elemento specificato nel formato PNG o JPG.

Esempio
Presupponendo che il primo elemento della libreria sia un elemento bitmap, il seguente codice lo esporta nel formato JPG:

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

bitmapItem.fileLastModifiedDate

Disponibilità
Flash CS4 Professional.

Uso
`bitmapItem.fileLastModifiedDate`

Descrizione
Proprietà di sola lettura; una stringa contenente un numero esadecimale che rappresenta il numero di secondi trascorsi tra il 1° gennaio 1970 e la data della modifica del file originale, nel momento in cui il file è stato importato nella libreria. Se il file non esiste più, il valore è "00000000".

Esempio
Presupponendo che il primo elemento della libreria sia un elemento bitmap, il seguente codice visualizza un numero esadecimale secondo quanto descritto sopra:

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

Consultate anche
`bitmapItem.sourceFileExists`, `bitmapItem.sourceFileIsCurrent`, `bitmapItem.sourceFilePath`, `FLfile.getModificationDate()`

bitmapItem.originalCompressionType

Disponibilità
Flash CS4 Professional.

Uso
`bitmapItem.originalCompressionType`

Descrizione
Proprietà di sola lettura; una stringa che specifica se l'elemento specificato è stato importato come file JPEG. I valori possibili per questa proprietà sono “photo” (per i file JPEG) e “lossless” (per i tipi di file non compressi come GIF e PNG).

Esempio
Presupponendo che il primo elemento della libreria sia un elemento bitmap, il seguente codice restituisce "photo" se il file è stato importato nella Librerie come file JPEG oppure "loseless" negli altri casi:

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

Consultate anche
`bitmapItem.compressionType`

bitmapItem.quality

Disponibilità
Flash MX 2004.

Uso
`bitmapItem.quality`

Descrizione
Proprietà; un numero intero che specifica la qualità della bitmap. Per utilizzare la qualità predefinita del documento, specificate -1; altrimenti, specificate un numero intero compreso tra 0 e 100. Disponibile solo per la compressione JPEG.

Esempio
Il codice seguente imposta su 65 la proprietà `quality` del primo elemento nella libreria del documento corrente:

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

bitmapItem.sourceFileExists

Disponibilità
Flash CS4 Professional.

Uso
`bitmapItem.sourceFileExists`

Descrizione
Proprietà di sola lettura; il valore booleano `true` se il file importato nella libreria esiste ancora nel percorso da cui è stato importato; `false` in caso contrario.

Esempio

Presupponendo che il primo elemento della libreria sia un elemento bitmap, il seguente codice restituisce "true" se il file importato nella libreria esiste ancora.

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

Consultate anche

`bitmapItem.sourceFileIsCurrent`,

`bitmapItem.sourceFilePath`

bitmapItem.sourceFileIsCurrent

Disponibilità

Flash CS4 Professional.

Uso

```
bitmapItem.sourceFileIsCurrent
```

Descrizione

Proprietà di sola lettura; il valore booleano `true` se la data di modifica del file dell'elemento della libreria corrisponde alla data di modifica su disco del file importato; `false` in caso contrario.

Esempio

Presupponendo che il primo elemento della libreria sia un elemento bitmap, il seguente codice restituisce "true" se il file importato non è stato modificato sul disco dopo l'importazione.

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

Consultate anche

`bitmapItem.fileLastModifiedDate`, `bitmapItem.sourceFilePath`

bitmapItem.sourceFilePath

Disponibilità

Flash CS4 Professional.

Uso

```
bitmapItem.sourceFilePath
```

Descrizione

Proprietà di sola lettura; una stringa, espressa come URI file:///, che rappresenta il percorso e il nome del file importato nella libreria.

Esempio

L'esempio seguente visualizza il percorso e il nome e il percorso del file di origine degli elementi della libreria di tipo `"bitmap"`:

```
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);
    }
}
```

Consultate anche

`bitmapItem.sourceFileExists`

bitmapItem.useDeblocking

Disponibilità

Flash CS4 Professional.

Uso

```
bitmapItem.useDeblocking
```

Descrizione

Proprietà; un valore booleano che specifica se il deblocking è attivato (`true`) o disattivato (`false`).

Esempio

Presupponendo che il primo elemento della libreria sia un elemento bitmap, il seguente codice attiva il deblocking per l'elemento:

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

bitmapItem.useImportedJPEGQuality

Disponibilità

Flash MX 2004.

Uso

```
bitmapItem.useImportedJPEGQuality
```

Descrizione

Proprietà; un valore booleano che specifica se deve essere utilizzata la qualità JPEG importata predefinita (`true`) o meno (`false`). Disponibile solo per la compressione JPEG.

Esempio

Il codice seguente imposta su `true` la proprietà `useImportedJPEGQuality` del primo elemento nella libreria del documento corrente:

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

Capitolo 6: Oggetto CompiledClipInstance

Ereditarietà Oggetto Element > Oggetto Instance > Oggetto CompiledClipInstance

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto CompiledClipInstance è una sottoclasse dell'oggetto Instance. Si tratta essenzialmente di un'istanza di un clip filmato che è stata convertita in un elemento di libreria di tipo clip compilato (consultate Oggetto Instance).

Riepilogo delle proprietà
Oltre a quelle dell'Oggetto Instance, l'oggetto CompiledClipInstance è dotato delle seguenti proprietà:

Proprietà	Descrizione
`compiledClipInstance.accName`	Una stringa equivalente al campo Nome del pannello Accessibilità.
`compiledClipInstance.actionScript`	Una stringa che rappresenta il codice ActionScript dell'istanza; equivalente a `symbolInstance.actionScript`.
`compiledClipInstance.description`	Una stringa equivalente al campo Descrizione del pannello Accessibilità.
`compiledClipInstance.forceSimple`	Un valore booleano che abilita e disabilita l'accessibilità degli elementi secondari dell'oggetto.
`compiledClipInstance.shortcut`	Una stringa equivalente al campo Tasto di scelta rapida del pannello Accessibilità.
`compiledClipInstance.silent`	Un valore booleano che abilita o disabilita l'accessibilità dell'oggetto; equivalente alla logica inversa dell'impostazione Rendi accessibile l'oggetto del pannello Accessibilità.
`compiledClipInstance.tabIndex`	Un numero intero equivalente al campo Indice tabulazione del pannello Accessibilità.

compiledClipInstance.accName

Disponibilità
Flash MX 2004.

Uso
```
compiledClipInstance.accName
```

Descrizione
Proprietà; una stringa equivalente al campo Nome del pannello Accessibilità. Gli screen reader identificano gli oggetti pronunciandone il nome.

Esempio

L'esempio seguente ottiene e imposta il nome assegnato al primo oggetto accessibile selezionato.

```
// 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

Disponibilità

Flash MX 2004.

Uso

`compiledClipInstance.actionScript`

Descrizione

Proprietà; una stringa che rappresenta il codice ActionScript dell'istanza; equivalente a `symbolInstance.actionScript`.

Esempio

Il codice seguente assegna ActionScript agli elementi specificati:

```
// 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

Disponibilità

Flash MX 2004.

Uso

`compiledClipInstance.description`

Descrizione

Proprietà; una stringa equivalente al campo Descrizione del pannello Accessibilità. La descrizione viene letta dallo screen reader.

Esempio

L'esempio seguente illustra come ottenere e impostare la proprietà `description`:

```
// 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

Disponibilità
Flash MX 2004.

Uso
`compiledClipInstance.forceSimple`

Descrizione
Proprietà; un valore booleano che abilita e disabilita l'accessibilità degli elementi secondari dell'oggetto. È equivalente alla logica inversa dell'impostazione Rendi accessibili gli oggetti secondari del pannello Accessibilità. Un valore `true` per `forceSimple` equivale all'opzione Rendi accessibili gli oggetti secondari deselezionata. Un valore `false` per `forceSimple` equivale alla stessa opzione selezionata.

Esempio
L'esempio seguente illustra come ottenere e impostare la proprietà `forceSimple`:

```
// 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

Disponibilità
Flash MX 2004.

Uso
`compiledClipInstance.shortcut`

Descrizione
Proprietà; una stringa equivalente al campo Tasto di scelta rapida del pannello Accessibilità. Il tasto di scelta rapida viene letto dallo screen reader. Questa proprietà non è disponibile per i campi di testo dinamici.

Esempio
L'esempio seguente illustra come ottenere e impostare la proprietà `shortcut`:

```
// 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

Disponibilità
Flash MX 2004.

Uso
`compiledClipInstance.silent`

Descrizione
Proprietà; un valore booleano che abilita o disabilita l'accessibilità dell'oggetto; equivalente alla logica inversa dell'impostazione Rendi accessibile l'oggetto del pannello Accessibilità. In altre parole, se `silent` è `true`, l'opzione Rendi accessibile l'oggetto è deselezionata. Se `silent` è `false`, l'opzione è selezionata.

Esempio
L'esempio seguente illustra come ottenere e impostare la proprietà `silent`:

```
// 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

Disponibilità
Flash MX 2004.

Uso
`compiledClipInstance.tabIndex`

Descrizione
Proprietà; un numero intero equivalente al campo Indice tabulazione del pannello Accessibilità. Crea l'ordine di tabulazione in base al quale viene eseguito l'accesso agli oggetti quando l'utente preme il tasto Tab.

Esempio
L'esempio seguente illustra come ottenere e impostare la proprietà `tabIndex`:

```
// 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;
```

Capitolo 7: Oggetto compilerErrors

Disponibilità
Flash CS3 Professional.

Descrizione
L'oggetto compilerErrors, che rappresenta il pannello Errori del compilatore, è una proprietà dell'oggetto flash (fl) alla quale è possibile accedere mediante `fl.compilerErrors` (consultate Oggetto Flash (fl)).

Riepilogo dei metodi
Con l'oggetto compilerErrors è possibile utilizzare i metodi seguenti.

Metodo	Descrizione
`compilerErrors.clear()`	Cancella il contenuto del pannello Errori del compilatore.
`compilerErrors.save()`	Salva il contenuto del pannello Errori del compilatore in un file di testo locale.

compilerErrors.clear()

Disponibilità
Flash CS3 Professional.

Uso
```
compilerErrors.clear()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; cancella il contenuto del pannello Errori del compilatore.

Esempio
L'esempio seguente cancella il contenuto del pannello Errori del compilatore:

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

Consultate anche
`compilerErrors.save()`

compilerErrors.save()

Disponibilità
Flash CS3 Professional.

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

Parametri
`fileURI` Una stringa, espressa nel formato file:///URI, che specifica il nome del file salvato. Se *fileURI* è già presente e non è stato ancora specificato un valore `true` per *bAppendToFile*, *fileURI* viene sovrascritto senza alcun messaggio di avvertimento.

`bAppendToFile` Valore booleano opzionale che specifica se il contenuto del pannello Errori del compilatore deve essere aggiunto a *fileURI* (`true`) o no (`false`). Il valore predefinito è `false`.

`bUseSystemEncoding` Valore booleano opzionale che specifica se salvare il testo del pannello Errori del compilatore utilizzando la codifica di sistema. Se il valore risulta `false` (impostazione predefinita), il pannello Errori del compilatore viene salvato mediante la codifica UTF-8, con caratteri BOM all'inizio del testo. Il valore predefinito è `false`.

Restituisce
Nulla.

Descrizione
Metodo, salva il contenuto del pannello Errori del compilatore in un file di testo locale.

Esempio
L'esempio che segue salva il contenuto del pannello Errori del compilatore nel file errors.log nella cartella C:\tests:

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

Consultate anche
`compilerErrors.clear()`

Capitolo 8: Oggetto ComponentInstance

Ereditarietà Oggetto Element > Oggetto Instance > Oggetto SymbolInstance > Oggetto ComponentInstance

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto ComponentInstance è una sottoclasse dell'oggetto SymbolInstance e rappresenta un componente in un fotogramma (consultate Oggetto SymbolInstance).

Riepilogo delle proprietà
Oltre a quelle dell'Oggetto SymbolInstance, l'oggetto ComponentInstance è dotato della seguente proprietà:

Proprietà	Descrizione
`componentInstance.parameters`	Di sola lettura; un array di proprietà ActionScript 2.0 a cui è possibile accedere dalla finestra di ispezione Proprietà dei componenti.

componentInstance.parameters

Disponibilità
Flash MX 2004.

Uso
```
componentInstance.parameters
```

Descrizione
Proprietà di sola lettura; un array di ActionScript 2.0 a cui è possibile accedere dalla finestra di ispezione Proprietà dei componenti. Consultate Oggetto Parameter.

Esempio
L'esempio seguente illustra come ottenere e impostare la proprietà `parameters`:

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

Consultate anche
Oggetto Parameter

Capitolo 9: Oggetto componentsPanel

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto componentsPanel, che rappresenta il pannello Componenti, è una proprietà dell'oggetto Flash (fl) alla quale è possibile accedere mediante `fl.componentsPanel` (consultate Oggetto Flash (fl)).

Riepilogo dei metodi
Con l'oggetto componentsPanel è possibile utilizzare i seguenti metodi:

Metodo	Descrizione
`componentsPanel.addItemToDocument()`	Aggiunge il componente specificato al documento nella posizione specificata.
`componentsPanel.reload()`	Aggiorna l'elenco dei componenti del pannello Componenti.

componentsPanel.addItemToDocument()

Disponibilità
Flash MX 2004.

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

Parametri
`position` Un punto (ad esempio `{x:0,y:100}`) che specifica la posizione in cui deve essere aggiunto il componente. Specificate *position* in relazione al punto centrale del componente e non al suo punto di registrazione (detto anche *punto di origine* o *punto zero*).

`categoryName` Una stringa che specifica il nome della categoria del componente (ad esempio, `"Data"`). I nomi di categoria validi sono elencati nel pannello Componenti.

`componentName` Una stringa che specifica il nome del componente nella categoria specificata (ad esempio, `"WebServiceConnector"`). I nomi di componente validi sono elencati nel pannello Componenti.

Restituisce
Nulla.

Descrizione
Aggiunge il componente specificato al documento nella posizione specificata.

Esempio
Gli esempi seguenti illustrano alcuni modi in cui può essere utilizzato questo metodo.

```
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()

Disponibilità
Flash 8.

Uso
`componentsPanel.reload()`

Parametri
Nessuno.

Restituisce
Un valore booleano `true` se l'elenco del pannello Componenti viene aggiornato; `false` in caso contrario.

Descrizione
Metodo; aggiorna l'elenco dei componenti del pannello Componenti.

Esempio
L'esempio seguente aggiorna il pannello Componenti:

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

Capitolo 10: Oggetto Contour

Disponibilità
Flash MX 2004.

Descrizione
Un oggetto Contour rappresenta un percorso chiuso di mezzi bordi sul contorno di una forma.

Riepilogo dei metodi
Con l'oggetto Contour potete utilizzare il metodo seguente:

Metodo	Descrizione
`contour.getHalfEdge()`	Restituisce un Oggetto HalfEdge sul contorno della selezione.

Riepilogo delle proprietà
Con l'oggetto Contour è possibile utilizzare le proprietà seguenti.

Proprietà	Descrizione
`contour.fill`	Un Oggetto Fill.
`contour.interior`	Sola lettura; il valore è `true` se il contorno racchiude un'area; altrimenti è `false`.
`contour.orientation`	Di sola lettura; un numero intero che indica l'orientamento del contorno.

contour.fill

Disponibilità
Flash CS4 Professional.

Uso
`contour.fill`

Descrizione
Proprietà; un Oggetto Fill.

Esempio
Presupponendo di avere un contorno con un riempimento selezionato, l'esempio seguente visualizza il colore di riempimento del contorno nel pannello Output:

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

contour.getHalfEdge()

Disponibilità

Flash MX 2004.

Uso

```
contour.getHalfEdge()
```

Parametri

Nessuno.

Restituisce

Un Oggetto HalfEdge.

Descrizione

Metodo; restituisce un Oggetto HalfEdge sul contorno della selezione.

Esempio

Questo esempio legge tutti i contorni della forma selezionata e mostra le coordinate dei vertici nel pannello Output:

```
// 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

Disponibilità
Flash MX 2004.

Uso
`contour.interior`

Descrizione
Proprietà di sola lettura; il valore è `true` se il contorno racchiude un'area; altrimenti è `false`.

Esempio
Questo esempio legge tutti i contorni della forma selezionata e mostra il valore della proprietà `interior` di ogni contorno nel pannello Output:

```
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

Disponibilità
Flash MX 2004.

Uso
`contour.orientation`

Descrizione
Proprietà di sola lettura; un numero intero che indica l'orientamento del contorno. Il valore del numero intero è -1 se l'orientamento è in senso antiorario, 1 se è in senso orario e 0 se è un contorno senza area.

Esempio
Questo esempio legge tutti i contorni della forma selezionata e mostra il valore della proprietà `orientation` di ogni contorno nel pannello Output:

```
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();
```

Capitolo 11: Oggetto Document

Disponibilità

Flash MX 2004.

Descrizione

L'oggetto Document rappresenta lo stage. Solo i file FLA vengono considerati documenti. Per restituire l'oggetto Document relativo al documento corrente, utilizzate `fl.getDocumentDOM()`.

Riepilogo dei metodi

Con l'oggetto Document potete utilizzare i metodi seguenti:

Metodo	Descrizione
`document.addDataToDocument()`	Memorizza i dati specificati in un documento.
`document.addDataToSelection()`	Memorizza i dati specificati negli oggetti selezionati.
`document.addFilter()`	Applica un filtro agli oggetti selezionati.
`document.addItem()`	Aggiunge all'oggetto Document specificato un elemento di un documento o di una libreria aperta.
`document.addNewLine()`	Aggiunge un nuovo percorso tra due punti.
`document.addNewOval()`	Aggiunge un nuovo oggetto Oval nel rettangolo di delimitazione specificato.
`document.addNewPrimitiveOval()`	Aggiunge un nuovo ovale di base che si adatti ai contorni specificati.
`document.addNewPrimitiveRectangle()`	Aggiunge un nuovo rettangolo di base che si adatti ai contorni specificati.
`document.addNewPublishProfile()`	Aggiunge un nuovo profilo di pubblicazione e lo rende il profilo corrente.
`document.addNewRectangle()`	Aggiunge un nuovo rettangolo o rettangolo arrotondato adattandolo ai contorni specificati.
`document.addNewScene()`	Aggiunge una nuova scena (Oggetto Timeline) come scena successiva a quella selezionata e sposta la selezione su di essa.
`document.addNewText()`	Inserisce un nuovo campo di testo vuoto.
`document.align()`	Allinea la selezione.
`document.allowScreens()`	Utilizzate questo metodo prima di utilizzare la proprietà `document.screenOutline`.
`document.arrange()`	Dispone la selezione sullo stage.
`document.breakApart()`	Esegue un'operazione di divisione della selezione corrente.
`document.canEditSymbol()`	Indica se sono attivati il menu Modifica simboli e la relativa funzionalità.
`document.canRevert()`	Determina se potete utilizzare correttamente il metodo `document.revert()` o `fl.revertDocument()`.

Metodo	Descrizione
`document.canSaveAVersion()`	Determina se è possibile salvare una versione del documento specificato sul server Version Cue.
`document.canTestMovie()`	Determina se potete utilizzare correttamente il metodo `document.testMovie()`.
`document.canTestScene()`	Determina se potete utilizzare correttamente il metodo `document.testScene()`.
`document.changeFilterOrder()`	Modifica l'indice del filtro nell'elenco dei filtri.
`document.clipCopy()`	Copia negli Appunti la selezione corrente del documento.
`document.clipCut()`	Taglia la selezione corrente dal documento e la inserisce negli Appunti.
`document.clipPaste()`	Incolla nel documento il contenuto degli Appunti.
`document.close()`	Chiude il documento specificato.
`document.convertLinesToFills()`	Converte le linee in riempimenti per gli oggetti selezionati.
`document.convertToSymbol()`	Converte in un nuovo simbolo gli elementi selezionati sullo stage.
`document.crop()`	Utilizza l'oggetto di disegno di primo livello selezionato per ritagliare tutti gli oggetti di disegno sottostanti.
`document.deleteEnvelope()`	Elimina l'involucro (ovvero il riquadro di delimitazione che contiene uno o più oggetti) dall'oggetto selezionato.
`document.deletePublishProfile()`	Se è presente più di un profilo, elimina quello attivo.
`document.deleteScene()`	Elimina la scena corrente (Oggetto Timeline) e, se la scena eliminata non era l'ultima, imposta quella immediatamente successiva come oggetto corrente della linea temporale.
`document.deleteSelection()`	Elimina la selezione corrente sullo stage.
`document.disableAllFilters()`	Disattiva tutti i filtri applicati agli oggetti selezionati.
`document.disableFilter()`	Disattiva il filtro specificato nell'elenco dei filtri.
`document.disableOtherFilters()`	Disattiva tutti i filtri a eccezione di quello che si trova nella posizione specificata nell'elenco dei filtri.
`document.distribute()`	Distribuisce la selezione.
`document.distributeToLayers()`	Esegue un'operazione di distribuzione sui livelli per la selezione corrente; equivale a selezionare l'opzione Distribuisci su livelli.
`document.documentHasData()`	Verifica la presenza nel documento di dati persistenti con il nome specificato.
`document.duplicatePublishProfile()`	Duplica il profilo attualmente attivo e rende attiva la versione duplicata.
`document.duplicateScene()`	Crea una copia della scena selezionata, assegnando alla nuova scena un nome univoco e rendendola la scena corrente.
`document.duplicateSelection()`	Duplica la selezione sullo stage.
`document.editScene()`	Seleziona la scena specificata per renderla modificabile.

Metodo	Descrizione
`document.enableAllFilters()`	Attiva tutti i filtri presenti nell'elenco dei filtri per gli oggetti selezionati.
`document.enableFilter()`	Attiva il filtro specificato per gli oggetti selezionati.
`document.enterEditMode()`	Attiva la modalità di modifica specificata dal parametro per lo strumento di creazione.
`document.exitEditMode()`	Esce dalla modalità di modifica dei simboli e rende attivo il livello immediatamente superiore nella modalità di modifica.
`document.exportPNG()`	Esporta il documento in uno o più file PNG.
`document.exportPublishProfile()`	Esporta il profilo attivo in un file XML.
`document.exportPublishProfileString()`	Restituisce una stringa che indica, nel formato XML, il profilo specificato.
`document.exportSWF()`	Esporta il documento nel formato SWF di Flash.
`document.getAlignToDocument()`	Equivale a recuperare il valore del pulsante Allo stage nel pannello Allinea.
`document.getBlendMode()`	Restituisce una stringa che specifica il metodo di fusione desiderato per gli oggetti selezionati.
`document.getCustomFill()`	Recupera l'oggetto Fill della forma selezionata o, se specificato, del pannello Strumenti e della finestra di ispezione Proprietà.
`document.getCustomStroke()`	Restituisce l'oggetto Stroke della forma selezionata o, se specificato, del pannello Strumenti e della finestra di ispezione Proprietà.
`document.getDataFromDocument()`	Recupera il valore dei dati specificati.
`document.getElementProperty()`	Ottiene la proprietà `Element` specificata per la selezione corrente.
`document.getElementTextAttr()`	Ottiene la proprietà `TextAttrs` specificata per gli oggetti Text selezionati.
`document.getFilters()`	Restituisce un array che contiene l'elenco dei filtri applicati agli oggetti attualmente selezionati.
`document.getMetadata()`	Restituisce una stringa che contiene i metadati XML associati al documento.
`document.getMobileSettings()`	Restituisce la stringa passata a `document.setMobileSettings()`.
`document.getPlayerVersion()`	Restituisce una stringa che rappresenta la versione del lettore previsto per il documento specificato.
`document.getSelectionRect()`	Ottiene il rettangolo di delimitazione della selezione corrente.
`document.getTextString()`	Ottiene il testo attualmente selezionato.
`document.getTimeline()`	Recupera l'Oggetto Timeline corrente nel documento.
`document.getTransformationPoint()`	Ottiene la posizione del punto di trasformazione della selezione corrente.
`document.group()`	Converte in gruppo la selezione corrente.
`document.importFile()`	Importa un file nel documento.

Metodo	Descrizione
`document.importPublishProfile()`	Importa un profilo da un file.
`document.importPublishProfileString()`	Importa una stringa XML che rappresenta un profilo di pubblicazione e lo imposta come profilo corrente.
`document.importSWF()`	Importa un file SWF nel documento.
`document.intersect()`	Crea un oggetto di disegno di intersezione da tutti gli oggetti di disegno selezionati.
`document.match()`	Rende uguali le dimensioni degli oggetti selezionati.
`document.mouseClick()`	Esegue un clic del mouse con lo strumento Selezione.
`document.mouseDblClk()`	Esegue un doppio clic del mouse con lo strumento Selezione.
`document.moveSelectedBezierPointsBy()`	Se la selezione contiene almeno un percorso con almeno un punto Bézier selezionato, questo metodo sposta tutti i punti Bézier selezionati su tutti i percorsi selezionati in base al valore specificato.
`document.moveSelectionBy()`	Sposta gli oggetti selezionati in base alla distanza specificata.
`document.optimizeCurves()`	Ottimizza la smussatura della selezione corrente, consentendo più passaggi (se specificato) per ottenere un risultato ottimale; equivale a selezionare Elabora > Forma > Ottimizza.
`document.publish()`	Pubblica il documento in base alle impostazioni di pubblicazione correnti (File > Impostazioni pubblicazione); equivale a selezionare File > Pubblica.
`document.punch()`	Sovrappone l'oggetto di disegno di primo livello selezionato a tutti gli oggetti di disegno sottostanti.
`document.removeAllFilters()`	Rimuove tutti i filtri dagli oggetti selezionati.
`document.removeDataFromDocument()`	Rimuove i dati persistenti con il nome specificato che sono stati associati al documento.
`document.removeDataFromSelection()`	Rimuove i dati persistenti con il nome specificato che sono stati associati alla selezione.
`document.removeFilter()`	Rimuove il filtro specificato dall'elenco dei filtri per gli oggetti selezionati.
`document.renamePublishProfile()`	Rinomina il profilo corrente.
`document.renameScene()`	Rinomina la scena selezionata nel pannello Scena.
`document.reorderScene()`	Sposta la scena specificata prima di un'altra scena specificata.
`document.resetOvalObject()`	Imposta tutti i valori nella finestra di ispezione Proprietà in base ai valori predefiniti nelle impostazioni dell'oggetto Oval.
`document.resetRectangleObject()`	Imposta tutti i valori nella finestra di ispezione Proprietà in base ai valori predefiniti nelle impostazioni dell'oggetto Rectangle.
`document.resetTransformation()`	Ripristina la matrice di trasformazione; equivale a selezionare Elabora > Trasforma > Elimina trasformazione.
`document.revert()`	Ripristina la versione salvata in precedenza del documento specificato; equivale a selezionare File > Ripristina.

Metodo	Descrizione
`document.revertToLastVersion()`	Ripristina il documento specificato alla versione memorizzata sul server Version Cue e visualizza eventuali errori nel pannello Output.
`document.rotate3DSelection()`	Applica una rotazione 3D alla selezione.
`document.rotateSelection()`	Ruota la selezione in base al numero di gradi specificato.
`document.save()`	Salva il documento nel percorso predefinito; equivale a selezionare File > Salva.
`document.saveAndCompact()`	Salva e compatta il file; equivale a selezionare File > Salva e compatta.
`document.saveAVersion()`	Salva una versione del documento specificato sul server Version Cue.
`document.scaleSelection()`	Scala la selezione in base al valore specificato; equivale a utilizzare lo strumento Trasformazione libera per modificare in scala l'oggetto.
`document.selectAll()`	Seleziona tutti gli elementi sullo stage; equivale a premere Ctrl+A (Windows) o Comando+A (Macintosh) o a selezionare Modifica > Seleziona tutto.
`document.selectNone()`	Deseleziona gli elementi selezionati.
`document.setAlignToDocument()`	Imposta le preferenze di `document.align()`, `document.distribute()`, `document.match()` e `document.space()` in relazione al documento; equivale ad attivare il pulsante Allo stage nel pannello Allinea.
`document.setBlendMode()`	Imposta il metodo di fusione per gli oggetti selezionati.
`document.setCustomFill()`	Configura le impostazioni di riempimento per il pannello Strumenti, per la finestra di ispezione Proprietà e per tutte le forme selezionate.
`document.setCustomStroke()`	Configura le impostazioni del tratto per il pannello Strumenti, per la finestra di ispezione Proprietà e per tutte le forme selezionate.
`document.setElementProperty()`	Imposta la proprietà `Element` specificata per gli oggetti selezionati nel documento.
`document.setElementTextAttr()`	Imposta sul valore specificato la proprietà `TextAttrs` specificata per gli elementi di testo selezionati.
`document.setFillColor()`	Sostituisce il colore di riempimento della selezione con il colore specificato.
`document.setFilterProperty()`	Imposta una proprietà Filter specificata per gli oggetti selezionati.
`document.setFilters()`	Applica i filtri agli oggetti selezionati.
`document.setInstanceAlpha()`	Imposta l'opacità dell'istanza.
`document.setInstanceBrightness()`	Imposta la luminosità dell'istanza.
`document.setInstanceTint()`	Imposta la tinta dell'istanza.
`document.setMetadata()`	Imposta i metadati XML per il documento specificato, sovrascrivendo quelli esistenti.

Metodo	Descrizione
`document.setMobileSettings()`	Imposta il valore di una stringa di impostazioni XML in un file FLA mobile.
`document.setOvalObjectProperty()`	Indica il valore di una specifica proprietà di oggetti Oval di base.
`document.setPlayerVersion()`	Imposta la versione del lettore Flash Player di destinazione per il documento specificato.
`document.setRectangleObjectProperty()`	Indica il valore di una proprietà specificata di oggetti Rectangle di base.
`document.setSelectionBounds()`	Sposta e ridimensiona la selezione con un'unica operazione.
`document.setSelectionRect()`	Disegna un perimetro di selezione rettangolare relativo allo stage, utilizzando le coordinate specificate.
`document.setStageVanishingPoint()`	Specifica il punto di fuga per la visualizzazione degli oggetti 3D.
`document.setStageViewAngle()`	Specifica l'angolo prospettico per la visualizzazione degli oggetti 3D.
`document.setStroke()`	Imposta il colore, la larghezza e lo stile dei tratti selezionati.
`document.setStrokeColor()`	Sostituisce il colore del tratto della selezione con il colore specificato.
`document.setStrokeSize()`	Sostituisce le dimensioni del tratto della selezione con le dimensioni specificate.
`document.setStrokeStyle()`	Sostituisce lo stile del tratto della selezione con lo stile specificato.
`document.setTextRectangle()`	Applica le dimensioni specificate al rettangolo di delimitazione dell'elemento di testo selezionato.
`document.setTextSelection()`	Imposta la selezione di testo del campo di testo selezionato sui valori specificati da ***`startIndex`*** e ***`endIndex`***.
`document.setTextString()`	Inserisce una stringa di testo.
`document.setTransformationPoint()`	Sposta il punto di trasformazione della selezione corrente.
`document.skewSelection()`	Inclina la selezione in base al valore specificato.
`document.smoothSelection()`	Smussa la curva di ogni contorno di riempimento o linea curva selezionata:
`document.space()`	Applica una spaziatura equidistante agli oggetti nella selezione.
`document.straightenSelection()`	Raddrizza i tratti selezionati; equivale a utilizzare il pulsante Raddrizza nel pannello Strumenti.
`document.swapElement()`	Scambia la selezione corrente con quella specificata.
`document.swapStrokeAndFill()`	Scambia i colori di tratto e riempimento.
`document.synchronizeWithHeadVersion()`	Sincronizza il documento specificato con la versione più recente presente sul server Version Cue e visualizza eventuali errori nel pannello Output.
`document.testMovie()`	Esegue l'operazione Prova filmato sul documento.

Metodo	Descrizione
`document.testScene()`	Esegue l'operazione Prova scena sulla scena corrente del documento.
`document.traceBitmap()`	Esegue un'operazione Ricalca bitmap sulla selezione corrente; equivale a selezionare Elabora > Bitmap > Ricalca bitmap.
`document.transformSelection()`	Applica una trasformazione generale alla selezione corrente applicando la matrice specificata negli argomenti.
`document.translate3DCenter()`	Imposta la posizione XYZ attorno a cui la selezione viene traslata o ruotata.
`document.translate3DSelection()`	Applica una traslazione 3D alla selezione.
`document.unGroup()`	Separa gli elementi della selezione corrente.
`document.union()`	Combina tutte le forme selezionate in un oggetto di disegno.
`document.unlockAllElements()`	Sblocca tutti gli elementi bloccati presenti nel fotogramma attualmente selezionato.
`document.xmlPanel()`	Inserisce una finestra di dialogo XMLUI.

Riepilogo delle proprietà

Con l'oggetto Document potete usare le proprietà seguenti.

Proprietà	Descrizione
`document.accName`	Una stringa equivalente al campo Nome del pannello Accessibilità.
`document.as3AutoDeclare`	Un valore booleano che descrive se le istanze collocate sullo stage devono essere aggiunte automaticamente alle classi Timeline definite dall'utente.
`document.as3Dialect`	Una stringa che descrive il "dialetto" ActionScript 3.0 usato nel documento specificato.
`document.as3ExportFrame`	Un numero intero che specifica il fotogramma in cui esportare le classi ActionScript3.0.
`document.as3StrictMode`	Un valore booleano che indica se il compilatore ActionScript 3.0 deve operare con l'opzione Modalità rigorosa attivata o disattivata.
`document.as3WarningsMode`	Un valore booleano che indica se il compilatore ActionScript 3.0 deve operare con l'opzione Modalità avvertenze attivata o disattivata.
`document.asVersion`	Un numero intero che specifica quale versione di ActionScript è utilizzata nel file specificato.
`document.autoLabel`	Un valore booleano equivalente alla casella di controllo Etichetta automatica del pannello Accessibilità.
`document.backgroundColor`	Una stringa, un valore esadecimale o un numero intero che rappresenta il colore di sfondo.
`document.currentPublishProfile`	Una stringa che specifica il nome del profilo di pubblicazione attivo per il documento specificato.
`document.currentTimeline`	Un numero intero che specifica l'indice della linea temporale attiva.
`document.description`	Una stringa equivalente al campo Descrizione del pannello Accessibilità.
`document.docClass`	Specifica la classe ActionScript 3.0 di primo livello associata al documento.

Proprietà	Descrizione
`document.externalLibraryPath`	Una stringa che contiene un elenco di elementi presenti nel percorso della libreria esterna di ActionScript 3.0 per il documento, che indica la posizione dei file SWC usati come librerie di runtime condivise.
`document.forceSimple`	Un valore booleano che specifica se gli elementi secondari dell'oggetto sono accessibili.
`document.frameRate`	Un valore float che specifica il numero di fotogrammi visualizzati al secondo durante la riproduzione del file SWF; il valore predefinito è 12.
`document.height`	Un numero intero che specifica l'altezza del documento (stage) espressa in pixel.
`document.id`	Un numero intero univoco (assegnato automaticamente) che identifica un documento durante una sessione Flash.
`document.library`	Sola lettura; l'Oggetto library di un documento.
`document.libraryPath`	Una stringa che contiene un elenco di elementi presenti nel percorso della libreria esterna di ActionScript 3.0 per il documento, che indica la posizione dei file SWC o delle cartelle SWC contenenti i file SWC.
`document.livePreview`	Un valore booleano che specifica se è attiva l'anteprima dal vivo.
`document.name`	Sola lettura; una stringa che rappresenta il nome di un documento (file FLA).
`document.path`	Sola lettura; una stringa che rappresenta il percorso del documento nel formato specifico di una piattaforma.
`document.pathURI`	Sola lettura; una stringa, espressa nel formato URI file:///, che rappresenta il percorso del documento.
`document.publishProfiles`	Sola lettura; un array dei nomi dei profili di pubblicazione per il documento.
`document.screenOutline`	Sola lettura; l'Oggetto ScreenOutline corrente per il documento.
`document.selection`	Un array degli oggetti selezionati nel documento.
`document.silent`	Un valore booleano che specifica se l'oggetto è accessibile.
`document.sourcePath`	Una stringa che contiene un elenco di elementi presenti nel percorso d'origine di ActionScript 3.0 per il documento, che indica la posizione dei file delle classi di ActionScript.
`document.timelines`	Sola lettura; un array di oggetti della linea temporale (consultate Oggetto Timeline).
`document.viewMatrix`	Sola lettura; un Oggetto Matrix.
`document.width`	Un numero intero che specifica la larghezza del documento (stage) espressa in pixel.
`document.zoomFactor`	Specifica la percentuale di ingrandimento dello stage durante la fase di creazione.

document.accName

Disponibilità

Flash MX 2004.

Uso

```
document.accName
```

Descrizione

Proprietà; una stringa equivalente al campo Nome del pannello Accessibilità. Gli screen reader identificano gli oggetti pronunciandone il nome.

Esempio

L'esempio seguente imposta `"Main Movie"` come nome assegnato a un documento accessibile:

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

L'esempio seguente ottiene il nome assegnato al documento accessibile:

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

document.addDataToDocument()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

`name` Una stringa che specifica il nome dei dati da aggiungere.

`type` Una stringa che definisce il tipo di dati da aggiungere. I valori accettabili sono `"integer"`, `"integerArray"`, `"double"`, `"doubleArray"`, `"string"` e `"byteArray"`.

`data` Il valore da aggiungere. I tipi validi dipendono dal parametro *type*.

Restituisce

Nulla.

Descrizione

Metodo; memorizza in un documento i dati specificati. I dati vengono scritti nel file FLA e sono disponibili per JavaScript quando il file viene riaperto.

Esempio

L'esempio seguente aggiunge un numero intero con valore 12 al documento corrente:

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

L'esempio seguente restituisce il valore presente in `"myData"` e visualizza il risultato nel pannello Output:

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

Consultate anche

`document.getDataFromDocument()`, `document.removeDataFromDocument()`

document.addDataToSelection()

Disponibilità
Flash MX 2004.

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

Parametri
`name` Una stringa che specifica il nome dei dati persistenti.

`type` Definisce il tipo di dati. I valori accettabili sono `"integer"`, `"integerArray"`, `"double"`, `"doubleArray"`, `"string"` e `"byteArray"`.

`data` Il valore da aggiungere. I tipi validi dipendono dal parametro *type*.

Restituisce
Nulla.

Descrizione
Metodo; memorizza i dati specificati negli oggetti selezionati. I dati vengono scritti nel file FLA e sono disponibili per JavaScript quando il file viene riaperto. Solo i simboli e le bitmap supportano i dati persistenti.

Esempio
L'esempio seguente aggiunge un numero intero con valore 12 all'oggetto corrente:

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

Consultate anche
`document.removeDataFromSelection()`

document.addFilter()

Disponibilità
Flash 8.

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

Parametri
`filterName` Una stringa che specifica il filtro da aggiungere all'elenco dei filtri e da attivare per gli oggetti selezionati. I valori accettabili sono `"adjustColorFilter"`, `"bevelFilter"`, `"blurFilter"`, `"dropShadowFilter"`, `"glowFilter"`, `"gradientBevelFilter"` e `"gradientGlowFilter"`.

Restituisce
Nulla.

Descrizione
Metodo; applica un filtro agli oggetti selezionati e posiziona il filtro alla fine dell'elenco dei filtri.

Esempio
L'esempio seguente applica un filtro alone agli oggetti selezionati:

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

Consultate anche
`document.changeFilterOrder()`, `document.disableFilter()`, `document.enableFilter()`, `document.getFilters()`, `document.removeFilter()`, `document.setBlendMode()`, `document.setFilterProperty()`

document.addItem()

Disponibilità
Flash MX 2004.

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

Parametri
`position` Un punto che specifica le coordinate *x* e *y* della posizione in cui deve essere aggiunto l'elemento. Utilizza il centro di un simbolo o l'angolo superiore sinistro di una bitmap o di un video.

`item` Un oggetto Item che specifica l'elemento da aggiungere e la libreria in cui trovarlo (consultate Oggetto Item).

Restituisce
Un valore booleano: `true` se l'esito è positivo; `false` in caso contrario.

Descrizione
Metodo; aggiunge all'oggetto Document specificato un elemento di un documento aperto o di una libreria aperta.

Esempio
L'esempio seguente aggiunge il primo elemento della libreria al primo documento nella posizione specificata per il simbolo, la bitmap o il video selezionato:

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

L'esempio seguente aggiunge al documento corrente il simbolo `myMovieClip` presente nella libreria dello stesso documento:

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

L'esempio seguente aggiunge al terzo documento presente nell'array di documenti il simbolo `myMovieClip` presente nel secondo documento dello stesso array:

```
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()

Disponibilità
Flash MX 2004.

Uso
`document.addNewLine(startPoint, endpoint)`

Parametri
`startpoint` Una coppia di numeri a virgola mobile che specificano le coordinate *x* e *y* del punto in cui inizia la linea.

`endpoint` Una coppia di numeri a virgola mobile che specificano le coordinate *x* e *y* del punto in cui termina la linea.

Restituisce
Nulla.

Descrizione
Metodo; aggiunge un nuovo percorso tra due punti. Il metodo utilizza gli attributi correnti del tratto del documento e aggiunge il percorso nel fotogramma e nel livello correnti. Equivale a fare clic sullo strumento Linea e a disegnare una linea.

Esempio
L'esempio seguente aggiunge una linea tra i punti iniziale e finale specificati:

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

document.addNewOval()

Disponibilità
Flash MX 2004.

Uso
`document.addNewOval(boundingRectangle [, bSuppressFill [, bSuppressStroke ]])`

Parametri
`boundingRectangle` Un rettangolo che specifica i contorni dell'ovale da aggiungere. Per informazioni sul formato di *boundingRectangle*, consultate `document.addNewRectangle()`.

`bSuppressFill` Un valore booleano che, se impostato su `true`, fa in modo che il metodo crei la forma senza riempimento. Il valore predefinito è `false`. Questo parametro è opzionale.

`bSuppressStroke` Un valore booleano che, se impostato su `true`, fa in modo che il metodo crei la forma senza tratto. Il valore predefinito è `false`. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; aggiunge un nuovo oggetto Oval nel rettangolo di delimitazione specificato. Questo metodo esegue la stessa operazione eseguita dallo strumento Ovale. Utilizza gli attributi predefiniti correnti del tratto e del riempimento del documento e aggiunge l'ovale nel fotogramma e nel livello correnti. Se sia *bSuppressFill* che *bSuppressStroke* sono impostati su `true`, il metodo non ha effetto.

Esempio
L'esempio seguente aggiunge un nuovo ovale all'interno delle coordinate specificate, ovvero 164 pixel in larghezza e 178 pixel in altezza:

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

L'esempio seguente disegna l'ovale senza riempimento:

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

L'esempio seguente disegna l'ovale senza tratto:

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

Consultate anche
`document.addNewPrimitiveOval()`

document.addNewPrimitiveOval()

Disponibilità
Flash CS4 Professional.

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

Parametri
`boundingRectangle` Un rettangolo che specifica i contorni all'interno dei quali viene aggiunto il nuovo ovale di base. Per informazioni sul formato di *boundingRectangle*, consultate document.addNewRectangle().

`bSuppressFill` Un valore booleano che, se impostato su `true`, fa in modo che il metodo crei l'ovale senza riempimento. Il valore predefinito è `false`. Questo parametro è opzionale.

`bSuppressStroke` Un valore booleano che, se impostato su `true`, fa in modo che il metodo crei l'ovale senza tratto. Il valore predefinito è `false`. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione

Metodo; aggiunge un nuovo ovale di base che si adatti ai contorni specificati. Questo metodo esegue la stessa operazione eseguita dallo strumento ovale di base. L'ovale di base usa gli attributi di tratto e riempimento del documento e viene aggiunto alla cornice e al livello correnti. Se sia *bSuppressFill* che *bSuppressStroke* sono impostati su `true`, il metodo non ha effetto.

Esempio

L'esempio seguente aggiunte ovali di base all'interno delle coordinate specificate, con e senza riempimenti e tratti:

```
// 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);
```

Consultate anche

`document.addNewOval()`

document.addNewPrimitiveRectangle()

Disponibilità

Flash CS4 Professional.

Uso

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

Parametri

`rect` Un rettangolo che specifica i contorni all'interno dei quali viene aggiunto il nuovo rettangolo di base. Per informazioni sul formato di *boundingRectangle*, consultate document.addNewRectangle().

`roundness` Un intero compreso tra 0 e 999 che rappresenta il numero di punti usati per specificare l'entità dell'arrotondamento degli angoli.

`bSuppressFill` Un valore booleano che, se impostato su `true`, fa in modo che il metodo crei il rettangolo senza riempimento. Il valore predefinito è `false`. Questo parametro è opzionale.

`bSuppressStroke` Un valore booleano che, se impostato su `true`, fa in modo che il metodo crei il rettangolo senza tratto. Il valore predefinito è `false`. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione

Metodo; aggiunge un nuovo rettangolo di base che si adatti ai contorni specificati. Questo metodo esegue la stessa operazione eseguita dallo strumento rettangolo di base. Il rettangolo di base usa gli attributi di tratto e riempimento del documento e viene aggiunto alla cornice e al livello correnti. Se sia *bSuppressFill* che *bSuppressStroke* sono impostati su `true`, il metodo non ha effetto.

Esempio

L'esempio seguente aggiunge rettangoli di base all'interno delle coordinate specificate, con e senza riempimenti e tratti e con differenti valori di arrotondamento:

```
// 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);
```

Consultate anche

`document.addNewRectangle()`

document.addNewPublishProfile()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

`profileName` Il nome univoco del nuovo profilo. Se non si specifica un nome, viene utilizzato un nome predefinito. Questo parametro è opzionale.

Restituisce

Un numero intero che corrisponde all'indice del nuovo profilo nell'elenco dei profili. Restituisce il valore -1 se non è possibile creare un nuovo profilo.

Descrizione

Metodo; aggiunge un nuovo profilo di pubblicazione e lo rende il profilo corrente.

Esempio

L'esempio seguente aggiunge un nuovo profilo di pubblicazione con un nome predefinito, quindi visualizza il nome del profilo nel pannello Output:

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

L'esempio seguente aggiunge un nuovo profilo di pubblicazione con il nome "`my profile`":

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

Consultate anche

`document.deletePublishProfile()`

document.addNewRectangle()

Disponibilità
Flash MX 2004.

Uso

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

Parametri

`boundingRectangle` Un rettangolo che specifica i contorni all'interno dei quali viene aggiunto il nuovo rettangolo, nel formato `{left:value1,top:value2,right:value3,bottom:value4}`. I valori `left` e `top` specificano la posizione dell'angolo superiore sinistro (ad esempio, `left:0,top:0` rappresenta l'angolo superiore sinistro dello stage), mentre i valori `right` e `bottom` indicano la posizione dell'angolo inferiore destro. Di conseguenza, la larghezza del rettangolo è la differenza tra i valori `left` e `right`, mentre l'altezza del rettangolo è la differenza tra i valori `top` e `bottom`.

In altre parole, i contorni del rettangolo non corrispondono tutti ai valori visualizzati nella finestra di ispezione Proprietà. I valori `left` e `top` corrispondono, rispettivamente, ai valori X e Y della finestra di ispezione Proprietà. Tuttavia, i valori `right` e `bottom` non corrispondono ai valori W e H della finestra di ispezione Proprietà. Esaminare, ad esempio, un rettangolo con i contorni seguenti:

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

I valori di questo rettangolo verrebbero visualizzati nel modo seguente nella finestra di ispezione Proprietà:

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

`roundness` Un valore intero compreso tra 0 e 999 che specifica l'arrotondamento da applicare agli angoli. Viene specificato in numero di punti: più alto è il valore, maggiore è l'arrotondamento.

`bSuppressFill` Un valore booleano che, se impostato su `true`, fa in modo che il metodo crei la forma senza riempimento. Il valore predefinito è `false`. Questo parametro è opzionale.

`bSuppressStroke` Un valore booleano che, se impostato su `true`, fa in modo che il metodo crei il rettangolo senza tratto. Il valore predefinito è `false`. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; aggiunge un nuovo rettangolo o rettangolo arrotondato adattandolo ai contorni specificati. Questo metodo esegue la stessa operazione eseguita dallo strumento Rettangolo. Utilizza gli attributi predefiniti correnti del tratto e del riempimento del documento e aggiunge il rettangolo nel fotogramma e nel livello correnti. Se sia *bSuppressFill* che *bSuppressStroke* sono impostati su `true`, il metodo non ha effetto.

Esempio
L'esempio seguente aggiunge un nuovo rettangolo senza angoli arrotondati all'interno delle coordinate specificate, ovvero 100 pixel in larghezza e in altezza:

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

L'esempio seguente aggiunge un nuovo rettangolo senza angoli arrotondati e senza riempimento e che misura 100 pixel in larghezza e 200 in altezza:

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

L'esempio seguente aggiunge un nuovo rettangolo senza angoli arrotondati e senza tratto e che misura 200 pixel in larghezza e 100 altezza:

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

Consultate anche

`document.addNewPrimitiveRectangle()`

document.addNewScene()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

`name` Specifica il nome della scena. Se non si specifica un nome, per la scena viene generato un nuovo nome.

Restituisce

Un valore booleano: `true` se l'aggiunta della scena ha esito positivo; `false` in caso contrario.

Descrizione

Metodo; aggiunge una nuova scena (`Oggetto Timeline`) come scena successiva a quella selezionata e sposta la selezione su di essa. Se il nome specificato esiste già, la scena non viene aggiunta e il metodo restituisce un errore.

Esempio

L'esempio seguente aggiunge la nuova scena `myScene` dopo quella corrente all'interno del documento corrente. La variabile `success` è `true` quando viene creata la nuova scena; in caso contrario è `false`.

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

L'esempio seguente aggiunge una nuova scena utilizzando la convenzione di denominazione predefinita. Se è presente una sola scena, la nuova scena viene denominata `"Scena 2"`.

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

document.addNewText()

Disponibilità

Flash MX 2004; parametro *text* opzionale aggiunto in FlashCS3 Professional.

Uso

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

Parametri

`boundingRectangle` Specifica dimensioni e posizione del campo di testo. Per informazioni sul formato di *boundingRectangle*, consultate `document.addNewRectangle()`.

`text` Una stringa opzionale che indica il testo da inserire nel campo. Omettendo questo parametro, la selezione nel pannello Strumenti passa allo strumento Testo. Immettere il valore di *text* per evitare il cambio dello strumento selezionato.

Restituisce

Nulla.

Descrizione

Metodo; inserisce un nuovo campo di testo e, se lo si desidera, vi immette del testo. Omettendo il parametro *text*, potete richiamare `document.setTextString()` per compilare il campo di testo.

Esempio

L'esempio seguente crea un nuovo campo di testo nell'angolo superiore sinistro dello stage, quindi imposta la stringa di testo su `"Hello World"`:

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

Consultate anche

`document.setTextString()`

document.align()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

`alignmode` Una stringa che specifica come allineare la selezione. I valori accettabili sono `"left"`, `"right"`, `"top"`, `"bottom"`, `"vertical center"` e `"horizontal center"`.

`bUseDocumentBounds` Un valore booleano che, se impostato su `true`, fa in modo che il metodo utilizzi i contorni del documento per l'allineamento. In caso contrario, il metodo utilizza i contorni degli oggetti selezionati. Il valore predefinito è `false`. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione
Metodo; allinea la selezione.

Esempio
L'esempio seguente allinea gli oggetti a sinistra e rispetto allo stage. Equivale ad attivare l'impostazione Allo stage nel pannello Allinea e a fare clic sul pulsante Allinea a sinistra:

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

Consultate anche
`document.distribute()`, `document.getAlignToDocument()`, `document.setAlignToDocument()`

document.allowScreens()

Disponibilità
Flash MX 2004.

Uso
`document.allowScreens()`

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se è possibile utilizzare `document.screenOutline` in sicurezza; `false` in caso contrario.

Descrizione
Metodo; da utilizzare prima della proprietà `document.screenOutline`. Se questo metodo restituisce il valore `true`, è possibile accedere in sicurezza a `document.screenOutline`; Flash visualizza un errore se si accede a `document.screenOutline` in un documento senza schermate.

Esempio
L'esempio seguente determina se è possibile utilizzare i metodi `screens` nel documento corrente:

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

Consultate anche
`document.screenOutline`

document.arrange()

Disponibilità
Flash MX 2004.

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

Parametri
`arrangeMode` Specifica la direzione in cui spostare la selezione. I valori accettabili sono `"back"`, `"backward"`, `"forward"` e `"front"`. Fornisce le stesse funzionalità delle opzioni del menu Elabora > Disponi.

Restituisce
Nulla.

Descrizione
Metodo; dispone la selezione sullo stage. Questo metodo funziona con tutti gli oggetti, a eccezione degli oggetti forma.

Esempio
L'esempio seguente sposta in primo piano la selezione corrente:

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

document.as3AutoDeclare

Disponibilità
Flash CS3 Professional.

Uso
```
document.as3AutoDeclare
```

Descrizione
Proprietà; un valore booleano che descrive se le istanze collocate sullo stage devono essere aggiunte automaticamente alle classi Timeline definite dall'utente. Il valore predefinito è `true`.

Esempio
Il seguente esempio indica che le istanze posizionate nello stage del documento corrente devono essere aggiunte manualmente alle classi Timeline definite dall'utente.

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

document.as3Dialect

Disponibilità
Flash CS3 Professional.

Uso

```
document.as3Dialect
```

Descrizione

Proprietà; una stringa che descrive il "dialetto" di ActionScript 3.0 usato nel documento specificato. Il valore predefinito è `"AS3"`. Per consentire l'uso di classi prototipo (come consentito dalle precedenti specifiche ECMAScript), impostate questo valore su `"ES"`.

Esempio

L'esempio che segue indica che nel documento corrente viene utilizzato il dialetto ECMAScript:

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

Consultate anche

`document.asVersion`

document.as3ExportFrame

Disponibilità

Flash CS3 Professional.

Uso

```
document.as3ExportFrame
```

Descrizione

Proprietà; un numero intero che specifica il fotogramma in cui si desidera esportare le classi ActionScript3.0. Per impostazione predefinita, le classi vengono esportate nel fotogramma1.

Esempio

Il seguente esempio modifica il fotogramma in cui vengono esportate le classi da1 (impostazione predefinita) a 5.

```
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

Disponibilità

Flash CS3 Professional.

Uso

```
document.as3StrictMode
```

Descrizione

Proprietà; un valore booleano che indica se il compilatore di ActionScript3.0 deve operare con l'opzione Modalità rigorosa attivata (`true`) o disattivata (`false`). La modalità rigorosa fa in modo che le avvertenze siano segnalate come errori; pertanto, in caso di presenza di tali errori la compilazione non viene completata correttamente. Il valore predefinito è `true`.

Esempio

L'esempio che segue disattiva l'opzione Modalità rigorosa del compilatore.

```
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);
```

Consultate anche

`document.as3WarningsMode`

document.as3WarningsMode

Disponibilità

Flash CS3 Professional.

Uso

```
document.as3WarningsMode
```

Descrizione

Proprietà; un valore booleano che indica se il compilatore ActionScript3.0 deve operare con l'opzione Modalità avvertenze attivata (`true`) o disattivata (`false`). La modalità avvertenze fa sì che le avvertenze siano segnalate durante la compilazione ed è utile per localizzare eventuali incompatibilità durante l'aggiornamento del codice ActionScript 2.0 ad ActionScript 3.0. Il valore predefinito è `true`.

Esempio

L'esempio che segue disattiva l'opzione Modalità avvertenze del compilatore.

```
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);
```

Consultate anche

`document.as3StrictMode`

document.asVersion

Disponibilità
Flash CS3 Professional.

Uso
`document.asVersion`

Descrizione
Proprietà; un numero intero che indica la versione di ActionScript utilizzata nel documento specificato. I valori accettabili sono compresi tra 1 e 3.

Per determinare la versione del lettore prevista per il document specificato, utilizzate `document.getPlayerVersion()`; questo metodo restituisce una stringa, pertanto può essere impiegato dai lettori Flash® Lite™.

Esempio
Se attualmente impostato per ActionScript 1.0, l'esempio che segue imposta la versione ActionScript 2.0 per il documento corrente.

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

Consultate anche
`document.as3Dialect`, `document.getPlayerVersion()`

document.autoLabel

Disponibilità
Flash MX 2004.

Uso
`document.autoLabel`

Descrizione
Proprietà; un valore booleano equivalente alla casella di controllo Etichetta automatica del pannello Accessibilità. È possibile utilizzarla per etichettare automaticamente gli oggetti sullo stage utilizzandone il testo.

Esempio
L'esempio seguente ottiene il valore della proprietà `autoLabel` e visualizza il risultato nel pannello Output:

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

L'esempio seguente imposta la proprietà `autoLabel` su `true` per etichettare automaticamente gli oggetti sullo stage:

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

document.backgroundColor

Disponibilità
Flash MX 2004.

Uso
```
document.backgroundColor
```

Descrizione
Proprietà; il colore di sfondo in uno dei seguenti formati:

- Una stringa nel formato `"#RRGGBB"` o `"#RRGGBBAA"`
- Un numero esadecimale nel formato `0xRRGGBB`
- Un numero intero che rappresenta l'equivalente decimale di un numero esadecimale.

Esempio
L'esempio seguente imposta il nero come colore di sfondo:

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

document.breakApart()

Disponibilità
Flash MX 2004.

Uso
```
document.breakApart()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; esegue un'operazione di divisione sulla selezione corrente.

Esempio
L'esempio seguente divide la selezione corrente:

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

document.canEditSymbol()

Disponibilità
Flash MX 2004.

Uso
`document.canEditSymbol()`

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se sono disponibili il menu Modifica simboli e la relativa funzionalità; `false` in caso contrario.

Descrizione
Metodo; indica se sono attivati il menu Modifica simboli e la relativa funzionalità. Questo metodo non specifica se la selezione può essere modificata. Non deve essere utilizzato per verificare se è consentito utilizzare `fl.getDocumentDOM().enterEditMode()`.

Esempio
L'esempio seguente visualizza nel pannello Output lo stato del menu Modifica simboli e della relativa funzionalità:

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

document.canRevert()

Disponibilità
Flash MX 2004.

Uso
`document.canRevert()`

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se è possibile utilizzare correttamente il metodo `document.revert()` o `fl.revertDocument()`; `false` in caso contrario.

Descrizione
Metodo; determina se è possibile utilizzare correttamente il metodo `document.revert()` o `fl.revertDocument()`.

Esempio

L'esempio seguente verifica se è possibile ripristinare la versione salvata in precedenza del documento corrente. In caso affermativo, `fl.getDocumentDOM().revert()` ripristina la versione salvata in precedenza.

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

document.canSaveAVersion()

Disponibilità

Flash CS3 Professional.

Uso

```
document.canSaveAVersion()
```

Parametri

Nessuno.

Restituisce

Il valore booleano `true` se la versione del file può essere salvata correttamente sul server Version Cue; `false` in caso contrario.

Descrizione

Metodo; determina se è possibile salvare una versione del documento specificato sul server Version Cue.

Esempio

L'esempio che segue verifica se è possibile utilizzare `document.saveAVersion()`. In caso affermativo, chiama il metodo.

```
if(fl.getDocumentDOM().canSaveAVersion()){
    fl.getDocumentDOM().saveAVersion;
}
```

Consultate anche

`document.revertToLastVersion()`, `document.saveAVersion()`

document.canTestMovie()

Disponibilità

Flash MX 2004.

Uso

```
document.canTestMovie()
```

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se è possibile utilizzare correttamente il metodo `document.testMovie()`; `false` in caso contrario.

Descrizione
Metodo; determina se è possibile utilizzare correttamente il metodo `document.testMovie()`.

Esempio
L'esempio seguente verifica se è possibile utilizzare `fl.getDocumentDOM().testMovie()`. In caso affermativo, chiama il metodo.

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

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

document.canTestScene()

Disponibilità
Flash MX 2004.

Uso
`document.canTestScene()`

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se è possibile utilizzare correttamente il metodo `document.testScene()`; `false` in caso contrario.

Descrizione
Metodo; determina se è possibile utilizzare correttamente il metodo `document.testScene()`.

Esempio
L'esempio seguente verifica se è possibile utilizzare correttamente fl.getDocumentDOM().testScene(). In caso affermativo, chiama il metodo.

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

Consultate anche

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

document.changeFilterOrder()

Disponibilità

Flash 8.

Uso

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

Parametri

`oldIndex` Un numero intero che rappresenta la posizione di indice a base zero corrente del filtro da riposizionare nell'elenco dei filtri.

`newIndex` Un numero intero che rappresenta la nuova posizione di indice del filtro nell'elenco.

Restituisce

Nulla.

Descrizione

Metodo; modifica l'indice del filtro nell'elenco dei filtri. I filtri sopra o sotto *newIndex* vengono spostati verso l'alto o verso il basso di conseguenza. Ad esempio, utilizzando i filtri illustrati di seguito, se si esegue il comando fl.getDocumentDOM().changeFilterOrder`(3,0)`, la disposizione dei filtri cambia nel modo seguente:

Prima	Dopo
`blurFilterdropShadowFilterglowFiltergradientBevelFilter`	`gradientBevelFilterblurFilterdropShadowFilterglowFilter`

Se quindi si esegue il comando fl.getDocumentDOM().changeFilterOrder`(0,2)`, la disposizione dei filtri cambia nel modo seguente:

Prima	Dopo
`gradientBevelFilterblurFilterdropShadowFilterglowFilter`	`blurFilterdropShadowFiltergradientBevelFilterglowFilter`

Esempio

L'esempio seguente sposta nella prima posizione dell'elenco dei filtri il filtro che attualmente si trova nella seconda posizione:

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

Consultate anche

`document.addFilter()`, `document.disableFilter()`, `document.enableFilter()`, `document.getFilters()`, `document.removeFilter()`, Oggetto Filter

document.clipCopy()

Disponibilità
Flash MX 2004.

Uso
```
document.clipCopy()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; copia negli Appunti la selezione corrente del documento

Per copiare una stringa negli Appunti, utilizzate `fl.clipCopyString()`.

Esempio
L'esempio seguente copia negli Appunti la selezione corrente del documento.

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

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

document.clipCut()

Disponibilità
Flash MX 2004.

Uso
```
document.clipCut()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; taglia la selezione corrente dal documento e la inserisce negli Appunti.

Esempio
L'esempio seguente taglia la selezione corrente dal documento e la inserisce negli Appunti.

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

Consultate anche

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

document.clipPaste()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

`bInPlace` Un valore booleano che, se impostato su `true`, fa in modo che il metodo incolli l'elemento nella stessa posizione dell'originale. Il valore predefinito è `false`, in base al quale il metodo incolla l'elemento al centro del documento. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione

Metodo; incolla nel documento il contenuto degli Appunti.

Esempio

L'esempio seguente incolla il contenuto degli Appunti al centro del documento:

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

L'esempio seguente incolla nel documento il contenuto degli Appunti nella stessa posizione dell'originale:

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

Consultate anche

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

document.close()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

`bPromptToSaveChanges` Un valore booleano che, se impostato su `true`, fa in modo che il metodo visualizzi una finestra di dialogo di richiesta in presenza di modifiche non salvate nel documento. Se *bPromptToSaveChanges* è impostato su `false`, non viene visualizzata alcuna richiesta di salvare i documenti modificati. Il valore predefinito è `true`. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione

Metodo; chiude il documento specificato.

Esempio

L'esempio seguente chiude il documento corrente e visualizza una finestra di dialogo che richiede all'utente di salvare le modifiche:

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

L'esempio seguente chiude il documento corrente senza salvare le modifiche:

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

document.convertLinesToFills()

Disponibilità

Flash MX 2004.

Uso

```
document.convertLinesToFills()
```

Parametri

Nessuno.

Restituisce

Nulla.

Descrizione

Metodo; converte le linee in riempimenti per gli oggetti selezionati.

Esempio

L'esempio seguente converte in riempimenti le linee selezionate:

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

document.convertToSymbol()

Disponibilità
Flash MX 2004.

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

Parametri
`type` Una stringa che specifica il tipo di simbolo da creare. I valori accettabili sono `"movie clip"`, `"button"` e `"graphic"`.

`name` Una stringa che specifica il nome univoco del nuovo simbolo. Se si specifica una stringa vuota, il metodo crea automaticamente un nome univoco per il simbolo.

`punto di registrazione` Specifica il punto che rappresenta la posizione 0,0 del simbolo. I valori accettabili sono: `"top left"`, `"top center"`, `"top right"`, `"center left"`, `"center"`, `"center right"`, `"bottom left"`, `"bottom center"` e `"bottom right"`.

Restituisce
Un oggetto per il simbolo creato oppure il valore `null` se non è possibile creare il simbolo.

Descrizione
Metodo; converte in un nuovo simbolo gli elementi selezionati sullo stage. Per informazioni sulla definizione del collegamento e sulle proprietà delle risorse condivise per un simbolo, consultate Oggetto Item.

Esempio
Gli esempi seguenti creano un simbolo di clip filmato con un nome specificato, un simbolo di pulsante con un nome specificato e un simbolo di clip filmato con un nome predefinito:

```
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()

Disponibilità
Flash 8.

Uso
```
document.crop()
```

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se l'esito è positivo; `false` in caso contrario.

Descrizione
Metodo; utilizza l'oggetto di disegno di primo livello selezionato per ritagliare tutti gli oggetti di disegno sottostanti. Il metodo restituisce `false` se non sono selezionati oggetti di disegno o se uno degli oggetti selezionati non è un oggetto di disegno.

Esempio
L'esempio seguente ritaglia gli oggetti selezionati:

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

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

document.currentPublishProfile

Disponibilità
Flash MX 2004.

Uso

```
document.currentPublishProfile
```

Descrizione
Proprietà; una stringa che specifica il nome del profilo di pubblicazione attivo per il documento specificato.

Esempio
L'esempio seguente aggiunge un nuovo profilo di pubblicazione con il nome predefinito, quindi visualizza il nome del profilo nel pannello Output:

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

L'esempio seguente imposta su `"Default"` il profilo di pubblicazione selezionato:

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

document.currentTimeline

Disponibilità
Flash MX 2004.

Uso

```
document.currentTimeline
```

Descrizione
Proprietà; un numero intero che specifica l'indice della linea temporale attiva. È possibile impostare la linea temporale attiva modificando il valore di questa proprietà; l'effetto è quasi uguale a quello che si ottiene chiamando `document.editScene()`. L'unica differenza è che non viene visualizzato un messaggio di errore se l'indice della linea temporale non è valido: semplicemente, la proprietà non viene impostata e l'azione fallisce senza che venga segnalato un errore.

Esempio
L'esempio seguente visualizza l'indice della linea temporale corrente:

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

L'esempio seguente passa dalla linea temporale principale attiva a una scena denominata `"myScene"`:

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

Consultate anche
`document.getTimeline()`

document.deleteEnvelope()

Disponibilità
Flash 8.

Uso
`document.deleteEnvelope()`

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se l'esito è positivo; `false` in caso contrario.

Descrizione
Metodo; elimina l'involucro (ovvero il riquadro di delimitazione che contiene uno o più oggetti) dagli oggetti selezionati.

Esempio
L'esempio seguente elimina l'involucro dagli oggetti selezionati:

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

Consultate anche

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

document.deletePublishProfile()

Disponibilità

Flash MX 2004.

Uso

```
document.deletePublishProfile()
```

Parametri

Nessuno.

Restituisce

Un numero intero che corrisponde all'indice del nuovo profilo corrente. Se non è disponibile un nuovo profilo, il metodo lascia invariato il profilo corrente e ne restituisce l'indice.

Descrizione

Metodo; elimina il profilo attivo, se ne è presente più di uno. Deve essere ancora presente almeno un profilo.

Esempio

L'esempio seguente elimina il profilo attivo, se ne è presente più di uno, e visualizza l'indice del nuovo profilo attivo:

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

Consultate anche

`document.addNewPublishProfile()`

document.deleteScene()

Disponibilità

Flash MX 2004.

Uso

```
document.deleteScene()
```

Parametri

Nessuno.

Restituisce

Un valore booleano: `true` se l'eliminazione della scena ha esito positivo; `false` in caso contrario.

Descrizione
Metodo; elimina la scena corrente (Oggetto Timeline) e, se la scena eliminata non era l'ultima, imposta quella immediatamente successiva come oggetto corrente della linea temporale. Se la scena eliminata era l'ultima, imposta il primo oggetto come oggetto corrente della linea temporale. Se nella linea temporale è presente solo un oggetto (scena), restituisce il valore `false`.

Esempio
Ipotizzando che nel documento corrente siano presenti tre scene (`Scene0`, `Scene1` e `Scene2`), l'esempio seguente rende corrente la scena `Scene2`, quindi la elimina:

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

document.deleteSelection()

Disponibilità
Flash MX 2004.

Uso
```
document.deleteSelection()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; elimina la selezione sullo stage. Visualizza un messaggio di errore se non è presente alcuna selezione.

Esempio
L'esempio seguente elimina la selezione corrente dal documento:

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

document.description

Disponibilità
Flash MX 2004.

Uso
```
document.description
```

Descrizione
Proprietà; una stringa equivalente al campo Descrizione del pannello Accessibilità. La descrizione viene letta dallo screen reader.

Esempio
L'esempio seguente imposta la descrizione del documento:

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

L'esempio seguente ottiene la descrizione del documento e la visualizza nel pannello Output:

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

document.disableAllFilters()

Disponibilità
Flash 8.

Uso
```
document.disableAllFilters()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; disattiva tutti i filtri applicati agli oggetti selezionati.

Esempio
L'esempio seguente disattiva tutti i filtri applicati agli oggetti selezionati:

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

Consultate anche
`document.addFilter()`, `document.changeFilterOrder()`, `document.disableFilter()`, `document.disableOtherFilters()`, `document.enableAllFilters()`, `document.getFilters()`, `document.removeAllFilters()`, Oggetto Filter

document.disableFilter()

Disponibilità
Flash 8.

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

Parametri
`filterIndex` Un numero intero che rappresenta l'indice a base zero del filtro nell'elenco dei filtri.

Restituisce
Nulla.

Descrizione
Metodo; disattiva il filtro specificato nell'elenco dei filtri.

Esempio
L'esempio seguente disattiva il primo e il terzo filtro (valori di indice 0 e 2) per gli oggetti selezionati nell'elenco dei filtri:

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

Consultate anche
`document.addFilter()`, `document.changeFilterOrder()`, `document.disableAllFilters()`, `document.disableOtherFilters()`, `document.enableFilter()`, `document.getFilters()`, `document.removeFilter()`, Oggetto Filter

document.disableOtherFilters()

Disponibilità
Flash 8.

Uso
`document.disableOtherFilters(enabledFilterIndex)`

Parametri
`enabledFilterIndex` Un numero intero che rappresenta l'indice a base zero del filtro che deve rimanere attivo dopo che vengono disattivati gli altri filtri.

Restituisce
Nulla.

Descrizione
Metodo; disattiva tutti i filtri a eccezione di quello che si trova nella posizione specificata nell'elenco dei filtri.

Esempio
L'esempio seguente disattiva tutti i filtri a eccezione del secondo filtro presente nell'elenco (indice con valore 1):

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

Consultate anche
`document.addFilter()`, `document.changeFilterOrder()`, `document.disableAllFilters()`, `document.disableFilter()`, `document.enableFilter()`, `document.getFilters()`, `document.removeFilter()`, Oggetto Filter

document.distribute()

Disponibilità
Flash MX 2004.

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

Parametri
`distributemode` Una stringa che specifica dove devono essere distribuiti gli oggetti selezionati. I valori accettabili sono `"left edge"`, `"horizontal center"`, `"right edge"`, `"top edge"`, `"vertical center"` e `"bottom edge"`.

`bUseDocumentBounds` Un valore booleano che, se impostato su `true`, distribuisce gli oggetti selezionati utilizzando i contorni del documento. In caso contrario, il metodo utilizza i contorni degli oggetti selezionati. Il valore predefinito è `false`.

Restituisce
Nulla.

Descrizione
Metodo; distribuisce la selezione.

Esempio
L'esempio seguente distribuisce gli oggetti selezionati lungo i bordi superiori:

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

L'esempio seguente distribuisce gli oggetti selezionati lungo i relativi bordi superiori e imposta esplicitamente il parametro *bUseDocumentBounds*:

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

L'esempio seguente distribuisce gli oggetti selezionati utilizzando come riferimento il loro bordo superiore rispetto ai contorni del documento:

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

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

document.distributeToLayers()

Disponibilità
Flash MX 2004.

Uso
```
document.distributeToLayers()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; esegue una distribuzione sui livelli per la selezione corrente; equivale a selezionare l'opzione Distribuisci su livelli. Visualizza un messaggio di errore se non è presente alcuna selezione.

Esempio
L'esempio seguente distribuisce sui livelli la selezione corrente:

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

document.docClass

Disponibilità
Flash CS3 Professional.

Uso

```
document.docClass
```

Descrizione
Proprietà; una stringa che specifica la classe ActionScript 3.0 di primo livello associata al documento. Se il documento non è configurato in modo da impiegare ActionScript3.0, la proprietà viene ignorata.

Esempio
L'esempio che segue indica che la classe di ActionScript3.0 associata al documento è com.mycompany.ManagerClass, definita in com/mycompany/ManagerClass.as:

```
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);
```

Consultate anche

```
item.linkageBaseClass
```

document.documentHasData()

Disponibilità
Flash MX 2004.

Uso

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

Parametri

`name` Una stringa che specifica il nome dei dati da verificare.

Restituisce

Un valore booleano: `true` se il documento contiene dati persistenti; `false` in caso contrario.

Descrizione

Metodo; verifica la presenza nel documento di dati persistenti con il nome specificato.

Esempio

L'esempio seguente verifica la presenza nel documento di dati persistenti denominati `"myData"`:

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

Consultate anche

`document.addDataToDocument()`, `document.getDataFromDocument()`,
`document.removeDataFromDocument()`

document.duplicatePublishProfile()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

`profileName` Una stringa che specifica il nome univoco del profilo duplicato. Se non si specifica un nome, il metodo utilizza il nome predefinito. Questo parametro è opzionale.

Restituisce

Un numero intero che corrisponde all'indice del nuovo profilo nell'elenco dei profili. Restituisce il valore -1 se non è possibile duplicare un nuovo profilo.

Descrizione

Metodo; duplica il profilo attivo e rende attiva la versione duplicata.

Esempio

L'esempio seguente duplica il profilo attivo e visualizza l'indice del nuovo profilo nel pannello Output.

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

document.duplicateScene()

Disponibilità
Flash MX 2004.

Uso
`document.duplicateScene()`

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se la duplicazione della scena ha esito positivo; `false` in caso contrario.

Descrizione
Metodo; crea una copia della scena selezionata, assegnando alla nuova scena un nome univoco e rendendola la scena corrente.

Esempio
L'esempio seguente duplica la seconda scena nel documento corrente:

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

document.duplicateSelection()

Disponibilità
Flash MX 2004.

Uso
`document.duplicateSelection()`

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; duplica la selezione sullo stage.

Esempio
L'esempio seguente duplica la selezione corrente; l'operazione è simile a fare clic su un elemento e trascinarlo tenendo premuto il tasto Alt:

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

document.editScene()

Disponibilità
Flash MX 2004.

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

Parametri
`index` Un numero intero a base zero che specifica quale scena deve essere modificata.

Restituisce
Nulla.

Descrizione
Metodo; seleziona la scena specificata per renderla modificabile.

Esempio
Supponendo che nel documento corrente siano presenti tre scene (`Scene0`, `Scene1` e `Scene2`), l'esempio seguente rende corrente la scena `Scene2`, quindi la elimina:

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

document.enableAllFilters()

Disponibilità
Flash 8.

Uso
```
document.enableAllFilters()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; attiva per gli oggetti selezionati tutti i filtri presenti nell'elenco dei filtri.

Esempio
L'esempio seguente attiva per gli oggetti selezionati tutti i filtri presenti nell'elenco dei filtri:

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

Consultate anche

`document.addFilter()`, `document.changeFilterOrder()`, `document.disableAllFilters()`, `document.enableFilter()`, `document.getFilters()`, `document.removeAllFilters()`, Oggetto Filter

document.enableFilter()

Disponibilità

Flash 8.

Uso

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

Parametri

`filterIndex` Un numero intero che specifica l'indice a base zero del filtro nell'elenco dei filtri da attivare.

Restituisce

Nulla.

Descrizione

Metodo; attiva il filtro specificato per gli oggetti selezionati.

Esempio

L'esempio seguente attiva il secondo filtro per gli oggetti selezionati:

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

Consultate anche

`document.addFilter()`, `document.changeFilterOrder()`, `document.disableFilter()`, `document.enableAllFilters()`, `document.getFilters()`, `document.removeFilter()`, Oggetto Filter

document.enterEditMode()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

`editMode` Una stringa che specifica la modalità di modifica. I valori accettabili sono `"inPlace"` e `"newWindow"`. Se non si specifica alcun parametro, la modalità di modifica dei simboli viene utilizzata come predefinita. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione
Metodo; attiva per lo strumento di creazione la modalità di modifica specificata dal parametro. Se non si specifica alcun parametro, il metodo attiva la modalità di modifica dei simboli per impostazione predefinita, che equivale a fare clic con il pulsante destro del mouse sul simbolo per aprire il menu di scelta rapida e selezionare Modifica.

Esempio
L'esempio seguente attiva la modalità Modifica in posizione di Flash per il simbolo selezionato:

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

L'esempio seguente attiva la modalità Modifica in nuova finestra di Flash per il simbolo selezionato:

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

Consultate anche
`document.exitEditMode()`

document.exitEditMode()

Disponibilità
Flash MX 2004.

Uso
```
document.exitEditMode()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; esce dalla modalità di modifica dei simboli e rende attivo il livello successivo per la modalità di modifica. Ad esempio, se si sta modificando un simbolo all'interno di un altro simbolo, questo metodo attiva il livello immediatamente superiore rispetto al simbolo, all'interno del simbolo principale.

Esempio
L'esempio seguente esce dalla modalità di modifica dei simboli:

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

Consultate anche
`document.enterEditMode()`

document.exportPNG()

Disponibilità
Flash 8.

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

Parametri
`fileURI` Una stringa, espressa nel formato file:///URI, che specifica il nome del file esportato. Se *fileURI* è una stringa vuota o non specificata, Flash visualizza la finestra di dialogo Esporta filmato.

`bCurrentPNGSettings` Un valore booleano che specifica se usare le impostazioni di pubblicazione PNG attuali (`true`) o visualizzare la finestra di dialogo Esporta PNG (`false`). Questo parametro è opzionale. Il valore predefinito è `false`.

`bCurrentFrame` Un valore booleano che specifica se esportare solo il fotogramma attuale (`true`) o tutti i fotogrammi, ciascuno in un file PNG distinto (`false`). Questo parametro è opzionale. Il valore predefinito è `false`.

Restituisce
Un valore booleano: `true` se il file viene correttamente esportato come file PNG; `false` in caso contrario.

Descrizione
Metodo; esporta il documento in uno o più file PNG. Se è stato specificato *fileURI* e il file esiste già, questo viene sovrascritto senza alcun messaggio di avvertimento.

Esempio
L'esempio seguente esporta il fotogramma attuale nel documento attuale in myFile.png usando le impostazioni di pubblicazione correnti:

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

document.exportPublishProfile()

Disponibilità
Flash MX 2004.

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

Parametri
`fileURI` Una stringa, espressa nel formato file:///URI, che specifica il percorso del file XML da cui viene esportato il profilo.

Restituisce
Nulla.

Descrizione
Metodo; esporta il profilo attivo in un file XML.

Esempio
L'esempio seguente esporta il profilo attivo nel file profile.xml contenuto nella cartella /Documents and Settings/nomeutente/Desktop presente nell'unità C:

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

Consultate anche
`document.exportPublishProfileString()`, `document.importPublishProfile()`

document.exportPublishProfileString()

Disponibilità
Flash CS4 Professional.

Uso

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

Parametri
`profileName` Una stringa che specifica il nome del profilo da esportare in una stringa XML. Questo parametro è opzionale.

Restituisce
Una stringa XML.

Descrizione
Metodo; restituisce una stringa che indica nel formato XML il profilo specificato. Se passate un valore per *profileName*, viene esportato il profilo corrente.

Esempio
L'esempio seguente memorizza una stringa XML che rappresenta il profilo corrente nella variabile `profileXML`, quindi la visualizza nel pannello Output:

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

Consultate anche
`document.exportPublishProfile()`, `document.importPublishProfileString()`

document.exportSWF()

Disponibilità
Flash MX 2004.

Uso

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

Parametri

`fileURI` Una stringa, espressa nel formato file:///URI, che specifica il nome del file esportato. Se *fileURI* è vuoto o non è specificato, Flash visualizza la finestra di dialogo Esporta filmato. Questo parametro è opzionale.

`bCurrentSettings` Un valore booleano che, se impostato su `true`, fa in modo che Flash utilizzi le impostazioni di pubblicazione SWF correnti. In caso contrario, viene visualizzata la finestra di dialogo Esporta Flash Player. Il valore predefinito è `false`. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione

Metodo; esporta il documento nel formato SWF di Flash.

Esempio

L'esempio seguente esporta il documento nella posizione specificata con le impostazioni di pubblicazione correnti:

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

L'esempio seguente visualizza la finestra di dialogo Esporta filmato e la finestra di dialogo Esporta Flash Player, quindi esporta il documento in base alle impostazioni specificate:

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

L'esempio seguente visualizza la finestra di dialogo Esporta filmato, quindi esporta il documento in base alle impostazioni specificate:

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

document.externalLibraryPath

Disponibilità

Flash CS4 Professional.

Uso

```
document.externalLibraryPath
```

Descrizione

Proprietà; una stringa che contiene un elenco di elementi presenti nel percorso della libreria esterna di ActionScript 3.0 per il documento, che indica la posizione dei file SWC usati come librerie di runtime condivise. Gli elementi nella stringa sono separati da punti e virgola. Nello strumento di creazione, gli elementi vengono specificati scegliendo File > Impostazioni di pubblicazione e quindi scegliendo Impostazioni per gli script ActionScript 3.0.

Esempio

L'esempio seguente imposta il percorso della libreria esterna del documento su "." e "../mySWCLibrary":

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

Consultate anche

`document.libraryPath`, `document.sourcePath`, `fl.externalLibraryPath`

document.forceSimple

Disponibilità

Flash MX 2004.

Uso

`document.forceSimple`

Descrizione

Proprietà; un valore booleano che specifica se gli elementi secondari dell'oggetto sono accessibili. È equivalente alla logica inversa dell'impostazione Rendi accessibili gli oggetti secondari del pannello Accessibilità. In altre parole, un valore `true` per `forceSimple` equivale all'opzione Rendi accessibili gli oggetti secondari deselezionata. Un valore `false` per `forceSimple` equivale alla stessa opzione selezionata.

Esempio

L'esempio che segue imposta la variabile `areChildrenAccessible` sul valore della proprietà `forceSimple`. Il valore `false` indica che gli elementi secondari sono accessibili.

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

L'esempio seguente imposta la proprietà `forceSimple` in modo che gli elementi secondari del documento siano accessibili:

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

document.frameRate

Disponibilità

Flash MX 2004.

Uso

`document.frameRate`

Descrizione

Proprietà; un valore float che specifica il numero di fotogrammi visualizzati al secondo durante la riproduzione del file SWF. Il valore predefinito è 12. L'impostazione di questa proprietà equivale a impostare la frequenza dei fotogrammi predefinita nella finestra di dialogo delle proprietà del documento (Elabora > Documento) nel file FLA.

Esempio

L'esempio seguente imposta la frequenza dei fotogrammi su 25,5 fotogrammi al secondo:

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

document.getAlignToDocument()

Disponibilità
Flash MX 2004.

Uso
```
document.getAlignToDocument()
```

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se la preferenza è impostata in modo da allineare gli oggetti allo stage; `false` in caso contrario.

Descrizione
Metodo; equivale a recuperare il valore del pulsante Allo stage nel pannello Allinea. Recupera la preferenza che è possibile usare per i metodi `document.align()`, `document.distribute()`, `document.match()` e `document.space()` nel documento.

Esempio
L'esempio seguente recupera il valore del pulsante Allo stage nel pannello Allinea. Se il valore restituito è `true`, il pulsante Allo stage è attivo; in caso contrario è disattivato.

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

Consultate anche
`document.setAlignToDocument()`

document.getBlendMode()

Disponibilità
Flash 8.

Uso
```
document.getBlendMode()
```

Parametri
Nessuno.

Restituisce
Una stringa che specifica il metodo di fusione desiderato per gli oggetti selezionati. Se sono selezionati più oggetti con metodi di fusione diversi, la stringa riflette il metodo di fusione dell'oggetto con la profondità maggiore.

Nota: *il valore restituito è imprevedibile se la selezione contiene oggetti che non supportano i metodi di fusione o che hanno il metodo di fusione "normal".*

Descrizione
Metodo; restituisce una stringa che specifica il metodo di fusione desiderato per gli oggetti selezionati.

Esempio
L'esempio seguente visualizza il nome del metodo di fusione nel pannello Output:

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

document.getCustomFill()

Disponibilità
Flash MX 2004.

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

Parametri
`objectToFill` Una stringa che specifica la posizione dell'oggetto Fill. I valori validi sono i seguenti:

- `"toolbar"` restituisce l'oggetto Fill del pannello Strumenti e della finestra di ispezione Proprietà.
- `"selection"` restituisce l'oggetto Fill della selezione.

  Se si omette questo parametro, il valore predefinito è `"selection"`. Se non si effettua alcuna selezione, il metodo restituisce il valore `undefined`. Questo parametro è opzionale.

Restituisce
L'Oggetto Fill specificato dal parametro *objectToFill* in caso di esito positivo; `undefined` in caso contrario.

Descrizione
Metodo; recupera l'oggetto Fill della forma selezionata o, se specificato, del pannello Strumenti e della finestra di ispezione Proprietà.

Esempio
L'esempio seguente ottiene l'oggetto Fill della selezione, quindi imposta il bianco come colore della selezione:

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

L'esempio seguente restituisce l'oggetto Fill del pannello Strumenti e della finestra di ispezione Proprietà, quindi sostituisce il campione di colore con un gradiente lineare:

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

Consultate anche

`document.setCustomFill()`

document.getCustomStroke()

Disponibilità

Flash MX 2004.

Uso

`document.getCustomStroke([locationOfStroke])`

Parametri

`locationOfStroke` Una stringa che specifica la posizione dell'oggetto Stroke. I valori validi sono i seguenti:

- `"toolbar"`, se impostato, restituisce l'oggetto Stroke del pannello Strumenti e della finestra di ispezione Proprietà.
- `"selection"`, se impostato, restituisce l'oggetto Stroke della selezione.

  Se si omette questo parametro, il valore predefinito è `"selection"`. Se non si effettua alcuna selezione, il metodo restituisce il valore `undefined`. Questo parametro è opzionale.

Restituisce

L'Oggetto Stroke specificato dal parametro *locationOfStroke* in caso di esito positivo; `undefined` in caso contrario.

Descrizione

Restituisce l'oggetto Stroke della forma selezionata o, se specificato, del pannello Strumenti e della finestra di ispezione Proprietà.

Esempio

L'esempio seguente restituisce le impostazioni correnti del tratto e imposta lo spessore del tratto su 2:

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

L'esempio seguente restituisce le impostazioni correnti del tratto per il pannello Strumenti e per la finestra di ispezione Proprietà e imposta il rosso come colore del tratto:

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

Consultate anche

`document.setCustomStroke()`

document.getDataFromDocument()

Disponibilità
Flash MX 2004.

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

Parametri
`name` Una stringa che specifica il nome dei dati da restituire.

Restituisce
I dati specificati.

Descrizione
Metodo; recupera il valore dei dati specificati. Il tipo restituito dipende dal tipo di dati che sono stati memorizzati.

Esempio
L'esempio seguente aggiunge al documento corrente un valore intero pari a 12 e utilizza questo metodo per visualizzare il valore nel pannello Output:

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

Consultate anche
`document.addDataToDocument()`, `document.documentHasData()`, `document.removeDataFromDocument()`

document.getElementProperty()

Disponibilità
Flash MX 2004.

Uso
```
document.getElementProperty(propertyName)
```

Parametri
`propertyName` Una stringa che specifica il nome della proprietà Element di cui deve essere recuperato il valore.

Restituisce
Il valore della proprietà specificata. Restituisce `null` se la proprietà è uno stato indeterminato, come quando si selezionano più elementi con valori di proprietà diversi. Restituisce `undefined` se la proprietà non è una proprietà valida dell'elemento selezionato.

Descrizione
Metodo; ottiene la proprietà `Element` specificata per la selezione corrente. Per un elenco dei valori accettabili, consultate la tabella di riepilogo delle proprietà per l'Oggetto Element.

Esempio

L'esempio seguente ottiene il valore `name` della proprietà Element per la selezione corrente:

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

Consultate anche

`document.setElementProperty()`

document.getElementTextAttr()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

`attrName` Una stringa che specifica il nome della proprietà `TextAttrs` da restituire. Per un elenco dei nomi delle proprietà e dei valori attesi, consultate la tabella di riepilogo delle proprietà per l'Oggetto TextAttrs.

`startIndex` Un numero intero che specifica l'indice del primo carattere, dove 0 (zero) specifica la prima posizione. Questo parametro è opzionale.

`endIndex` Un numero intero che specifica l'indice dell'ultimo carattere. Questo parametro è opzionale.

Restituisce

Se è selezionato un solo campo di testo, la proprietà viene restituita se all'interno del testo è utilizzato un solo valore. Restituisce `undefined` se nel campo di testo sono utilizzati diversi valori. Se sono selezionati diversi campi di testo e se tutti i valori per l'allineamento del testo sono uguali, il metodo restituisce questo valore. Se sono selezionati diversi campi di testo ma i valori per l'allineamento del testo non sono tutti uguali, il metodo restituisce `undefined`. Se gli argomenti opzionali non vengono passati, queste regole agiscono sull'intervallo di testo selezionato o sull'intero campo di testo se il testo non è attualmente in corso di modifica. Se viene passato solo *startIndex*, viene restituita la proprietà del carattere a destra dell'indice, se i valori di tutti gli oggetti Text selezionati corrispondono. Se vengono passati *startIndex* e *endIndex*, il valore restituito rispecchia l'intero intervallo di caratteri da *startIndex* a *endIndex* escluso.

Descrizione

Metodo; ottiene una proprietà `TextAttrs` specifica degli oggetti Text selezionati. Gli oggetti di testo che non sono campi di testo vengono ignorati. Per un elenco dei nomi delle proprietà e dei valori attesi, consultate la tabella di riepilogo delle proprietà per l'Oggetto TextAttrs. Consultate anche `document.setElementTextAttr()`.

Esempio

L'esempio seguente ottiene le dimensioni dei campi di testo selezionati:

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

L'esempio seguente ottiene il colore del carattere in corrispondenza dell'indice 3 nei campi di testo selezionati:

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

L'esempio seguente ottiene il nome del carattere del testo compreso tra l'indice 2 e l'indice 10 escluso dei campi di testo selezionati:

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

document.getFilters()

Disponibilità
Flash 8.

Uso
```
document.getFilters()
```

Parametri
Nessuno.

Restituisce
Un array che contiene l'elenco dei filtri applicati agli oggetti selezionati.

Descrizione
Metodo; restituisce un array che contiene l'elenco dei filtri applicati agli oggetti selezionati. Se sono selezionati più oggetti che non dispongono di filtri identici, questo metodo restituisce l'elenco dei filtri applicati al primo oggetto selezionato.

Esempio
Consultate `document.setFilters()`.

Consultate anche
`document.addFilter()`, `document.changeFilterOrder()`, `document.setFilters()`, Oggetto Filter

document.getMetadata()

Disponibilità
Flash 8.

Uso
```
document.getMetadata()
```

Parametri
Nessuno.

Restituisce
Una stringa che contiene i metadati XML associati al documento o una stringa vuota se non sono presenti metadati.

Descrizione
Metodo; restituisce una stringa che contiene i metadati XML associati al documento o una stringa vuota se non sono presenti metadati.

Esempio
L'esempio seguente visualizza i metadati XML del documento corrente nel pannello Output:

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

Consultate anche
document.setMetadata()

document.getMobileSettings()

Disponibilità
Flash CS3 Professional.

Uso
```
document.getMobileSettings()
```

Parametri
Nessuno.

Restituisce
Una stringa che rappresenta le impostazioni XML del documento. Se non è stato impostato alcun valore, restituisce una stringa vuota.

Descrizione
Metodo; restituisce le impostazioni XML mobili per il documento.

Esempio
L'esempio seguente visualizza la stringa di impostazioni XML per il documento corrente:

```
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>"
```

Consultate anche
document.setMobileSettings()

document.getPlayerVersion()

Disponibilità
Flash CS3 Professional.

Uso
`document.getPlayerVersion()`

Parametri
Nessuno.

Restituisce
Una stringa che rappresenta la versione di Flash Player specificata tramite `document.setPlayerVersion()`. Se non è impostato alcun valore, restituisce il valore specificato nella finestra di dialogo Impostazioni pubblicazione.

Descrizione
Metodo; restituisce una stringa che rappresenta la versione del lettore prevista per il documento specificato. Per un elenco di valori che possono essere restituiti da questo metodo, consultate `document.setPlayerVersion()`.

Per determinare la versione di ActionScript prevista per il file specificato, utilizzate `document.asVersion`.

Esempio
L'esempio che segue illustra come selezionare versioni specifiche del lettore per il documento corrente e come richiamare i rispettivi valori:

```
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""
```

Consultate anche
`document.setPlayerVersion()`

document.getSelectionRect()

Disponibilità
Flash MX 2004.

Uso
`document.getSelectionRect()`

Parametri
Nessuno.

Restituisce
Il rettangolo di delimitazione della selezione corrente, oppure 0 se non è stata effettuata alcuna selezione. Per informazioni sul formato del valore restituito, consultate `document.addNewRectangle()`.

Descrizione
Metodo; ottiene il rettangolo di delimitazione della selezione corrente. Se una selezione non è rettangolare, viene restituito il rettangolo minimo in grado di contenerla. Il rettangolo si basa sullo spazio del documento o, in modalità di modifica, sul punto di registrazione (detto anche *punto di origine* o *punto zero*) del simbolo che si sta modificando.

Esempio
L'esempio seguente ottiene il rettangolo di delimitazione per la selezione corrente, quindi ne seleziona le proprietà:

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

Consultate anche
`document.selection`, `document.setSelectionRect()`

document.getTextString()

Disponibilità
Flash MX 2004.

Uso

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

Parametri
`startIndex` Un numero intero che corrisponde a un indice del primo carattere da ottenere. Questo parametro è opzionale.

`endIndex` Un numero intero che corrisponde a un indice dell'ultimo carattere da ottenere. Questo parametro è opzionale.

Restituisce
Una stringa che contiene il testo selezionato.

Descrizione
Metodo; ottiene il testo selezionato. Se i parametri opzionali non vengono passati, viene utilizzata la selezione di testo corrente. Se al momento non è aperto alcun testo per la modifica, viene restituito l'intero testo. Se viene passato solo *startIndex*, viene restituita la stringa che inizia in corrispondenza di tale indice e che termina alla fine del campo. Se vengono passati *startIndex* e *endIndex*, viene restituita la stringa che inizia da *startIndex* e termina a *endIndex* escluso.

Se sono selezionati diversi campi di testo, viene restituita la concatenazione di tutte le stringhe.

Esempio
L'esempio seguente ottiene la stringa nei campi di testo selezionati:

```
fl.getDocumentDOM().getTextString();
```

L'esempio seguente ottiene la stringa in corrispondenza dell'indice di carattere 5 nei campi di testo selezionati:

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

L'esempio seguente ottiene la stringa compresa tra l'indice di carattere 2 e l'indice di carattere 10 escluso:

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

Consultate anche

`document.setTextString()`

document.getTimeline()

Disponibilità

Flash MX 2004.

Uso

```
document.getTimeline()
```

Parametri

Nessuno.

Restituisce

L'oggetto Timeline corrente.

Descrizione

Metodo; recupera l'Oggetto Timeline corrente nel documento. La linea temporale corrente può essere la scena corrente, il simbolo in fase di modifica o la schermata corrente.

Esempio

L'esempio seguente ottiene l'oggetto Timeline e restituisce il numero di fotogrammi presenti nel livello più lungo:

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

L'esempio seguente attiva la modalità Modifica in posizione per il simbolo selezionato sullo stage e inserisce un fotogramma sulla relativa linea temporale.

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

L'esempio seguente ottiene l'oggetto Timeline e ne visualizza il nome:

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

Consultate anche

`document.currentTimeline`, `document.timelines`, `symbolItem.timeline`

document.getTransformationPoint()

Disponibilità
Flash MX 2004.

Uso
`document.getTransformationPoint()`

Parametri
Nessuno.

Restituisce
Un punto (ad esempio `{x:10,y:20}`, dove `x` e `y` sono numeri a virgola mobile) che specifica la posizione del punto di trasformazione (detto anche *punto di origine* o *punto zero*) all'interno del sistema di coordinate dell'elemento selezionato.

Descrizione
Metodo; ottiene la posizione del punto di trasformazione della selezione corrente. Questo punto può essere utilizzato per trasformazioni come rotazioni e inclinazioni.

Nota: *i punti di trasformazione sono relativi a posizioni differenti, a seconda del tipo di elemento selezionato. Per ulteriori informazioni, consultate* `document.setTransformationPoint()`.

Esempio
L'esempio seguente ottiene il punto di trasformazione per la selezione corrente. La proprietà `transPoint.x` fornisce la coordinata *x* del punto di trasformazione, mentre la proprietà `transPoint.y` fornisce la coordinata *y*.

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

Consultate anche
`document.setTransformationPoint()`, `element.getTransformationPoint()`

document.group()

Disponibilità
Flash MX 2004.

Uso
`document.group()`

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; converte in gruppo la selezione corrente.

Esempio
L'esempio seguente converte in gruppo gli oggetti presenti nella selezione corrente:

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

Consultate anche
`document.unGroup()`

document.height

Disponibilità
Flash MX 2004.

Uso
`document.height`

Descrizione
Proprietà; un numero intero che specifica l'altezza del documento (stage) espressa in pixel.

Esempio
L'esempio seguente imposta l'altezza dello stage su 400 pixel:

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

Consultate anche
`document.width`

document.id

Disponibilità
Flash CS3 Professional.

Uso
`document.id`

Descrizione
Proprietà di sola lettura; un numero intero univoco (assegnato automaticamente) che identifica un documento durante una sessione Flash. Questa proprietà può essere impiegata insieme a `fl.findDocumentDOM()` per specificare il documento oggetto di una particolare azione.

Esempio
L'esempio seguente visualizza l'ID del documento corrente:

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

Consultate anche

`fl.findDocumentDOM()`

document.importFile()

Disponibilità

Flash 8.

Uso

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

Parametri

`fileURI` Una stringa, espressa nel formato file:///URI, che specifica il percorso del file da importare.

`importToLibrary` Valore booleano che specifica se importare il file solo nella libreria del documento (`true`) o archiviarne una copia anche nello stage (`false`). Il valore predefinito è `false`.

Restituisce

Nulla.

Descrizione

Metodo; importa un file nel documento. Questo metodo esegue la stessa operazione delle opzioni di menu Importa nella libreria e Importa nello stage. Per importare un profilo di pubblicazione, usate `document.importPublishProfile()`.

Esempio

L'esempio seguente consente all'utente di cercare un file da importare nello stage:

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

Consultate anche

`document.importSWF()`, `fl.browseForFileURL()`

document.importPublishProfile()

Disponibilità

Flash MX 2004.

Uso

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

Parametri

`fileURI` Una stringa, espressa nel formato file:///URI, che specifica il percorso del file XML per il profilo da importare.

Restituisce

Un numero intero che corrisponde all'indice del profilo importato nell'elenco dei profili. Restituisce il valore -1 se non è possibile importare il profilo.

Descrizione

Metodo; importa un profilo da un file.

Esempio

L'esempio seguente importa il profilo contenuto nel file profile.xml e ne visualizza l'indice nell'elenco dei profili:

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

document.importPublishProfileString()

Disponibilità

Flash CS4 Professional.

Uso

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

Parametri

`xmlString` Una stringa che contiene i dati XML da importare come profilo corrente.

Restituisce

Un valore booleano: `true` se la stringa viene importata con successo; `false` in caso contrario.

Descrizione

Metodo; importa una stringa XML che rappresenta un profilo di pubblicazione e lo imposta come profilo corrente. Per generare una stringa XML da importare, usate `document.exportPublishProfileString()` prima di questo metodo.

Esempio

Nell'esempio seguente il profilo predefinito viene esportato come stringa XML. Per modificare la stringa XML, viene usato il comando standard `replace` di JavaScript. La stringa viene quindi importata e il valore predefinito per l'output di ActionScript 3 è impostato su ActionScript 1.

```
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()

Disponibilità
Flash MX 2004.

Uso
`document.importSWF(fileURI)`

Parametri
`fileURI` Una stringa, espressa nel formato file:///URI, che specifica il file che deve essere importato dal file SWF.

Restituisce
Nulla.

Descrizione
Metodo; importa un file SWF nel documento. Equivale a utilizzare l'opzione di menu Importa per specificare un file SWF. In Flash8 e versioni successive, è anche possibile usare `document.importFile()` per importare un file SWF (e qualsiasi altro tipo di file).

Esempio
L'esempio seguente importa il file `"mySwf.swf"` dalla cartella Configuration di Flash:

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

Consultate anche
`document.importFile()`

document.intersect()

Disponibilità
Flash 8.

Uso
`document.intersect()`

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se l'esito è positivo; `false` in caso contrario.

Descrizione
Metodo; crea un oggetto di disegno di intersezione utilizzando tutti gli oggetti di disegno selezionati. Il metodo restituisce `false` se non sono selezionati oggetti di disegno o se uno degli oggetti selezionati non è un oggetto di disegno.

Esempio
L'esempio seguente crea un oggetto di disegno di intersezione utilizzando tutti gli oggetti di disegno selezionati:

```
fl.getDocumentDOM().intersect();
```

Consultate anche
`document.crop()`, `document.deleteEnvelope()`, `document.punch()`, `document.union()`, `shape.isDrawingObject`

document.library

Disponibilità
Flash MX 2004.

Uso
`document.library`

Descrizione
Proprietà di sola lettura; l'Oggetto library di un documento.

Esempio
L'esempio seguente ottiene la libreria del documento attivo.

```
var myCurrentLib = fl.getDocumentDOM().library;
```

L'esempio seguente presuppone che il documento attivo non sia `fl.documents[1]` e ottiene la libreria per una libreria non attiva o per una libreria aperta come libreria esterna mediante File > Apri:

```
var externalLib = fl.documents[1].library;
```

document.libraryPath

Disponibilità
Flash CS4 Professional.

Uso
`document.libraryPath`

Descrizione
Proprietà; una stringa che contiene un elenco di elementi presenti nel percorso della libreria esterna di ActionScript 3.0 per il documento, che indica la posizione dei file o delle cartelle SWC contenenti i file SWC. Gli elementi nella stringa sono separati da punti e virgola. Nello strumento di creazione, gli elementi vengono specificati scegliendo File > Impostazioni di pubblicazione e quindi scegliendo Impostazioni per gli script ActionScript 3.0.

Esempio
Il codice seguente aggiunge la cartella ../Files al percorso della libreria del documento e quindi visualizza il percorso della libreria nel pannello Output:

```
var myDoc = fl.getDocumentDOM()
fl.trace(myDoc.libraryPath);
myDoc.libraryPath = "../Files;" + myDoc.libraryPath;
fl.trace(myDoc.libraryPath);
```

Consultate anche

`document.externalLibraryPath`,`document.sourcePath`, `fl.libraryPath`

document.livePreview

Disponibilità

Flash MX 2004.

Uso

```
document.livePreview
```

Descrizione

Proprietà; un valore booleano che specifica se l'anteprima dal vivo è attiva. Se è impostata su `true`, i componenti vengono visualizzati sullo stage così come compaiono nel contenuto Flash pubblicato, comprese le loro dimensioni approssimative. Se è impostata su `false`, i componenti vengono visualizzati solo come contorni. Il valore predefinito è `true`.

Esempio

L'esempio seguente imposta l'anteprima dal vivo su `false`:

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

document.match()

Disponibilità

Flash MX 2004.

Uso

```
document.match(bWidth, bHeight [, bUseDocumentBounds])
```

Parametri

`bWidth` Un valore booleano che, se impostato su `true`, fa in modo che il metodo renda uguale la larghezza degli elementi selezionati.

`bHeight` Un valore booleano che, se impostato su `true`, fa in modo che il metodo renda uguale l'altezza degli elementi selezionati.

`bUseDocumentBounds` Un valore booleano che, se impostato su `true`, fa in modo che il metodo faccia combaciare le dimensioni degli oggetti con i contorni del documento. In caso contrario, il metodo utilizza i contorni dell'oggetto più grande. Il valore predefinito è `false`. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; rende uguali le dimensioni degli oggetti selezionati.

Esempio
L'esempio seguente fa corrispondere la larghezza solo degli oggetti selezionati:

```
fl.getDocumentDOM().match(true,false);
```

L'esempio seguente fa corrispondere solo l'altezza:

```
fl.getDocumentDOM().match(false,true);
```

L'esempio seguente fa corrispondere la larghezza solo ai contorni del documento:

```
fl.getDocumentDOM().match(true,false,true);
```

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

document.mouseClick()

Disponibilità
Flash MX 2004.

Uso

```
document.mouseClick(position, bToggleSel, bShiftSel)
```

Parametri
`position` Una coppia di numeri a virgola mobile che specificano le coordinate *x* e *y* del clic, espresse in pixel.

`bToggleSel` Un valore booleano che indica lo stato del tasto Maiusc: `true`) se è premuto e `false` se non è premuto.

`bShiftSel` Un valore booleano che indica lo stato della preferenza Seleziona con Maiusc dell'applicazione:`true`) se è attivata e `false` se è disattivata.

Restituisce
Nulla.

Descrizione
Metodo; esegue un clic del mouse con lo strumento Selezione.

Esempio
L'esempio seguente esegue un clic del mouse nel punto specificato:

```
fl.getDocumentDOM().mouseClick({x:300, y:200}, false, false);
```

Consultate anche
`document.mouseDblClk()`

document.mouseDblClk()

Disponibilità
Flash MX 2004.

Uso
```
document.mouseDblClk(position, bAltDown, bShiftDown, bShiftSelect)
```

Parametri
`position` Una coppia di numeri a virgola mobile che specificano le coordinate *x* e *y* del clic, espresse in pixel.

`bAltdown` Un valore booleano che registra se il tasto Alt è premuto nel momento dell'evento:`true`) se è premuto e `false` se non è premuto.

`bShiftDown` Un valore booleano che registra se il tasto Maiusc è premuto nel momento dell'evento:`true`) se è premuto e `false` se non è premuto.

`bShiftSelect` Un valore booleano che indica lo stato della preferenza Seleziona con Maiusc dell'applicazione:`true`) se è attivata e `false` se è disattivata.

Restituisce
Nulla.

Descrizione
Metodo; esegue un doppio clic del mouse con lo strumento Selezione.

Esempio
L'esempio seguente esegue un doppio clic del mouse nel punto specificato:

```
fl.getDocumentDOM().mouseDblClk({x:392.9, y:73}, false, false, true);
```

Consultate anche
`document.mouseClick()`

document.moveSelectedBezierPointsBy()

Disponibilità
Flash MX 2004.

Uso
```
document.moveSelectedBezierPointsBy(delta)
```

Parametri
`delta` Una coppia di numeri a virgola mobile che specificano le coordinate *x* e *y* espresse in pixel in base alle quali vengono spostati i punti Bezier selezionati. Ad esempio, se si passa (`{x:1,y:2}`) viene specificato un punto spostato di un pixel verso destra e di due pixel verso il basso rispetto alla posizione corrente.

Restituisce
Nulla.

Descrizione
Metodo; se la selezione contiene almeno un percorso con almeno un punto Bézier selezionato, sposta tutti i punti Bézier selezionati su tutti i percorsi selezionati in base al valore specificato.

Esempio
L'esempio seguente sposta i punti Bézier selezionati di 10 pixel verso destra e 5 pixel verso il basso:

```
fl.getDocumentDOM().moveSelectedBezierPointsBy({x:10, y:5});
```

document.moveSelectionBy()

Disponibilità
Flash MX 2004.

Uso
```
document.moveSelectionBy(distanceToMove)
```

Parametri
`distanceToMove` Una coppia di numeri a virgola mobile che specificano le coordinate *x* e *y* in base alle quali il metodo sposta la selezione. Ad esempio, se si passa (`{x:1,y:2}`) viene specificato un punto spostato di un pixel verso destra e di due pixel verso il basso rispetto alla posizione corrente.

Restituisce
Nulla.

Descrizione
Metodo; sposta gli oggetti selezionati in base alla distanza specificata.

Nota: *quando l'utente sposta l'elemento mediante i tasti freccia, il pannello Cronologia combina tutte le pressioni di un tasto freccia in un unico movimento. Quando l'utente preme ripetutamente i tasti freccia anziché effettuare singoli spostamenti nel pannello Cronologia, il metodo esegue un solo spostamento e gli argomenti vengono aggiornati per rispecchiare la pressione ripetuta dei tasti freccia.*

Per informazioni su come eseguire una selezione, consultate `document.setSelectionRect()`, `document.mouseClick()`, `document.mouseDblClk()` e Oggetto Element.

Esempio
L'esempio seguente sposta l'elemento selezionato di 62 pixel verso destra e di 84 pixel verso il basso:

```
fl.getDocumentDOM().moveSelectionBy({x:62, y:84});
```

document.name

Disponibilità
Flash MX 2004.

Uso
`document.name`

Descrizione
Proprietà di sola lettura; una stringa che rappresenta il nome di un documento (file FLA).

Esempio
L'esempio seguente imposta la variabile `fileName` sul nome di file del primo documento dell'array di documenti:

```
var fileName = flash.documents[0].name;
```

L'esempio seguente visualizza i nomi di tutti i documenti aperti nel pannello Output:

```
var openDocs = fl.documents;
for(var i=0;i < openDocs.length; i++){
    fl.trace(i + " " + openDocs[i].name +"\n");
}
```

document.optimizeCurves()

Disponibilità
Flash MX 2004.

Uso
`document.optimizeCurves(smoothing, bUseMultiplePasses)`

Parametri
`smoothing` Un numero intero compreso tra 0 e 100, dove 0 rappresenta l'assenza di smussatura e 100 la smussatura massima.

`bUseMultiplePasses` Un valore booleano che, se impostato su `true`, indica che il metodo deve utilizzare più passaggi. Questa impostazione è più lenta ma produce un risultato migliore. Equivale a fare clic sul pulsante Usa più passaggi nella finestra di dialogo Ottimizza curve.

Restituisce
Nulla.

Descrizione
Metodo; ottimizza la smussatura della selezione corrente, consentendo più passaggi (se specificato) per ottenere un risultato ottimale. Equivale a selezionare Elabora > Forma > Ottimizza.

Esempio
L'esempio seguente ottimizza la curva della selezione corrente a 50º di smussatura con più passaggi:

```
fl.getDocumentDOM().optimizeCurves(50, true);
```

document.path

Disponibilità
Flash MX 2004.

Uso
`document.path`

Descrizione
Proprietà di sola lettura; una stringa che rappresenta il percorso del documento nel formato specifico di una piattaforma. Se il documento non è mai stato salvato, la proprietà è `undefined`.

Esempio
L'esempio seguente visualizza il percorso del primo documento nell'array di documenti nel pannello Output. Dovete salvare il documento prima di eseguire questo script. Nell'esempio, il file è chiamato test.fla e viene salvato nella cartella Documenti di Windows.

```
var filePath = flash.documents[0].path;
fl.trace(filePath);
// displays C:\Documents and Settings\<user name>\My Documents\test.fla
```

Consultate anche
`document.pathURI`

document.pathURI

Disponibilità
Flash CS4 Professional.

Uso
`document.pathURI`

Descrizione
Proprietà di sola lettura; una stringa, espressa nel formato file:///URI, che rappresenta il percorso del documento. Se il documento non è mai stato salvato, la proprietà è `undefined`.

Esempio
L'esempio seguente visualizza il percorso del primo documento nell'array di documenti espresso come stringa file:///URI nel pannello Output. Dovete salvare il documento prima di eseguire questo script. Nell'esempio, il file è chiamato test.fla e viene salvato nella cartella Documenti di Windows.

```
var filePathURI = flash.documents[0].pathURI;
fl.trace(filePathURI);
// displays file:///C|/Documents%20and%20Settings/<userName>/My%20Documents/test.fla
```

Consultate anche
document.path

document.publish()

Disponibilità
Flash MX 2004.

Uso
```
document.publish()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; pubblica il documento in base alle impostazioni di pubblicazione correnti (File > Impostazioni pubblicazione). Equivale a selezionare File > Pubblica.

Esempio
L'esempio seguente pubblica il documento corrente:

```
fl.getDocumentDOM().publish();
```

document.publishProfiles

Disponibilità
Flash MX 2004.

Uso
```
document.publishProfiles
```

Descrizione
Proprietà di sola lettura; un array dei nomi dei profili di pubblicazione per il documento.

Esempio
L'esempio seguente visualizza i nomi dei profili di pubblicazione per il documento:

```
var myPubProfiles = fl.getDocumentDOM().publishProfiles;
for (var i=0; i < myPubProfiles.length; i++){
     fl.trace(myPubProfiles[i]);
}
```

document.punch()

Disponibilità
Flash 8.

Uso
```
document.punch()
```

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se l'esito è positivo; `false` in caso contrario.

Descrizione
Metodo; sovrappone l'oggetto di disegno di primo livello selezionato a tutti gli oggetti di disegno sottostanti. Il metodo restituisce `false` se non sono selezionati oggetti di disegno o se uno degli oggetti selezionati non è un oggetto di disegno.

Esempio
L'esempio seguente effettua una sovrapposizione sugli oggetti di disegno che si trovano sotto l'oggetto di disegno selezionato:

```
fl.getDocumentDOM().punch();
```

Consultate anche
`document.crop()`, `document.deleteEnvelope()`, `document.intersect()`, `document.union()`, `shape.isDrawingObject`

document.removeAllFilters()

Disponibilità
Flash 8.

Uso
```
document.removeAllFilters()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; rimuove tutti i filtri dagli oggetti selezionati.

Esempio
L'esempio seguente rimuove tutti i filtri dagli oggetti selezionati:

```
fl.getDocumentDOM().removeAllFilters();
```

Consultate anche
`document.addFilter()`, `document.changeFilterOrder()`, `document.disableAllFilters()`, `document.getFilters()`, `document.removeFilter()`, Oggetto Filter

document.removeDataFromDocument()

Disponibilità
Flash MX 2004.

Uso
`document.removeDataFromDocument(name)`

Parametri
`name` Una stringa che specifica il nome dei dati da rimuovere.

Restituisce
Nulla.

Descrizione
Metodo; rimuove i dati persistenti con il nome specificato che sono stati associati al documento.

Esempio
L'esempio seguente rimuove dal documento i dati persistenti denominati `"myData"`:

```
fl.getDocumentDOM().removeDataFromDocument("myData");
```

Consultate anche
`document.addDataToDocument()`, `document.documentHasData()`, `document.getDataFromDocument()`

document.removeDataFromSelection()

Disponibilità
Flash MX 2004.

Uso
`document.removeDataFromSelection(name)`

Parametri
`name` Una stringa che specifica il nome dei dati persistenti da rimuovere.

Restituisce
Nulla.

Descrizione
Metodo; rimuove i dati persistenti con il nome specificato che sono stati associati alla selezione.

Esempio
L'esempio seguente rimuove dalla selezione i dati persistenti denominati `"myData"`:

```
fl.getDocumentDOM().removeDataFromSelection("myData");
```

Consultate anche
`document.addDataToSelection()`

document.removeFilter()

Disponibilità
Flash 8.

Uso
`document.removeFilter(filterIndex)`

Parametri
`filterIndex` Un numero intero che specifica l'indice a base zero del filtro da rimuovere dagli oggetti selezionati.

Restituisce
Nulla.

Descrizione
Metodo; rimuove il filtro specificato dall'elenco dei filtri per gli oggetti selezionati.

Esempio
L'esempio seguente rimuove il primo filtro (valore di indice 0) nell'elenco dei filtri per gli oggetti selezionati:

```
fl.getDocumentDOM().removeFilter(0);
```

Consultate anche
`document.addFilter()`, `document.changeFilterOrder()`, `document.disableFilter()`, `document.getFilters()`, `document.removeAllFilters()`, Oggetto Filter

document.renamePublishProfile()

Disponibilità
Flash MX 2004.

Uso
```
document.renamePublishProfile([profileNewName])
```

Parametri
`profileNewName` Un parametro opzionale che specifica il nuovo nome del profilo. Il nome deve essere univoco. Se non si specifica un nome, viene utilizzato un nome predefinito.

Restituisce
Un valore booleano: `true` se la modifica del nome ha esito positivo; `false` in caso contrario.

Descrizione
Metodo; rinomina il profilo corrente.

Esempio
L'esempio seguente rinomina il profilo corrente utilizzando un nome predefinito e lo visualizza:

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

document.renameScene()

Disponibilità
Flash MX 2004.

Uso
```
document.renameScene(name)
```

Parametri
`name` Una stringa che specifica il nuovo nome della scena.

Restituisce
Un valore booleano: `true` se la modifica del nome ha esito positivo; `false` in caso contrario. Se ad esempio il nuovo nome non è univoco, il metodo restituisce il valore `false`.

Descrizione
Metodo; rinomina la scena selezionata nel pannello Scena. Il nuovo nome della scena selezionata deve essere univoco.

Esempio
L'esempio seguente rinomina la scena corrente utilizzando il nome `"new name"`:

```
var success = fl.getDocumentDOM().renameScene("new name");
```

document.reorderScene()

Disponibilità
Flash MX 2004.

Uso

```
document.reorderScene(sceneToMove, sceneToPutItBefore)
```

Parametri

`sceneToMove` Un numero intero che specifica la scena da spostare, dove 0 (zero) specifica la prima scena.

`sceneToPutItBefore` Un numero intero che specifica la scena prima della quale si desidera spostare la scena specificata da *sceneToMove*. Specificate 0 (zero) per la prima scena. Ad esempio, se si specifica 1 per *sceneToMove* e 0 per *sceneToPutItBefore*, la seconda scena viene posizionata prima della prima scena. Specificate -1 per spostare la scena alla fine.

Restituisce

Nulla.

Descrizione

Metodo; sposta la scena specificata prima di un'altra scena specificata.

Esempio

L'esempio seguente sposta la seconda scena prima della prima scena:

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

document.resetOvalObject()

Disponibilità

Flash CS3 Professional.

Uso

```
document.resetOvalObject()
```

Parametri

Nessuno.

Restituisce

Nulla.

Descrizione

Metodo; imposta tutti i valori nella finestra di ispezione Proprietà in base ai valori predefiniti nelle impostazioni dell'oggetto Oval. Se sono selezionati oggetti Oval, le relative proprietà vengono riportate ai valori predefiniti.

Esempio

L'esempio che segue ripristina i valori predefiniti delle proprietà degli oggetti Oval nel documento corrente:

```
fl.getDocumentDOM().resetOvalObject();
```

Consultate anche

`document.resetRectangleObject()`

document.resetRectangleObject()

Disponibilità
Flash CS3 Professional.

Uso
```
document.resetRectangleObject()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; imposta tutti i valori nella finestra di ispezione Proprietà in base ai valori predefiniti nelle impostazioni dell'oggetto Rectangle. Se sono selezionati oggetti Rectangle, le relative proprietà vengono riportate ai valori predefiniti.

Esempio
L'esempio che segue ripristina i valori predefiniti delle proprietà degli oggetti Rectangle nel documento corrente:

```
fl.getDocumentDOM().resetRectangleObject();
```

Consultate anche
```
document.resetOvalObject()
```

document.resetTransformation()

Disponibilità
Flash MX 2004.

Uso
```
document.resetTransformation()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; reimposta la matrice di trasformazione. Equivale a selezionare Elabora > Trasforma > Elimina trasformazione.

Esempio
L'esempio seguente reimposta la matrice di trasformazione per la selezione corrente.

```
fl.getDocumentDOM().resetTransformation();
```

document.revert()

Disponibilità
Flash MX 2004.

Uso
```
document.revert()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; ripristina la versione salvata in precedenza del documento specificato. Equivale a selezionare File > Ripristina.

Esempio
L'esempio seguente ripristina la versione salvata in precedenza del documento corrente.

```
fl.getDocumentDOM().revert();
```

Consultate anche
`document.canRevert()`, `fl.revertDocument()`

document.revertToLastVersion()

Disponibilità
Flash CS3 Professional.

Uso
```
document.revertToLastVersion()
```

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se il documento viene ripristinato correttamente; `false` in caso contrario.

Descrizione
Metodo; se il file può essere ripristinato, visualizza una finestra di dialogo che permette all'utente di confermare il ripristino del file. Se l'utente conferma l'operazione, il metodo ripristina il file alla versione memorizzata sul server Version Cue e visualizza eventuali errori nel pannello Output.

Esempio
L'esempio che segue ripristina il documento corrente all'ultima versione memorizzata sul server Version Cue:

```
fl.getDocumentDOM().revertToLastVersion();
```

Consultate anche
`document.canSaveAVersion()`, `document.saveAVersion()`, `document.synchronizeWithHeadVersion()`, `fl.revertDocumentToLastVersion()`

document.rotate3DSelection()

Disponibilità
Flash CS4 Professional.

Uso
```
document.rotate3DSelection(xyzCoordinate, bGlobalTransform)
```

Parametri
`xyzCoordinate` Un punto in coordinate XYZ che specifica gli assi di rotazione 3D.

`bGlobalTransform` Un valore booleano che specifica se la modalità di trasformazione deve essere globale (`true`) o locale (`false`).

Restituisce
Nulla.

Descrizione
Metodo; applica una rotazione 3D alla selezione. Questo metodo è disponibile solo per i clip video.

Esempio
Nell'esempio seguente, la selezione viene prima ruotata rispetto allo stage (globalmente) e quindi rispetto a se stessa (localmente).

```
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()

Disponibilità
Flash MX 2004.

Uso

```
document.rotateSelection(angle [, rotationPoint])
```

Parametri

`angle` Un valore a virgola mobile che specifica l'angolo della rotazione.

`rotationPoint` Una stringa che specifica quale lato del riquadro di delimitazione ruotare. I valori accettabili sono `"top right"`, `"top left"`, `"bottom right"`, `"bottom left"`, `"top center"`, `"right center"`, `"bottom center"` e `"left center"`. Se non viene specificato, il metodo utilizza il punto di trasformazione. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione

Metodo; ruota la selezione in base al numero di gradi specificato. Equivale a utilizzare lo strumento Trasformazione libera per ruotare l'oggetto.

Esempio

L'esempio seguente ruota la selezione di 45º rispetto al punto di trasformazione:

```
fl.getDocumentDOM().rotateSelection(45);
```

L'esempio seguente ruota la selezione di 45º rispetto all'angolo inferiore sinistro:

```
fl.getDocumentDOM().rotateSelection(45, "bottom left");
```

document.save()

Disponibilità

Flash MX 2004.

Uso

```
document.save([bOkToSaveAs])
```

Parametri

`bOkToSaveAs` Un parametro opzionale che, se impostato su `true` oppure omesso, e nel caso il file non sia mai stato salvato, apre la finestra di dialogo Salva con nome. Se è impostato su `false` e il file non è mai stato salvato, il file non viene salvato.

Restituisce

Un valore booleano: `true` se l'operazione di salvataggio ha esito positivo; `false` in caso contrario.

Descrizione

Metodo; salva il documento nella posizione predefinita. Equivale a selezionare File > Salva.

Per specificare un diverso nome per il file (invece di utilizzare quello attuale), usate `fl.saveDocument()`.

Nota: *se il file è nuovo e non è stato modificato o salvato, oppure se non è stato modificato dopo l'ultimo salvataggio eseguito, il metodo non ha alcun effetto e viene restituito il valore* `false`*. Per rendere possibile il salvataggio di un file che non è mai stato salvato o che non è stato modificato, usate* `document.saveAndCompact()` *o* `fl.saveDocumentAs()`*.*

Esempio

L'esempio seguente salva il documento corrente nella posizione predefinita:

```
fl.getDocumentDOM().save();
```

Consultate anche

`document.saveAndCompact()`, `fl.saveAll()`, `fl.saveDocument()`, `fl.saveDocumentAs()`

document.saveAndCompact()

Disponibilità

Flash MX 2004.

Uso

```
document.saveAndCompact([bOkToSaveAs])
```

Parametri

`bOkToSaveAs` Un parametro opzionale che, se impostato su `true` oppure omesso, e nel caso il file non sia mai stato salvato, apre la finestra di dialogo Salva con nome. Se è impostato su `false` e il file non è mai stato salvato, il file non viene salvato. Il valore predefinito è `true`.

Restituisce

Un valore booleano: `true` se l'operazione di salvataggio e compressione ha esito positivo; `false` in caso contrario.

Descrizione

Metodo; salva e comprime il file. Equivale a selezionare File > Salva e comprimi.

Nota: *se il file non è mai stato salvato, questo metodo restituisce il valore* `true` *anche se l'utente esce dalla finestra di dialogo Salva con nome mediante il pulsante Annulla. Per determinare esattamente se il file è stato salvato, usate* `fl.saveDocumentAs()`*.*

Esempio

L'esempio seguente salva e comprime il documento corrente:

```
fl.getDocumentDOM().saveAndCompact();
```

Consultate anche

`document.save()`, `fl.saveDocumentAs()`, `fl.saveDocument()`, `fl.saveAll()`

document.saveAVersion()

Disponibilità
Flash CS3 Professional.

Uso
```
document.saveAVersion()
```

Parametri
Nessuno.

Restituisce
Il valore booleano `true` se la versione del documento viene salvata correttamente sul server Version Cue; `false` in caso contrario.

Descrizione
Metodo; se il file può essere salvato sul server Version Cue, viene visualizzata una finestra di dialogo che permette all'utente di immettere i propri commenti sulla versione, salva una versione del documento specificato sul server e infine mostra eventuali errori nel pannello Output.

Nota: *qualora Flash non sia in grado di salvare il file perché le credenziali del server non sono state inserite nella cache della sessione dell'applicazione corrente, nel pannello Output viene visualizzato un errore di autenticazione. In questo caso l'utente deve utilizzare la finestra di dialogo File > Apri per aprire l'area di lavoro Version Cue con le credenziali corrette. Vengono quindi eseguite ulteriori chiamate alle API JavaScript su questo server.*

Esempio
```
See document.canSaveAVersion().
```

Consultate anche
`document.canSaveAVersion()`, `document.revertToLastVersion()`,
`document.synchronizeWithHeadVersion()`

document.scaleSelection()

Disponibilità
Flash MX 2004.

Uso
```
document.scaleSelection(xScale, yScale [, whichCorner])
```

Parametri
`xScale` Un valore a virgola mobile che specifica il valore *x* in base al quale effettuare la modifica in scala.

`yScale` Un valore a virgola mobile che specifica il valore *y* in base al quale effettuare la modifica in scala.

`whichCorner` Un valore di stringa che specifica il bordo attorno al quale viene eseguita la trasformazione. Se viene omesso, la modifica in scala viene eseguita attorno al punto di trasformazione. I valori accettabili sono: `bottom left"`,

`"bottom right"`, `"top right"`, `"top left"`, `"top center"`, `"right center"`, `"bottom center"` e `"left center"`. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; esegue la modifica in scala della selezione in base a un valore specificato. Equivale a utilizzare lo strumento Trasformazione libera per eseguire la modifica in scala dell'oggetto.

Esempio
L'esempio seguente espande la larghezza della selezione corrente per raddoppiare la larghezza originale e dimezza l'altezza:

```
fl.getDocumentDOM().scaleSelection(2.0, 0.5);
```

L'esempio seguente riflette la selezione in verticale:

```
fl.getDocumentDOM().scaleSelection(1, -1);
```

L'esempio seguente riflette la selezione in orizzontale:

```
fl.getDocumentDOM().scaleSelection(-1, 1);
```

L'esempio seguente scala in verticale la selezione corrente di 1,9 rispetto alla parte superiore centrale:

```
fl.getDocumentDOM().scaleSelection(1, 1.90, 'top center');
```

document.screenOutline

Disponibilità
Flash MX 2004.

Uso
`document.screenOutline`

Descrizione
Proprietà di sola lettura; l'oggetto ScreenOutline corrente per il documento. Prima di accedere per la prima volta all'oggetto, assicurarsi di utilizzare `document.allowScreens()` per determinare se la proprietà esiste.

Esempio
L'esempio seguente visualizza l'array di valori nella proprietà `screenOutline`:

```
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);
```

Consultate anche
`document.allowScreens()`, Oggetto ScreenOutline

document.selectAll()

Disponibilità
Flash MX 2004.

Uso
```
document.selectAll()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; seleziona tutti gli elementi sullo stage. Equivale a premere Ctrl+A (Windows) o Comando+A (Macintosh) o a selezionare Modifica > Seleziona tutto.

Esempio
L'esempio seguente seleziona tutto ciò che è attualmente visibile per l'utente:

```
fl.getDocumentDOM().selectAll();
```

Consultate anche
`document.selection`, `document.selectNone()`

document.selection

Disponibilità
Flash MX 2004.

Uso
```
document.selection
```

Descrizione
Proprietà; un array degli oggetti selezionati nel documento. Se non è selezionato alcun oggetto, restituisce un array con lunghezza pari a zero. Se nessun documento è aperto, restituisce il valore `null`.

Per aggiungere oggetti all'array, è prima necessario selezionarli in uno dei modi seguenti:

- Selezionate manualmente gli oggetti sullo stage.
- Usate uno dei metodi di selezione, ad esempio `document.setSelectionRect()`, `document.setSelectionBounds()`, `document.mouseClick()`, `document.mouseDblClk()` o `document.selectAll()`.
- Selezionate manualmente uno o più fotogrammi.

- Usate uno dei metodi dell'Oggetto Timeline per selezionare i fotogrammi, ad esempio `timeline.getSelectedFrames()`, `timeline.setSelectedFrames()` o `timeline.selectAllFrames()`.
- Specificate tutti gli elementi in un particolare fotogramma (consultate Oggetto Element). Fate riferimento al primo esempio riportato di seguito.
- Create un array di uno o più elementi, quindi assegnatelo all'array `document.selection`. Fate riferimento al terzo esempio riportato di seguito.

Esempio

L'esempio seguente assegna alla selezione corrente tutti gli elementi presenti nel fotogramma 11 (è importante ricordare che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().currentFrame = 10;
fl.getDocumentDOM().selection =
fl.getDocumentDOM().getTimeline().layers[0].frames[10].elements;
```

L'esempio seguente crea un rettangolo nell'angolo superiore sinistro dello stage e una stringa di testo sotto il rettangolo stesso. Quindi seleziona entrambi gli oggetti mediante `document.setSelectionRect()` e li aggiunge all'array `document.selection`. Infine, visualizza il contenuto di `document.selection` nel pannello Output.

```
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]);
}
```

Segue un esempio avanzato che mostra come spostarsi lungo l'array dei livelli e l'array degli elementi per individuare le istanze di un particolare simbolo e selezionarle. L'esempio può includere anche la ricerca di più fotogrammi o scene. Assegna alla selezione tutte le istanze del clip filmato `myMovieClip` presente nel primo fotogramma:

```
// 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()

Disponibilità
Flash MX 2004.

Uso
`document.selectNone()`

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; deseleziona gli elementi selezionati.

Esempio
L'esempio seguente deseleziona gli elementi eventualmente selezionati:

```
fl.getDocumentDOM().selectNone();
```

Consultate anche

`document.selectAll()`, `document.selection`

document.setAlignToDocument()

Disponibilità

Flash MX 2004.

Uso

```
document.setAlignToDocument(bToStage)
```

Parametri

`bToStage` Un valore booleano che, se impostato su `true`, allinea gli oggetti allo stage. Se è impostato su `false`, non li allinea.

Restituisce

Nulla.

Descrizione

Metodo; recupera la preferenza per `document.align()`, `document.distribute()`, `document.match()` e `document.space()` da usare nel documento. Equivale ad attivare il pulsante Allo stage nel pannello Allinea.

Esempio

L'esempio seguente attiva il pulsante Allo stage nel pannello Allinea, affinché gli oggetti vengano allineati con lo stage.

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

Consultate anche

`document.getAlignToDocument()`

document.setBlendMode()

Disponibilità

Flash 8.

Uso

```
document.setBlendMode(mode)
```

Parametri

`mode` Una stringa che rappresenta il metodo di fusione desiderato per gli oggetti selezionati. I valori accettabili sono `"normal"`, `"layer"`, `"multiply"`, `"screen"`, `"overlay"`, `"hardlight"`, `"lighten"`, `"darken"`, `"difference"`, `"add"`, `"subtract"`, `"invert"`, `"alpha"` ed `"erase"`.

Restituisce
Nulla.

Descrizione
Metodo; imposta il metodo di fusione per gli oggetti selezionati.

Esempio
L'esempio seguente imposta su `"add"` il metodo di fusione per l'oggetto selezionato.

```
fl.getDocumentDOM().setBlendMode("add");
```

Consultate anche
`document.addFilter()`, `document.setFilterProperty()`, `symbolInstance.blendMode`

document.setCustomFill()

Disponibilità
Flash MX 2004.

Uso
`document.setCustomFill(fill)`

Parametri
`fill` Un oggetto Fill che specifica le impostazioni di riempimento da utilizzare. Consultate Oggetto Fill.

Restituisce
Nulla.

Descrizione
Metodo; configura le impostazioni di riempimento per il pannello Strumenti, per la finestra di ispezione Proprietà e per tutte le forme selezionate. In tal modo, uno script configura le impostazioni di riempimento prima di disegnare l'oggetto, anziché disegnare l'oggetto, selezionarlo e infine modificarne le impostazioni di riempimento. Inoltre consente a uno script di modificare le impostazioni di riempimento del pannello Strumenti e della finestra di ispezione Proprietà.

Esempio
L'esempio seguente imposta il bianco come colore del campione di colore di riempimento nel pannello Strumenti, nella finestra di ispezione Proprietà e in tutte le forme selezionate:

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

Consultate anche
`document.getCustomFill()`

document.setCustomStroke()

Disponibilità
Flash MX 2004.

Uso
`document.setCustomStroke(stroke)`

Parametri
`stroke` Un Oggetto Stroke.

Restituisce
Nulla.

Descrizione
Metodo; configura le impostazioni del tratto per il pannello Strumenti, per la finestra di ispezione Proprietà e per tutte le forme selezionate. In tal modo, uno script configura le impostazioni del tratto prima di disegnare l'oggetto, anziché disegnare l'oggetto, selezionarlo e infine modificarne le impostazioni del tratto. Inoltre consente a uno script di modificare le impostazioni del tratto del pannello Strumenti e della finestra di ispezione Proprietà.

Esempio
L'esempio seguente modifica le impostazioni dello spessore del tratto per il pannello Strumenti, per la finestra di ispezione Proprietà e per tutte le forme selezionate.

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

Consultate anche
`document.getCustomStroke()`

document.setElementProperty()

Disponibilità
Flash MX 2004.

Uso
`document.setElementProperty(property, value)`

Parametri
`property` Una stringa che specifica il nome della proprietà `Element` da impostare. Per un elenco completo di proprietà e valori, consultate la tabella di riepilogo delle proprietà per l'Oggetto Element.

Non potete usare questo metodo per impostare i valori delle proprietà di sola lettura, ad esempio `element.elementType`, `element.top` o `element.left`.

`value` Un numero intero che specifica il valore da impostare nella proprietà Element specificata.

Restituisce
Nulla.

Descrizione
Metodo; imposta la proprietà `Element` specificata sugli oggetti selezionati nel documento. Non esegue alcuna operazione se non è presente una selezione.

Esempio
L'esempio seguente imposta per tutti gli oggetti selezionati una larghezza pari a 100 e un'altezza pari a 50:

```
fl.getDocumentDOM().setElementProperty("width", 100);
fl.getDocumentDOM().setElementProperty("height", 50);
```

document.setElementTextAttr()

Disponibilità
Flash MX 2004.

Uso
```
document.setElementTextAttr(attrName, attrValue [, startIndex [, endIndex]])
```

Parametri
`attrName` Una stringa che specifica il nome della proprietà `TextAttrs` da modificare.

`attrValue` Il valore su cui impostare la proprietà `TextAttrs`. Per un elenco dei nomi delle proprietà e dei valori attesi, consultate la tabella di riepilogo delle proprietà per l'Oggetto TextAttrs.

`startIndex` Un numero intero che specifica l'indice del primo carattere modificato. Questo parametro è opzionale.

`endIndex` Un numero intero che specifica l'indice dell'ultimo carattere modificato. Questo parametro è opzionale.

Restituisce
Un valore booleano: `true` se almeno una proprietà dell'attributo di testo viene modificata; `false` in caso contrario.

Descrizione
Metodo; imposta sul valore specificato la proprietà `TextAttrs` specificata per gli elementi di testo selezionati. Per un elenco dei nomi delle proprietà e dei valori consentiti, consultate la tabella di riepilogo delle proprietà per l'Oggetto TextAttrs. Se i parametri opzionali non vengono passati, il metodo imposta lo stile dell'intervallo di testo selezionato o dell'intero campo di testo se non è selezionato alcun testo. Se viene passato solo *startIndex*, il metodo imposta gli attributi di tale carattere. Se vengono passati *startIndex* e *endIndex*, il metodo imposta gli attributi sui caratteri compresi tra *startIndex* ed *endIndex* escluso. Se sono specificati gli stili di paragrafo, il metodo ha effetto anche su tutti i paragrafi all'interno dell'intervallo.

Esempio
Gli esempi seguenti impostano gli attributi di testo `fillColor, italic e bold` per gli elementi di testo selezionati:

```
var success = fl.getDocumentDOM().setElementTextAttr("fillColor", "#00ff00");
var pass = fl.getDocumentDOM().setElementTextAttr("italic", true, 10);
var ok = fl.getDocumentDOM().setElementTextAttr("bold", true, 5, 15);
```

document.setFillColor()

Disponibilità
Flash MX 2004.

Uso
```
document.setFillColor(color)
```

Parametri
`color` Il colore di riempimento, in uno dei seguenti formati:

- Una stringa nel formato `"#RRGGBB"` o `"#RRGGBBAA"`
- Un numero esadecimale nel formato `0xRRGGBB`
- Un numero intero che rappresenta l'equivalente decimale di un numero esadecimale.

  Se impostato su `null`, non viene impostato alcun colore di riempimento. Equivale a impostare su nessun riempimento il campione di colore Riempimento nell'interfaccia utente.

Restituisce
Nulla.

Descrizione
Metodo; sostituisce il colore di riempimento della selezione con il colore specificato. Per informazioni su come modificare il colore di riempimento nel pannello Strumenti e nella finestra di ispezione Proprietà, consultate `document.setCustomFill()`.

Esempio
Le prime tre istruzioni nell'esempio seguente impostano il colore di riempimento utilizzando ciascuno dei tre formati di definizione del colore. La quarta istruzione imposta il riempimento su nessun riempimento.

```
fl.getDocumentDOM().setFillColor("#cc00cc");
fl.getDocumentDOM().setFillColor(0xcc00cc);
fl.getDocumentDOM().setFillColor(120000);
fl.getDocumentDOM().setFillColor(null);
```

document.setFilterProperty()

Disponibilità
Flash 8.

Uso
```
document.setFilterProperty(property, filterIndex, value)
```

Parametri
`property` Una stringa che specifica la proprietà da impostare. I valori accettabili sono `"blurX"`, `"blurY"`, `"quality"`, `angle"`, `"distance"`, `"strength"`, `"knockout"`, `"inner"`, `"bevelType"`, `"color"`, `"shadowColor"` e `"highlightColor"`.

filterIndex Un numero intero che specifica l'indice a base zero del filtro nell'elenco dei filtri.

value Un numero o una stringa che specifica il valore da impostare per la proprietà Filter specificata. I valori accettabili dipendono dalla proprietà e dal filtro che si imposta.

Restituisce
Nulla.

Descrizione
Metodo; imposta una proprietà Filter specificata per gli oggetti attualmente selezionati (supponendo che l'oggetto supporti il filtro specificato).

Esempio
L'esempio seguente imposta la proprietà `quality` su 2 per il secondo filtro (valore di indice 1) nell'elenco dei filtri per gli oggetti selezionati, quindi imposta la proprietà `shadowColor` del primo filtro nell'elenco dei filtri per gli oggetti selezionati:

```
fl.getDocumentDOM().setFilterProperty("quality", 1, 2);
fl.getDocumentDOM().setFilterProperty("shadowColor", 0, "#FF00FF");
```

Consultate anche
`document.addFilter()`, `document.getFilters()`, `document.setBlendMode()`, `document.setFilters()`, Oggetto Filter

document.setFilters()

Disponibilità
Flash 8.

Uso
```
document.setFilters(filterArray)
```

Parametri
filterArray L'array di filtri corrente specificato.

Restituisce
Nulla.

Descrizione
Metodo; applica i filtri agli oggetti selezionati. Utilizzate questo metodo dopo aver chiamato `document.getFilters()` e aver apportato ai filtri tutte le modifiche desiderate.

Esempio
L'esempio seguente ottiene i filtri applicati all'oggetto selezionato e imposta la proprietà `blurX` di tutti i filtri di sfocatura su 50:

```
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);
```

Consultate anche

`document.addFilter()`, `document.getFilters()`, `document.setFilterProperty()`, Oggetto Filter

document.setInstanceAlpha()

Disponibilità

Flash MX 2004.

Uso

```
document.setInstanceAlpha(opacity)
```

Parametri

`opacity` Un numero intero compreso tra 0 (trasparente) e 100 (completamente saturo) che regola la trasparenza dell'istanza.

Restituisce

Nulla.

Descrizione

Metodo; imposta l'opacità dell'istanza.

Esempio

L'esempio seguente imposta l'opacità della tinta sul valore 50:

```
fl.getDocumentDOM().setInstanceAlpha(50);
```

document.setInstanceBrightness()

Disponibilità

Flash MX 2004.

Uso

```
document.setInstanceBrightness(brightness)
```

Parametri

`brightness` Un numero intero che specifica un valore di luminosità compreso tra -100 (nero) e 100 (bianco).

Restituisce
Nulla.

Descrizione
Metodo; imposta la luminosità dell'istanza.

Esempio
L'esempio seguente imposta la luminosità dell'istanza sul valore 50:

```
fl.getDocumentDOM().setInstanceBrightness(50);
```

document.setInstanceTint()

Disponibilità
Flash MX 2004.

Uso
```
document.setInstanceTint( color, strength )
```

Parametri
`color` Il colore della tinta, in uno dei seguenti formati:

- Una stringa nel formato `"#RRGGBB"` o `"#RRGGBBAA"`
- Un numero esadecimale nel formato `0xRRGGBB`
- Un numero intero che rappresenta l'equivalente decimale di un numero esadecimale.

`strength` Un valore intero compreso tra 0 e 100 che specifica l'opacità della tinta.

Restituisce
Nulla.

Descrizione
Metodo; imposta la tinta dell'istanza.

Esempio
L'esempio seguente imposta la tinta rossa per l'istanza selezionata con un valore di opacità pari a 50:

```
fl.getDocumentDOM().setInstanceTint(0xff0000, 50);
```

document.setMetadata()

Disponibilità
Flash 8.

Uso
```
document.setMetadata(strMetadata)
```

Parametri

`strMetadata` Una stringa che contiene i metadati XML associati al documento. Per ulteriori informazioni, fate riferimento alla seguente descrizione.

Restituisce

Un valore booleano: `true` se l'esito è positivo; `false` in caso contrario.

Descrizione

Metodo; imposta i metadati XML per il documento specificato, sovrascrivendo quelli esistenti. Il codice XML passato come *strMetadata* viene convalidato ed è possibile che venga riscritto prima di essere memorizzato. Se non possono essere convalidati come XML valido o se violano determinate regole, i metadati XML non vengono impostati e viene restituito il valore `false`. Se viene restituito `false`, non è possibile ottenere ulteriori dettagli sull'errore.

Nota: *anche se viene restituito `true`, il codice XML impostato potrebbe non essere esattamente la stessa stringa che avete passato. Per ottenere l'esatto valore che era stato impostato per il codice XML, usate `document.getMetadata()`.*

I metadati sono in formato RDF, conforme alla specifica XMP. Per ulteriori informazioni su RDF e XMP, consultate le risorse seguenti.

- Per un'introduzione al formato RDF visitate il sito all'indirizzo www.w3.org/TR/rdf-primer/
- La specifica RDF all'indirizzo www.w/19990222.org/TR/rdf-primer/
- La home page per la specifica XMP all'indirizzo http://www.adobe.com/it/enterprise/xmp.html/

Esempio

L'esempio seguente mostra diversi modi validi di rappresentare gli stessi dati. In tutti questi casi, tranne il secondo, se i dati sono stati inviati a `Document.setMetadata()`, non vengono riscritti ad eccezione delle interruzioni di riga, che vengono rimosse.

Nel primo esempio, i metadati dispongono di tag, con schemi diversi posti in tag `rdf:Description` distinti:

```
<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>
```

Nel secondo esempio, i metadati dispongono di tag, ma con tutti gli schemi diversi in un singolo tag `rdf:Description`. Questo esempio include inoltre commenti che verranno ignorati e scartati da `Document.setMetadata()`:

```
<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>
```

Nel terzo esempio, i metadati dispongono di attributi e tutti gli schemi diversi si trovano in un singolo tag `rdf:Description`:

```
<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>
```

Consultate anche

`document.getMetadata()`

document.setMobileSettings()

Disponibilità

Flash CS3 Professional.

Uso

```
document.setMobileSettings(xmlString)
```

Parametri

`xmlString` Una stringa che descrive le impostazioni XML in un file FLA mobile.

Restituisce

Il valore `true` se le impostazioni sono state impostate correttamente; `false` in caso contrario.

Descrizione

Metodo; imposta il valore di una stringa di impostazioni XML in un file FLA mobile (la maggior parte dei file FLA mobili dispongono di una stringa XML che descrive le impostazioni all'interno del documento).

Esempio

L'esempio che segue definisce la stringa di impostazioni XML per un file FLA mobile. Notate che l'esempio seguente rappresenta una singola riga di codice.

```
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>");
```

Consultate anche

`document.getMobileSettings()`

document.setOvalObjectProperty()

Disponibilità

Flash CS3 Professional.

Uso

```
document.setOvalObjectProperty(propertyName, value)
```

Parametri

`propertyName` Una stringa che specifica la proprietà da impostare. Per i valori accettabili, consultate la tabella di riepilogo delle proprietà per l'Oggetto Oval.

`value` Il valore da assegnare alla proprietà. I valori accettabili variano a seconda della proprietà specificata in *propertyName*.

Restituisce

Nulla.

Descrizione

Metodo; indica il valore di una proprietà specifica di oggetti Oval di base.

Esempio

Per vedere degli esempi, fate riferimento alle singole proprietà in Oggetto Oval.

Consultate anche

Oggetto Oval, `shape.isOvalObject`

document.setPlayerVersion()

Disponibilità

Flash CS3 Professional.

Uso

```
document.setPlayerVersion(version)
```

Parametri

`version` Una stringa che rappresenta la versione del lettore Flash Player di destinazione per il documento specificato. I valori accettabili sono `"FlashLite"`, `"FlashLite11"`, `"FlashLite20"`, `"FlashLite30"`, `"1"`, `"2"`, `"3"`, `"4"`, `"5"`, `"6"`, `"7"`, `"8"`, `"9"`, `"FlashPlayer10"` e `"AdobeAIR1_1"`.

Restituisce

Il valore `true` se la versione del lettore è stata impostata correttamente; `false` in caso contrario.

Descrizione

Metodo; imposta la versione del lettore Flash Player di destinazione per il documento specificato. Si tratta dello stesso valore impostato nella finestra di dialogo Impostazioni pubblicazione.

Esempio

L'esempio che segue utilizza Flash Player 6 come versione del lettore per il documento attuale:

```
fl.getDocumentDOM().setPlayerVersion("6");
```

Consultate anche

`document.getPlayerVersion()`

document.setRectangleObjectProperty()

Disponibilità

Flash CS3 Professional.

Uso

```
document.setRectangleObjectProperty(propertyName, value)
```

Parametri

`propertyName` Una stringa che specifica la proprietà da impostare. Per i valori accettabili, consultate la tabella di riepilogo delle proprietà per l'Oggetto Rectangle.

`value` Il valore da assegnare alla proprietà. I valori accettabili variano a seconda della proprietà specificata in *propertyName*.

Restituisce

Nulla.

Descrizione

Metodo; indica il valore di una proprietà specifica di oggetti Rectangle di base.

Esempio

Per vedere degli esempi, fate riferimento alle singole proprietà in Oggetto Rectangle.

Consultate anche

Oggetto Rectangle, shape.isRectangleObject

document.setSelectionBounds()

Disponibilità

Flash MX 2004; parametro *bContactSensitiveSelection* aggiunto in Flash 8.

Uso

```
document.setSelectionBounds(boundingRectangle [, bContactSensitiveSelection])
```

Parametri

`boundingRectangle` Un rettangolo che specifica la nuova posizione e le dimensioni della selezione. Per informazioni sul formato di *boundingRectangle*, consultate `document.addNewRectangle()`.

`bContactSensitiveSelection` Un valore booleano che indica se la modalità di selezione Sensibile al contatto è attiva (`true`) o inattiva (`false`) per la selezione degli oggetti. Il valore predefinito è `false`.

Restituisce

Nulla.

Descrizione

Metodo; sposta e ridimensiona la selezione con un'unica operazione.

Se passate un valore per *bContactSensitiveSelection*, tale valore è valido solo per questo metodo e non incide sulla modalità di selezione Sensibile al contatto del documento (consultate `fl.contactSensitiveSelection`).

Esempio

L'esempio seguente sposta la selezione corrente al punto 10,20 e assegna ad essa le nuove dimensioni 100, 200:

```
var l = 10;
var t = 20;
fl.getDocumentDOM().setSelectionBounds({left:l, top:t, right:(100+l), bottom:(200+t)});
```

Consultate anche

`document.selection`, `document.setSelectionRect()`

document.setSelectionRect()

Disponibilità

Flash MX 2004; parametro *bContactSensitiveSelection* aggiunto in Flash 8.

Uso

```
document.setSelectionRect(rect [, bReplaceCurrentSelection [, bContactSensitiveSelection]])
```

Parametri

`rect` Un oggetto rettangolo da impostare come selezionato. Per informazioni sul formato di *rect*, consultate document.addNewRectangle().

`bReplaceCurrentSelection` Un valore booleano che specifica se il metodo sostituisce la selezione corrente (`true`) o effettua un'aggiunta alla selezione corrente (`false`). Il valore predefinito è `true`.

`bContactSensitiveSelection` Un valore booleano che indica se la modalità di selezione Sensibile al contatto è attiva (`true`) o inattiva (`false`) per la selezione degli oggetti. Il valore predefinito è `false`.

Restituisce
Nulla.

Descrizione
Metodo; disegna un perimetro di selezione rettangolare relativo allo stage utilizzando le coordinate specificate. È un metodo diverso da `document.getSelectionRect()`, in cui il rettangolo è relativo all'oggetto in corso di modifica.

Equivale a trascinare un rettangolo con lo strumento Selezione. Per selezionare un'istanza, racchiuderla completamente nel rettangolo.

Se passate un valore per *bContactSensitiveSelection*, tale valore è valido solo per questo metodo e non incide sulla modalità di selezione Sensibile al contatto del documento (consultate `fl.contactSensitiveSelection`).

Nota: *se si ripete `setSelectionRect()` mediante il pannello o la voce di menu Cronologia, viene ripetuta l'operazione precedente a `setSelectionRect()`.*

Esempio
Nell'esempio seguente la seconda selezione sostituisce la prima:

```
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);
```

Nell'esempio seguente la seconda selezione viene aggiunta alla prima. Equivale a selezionare un altro oggetto tenendo premuto il tasto Maiusc.

```
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);
```

Consultate anche
`document.getSelectionRect()`, `document.selection`, `document.setSelectionBounds()`

document.setStageVanishingPoint()

Disponibilità
Flash CS4 Professional.

Uso
`document.setStageVanishingPoint(point)`

Parametri
`point` Un punto che specifica le coordinate *x* e *y* della posizione in cui impostare il fuoco prospettico per la visualizzazione degli oggetti 3D.

Restituisce
Nulla.

Descrizione
Specifica il fuoco prospettico per la visualizzazione degli oggetti 3D.

Esempio
L'esempio seguente imposta il fuoco prospettico dello stage:

```
fl.getDocumentDOM().setStageVanishingPoint({x:45, y:45});
```

document.setStageViewAngle()

Disponibilità
Flash CS4 Professional.

Uso

```
document.setStageViewAngle(angle)
```

Parametri
`angle` Un valore a virgola mobile compreso tra 0,0 e 179,0.

Restituisce
Nulla.

Descrizione
Specifica l'angolo prospettiva per la visualizzazione degli oggetti 3D.

Esempio
L'esempio seguente imposta l'angolo prospettiva dello stage su 70 gradi:

```
fl.getDocumentDOM().setStageViewAngle(70);
```

document.setStroke()

Disponibilità
Flash MX 2004.

Uso

```
document.setStroke(color, size, strokeType)
```

Parametri
`color` Il colore del tratto, in uno dei seguenti formati:

- Una stringa nel formato `"#RRGGBB"` o `"#RRGGBBAA"`
- Un numero esadecimale nel formato `0xRRGGBB`
- Un numero intero che rappresenta l'equivalente decimale di un numero esadecimale.

`size` Un valore a virgola mobile che specifica le nuove dimensioni del tratto per la selezione.

strokeType Un rettangolo che specifica il nuovo tipo di tratto per la selezione. I valori accettabili sono `"hairline"`, `"solid"`, `"dashed"`, `"dotted"`, `"ragged"`, `"stipple"` e `"hatched"`.

Restituisce
Nulla.

Descrizione
Metodo; imposta il colore, la larghezza e lo stile del tratto selezionato. Per informazioni su come modificare il tratto nel pannello Strumenti e nella finestra di ispezione Proprietà, consultate `document.setCustomStroke()`.

Esempio
L'esempio seguente imposta per il tratto il colore rosso, la dimensione 3,25 e il tipo tratteggiato:

```
fl.getDocumentDOM().setStroke("#ff0000", 3.25, "dashed");
```

document.setStrokeColor()

Disponibilità
Flash MX 2004.

Uso
`document.setStrokeColor(color)`

Parametri
color Il colore del tratto, in uno dei seguenti formati:

- Una stringa nel formato `"#RRGGBB"` o `"#RRGGBBAA"`
- Un numero esadecimale nel formato `0xRRGGBB`
- Un numero intero che rappresenta l'equivalente decimale di un numero esadecimale.

Restituisce
Nulla.

Descrizione
Metodo; sostituisce il colore del tratto della selezione con il colore specificato. Per informazioni su come modificare il tratto nel pannello Strumenti e nella finestra di ispezione Proprietà, consultate `document.setCustomStroke()`.

Esempio
Le tre istruzioni nell'esempio seguente impostano il colore del tratto utilizzando ognuno dei tre formati di specificazione del colore.

```
fl.getDocumentDOM().setStrokeColor("#cc00cc");
fl.getDocumentDOM().setStrokeColor(0xcc00cc);
fl.getDocumentDOM().setStrokeColor(120000);
```

document.setStrokeSize()

Disponibilità
Flash MX 2004.

Uso
```
document.setStrokeSize(size)
```

Parametri
`size` Un valore a virgola mobile compreso tra 0,25 e 10 che specifica le dimensioni del tratto. Il metodo considera un massimo di due decimali.

Restituisce
Nulla.

Descrizione
Metodo; sostituisce le dimensioni del tratto della selezione con quelle specificate. Per informazioni su come modificare il tratto nel pannello Strumenti e nella finestra di ispezione Proprietà, consultate `document.setCustomStroke()`.

Esempio
L'esempio seguente imposta su 5 le dimensioni del tratto della selezione:

```
fl.getDocumentDOM().setStrokeSize(5);
```

document.setStrokeStyle()

Disponibilità
Flash MX 2004.

Uso
```
document.setStrokeStyle(strokeType)
```

Parametri
`strokeType` Una stringa che specifica lo stile del tratto per la selezione. I valori accettabili sono `"hairline"`, `"solid"`, `"dashed"`, `"dotted"`, `"ragged"`, `"stipple"` e `"hatched"`.

Restituisce
Nulla.

Descrizione
Metodo; sostituisce lo stile del tratto della selezione con lo stile specificato. Per informazioni su come modificare il tratto nel pannello Strumenti e nella finestra di ispezione Proprietà, consultate `document.setCustomStroke()`.

Esempio
L'esempio seguente imposta su `"dashed"` lo stile del tratto della selezione:

```
fl.getDocumentDOM().setStrokeStyle("dashed");
```

document.setTextRectangle()

Disponibilità
Flash MX 2004.

Uso
```
document.setTextRectangle(boundingRectangle)
```

Parametri
`boundingRectangle` Un rettangolo che specifica le nuove dimensioni per il flusso dell'elemento di testo. Per informazioni sul formato di *boundingRectangle*, consultate document.addNewRectangle().

Restituisce
Un valore booleano: `true` se vengono modificate le dimensioni di almeno un campo di testo; `false` in caso contrario.

Descrizione
Metodo; applica le dimensioni specificate al rettangolo di delimitazione dell'elemento di testo selezionato. Questo metodo fa in modo che al testo venga applicato un flusso all'interno del nuovo rettangolo; l'elemento di testo non viene scalato o trasformato. I valori passati in *boundingRectangle* vengono utilizzati nel modo seguente:

- Se il testo è orizzontale e statico, il metodo prende in considerazione solo il valore della larghezza passato in *boundingRectangle*; l'altezza viene calcolata automaticamente in modo che si adatti a tutto il testo.
- Se il testo è verticale e quindi statico, il metodo prende in considerazione solo il valore dell'altezza passato in *boundingRectangle*; la larghezza viene calcolata automaticamente in modo che si adatti a tutto il testo.
- Se il testo è dinamico o di input, il metodo prende in considerazione entrambi i valori della larghezza e dell'altezza passati in *boundingRectangle*; il rettangolo risultante potrebbe essere più grande del testo. Tuttavia, se i parametri specificano dimensioni del rettangolo troppo ridotte per adattarsi a tutto il testo, il metodo prende in considerazione solo il valore della larghezza passato in *boundingRectangle*, mentre l'altezza viene calcolata automaticamente in modo che si adatti a tutto il testo.

Esempio
L'esempio seguente applica le dimensioni specificate al rettangolo di delimitazione del testo:

```
fl.getDocumentDOM().setTextRectangle({left:0, top:0, right:50, bottom:200})
```

document.setTextSelection()

Disponibilità
Flash MX 2004.

Uso
```
document.setTextSelection(startIndex, endIndex)
```

Parametri
`startIndex` Un numero intero che specifica la posizione del primo carattere da selezionare. La posizione del primo carattere è 0 (zero).

endIndex Un numero intero che specifica la posizione finale della selezione fino a *endIndex* escluso. La posizione del primo carattere è 0 (zero).

Restituisce
Un valore booleano: `true` se il metodo è in grado di impostare correttamente la selezione di testo; `false` in caso contrario.

Descrizione
Metodo; imposta la selezione di testo del campo di testo selezionato sui valori specificati da *startIndex* ed *endIndex*. Viene attivata la modifica del testo, se non è già attiva.

Esempio
L'esempio seguente seleziona il testo dal 6° al 25° carattere:

```
fl.document.setTextSelection(5, 25);
```

document.setTextString()

Disponibilità
Flash MX 2004.

Uso

```
document.setTextString(text [, startIndex [, endIndex]])
```

Parametri
text La stringa dei caratteri da inserire nel campo di testo.

startIndex Un numero intero che specifica il primo carattere da sostituire. La posizione del primo carattere è 0 (zero). Questo parametro è opzionale.

endIndex Un numero intero che specifica l'ultimo carattere da sostituire. Questo parametro è opzionale.

Restituisce
Un valore booleano: `true` se vengono modificate le dimensioni di almeno una stringa di testo; `false` in caso contrario.

Descrizione
Metodo; inserisce una stringa di testo. Se i parametri opzionali non vengono passati, viene sostituita la selezione di testo esistente; se l'oggetto Text non è attualmente in corso di modifica, viene sostituito l'intero testo. Se viene passato solo *startIndex*, la stringa passata viene inserita in questa posizione. Se vengono passati *startIndex* ed *endIndex*, la stringa passata sostituisce il segmento di testo compreso tra *startIndex* ed *endIndex* escluso.

Esempio
L'esempio seguente sostituisce la selezione di testo corrente con "Hello World":

```
var success = fl.getDocumentDOM().setTextString("Hello World!");
```

L'esempio seguente inserisce "hello" nella posizione 6 della selezione di testo corrente:

```
var pass = fl.getDocumentDOM().setTextString("hello", 6);
```

L'esempio seguente inserisce nella selezione di testo corrente il termine "Howdy" tra la posizione 2 e la posizione 7 esclusa:

```
var ok = fl.getDocumentDOM().setTextString("Howdy", 2, 7);
```

Consultate anche

`document.getTextString()`

document.setTransformationPoint()

Disponibilità

Flash MX 2004.

Uso

```
document.setTransformationPoint( transformationPoint )
```

Parametri

`transformationPoint` Un punto (ad esempio `{x:10,y:20}`, dove `x` e `y` sono numeri a virgola mobile) che specifica i valori del punto di trasformazione di ciascuno dei seguenti elementi:

- Forme: *transformationPoint* viene impostato rispetto al documento; 0,0 corrisponde all'angolo superiore sinistro dello stage.
- Simboli: *transformationPoint* viene impostato rispetto al punto di registrazione del simbolo; 0,0 corrisponde al punto di registrazione.
- Testo: *transformationPoint* viene impostato rispetto al campo di testo; 0,0 corrisponde all'angolo superiore sinistro del campo di testo.
- Bitmap/video: *transformationPoint* viene impostato rispetto al bitmap/video; 0,0 corrisponde all'angolo superiore sinistro della bitmap o del video.
- Oggetti di disegno, ovali e rettangoli di base: *transformationPoint* viene impostato rispetto al documento; 0,0 corrisponde all'angolo superiore sinistro dello stage. Per impostare *transformationPoint* rispetto al punto centrale dell'oggetto, della forma di base o del gruppo, usate `element.setTransformationPoint()`.

Restituisce

Nulla.

Descrizione

Metodo; imposta la posizione del punto di trasformazione della selezione corrente.

Esempio

L'esempio seguente imposta su 100, 200 il punto di trasformazione della selezione corrente:

```
fl.getDocumentDOM().setTransformationPoint({x:100, y:200});
```

Consultate anche

`document.getTransformationPoint()`, `element.setTransformationPoint()`

document.silent

Disponibilità
Flash MX 2004.

Uso
```
document.silent
```

Descrizione
Proprietà; un valore booleano che specifica se l'oggetto è accessibile. È equivalente alla logica inversa dell'impostazione Rendi accessibile il filmato del pannello Accessibilità. In altre parole, il valore `true` per `document.silent` equivale all'opzione Rendi accessibile il filmato deselezionata. Il valore `false` equivale all'opzione selezionata.

Esempio
Questo esempio imposta la variabile `isSilent` sul valore della proprietà `silent`:

```
var isSilent = fl.getDocumentDOM().silent;
```

L'esempio seguente imposta la proprietà `silent` su `false` per indicare che il documento è accessibile:

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

document.skewSelection()

Disponibilità
Flash MX 2004.

Uso
```
document.skewSelection(xSkew, ySkew [, whichEdge])
```

Parametri
`xSkew` Un valore a virgola mobile che specifica il valore *x* dell'inclinazione da applicare, espressa in gradi.

`ySkew` Un valore a virgola mobile che specifica il valore *y* dell'inclinazione da applicare, espressa in gradi.

`whichEdge` Una stringa che specifica il bordo su cui viene eseguita la trasformazione; se omessa, l'inclinazione viene applicata al punto di trasformazione. I valori accettabili sono `"top center"`, `"right center"`, `"bottom center"` e `"left center"`. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; inclina la selezione in base a un valore specificato. Equivale a utilizzare lo strumento Trasformazione libera per inclinare l'oggetto.

Esempio
Gli esempi seguenti inclinano l'oggetto selezionato di 2,0 gradi in senso verticale e 1,5 gradi in senso orizzontale. Il secondo esempio trasforma gli oggetti prendendo come riferimento il bordo superiore centrale:

```
fl.getDocumentDOM().skewSelection(2.0, 1.5);
fl.getDocumentDOM().skewSelection(2.0, 1.5, "top center");
```

document.smoothSelection()

Disponibilità
Flash MX 2004.

Uso
```
document.smoothSelection()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; smussa la curva di ogni contorno di riempimento o linea curva selezionato. Equivale a utilizzare lo strumento Attenua del pannello Strumenti.

Esempio
L'esempio seguente smussa la curva della selezione corrente:

```
fl.getDocumentDOM().smoothSelection();
```

document.sourcePath

Disponibilità
Flash CS4 Professional.

Uso
```
document.sourcePath
```

Descrizione
Proprietà; una stringa che contiene un elenco di elementi presenti nel percorso d'origine di ActionScript 3.0 per il documento che indica la posizione dei file delle classi di ActionScript. Gli elementi nella stringa sono separati da punti e virgola. Nello strumento di creazione, gli elementi vengono specificati scegliendo File > Impostazioni di pubblicazione e quindi scegliendo Impostazioni per gli script ActionScript 3.0.

Esempio
L'esempio seguente imposta la cartella di file ./Class sul percorso d'origine del documento:

```
var myDoc = fl.getDocumentDOM();
fl.trace(myDoc.sourcePath);
myDoc.sourcePath = "./Class files;" + myDoc.sourcePath;
fl.trace(myDoc.sourcePath);
```

Consultate anche

document.externalLibraryPath,document.libraryPath, fl.sourcePath

document.space()

Disponibilità

Flash MX 2004.

Uso

```
document.space(direction [, bUseDocumentBounds])
```

Parametri

direction Una stringa che specifica la direzione in cui deve essere applicata la spaziatura agli oggetti compresi nella selezione. I valori accettabili sono "horizontal" e "vertical".

bUseDocumentBounds Un valore booleano che, se impostato su true, applica la spaziatura agli oggetti rispetto ai contorni del documento. In caso contrario, il metodo utilizza i contorni degli oggetti selezionati. Il valore predefinito è false. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione

Metodo; applica una spaziatura equidistante agli oggetti compresi nella selezione.

Esempio

L'esempio seguente applica agli oggetti una spaziatura orizzontale rispetto allo stage:

```
fl.getDocumentDOM().space("horizontal",true);
```

L'esempio seguente applica agli oggetti una spaziatura orizzontale l'uno rispetto all'altro:

```
fl.getDocumentDOM().space("horizontal");
```

L'esempio seguente applica agli oggetti una spaziatura orizzontale l'uno rispetto all'altro, con *bUseDcoumentBounds* impostato espressamente su false:

```
fl.getDocumentDOM().space("horizontal",false);
```

Consultate anche

document.getAlignToDocument(), document.setAlignToDocument()

document.straightenSelection()

Disponibilità
Flash MX 2004.

Uso
```
document.straightenSelection()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; raddrizza i tratti selezionati. Equivale a utilizzare il pulsante Raddrizza nel pannello Strumenti.

Esempio
L'esempio seguente raddrizza la curva della selezione corrente:

```
fl.getDocumentDOM().straightenSelection();
```

document.swapElement()

Disponibilità
Flash MX 2004.

Uso
```
document.swapElement(name)
```

Parametri
`name` Una stringa che specifica il nome dell'elemento della libreria da utilizzare.

Restituisce
Nulla.

Descrizione
Metodo; scambia la selezione corrente con quella specificata. La selezione deve contenere un oggetto grafico, un pulsante, un clip filmato, un video o una bitmap. Questo metodo visualizza un messaggio di errore se non è selezionato alcun oggetto o se l'oggetto non viene trovato.

Esempio
L'esempio seguente scambia la selezione corrente con l'elemento `Symbol 1` presente nella libreria:

```
fl.getDocumentDOM().swapElement('Symbol 1');
```

document.swapStrokeAndFill()

Disponibilità
Flash 8.

Uso
```
document.swapStrokeAndFill()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; scambia i colori di tratto e riempimento.

Esempio
L'esempio seguente scambia i colori di tratto e riempimento nel documento corrente:

```
fl.getDocumentDOM().swapStrokeAndFill();
```

document.synchronizeWithHeadVersion()

Disponibilità
Flash CS3 Professional.

Uso
```
fldocument.getDocumentDOMsynchronizeWithHeadVersion().swapStrokeAndFill();
```

Parametri
Nessuno.

Restituisce
Il valore booleano `true` se il file specificato è stato sincronizzato correttamente con il server Version Cue; `false` in caso contrario.

Descrizione
Metodo; sincronizza il documento specificato con la versione più recente presente sul server Version Cue e visualizza eventuali errori nel pannello Output.

Questo metodo funziona solamente con i documenti attualmente aperti. Per richiamare la versione più recente di un file attualmente non aperto, usate `fl.downloadLatestVersion()`.

Esempio
L'esempio che segue sincronizza il documento corrente con l'ultima versione memorizzata sul server Version Cue:

```
fl.getDocumentDOM().synchronizeWithHeadVersion();
```

Consultate anche

`document.canSaveAVersion()`, `fl.downloadLatestVersion()`, `document.revertToLastVersion()`, `document.saveAVersion()`, `fl.synchronizeDocumentWithHeadVersion()`

document.testMovie()

Disponibilità
Flash MX 2004.

Uso

```
document.testMovie()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; esegue l'operazione Prova filmato sul documento.

Esempio
L'esempio seguente prova il filmato del documento corrente:

```
fl.getDocumentDOM().testMovie();
```

Consultate anche

`document.canTestMovie()`, `document.testScene()`

document.testScene()

Disponibilità
Flash MX 2004.

Uso

```
document.testScene()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; esegue l'operazione Prova scena sulla scena corrente del documento.

Esempio
L'esempio seguente prova la scena corrente nel documento:

```
fl.getDocumentDOM().testScene();
```

Consultate anche
`document.canTestScene()`, `document.testMovie()`

document.timelines

Disponibilità
Flash MX 2004.

Uso
`document.timelines`

Descrizione
Proprietà di sola lettura; un array degli oggetti della linea temporale (consultate Oggetto Timeline).

Esempio
L'esempio seguente ottiene l'array delle linee temporali correnti nel documento attivo e ne visualizza i nomi nel pannello Output:

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

Consultate anche
`document.currentTimeline`, `document.getTimeline()`

document.traceBitmap()

Disponibilità
Flash MX 2004.

Uso
`document.traceBitmap(threshold, minimumArea, curveFit, cornerThreshold)`

Parametri

`threshold` Un numero intero che controlla il numero di colori presenti nella bitmap ricalcata. I valori accettabili sono numeri interi compresi tra 0 e 500.

`minimumArea` Un numero intero che specifica il raggio, espresso in pixel. I valori accettabili sono numeri interi compresi tra 1 e 1000.

`curveFit` Una stringa che specifica la smussatura con cui vengono disegnati i contorni. I valori accettabili sono `"pixels"`, `"very tight"`, `"tight"`, `"normal"`, `"smooth"` e `"very smooth"`.

`cornerThreshold` Una stringa simile a *curveFit*, ma che agisce sugli angoli di un'immagine bitmap. I valori accettabili sono `"many corners"`, `"normal"` e `"few corners"`.

Restituisce

Nulla.

Descrizione

Metodo; esegue un ricalco bitmap sulla selezione corrente. Equivale a selezionare Elabora > Bitmap > Ricalca bitmap.

Esempio

L'esempio seguente ricalca la bitmap selezionata utilizzando i parametri specificati:

```
fl.getDocumentDOM().traceBitmap(0, 500, 'normal', 'normal');
```

document.translate3DCenter()

Disponibilità

Flash CS4 Professional.

Uso

```
document.translate3DCenter(xyzCoordinate)
```

Parametri

`xyzCoordinate` Una coordinata XYZ che specifica il punto centrale della rotazione o della traslazione 3D.

Restituisce

Nulla.

Descrizione

Metodo: imposta la posizione XYZ attorno a cui la selezione viene traslata o ruotata. Questo metodo è disponibile solo per i clip video.

Esempio

Nell'esempio seguente vengono specificati gli assi XYZ per la traslazione 3D.

```
fl.getDocumentDOM().translate3DCenter({x:180, y:18,z:-30});
```

document.translate3DSelection()

Disponibilità
Flash CS4 Professional.

Uso
```
document.translate3DSelection(xyzCoordinate, bGlobalTransform)
```

Parametri
`xyzCoordinate` Una coordinata XYZ che specifica gli assi della traslazione 3D.

`bGlobalTransform` Un valore booleano che specifica se la modalità di trasformazione deve essere globale (`true`) o locale (`false`).

Restituisce
Nulla.

Descrizione
Metodo; applica una traslazione 3D alla selezione. Questo metodo è disponibile solo per i clip video.

Esempio
Nell'esempio seguente la selezione viene prima traslata rispetto allo stage (globalmente) e quindi rispetto a se stessa (localmente).

```
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);
```

Consultate anche
document.translate3DCenter()

document.transformSelection()

Disponibilità
Flash MX 2004.

Uso
```
document.transformSelection(a, b, c, d)
```

Parametri
`a` Un numero a virgola mobile che specifica l'elemento (0,0) della matrice di trasformazione.

`b` Un numero a virgola mobile che specifica l'elemento (0,1) della matrice di trasformazione.

`c` Un numero a virgola mobile che specifica l'elemento (1,0) della matrice di trasformazione.

`d` Un numero a virgola mobile che specifica l'elemento (1,1) della matrice di trasformazione.

Restituisce
Nulla.

Descrizione
Metodo; esegue una trasformazione generale sulla selezione corrente applicando la matrice specificata negli argomenti. Per ulteriori informazioni, consultate la proprietà `element.matrix`

Esempio
L'esempio seguente allunga la selezione in base a un fattore 2 nella direzione x:

```
fl.getDocumentDOM().transformSelection(2.0, 0.0, 0.0, 1.0);
```

document.unGroup()

Disponibilità
Flash MX 2004.

Uso
`document.unGroup()`

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; separa gli elementi della selezione corrente.

Esempio
L'esempio seguente separa gli elementi presenti nella selezione corrente:

```
fl.getDocumentDOM().unGroup();
```

Consultate anche
`document.group()`

document.union()

Disponibilità
Flash 8.

Uso
`document.union()`

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se l'esito è positivo; `false` in caso contrario.

Descrizione
Metodo; combina tutte le forme selezionate in un oggetto di disegno.

Esempio
L'esempio seguente combina tutte le forme selezionate in un oggetto di disegno:

```
fl.getDocumentDOM().union();
```

Consultate anche
`document.crop()`, `document.deleteEnvelope()`, `document.intersect()`, `document.punch()`, `shape.isDrawingObject`

document.unlockAllElements()

Disponibilità
Flash MX 2004.

Uso
```
document.unlockAllElements()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; sblocca tutti gli elementi bloccati presenti nel fotogramma selezionato.

Esempio
L'esempio seguente sblocca tutti gli oggetti bloccati presenti nel fotogramma corrente:

```
fl.getDocumentDOM().unlockAllElements();
```

Consultate anche
`element.locked`

document.viewMatrix

Disponibilità
Flash MX 2004.

Uso
`document.viewMatrix`

Descrizione
Proprietà di sola lettura; un oggetto Matrix. `viewMatrix` viene utilizzata per passare dallo spazio dell'oggetto a quello del documento quando il documento è in modalità di modifica. Così come viene ricevuta dallo strumento, la posizione del mouse è relativa all'oggetto che è in corso di modifica. Consultate Oggetto Matrix.

Ad esempio, se si crea un simbolo, si fa doppio clic per modificarlo e si disegna con lo strumento PolyStar, il punto (0,0) corrisponde al punto di registrazione del simbolo. Tuttavia, poiché l'oggetto drawingLayer prevede dei valori nello spazio del documento, se disegnate una linea partendo da (0,0) mediante drawingLayer, la linea inizia nell'angolo superiore sinistro dello stage. La proprietà `viewMatrix` fornisce un modo per passare dallo spazio dell'oggetto in corso di modifica allo spazio del documento.

Esempio
L'esempio seguente ottiene il valore della proprietà `viewMatrix`:

```
var mat = fl.getDocumentDOM().viewMatrix;
```

document.width

Disponibilità
Flash MX 2004.

Uso
`document.width`

Descrizione
Proprietà; un numero intero che specifica la larghezza del documento (stage) espressa in pixel.

Esempio
L'esempio seguente imposta la larghezza dello stage su 400 pixel:

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

Consultate anche
`document.height`

document.xmlPanel()

Disponibilità
Flash MX 2004.

Uso
`document.xmlPanel(fileURI)`

Parametri
`fileURI` Una stringa, espressa nel formato file:///URI, che specifica il percorso del file XML che definisce i controlli nel pannello. È necessario specificare il percorso completo.

Restituisce
Un oggetto in cui sono definite le proprietà per tutti i controlli definiti nel file XML. Tutte le proprietà vengono restituite sotto forma di stringhe. L'oggetto restituito contiene la proprietà predefinita `"dismiss"` che contiene il valore di stringa `"accept"` o `"cancel"`.

Descrizione
Metodo; inserisce una finestra di dialogo XMLUI. Consultate `fl.xmlui`.

Esempio
L'esempio seguente carica il file Test.xml e visualizza ogni proprietà contenuta al suo interno:

```
var obj = fl.getDocumentDOM().xmlPanel(fl.configURI + "Commands/Test.xml");
for (var prop in obj) {
    fl.trace("property " + prop + " = " + obj[prop]);
}
```

document.zoomFactor

Disponibilità
Flash 8.

Uso
`document.zoomFactor`

Descrizione
Proprietà; specifica la percentuale di ingrandimento dello stage durante la fase di creazione. Il valore 1 equivale a un ingrandimento del 100%, il valore 8 equivale all'800%, il valore 0,5 equivale al 50% e così via.

Esempio
L'esempio seguente imposta un ingrandimento del 200% per lo stage:

```
fl.getDocumentDOM().zoomFactor = 2;
```

Capitolo 12: Oggetto drawingLayer

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto drawingLayer è accessibile da JavaScript come elemento secondario dell'oggetto Flash. Viene utilizzato per gli strumenti estensibili quando l'utente desidera disegnare temporaneamente durante un'operazione di trascinamento (ad esempio, la creazione di un perimetro di selezione). Dovete chiamare `drawingLayer.beginFrame()` prima di chiamare qualunque altro metodo di drawingLayer.

Riepilogo dei metodi
I seguenti metodi sono disponibili per l'oggetto drawingLayer:

Metodo	Descrizione
`drawingLayer.beginDraw()`	Attiva la modalità di disegno di Flash.
`drawingLayer.beginFrame()`	Cancella quanto disegnato in precedenza mediante drawingLayer e si prepara per altri comandi di disegno.
`drawingLayer.cubicCurveTo()`	Disegna una curva cubica a partire dal punto in cui si trova la penna, utilizzando i parametri come coordinate del segmento cubico.
`drawingLayer.curveTo()`	Disegna un segmento di curva quadratica che parte dalla posizione di disegno corrente e termina nel punto specificato.
`drawingLayer.drawPath()`	Disegna il percorso specificato.
`drawingLayer.endDraw()`	Disattiva la modalità di disegno.
`drawingLayer.endFrame()`	Segnala la fine di un gruppo di comandi di disegno.
`drawingLayer.lineTo()`	Disegna una linea dalla posizione di disegno corrente al punto (*x*,*y*).
`drawingLayer.moveTo()`	Imposta la posizione di disegno corrente.
`drawingLayer.newPath()`	Restituisce un nuovo Oggetto Path.
`drawingLayer.setColor()`	Imposta il colore dei dati disegnati successivamente.
`drawingLayer.setFill()`	Questo metodo non è disponibile.
`drawingLayer.setStroke()`	Questo metodo non è disponibile.

drawingLayer.beginDraw()

Disponibilità
Flash MX 2004.

Uso
```
drawingLayer.beginDraw([persistentDraw])
```

Parametri

`persistentDraw` Un valore booleano (opzionale). Se impostato su `true`, indica che il disegno nell'ultimo fotogramma rimane sullo stage finché non viene effettuata una nuova chiamata a `beginDraw()` o `beginFrame()`. In questo contesto, il termine *frame (fotogramma)* si riferisce al punto in cui si inizia e si finisce di disegnare e non all'elemento della linea temporale. Ad esempio, quando si disegna un rettangolo, è possibile visualizzare un'anteprima del contorno della forma mentre si trascina il mouse. Per fare in modo che l'anteprima della forma rimanga visibile anche dopo che si è rilasciato il mouse, impostate *persistentDraw* su `true`.

Restituisce

Nulla.

Descrizione

Metodo; attiva la modalità di disegno di Flash, che viene utilizzata per disegnare temporaneamente mentre il pulsante del mouse è premuto. Generalmente si utilizza questo metodo solo durante la creazione di oggetti estensibili.

Esempio

L'esempio seguente attiva la modalità di disegno di Flash:

```
fl.drawingLayer.beginDraw();
```

drawingLayer.beginFrame()

Disponibilità

Flash MX 2004.

Uso

```
drawingLayer.beginFrame()
```

Parametri

Nessuno.

Restituisce

Nulla.

Descrizione

Metodo; cancella quanto disegnato in precedenza mediante drawingLayer e si prepara per altri comandi di disegno. Deve essere chiamato dopo `drawingLayer.beginDraw()`. Tutto quanto è disegnato tra `drawingLayer.beginFrame()` e `drawingLayer.endFrame()` rimane sullo stage fino a quando non vengono chiamati i metodi `beginFrame()` e `endFrame()`. In questo contesto, il termine *frame (fotogramma)* si riferisce al punto in cui si inizia e si finisce di disegnare e non all'elemento della linea temporale. Generalmente si utilizza questo metodo solo durante la creazione di oggetti estensibili. Consultate `drawingLayer.beginDraw()`.

drawingLayer.cubicCurveTo()

Disponibilità
Flash MX 2004.

Uso
```
drawingLayer.cubicCurveTo(x1Ctrl, y1Ctrl, x2Ctl, y2Ctl, xEnd, yEnd)
```

Parametri
x1Ctl Un valore a virgola mobile che corrisponde alla posizione *x* del primo punto di controllo.

y1Ctl Un valore a virgola mobile che corrisponde alla posizione *y* del primo punto di controllo.

x2Ctl Un valore a virgola mobile che corrisponde alla posizione *x* del punto di controllo centrale.

y2Ctl Un valore a virgola mobile che corrisponde alla posizione *y* del punto di controllo centrale.

xEnd Un valore a virgola mobile che corrisponde alla posizione *x* del punto di controllo finale.

yEnd Un valore a virgola mobile che corrisponde alla posizione *y* del punto di controllo finale.

Restituisce
Nulla.

Descrizione
Metodo; disegna una curva cubica a partire dal punto in cui si trova la penna utilizzando i parametri come coordinate del segmento cubico. Generalmente si utilizza questo metodo solo durante la creazione di oggetti estensibili.

Esempio
L'esempio seguente disegna una curva cubica utilizzando i punti di controllo specificati:

```
fl.drawingLayer.cubicCurveTo(0, 0, 1, 1, 2, 0);
```

drawingLayer.curveTo()

Disponibilità
Flash MX 2004.

Uso
```
drawingLayer.curveTo(xCtl, yCtl, xEnd, yEnd)
```

Parametri
xCtl Un valore a virgola mobile che corrisponde alla posizione *x* del punto di controllo.

yCtl Un valore a virgola mobile che corrisponde alla posizione *y* del punto di controllo.

xEnd Un valore a virgola mobile che corrisponde alla posizione *x* del punto di controllo finale.

yEnd Un valore a virgola mobile che corrisponde alla posizione *y* del punto di controllo finale.

Restituisce
Nulla.

Descrizione
Metodo; disegna un segmento di curva quadratica che parte dalla posizione di disegno corrente e termina nel punto specificato. Generalmente si utilizza questo metodo solo durante la creazione di oggetti estensibili.

Esempio
L'esempio seguente disegna una curva quadratica utilizzando i punti di controllo specificati:

```
fl.drawingLayer.curveTo(0, 0, 2, 0);
```

drawingLayer.drawPath()

Disponibilità
Flash MX 2004.

Uso
```
drawingLayer.drawPath(path)
```

Parametri
path Un Oggetto Path da disegnare.

Restituisce
Nulla.

Descrizione
Metodo; disegna il percorso specificato dal parametro *path*. Generalmente si utilizza questo metodo solo durante la creazione di oggetti estensibili.

Esempio
L'esempio seguente disegna un tracciato specificato dall'oggetto Path di nome `gamePath`:

```
fl.drawingLayer.drawPath(gamePath);
```

drawingLayer.endDraw()

Disponibilità
Flash MX 2004.

Uso
```
drawingLayer.endDraw()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; esce dalla modalità di disegno. La modalità di disegno viene utilizzata per disegnare temporaneamente mentre il pulsante del mouse è premuto. Generalmente si utilizza questo metodo solo durante la creazione di oggetti estensibili.

Esempio
L'esempio seguente disattiva la modalità di disegno:

```
fl.drawingLayer.endDraw();
```

drawingLayer.endFrame()

Disponibilità
Flash MX 2004.

Uso
```
drawingLayer.endFrame()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; segnala la fine di un gruppo di comandi di disegno. Un gruppo di comandi di disegno fa riferimento a tutto quanto è stato disegnato tra `drawingLayer.beginFrame()` e `drawingLayer.endFrame()`. La successiva chiamata a `drawingLayer.beginFrame()` cancella tutto quanto è stato disegnato in questo gruppo di comandi. Generalmente si utilizza questo metodo solo durante la creazione di oggetti estensibili.

drawingLayer.lineTo()

Disponibilità
Flash MX 2004.

Uso
```
drawingLayer.lineTo(x, y)
```

Parametri
x Un valore a virgola mobile che corrisponde alla coordinata *x* del punto finale della linea da disegnare.

y Un valore a virgola mobile che corrisponde alla coordinata *y* del punto finale della linea da disegnare.

Restituisce
Nulla.

Descrizione
Metodo; disegna una linea dalla posizione di disegno corrente fino al punto (*x,y*). Generalmente si utilizza questo metodo solo durante la creazione di oggetti estensibili.

Esempio
L'esempio seguente disegna una linea partendo dalla posizione di disegno corrente e fino al punto (20,30):

```
fl.drawingLayer.lineTo(20, 30);
```

drawingLayer.moveTo()

Disponibilità
Flash MX 2004.

Uso
```
drawingLayer.moveTo(x, y)
```

Parametri
x Un valore a virgola mobile che specifica la coordinata *x* del punto da cui si inizia a disegnare.

y Un valore a virgola mobile che specifica la coordinata *y* del punto da cui si inizia a disegnare.

Restituisce
Nulla.

Descrizione
Metodo; imposta la posizione di disegno corrente. Generalmente si utilizza questo metodo solo durante la creazione di oggetti estensibili.

Esempio
L'esempio seguente imposta la posizione di disegno corrente al punto (10,15):

```
fl.drawingLayer.moveTo(10, 15);
```

drawingLayer.newPath()

Disponibilità
Flash MX 2004.

Uso
```
drawingLayer.newPath()
```

Parametri
Nessuno.

Restituisce
Un oggetto Path.

Descrizione
Metodo; restituisce un nuovo oggetto Path. Generalmente si utilizza questo metodo solo durante la creazione di oggetti estensibili. Consultate Oggetto Path.

Esempio
L'esempio seguente restituisce un oggetto Path:

```
fl.drawingLayer.newPath();
```

drawingLayer.setColor()

Disponibilità
Flash MX 2004.

Uso
`drawingLayer.setColor(color)`

Parametri
`color` Il colore dei dati disegnati successivamente, in uno dei seguenti formati:

- Una stringa nel formato `"#RRGGBB"` o `"#RRGGBBAA"`
- Un numero esadecimale nel formato `0xRRGGBB`
- Un numero intero che rappresenta l'equivalente decimale di un numero esadecimale.

Restituisce
Nulla.

Descrizione
Metodo; imposta il colore dei dati disegnati successivamente. Si applica solo ai dati di tipo persistente. Per utilizzare questo metodo, il parametro passato a `drawingLayer.beginDraw()` deve essere impostato su `true`. Generalmente si utilizza questo metodo solo durante la creazione di oggetti estensibili. Consultate `drawingLayer.beginDraw()`.

Esempio
L'esempio seguente disegna una linea rossa sullo stage:

```
fl.drawingLayer.beginDraw( true );
fl.drawingLayer.beginFrame();
fl.drawingLayer.setColor( "#ff0000" );
fl.drawingLayer.moveTo(0,0);
fl.drawingLayer.lineTo(100,100);
fl.drawingLayer.endFrame();
fl.drawingLayer.endDraw();
```

drawingLayer.setFill()

Questo metodo non è disponibile.

drawingLayer.setStroke()

Questo metodo non è disponibile.

Capitolo 13: Oggetto Edge

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto Edge rappresenta un bordo di una forma sullo stage.

Riepilogo dei metodi
I seguenti metodi sono disponibili per l'oggetto Edge:

Metodo	Descrizione
`edge.getControl()`	Ottiene un oggetto point impostato sulla posizione del punto di controllo specificato del bordo.
`edge.getHalfEdge()`	Restituisce un Oggetto HalfEdge.
`edge.setControl()`	Imposta la posizione del punto di controllo del bordo.
`edge.splitEdge()`	Suddivide il bordo in due parti.

Riepilogo delle proprietà
Le seguenti proprietà sono disponibili per l'oggetto Edge:

Proprietà	Descrizione
`edge.cubicSegmentIndex`	Un numero intero che specifica il valore dell'indice di un segmento cubico del bordo.
`edge.id`	Sola lettura; un numero intero che rappresenta un identificatore univoco per il bordo.
`edge.isLine`	Sola lettura; un numero intero con valore 0 o 1.
`edge.stroke`	Un Oggetto Stroke.

edge.cubicSegmentIndex

Disponibilità
Flash CS4 Professional.

Uso
```
edge.cubicSegmentIndex
```

Descrizione
Proprietà di sola lettura; un numero intero che specifica il valore dell'indice di un segmento cubico del bordo (consultate shape.getCubicSegmentPoints()).

Esempio
Il codice seguente restituisce i valori degli indici di tutti i segmenti cubici per il bordo specificato:

```
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()

Disponibilità
Flash MX 2004.

Uso
`edge.getControl(i)`

Parametri
i Un numero intero che specifica quale punto di controllo del bordo deve essere restituito. Specificate 0 per il primo punto di controllo, 1 per quello centrale o 2 per quello finale. Se la proprietà edge.isLine è `true`, il punto di controllo centrale viene impostato sul punto centrale del segmento che unisce i punti di controllo iniziale e finale.

Restituisce
Il punto di controllo specificato.

Descrizione
Metodo; ottiene un oggetto point impostato sulla posizione del punto di controllo specificato del bordo.

Esempio
L'esempio seguente memorizza nella variabile `pt` il primo punto di controllo della forma specificata:

```
var shape = fl.getDocumentDOM().selection[0];
var pt = shape.edges[0].getControl(0);
```

edge.getHalfEdge()

Disponibilità
Flash MX 2004.

Uso
`edge.getHalfEdge(index)`

Parametri
index Un numero intero che specifica quale mezzo bordo deve essere restituito. Il valore di *index* deve essere 0 per il primo mezzo bordo oppure 1 per il secondo.

Restituisce
Un oggetto HalfEdge.

Descrizione
Metodo; restituisce un Oggetto HalfEdge.

Esempio
L'esempio seguente memorizza nelle variabili `hEdge0` e `hEdge1` i mezzi bordi del bordo specificato:

```
var shape = fl.getDocumentDOM().selection[0];
var edge = shape.edges[0];
var hEdge0 = edge.getHalfEdge(0);
var hEdge1 = edge.getHalfEdge(1);
```

edge.id

Disponibilità
Flash MX 2004.

Uso
`edge.id`

Descrizione
Proprietà di sola lettura; un numero intero che rappresenta un identificatore univoco per il bordo.

Esempio
L'esempio seguente memorizza un identificatore unico per il bordo specificato nella variabile `my_shape_id`:

```
var shape = fl.getDocumentDOM().selection[0];
var my_shape_id = shape.edges[0].id;
```

edge.isLine

Disponibilità
Flash MX 2004.

Uso
`edge.isLine`

Descrizione
Proprietà di sola lettura; un numero intero con valore 0 o 1. Il valore 1 indica che il bordo è una linea retta. In tal caso, il punto di controllo centrale divide a metà la linea che unisce i due punti agli estremi.

Esempio
L'esempio seguente determina se il bordo specificato è una linea retta, e mostra il valore 1 (se è una linea retta) o 0 (se non è una linea retta) nel pannello Output:

```
var shape = fl.getDocumentDOM().selection[0];
fl.trace(shape.edges[0].isLine);
```

edge.setControl()

Disponibilità
Flash MX 2004.

Uso
```
edge.setControl(index, x, y)
```

Parametri
index Un numero intero che specifica il punto di controllo da impostare. Utilizzate il valore 0, 1 o 2 per specificare rispettivamente il punto iniziale, centrale o finale.

x Un valore a virgola mobile che specifica la posizione orizzontale del punto di controllo. Se lo stage è in modalità Modifica o Modifica in posizione, la coordinata del punto è relativa all'oggetto che si sta modificando. Altrimenti, è relativa allo stage.

y Un valore a virgola mobile che specifica la posizione verticale del punto di controllo. Se lo stage è in modalità Modifica o Modifica in posizione, la coordinata del punto è relativa all'oggetto che si sta modificando. Altrimenti, è relativa allo stage.

Restituisce
Nulla.

Descrizione
Metodo; imposta la posizione del punto di controllo del bordo. È necessario chiamare `shape.beginEdit()` prima di utilizzare questo metodo. Consultate `shape.beginEdit()`.

Esempio
L'esempio seguente imposta le coordinate (0,1) per il punto di controllo iniziale del bordo specificato:

```
x = 0; y = 1;
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit();
shape.edges[0].setControl(0, x, y);
shape.endEdit();
```

edge.splitEdge()

Disponibilità
Flash MX 2004.

Uso
```
edge.splitEdge(t)
```

Parametri

t Un valore a virgola mobile compreso tra 0 e 1 che specifica i punti dove dividere il bordo. Un valore 0 rappresenta un punto finale, mentre 1 rappresenta l'altro. Ad esempio, se si passa il valore 0,5 e il bordo è una linea retta, il bordo viene diviso esattamente nel centro. Se il bordo è una curva, il valore 0,5 rappresenta il centro parametrico della curva.

Restituisce

Nulla.

Descrizione

Metodo; suddivide il bordo in due parti. È necessario chiamare `shape.beginEdit()` prima di utilizzare questo metodo.

Esempio

L'esempio seguente divide a metà il bordo specificato:

```
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit()
shape.edges[0].splitEdge( 0.5 );
shape.endEdit()
```

edge.stroke

Disponibilità

Flash CS4 Professional.

Uso

```
edge.stroke
```

Descrizione

Proprietà; un Oggetto Stroke.

Esempio

L'esempio seguente visualizza il colore del tratto del primo bordo dell'oggetto selezionato:

```
var shape = fl.getDocumentDOM().selection[0];
fl.trace(shape.edges[0].stroke.color);
```

Capitolo 14: Oggetto Element

Disponibilità

Flash MX 2004.

Descrizione

Tutti gli elementi che compaiono sullo stage appartengono al tipo Element. L'esempio di codice seguente consente di selezionare un elemento:

```
var el = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];
```

Riepilogo dei metodi

I seguenti metodi sono disponibili per l'oggetto Element:

Metodo	Descrizione
`element.getPersistentData()`	Recupera il valore dei dati specificati dal parametro ***name***.
`element.getTransformationPoint()`	Ottiene il valore del punto di trasformazione dell'elemento specificato.
`element.hasPersistentData()`	Determina se i dati indicati sono stati associati all'elemento specificato.
`element.removePersistentData()`	Rimuove gli eventuali dati persistenti con il nome specificato che sono stati associati all'oggetto.
`element.setPersistentData()`	Memorizza i dati con un elemento.
`element.setTransformationPoint()`	Imposta la posizione del punto di trasformazione dell'elemento.

Riepilogo delle proprietà

Le seguenti proprietà sono disponibili per l'oggetto Element:

Proprietà	Descrizione
`element.depth`	Di sola lettura; un numero intero con un valore superiore a 0 per la profondità dell'oggetto visualizzato.
`element.elementType`	Di sola lettura; una stringa che rappresenta il tipo dell'elemento specificato.
`element.height`	Un valore float che specifica l'altezza dell'elemento, espressa in pixel.
`element.layer`	Sola lettura: rappresenta l'Oggetto Layer sul quale si trova l'elemento.
`element.left`	Di sola lettura; un valore float che rappresenta il lato sinistro dell'elemento.
`element.locked`	Un valore booleano: `true` se l'elemento è bloccato, `false` in caso contrario.
`element.matrix`	Un Oggetto Matrix. La matrice ha le proprietà `a`, `b`, `c`, `d`, `tx` e `ty`. Le proprietà `a`, `b`, `c` e `d` sono numeri a virgola mobile; `tx` e `ty` sono coordinate.
`element.name`	Una stringa che specifica il nome dell'elemento, normalmente definito nome dell'istanza.
`element.rotation`	Un numero intero o float compreso tra -180 e 180 che specifica la rotazione oraria dell'oggetto in gradi.
`element.scaleX`	Un valore float che specifica il valore della scala *x* di simboli, oggetti di disegno e rettangoli e ovali di base.

Proprietà	Descrizione
element.scaleY	Un valore float che specifica il valore della scala *y* di simboli, oggetti di disegno e rettangoli e ovali di base.
element.selected	Un valore booleano che specifica se l'elemento è selezionato o non selezionato.
element.skewX	Un valore float compreso tra -180 e 180 che specifica il valore dell'inclinazione *x* di simboli, oggetti di disegno e rettangoli e ovali di base.
element.skewY	Un valore float compreso tra -180 e 180 che specifica il valore dell'inclinazione *y* di simboli, oggetti di disegno e rettangoli e ovali di base.
element.top	Di sola lettura; lato superiore dell'elemento.
element.transformX	Un numero a virgola mobile che specifica il valore *x* del punto di trasformazione dell'elemento selezionato all'interno del sistema di coordinate dell'elemento principale.
element.transformY	Un numero a virgola mobile che specifica il valore *y* del punto di trasformazione dell'elemento selezionato all'interno del sistema di coordinate dell'elemento principale.
element.width	Un valore float che specifica la larghezza dell'elemento, espressa in pixel.
element.x	Un valore float che specifica il valore *x* del punto di registrazione dell'elemento selezionato.
element.y	Un valore float che specifica il valore *y* del punto di registrazione dell'elemento selezionato.

element.depth

Disponibilità

Flash MX 2004.

Uso

```
element.depth
```

Descrizione

Proprietà di sola lettura; un numero intero con un valore superiore a 0 per la profondità dell'oggetto visualizzato. L'ordine di disegno degli oggetti sullo stage specifica quale di essi si trova sopra tutti gli altri. L'ordine degli oggetti può essere gestito anche mediante l'opzione di menu Elabora > Disponi.

Esempio

L'esempio seguente visualizza la profondità dell'elemento specificato nel pannello Output:

```
// Select an object and run this script.
fl.trace("Depth of selected object: " + fl.getDocumentDOM().selection[0].depth);
```

Consultate l'esempio per element.elementType.

element.elementType

Disponibilità

Flash MX 2004.

Uso

```
element.elementType
```

Descrizione

Proprietà di sola lettura; una stringa che rappresenta il tipo dell'elemento specificato. Il valore è uno dei seguenti: `"shape"`, `"text"`, `"instance"` o `"shapeObj"`. Il tipo `"shapeObj"` viene creato nel caso di uno strumento estensibile.

Esempio

L'esempio seguente memorizza il tipo del primo elemento nella variabile `eType`:

```
// 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
```

L'esempio seguente visualizza diverse proprietà per tutti gli elementi presenti sul livello o sul fotogramma corrente:

```
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()

Disponibilità

Flash MX 2004.

Uso

```
element.getPersistentData(name)
```

Parametri

`name` Una stringa che identifica i dati da restituire.

Restituisce

I dati specificati dal parametro *name* oppure 0 se non esistono dati.

Descrizione

Metodo; recupera il valore dei dati specificati dal parametro *name*. Il tipo di dati dipende da quali dati sono stati memorizzati (consultate `element.setPersistentData()`). Solo i simboli e le bitmap supportano i dati persistenti.

Esempio

L'esempio seguente imposta e ottiene i dati per l'elemento specificato e ne visualizza il valore nel pannello Output, quindi rimuove i dati:

```
// 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()

Disponibilità

Flash CS3 Professional.

Uso

`element.getTransformationPoint()`

Parametri

Nessuno.

Restituisce

Un punto (ad esempio `{x:10,y:20}`, dove `x` e `y` sono numeri a virgola mobile) che specifica la posizione del punto di trasformazione (detto anche *punto di origine* o *punto zero*) all'interno del sistema di coordinate dell'elemento.

Descrizione

Metodo; ottiene il valore del punto di trasformazione dell'elemento specificato.

I punti di trasformazione sono relativi a posizioni differenti, a seconda del tipo di elemento selezionato. Per ulteriori informazioni, consultate `element.setTransformationPoint()`.

Esempio

L'esempio che segue ottiene il punto di trasformazione per il terzo elemento del nono fotogramma al primo livello del documento. La proprietà `transPoint.x` fornisce la coordinata *x* del punto di trasformazione, mentre la proprietà `transPoint.y` fornisce la coordinata *y*.

```
var transPoint =
fl.getDocumentDOM().getTimeline().layers[0].frames[8].elements[2].getTransformationPoint();
```

Consultate anche

`document.getTransformationPoint()`, `element.setTransformationPoint()`, `element.transformX`, `element.transformY`

element.hasPersistentData()

Disponibilità

Flash MX 2004.

Uso
```
element.hasPersistentData(name)
```

Parametri
`name` Una stringa che specifica il nome dei dati da verificare.

Restituisce
Un valore booleano: `true` se i dati specificati sono associati all'oggetto; `false` in caso contrario.

Descrizione
Metodo; determina se i dati specificati sono stati associati all'elemento specificato. Solo i simboli e le bitmap supportano i dati persistenti.

Esempio
Consultate `element.getPersistentData()`.

element.height

Disponibilità
Flash MX 2004.

Uso
```
element.height
```

Descrizione
Proprietà; un valore float che specifica l'altezza dell'elemento, espressa in pixel.

Non utilizzare questa proprietà per ridimensionare un campo di testo. Selezionate il campo di testo e usate `document.setTextRectangle()`. Se si utilizza questa proprietà con un campo di testo, il testo viene modificato in scala.

Esempio
L'esempio seguente imposta l'altezza dell'elemento specificato su 100 pixel:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].height = 100;
```

element.layer

Disponibilità
Flash 8.

Uso
```
element.layer
```

Descrizione
Proprietà di sola lettura: rappresenta l'Oggetto Layer sul quale si trova l'elemento.

Esempio
L'esempio seguente memorizza l'oggetto Layer che contiene l'elemento nella variabile `theLayer`:

```
var theLayer = element.layer;
```

element.left

Disponibilità
Flash MX 2004.

Uso
`element.left`

Descrizione
Proprietà di sola lettura; un valore float che rappresenta il lato sinistro dell'elemento. Il valore di `element.left` è relativo alla parte superiore sinistra dello stage per gli elementi che si trovano in una scena, oppure al punto di registrazione (detto anche *punto di origine* o *punto zero*) del simbolo, se l'elemento è memorizzato all'interno di un simbolo. Usate `document.setSelectionBounds()` o `document.moveSelectionBy()` per impostare questa proprietà.

Esempio
L'esempio seguente illustra il modo in cui il valore della proprietà viene modificato quando viene spostato un elemento:

```
// 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);
```

Consultate l'esempio per `element.elementType`.

element.locked

Disponibilità
Flash MX 2004.

Uso
`element.locked`

Descrizione
Proprietà; un valore booleano: `true` se l'elemento è bloccato, `false` in caso contrario. Se il valore di `element.elementType` è `"shape"`, questa proprietà viene ignorata.

Esempio

L'esempio seguente blocca il primo elemento presente nel primo fotogramma del primo livello:

```
// Similar to Modify > Arrange > Lock:
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].locked = true;
```

element.matrix

Disponibilità

Flash MX 2004.

Uso

```
element.matrix
```

Descrizione

Proprietà; un oggetto Matrix. Una matrice ha le proprietà `a`, `b`, `c`, `d`, `tx` e `ty`. Le proprietà `a`, `b`, `c` e `d` sono valori a virgola mobile; le proprietà `tx` e `ty` sono coordinate. Consultate Oggetto Matrix.

Esempio

L'esempio seguente sposta l'elemento specificato di 10 pixel per *x* e 20 pixel per *y*:

```
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

Disponibilità

Flash MX 2004.

Uso

```
element.name
```

Descrizione

Proprietà; una stringa che specifica il nome dell'elemento, normalmente definito nome dell'istanza. Se il valore di `element.elementType` è `"shape"`, questa proprietà viene ignorata. Consultate `element.elementType`.

Esempio

L'esempio seguente imposta il nome dell'istanza del primo elemento del fotogramma 1 del primo livello `"clip_mc"`:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].name = "clip_mc";
```

Consultate l'esempio per `element.elementType`.

element.removePersistentData()

Disponibilità
Flash MX 2004.

Uso
```
element.removePersistentData(name)
```

Parametri
`name` Una stringa che specifica il nome dei dati da rimuovere.

Restituisce
Nulla.

Descrizione
Metodo; rimuove gli eventuali dati persistenti con il nome specificato che sono stati associati all'oggetto. Solo i simboli e le bitmap supportano i dati persistenti.

Esempio
Consultate `element.getPersistentData()`.

element.rotation

Disponibilità
Flash CS3 Professional.

Uso
```
element.rotation
```

Descrizione
Proprietà; un numero intero o float compreso tra -180 e 180 che specifica la rotazione in senso orario dell'oggetto in gradi.

Esempio
L'esempio seguente imposta la rotazione dell'elemento attualmente selezionato su 45 gradi:

```
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

Disponibilità
Flash CS3 Professional.

Uso

```
element.scaleX
```

Descrizione

Proprietà; un valore float che specifica il valore della scala *x* di simboli, oggetti di disegno e rettangoli e ovali di base. Il valore 1 indica una scala pari al 100 percento.

Esempio

L'esempio che segue imposta il valore della scala *x* per la selezione corrente su 2 (raddoppiandone il valore):

```
var element = fl.getDocumentDOM().selection[0];
element.scaleX = 2;
```

Consultate anche

element.scaleY

element.scaleY

Disponibilità

Flash CS3 Professional.

Uso

```
element.scaleY
```

Descrizione

Proprietà; un valore float che specifica il valore della scala *y* di simboli, oggetti di disegno e rettangoli e ovali di base. Il valore 1 indica una scala pari al 100 percento.

Esempio

L'esempio che segue imposta il valore della scala *y* per la selezione corrente su 2 (raddoppiandone il valore):

```
var element = fl.getDocumentDOM().selection[0];
element.scaleY = 2;
```

Consultate anche

element.scaleX

element.selected

Disponibilità

Flash 8.

Uso

```
element.selected
```

Descrizione
Proprietà; un valore booleano che specifica se il pulsante è selezionato (`true`) o non selezionato (`false`).

Esempio
L'esempio seguente seleziona l'elemento:

```
element.selected = true;
```

element.setPersistentData()

Disponibilità
Flash MX 2004.

Uso
```
element.setPersistentData(name, type, value)
```

Parametri
`name` Una stringa che specifica il nome da associare ai dati. Questo nome viene utilizzato per recuperare i dati.

`type` Una stringa che definisce il tipo dei dati. I valori validi sono `"integer"`, `"integerArray"`, `"double"`, `"doubleArray"`, `"string"` e `"byteArray"`.

`value` Specifica il valore da associare all'oggetto. Il tipo di dati di *value* dipende dal valore del parametro *type*. Il valore specificato deve essere appropriato per il tipo di dati specificato dal parametro *type*.

Restituisce
Nulla.

Descrizione
Metodo; memorizza i dati con un elemento. I dati sono disponibili quando viene riaperto il file FLA che contiene l'elemento. Solo i simboli e le bitmap supportano i dati persistenti.

Esempio
Consultate `element.getPersistentData()`.

element.setTransformationPoint()

Disponibilità
Flash CS3 Professional.

Uso
```
element.setTransformationPoint(transformationPoint)
```

Parametri

`transformationPoint` Un punto (ad esempio `{x:10,y:20}` dove `x` e `y` sono numeri a virgola mobile) che specifica i valori per il punto di trasformazione di un elemento o gruppo.

- Forme: *transformationPoint* viene impostato rispetto al documento; 0,0 corrisponde all'angolo superiore sinistro dello stage.
- Simboli: *transformationPoint* viene impostato rispetto al punto di registrazione del simbolo; 0,0 corrisponde al punto di registrazione.
- Testo: *transformationPoint* viene impostato rispetto al campo di testo; 0,0 corrisponde all'angolo superiore sinistro del campo di testo.
- Bitmap/video: *transformationPoint* viene impostato rispetto al bitmap/video; 0,0 corrisponde all'angolo superiore sinistro della bitmap o del video.
- Oggetti di disegno, ovali e rettangoli di base e gruppi: *transformationPoint* viene impostato rispetto al centro dell'elemento o gruppo; 0,0 corrisponde al punto centrale dell'elemento o gruppo.

Restituisce

Nulla.

Descrizione

Metodo; imposta la posizione del punto di trasformazione dell'elemento.

Questo metodo è pressoché identico a `document.setTransformationPoint()`. Le differenze tra i due metodi sono le seguenti:

- Il punto di trasformazione di oggetti di disegno, oggetti di base e gruppi viene impostato rispetto al centro dell'elemento o gruppo invece che rispetto allo stage.
- Potete impostare i punti di trasformazione degli elementi senza doverli selezionare preventivamente.

  Questo metodo sposta il punto di trasformazione ma non l'elemento. Al contrario, le proprietà `element.transformX` ed `element.transformY` spostano l'elemento.

Esempio

L'esempio seguente imposta su 100, 200 il punto di trasformazione del terzo elemento dello stage:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[2].setTransformationPoint({x:
100, y:200});
```

Consultate anche

`document.setTransformationPoint()`, `element.getTransformationPoint()`, `element.transformX`, `element.transformY`

element.skewX

Disponibilità

Flash CS3 Professional.

Uso

```
element.skewX
```

Descrizione
Proprietà; un valore float compreso tra -180 e 180 che specifica il valore dell'inclinazione *x* di simboli, oggetti di disegno e rettangoli e ovali di base.

Esempio
L'esempio che segue imposta il valore dell'inclinazione *x* della selezione corrente su 10:

```
var element = fl.getDocumentDOM().selection[0];
element.skewX = 10;
```

Consultate anche
`document.setTransformationPoint()`, `element.skewY`

element.skewY

Disponibilità
Flash CS3 Professional.

Uso
`element.skewY`

Descrizione
Proprietà; un valore float compreso tra -180 e 180 che specifica il valore dell'inclinazione *y* di simboli, oggetti di disegno e rettangoli e ovali di base.

Esempio
L'esempio che segue imposta il valore dell'inclinazione *y* della selezione corrente su 10:

```
var element = fl.getDocumentDOM().selection[0];
element.skewY = 10;
```

Consultate anche
`document.setTransformationPoint()`, `element.skewX`

element.top

Disponibilità
Flash MX 2004.

Uso
`element.top`

Descrizione

Proprietà di sola lettura; specifica il lato superiore dell'elemento. Il valore di `element.top` è relativo alla parte superiore sinistra dello stage per gli elementi che si trovano in una scena e al punto di registrazione del simbolo se l'elemento è memorizzato all'interno di un simbolo. Usate `document.setSelectionBounds()` o `document.moveSelectionBy()` per impostare questa proprietà.

Esempio

L'esempio seguente illustra il modo in cui il valore della proprietà viene modificato quando viene spostato un elemento:

```
// 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);
```

Consultate l'esempio per `element.elementType`.

element.transformX

Disponibilità

Flash CS3 Professional.

Uso

```
element.transformX
```

Descrizione

Proprietà; un numero a virgola mobile che specifica il valore *x* del punto di trasformazione dell'elemento selezionato all'interno del sistema di coordinate dell'elemento principale. Impostando questa proprietà su un nuovo valore causa lo spostamento dell'elemento. Al contrario, il metodo `element.setTransformationPoint()` sposta il punto di trasformazione ma non l'elemento.

Esempio

Consultate anche

`element.getTransformationPoint()`, `element.setTransformationPoint()`, `element.transformY`

element.transformY

Disponibilità

Flash CS3 Professional.

Uso

```
element.transformY
```

Descrizione
Proprietà; un numero a virgola mobile che specifica il valore *y* del punto di trasformazione dell'elemento selezionato all'interno del sistema di coordinate dell'elemento principale. Impostando questa proprietà su un nuovo valore causa lo spostamento dell'elemento. Al contrario, il metodo `element.setTransformationPoint()` sposta il punto di trasformazione ma non l'elemento.

Consultate anche
`element.getTransformationPoint()`, `element.setTransformationPoint()`, `element.transformX`

element.width

Disponibilità
Flash MX 2004.

Uso
```
element.width
```

Descrizione
Proprietà; un valore float che specifica la larghezza dell'elemento, espressa in pixel.

Non utilizzare questa proprietà per ridimensionare un campo di testo. Selezionate il campo di testo e usate `document.setTextRectangle()`. Se si utilizza questa proprietà con un campo di testo, il testo viene modificato in scala.

Esempio
L'esempio seguente imposta la larghezza dell'elemento specificato su 100 pixel:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].width= 100;
```

element.x

Disponibilità
Flash CS3 Professional.

Uso
```
element.x
```

Descrizione
Proprietà; un valore float che specifica il valore *x* del punto di registrazione dell'elemento selezionato.

Esempio
L'esempio seguente imposta su 100,200 il valore del punto di registrazione dell'elemento specificato:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].x= 100;
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].y= 200;
```

Consultate anche

`element.y`

element.y

Disponibilità

Flash CS3 Professional.

Uso

```
element.y
```

Descrizione

Proprietà; un valore float che specifica il valore *y* del punto di registrazione dell'elemento selezionato.

Esempio

Consultate `element.x`

Capitolo 15: Oggetto Fill

Disponibilità
Flash MX 2004.

Descrizione
Questo oggetto contiene tutte le proprietà dell'impostazione Colore di riempimento del pannello Strumenti o di una forma selezionata. Per recuperare un oggetto Fill, usate `document.getCustomFill()`.

Riepilogo delle proprietà
Le seguenti proprietà sono disponibili per l'oggetto Fill:

Proprietà	Descrizione
`fill.bitmapIsClipped`	Un valore booleano che specifica se il riempimento bitmap di una forma filtro più grande della bitmap viene ritagliato o ripetuto.
`fill.bitmapPath`	Una stringa che specifica percorso e nome del riempimento bitmap nella libreria.
`fill.color`	Una stringa, un valore esadecimale o un numero intero che rappresenta il colore di riempimento.
`fill.colorArray`	Un array di colori sotto forma di gradiente.
`fill.focalPoint`	Un numero intero che specifica l'offset orizzontale del punto focale del gradiente rispetto al punto di trasformazione.
`fill.linearRGB`	Un valore booleano che specifica se per un riempimento viene eseguito il rendering come gradiente RGB lineare o radiale.
`fill.matrix`	Un Oggetto Matrix che definisce la posizione, l'orientamento e le scale dei riempimenti a gradiente.
`fill.overflow`	Una stringa che specifica il comportamento del limite di riempimento di un gradiente.
`fill.posArray`	Un array di numeri interi, ciascuno dei quali è compreso nell'intervallo tra 0 e 255, che indica la posizione del colore corrispondente.
`fill.style`	Una stringa che contiene lo stile del riempimento.

fill.bitmapIsClipped

Disponibilità
Flash CS4 Professional.

Uso
```
fill.bitmapIsClipped
```

Descrizione
Un valore booleano che specifica se il riempimento bitmap di una forma filtro più grande della bitmap viene ritagliato (`true`) o ripetuto (`false`). Questa proprietà è disponibile solo se il valore della proprietà `fill.style` è `"bitmap"`. Se la forma è più piccola della bitmap, il valore è `false`.

Esempio

L'esempio seguente indica se il riempimento bitmap è stato ritagliato, se necessario, nel pannello Output:

```
var fill = fl.getDocumentDOM().getCustomFill();
if (fill.style == "bitmap")
    fl.trace("Fill image is clipped: " + fill.bitmapIsClipped);
```

Consultate anche

`fill.bitmapPath`

fill.bitmapPath

Disponibilità

Flash CS4 Professional.

Uso

`fill.bitmapPath`

Descrizione

Una stringa che specifica percorso e nome del riempimento bitmap nella libreria. Questa proprietà è disponibile solo se il valore della proprietà `fill.style` è `"bitmap"`.

Esempio

L'esempio seguente imposta lo stile del riempimento dell'elemento specificato con un'immagine bitmap della Libreria:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style = "bitmap";
fill.bitmapPath = "myBitmap.jpg";
fl.getDocumentDOM().setCustomFill(fill);
```

Consultate anche

`fill.bitmapIsClipped`

fill.color

Disponibilità

Flash MX 2004.

Uso

`fill.color`

Descrizione

Proprietà; il colore di riempimento in uno dei seguenti formati:

- Una stringa nel formato `"#RRGGBB"` o `"#RRGGBBAA"`
- Un numero esadecimale nel formato `0xRRGGBB`

- Un numero intero che rappresenta l'equivalente decimale di un numero esadecimale.

Esempio
L'esempio seguente imposta il colore di riempimento della selezione corrente:

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

fill.colorArray

Disponibilità
Flash MX 2004.

Uso
`fill.colorArray`

Descrizione
Proprietà; un array di colori sotto forma di gradiente, espressi in numeri interi. Questa proprietà è disponibile solo se il valore della proprietà `fill.style` è `"radialGradient"` o `"linearGradient"`. Consultate `fill.style`

Esempio
L'esempio seguente visualizza, se appropriato, l'array di colori della selezione corrente nel pannello Output:

```
var fill = fl.getDocumentDOM().getCustomFill();
if(fill.style == "linearGradient" || fill.style == "radialGradient")
    alert(fill.colorArray);
```

L'esempio seguente imposta il riempimento con il gradiente lineare specificato:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style = "linearGradient";
fill.colorArray = ["#00ff00","#ff00ff"];
fill.posArray = [0, 255];
fl.getDocumentDOM().setCustomFill(fill);
```

fill.focalPoint

Disponibilità
Flash 8.

Uso
`fill.focalPoint`

Descrizione
Proprietà; un numero intero che specifica l'offset orizzontale del punto focale del gradiente rispetto al punto di trasformazione. Ad esempio, il valore 10 colloca il punto focale a 10/255 della distanza tra il punto di trasformazione e il bordo del gradiente. Il valore -255 colloca il punto focale sul bordo sinistro del gradiente. Il valore predefinito è 0.

Questa proprietà è disponibile solo se il valore della proprietà `fill.style` è `"radialGradient"`.

Esempio

L'esempio seguente imposta il punto focale di un gradiente radiale per la selezione corrente su 100 pixel a destra del centro della forma:

```
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

Disponibilità

Flash 8.

Uso

```
fill.linearRGB
```

Descrizione

Proprietà; un valore booleano che specifica se per un riempimento viene eseguito il rendering come gradiente RGB lineare o radiale. Impostate questa proprietà su `true` per specificare un'interpolazione lineare di un gradiente oppure su `false` per specificare un'interpolazione radiale. Il valore predefinito è `false`.

Esempio

L'esempio seguente specifica che per il gradiente della selezione corrente deve essere effettuato il rendering mediante la modalità RGB lineare:

```
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

Disponibilità

Flash MX 2004.

Uso

```
fill.matrix
```

Descrizione
Proprietà; un Oggetto Matrix che definisce la posizione, l'orientamento e le scale dei riempimenti a gradiente.

Esempio
L'esempio seguente usa la proprietà `fill.matrix` per specificare un riempimento del gradiente per la selezione corrente:

```
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

Disponibilità
Flash 8.

Uso
`fill.overflow`

Descrizione
Proprietà; una stringa che specifica il comportamento del limite di riempimento di un gradiente. I valori accettabili sono `"extend"`, `"repeat"` e `"reflect"`; le stringhe non fanno distinzione tra maiuscole e minuscole. Il valore predefinito è `"extend"`.

Esempio
L'esempio seguente specifica che il comportamento del limite di riempimento per la selezione corrente deve essere `"extend"`:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.overflow = "extend";
fl.getDocumentDOM().setCustomFill(fill);
```

fill.posArray

Disponibilità
Flash MX 2004.

Uso
`fill.posArray`

Descrizione
Proprietà; un array di numeri interi, ciascuno dei quali è compreso nell'intervallo tra 0 e 255, che indica la posizione del colore corrispondente. Questa proprietà è disponibile solo se il valore della proprietà `fill.style` è `"radialGradient"` o `"linearGradient"`.

Esempio
L'esempio seguente specifica i colori da utilizzare in un gradiente lineare per la selezione corrente:

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

fill.style

Disponibilità
Flash MX 2004. Valore "bitmap" aggiunto in Flash CS4 Professional.

Uso
`fill.style`

Descrizione
Proprietà; una stringa che specifica lo stile del riempimento. I valori accettabili sono `"bitmap"`, `"solid"`, `"linearGradient"`, `"radialGradient"` e `"noFill"`.

Se questo valore è `"linearGradient"` o `"radialGradient"`, sono disponibili anche le proprietà `fill.colorArray` e `fill.posArray`. Se questo valore è `"bitmap"`, sono disponibili anche le proprietà `fill.bitmapIsClipped` e `fill.bitmapPath`.

Esempio
L'esempio seguente specifica i colori da utilizzare in un gradiente lineare per la selezione corrente:

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

Capitolo 16: Oggetto Filter

Disponibilità

Flash 8.

Descrizione

Questo oggetto contiene tutte le proprietà per tutti i filtri. La proprietà `filter.name` specifica il tipo di filtro e determina le proprietà applicabili a ogni filtro. Consultate `filter.name`.

Per restituire l'elenco dei filtri per uno o più oggetti, utilizzate `document.getFilters()`. Per applicare i filtri a uno o più oggetti, utilizzate `document.setFilters()`. Consultate `document.getFilters()` e `document.setFilters()`.

Riepilogo delle proprietà

Con l'oggetto Filter è possibile utilizzare le proprietà seguenti:

Proprietà	Descrizione
`filter.angle`	Un valore float che specifica l'angolo dell'ombra o del colore di evidenziazione espresso in gradi.
`filter.blurX`	Un valore float che specifica la quantità di sfocatura nella direzione x, espressa in pixel.
`filter.blurY`	Un valore float che specifica la quantità di sfocatura nella direzione y.
`filter.brightness`	Un valore float che specifica la luminosità del filtro.
`filter.color`	Una stringa, un valore esadecimale o un numero intero che rappresenta il colore del filtro.
`filter.contrast`	Un valore float che specifica il valore di contrasto del filtro.
`filter.distance`	Un valore float che specifica la distanza tra l'effetto del filtro e un oggetto, espressa in pixel.
`filter.enabled`	Un valore booleano che specifica se il filtro specificato è abilitato.
`filter.hideObject`	Un valore booleano che specifica se l'immagine di origine è nascosta.
`filter.highlightColor`	Una stringa, un valore esadecimale o un numero intero che rappresenta il colore di evidenziazione.
`filter.hue`	Un valore float che specifica la tonalità del filtro.
`filter.inner`	Un valore booleano che specifica se l'ombra è interna.
`filter.knockout`	Un valore booleano che specifica se il filtro è di foratura.
`filter.name`	Di sola lettura; una stringa che specifica il tipo di filtro.
`filter.quality`	Una stringa che specifica la qualità della sfocatura.
`filter.saturation`	Un valore float che specifica il valore di saturazione del filtro.
`filter.shadowColor`	Una stringa, un valore esadecimale o un numero intero che rappresenta il colore dell'ombra.
`filter.strength`	Un numero intero che specifica la percentuale di intensità del filtro.
`filter.type`	Una stringa che specifica il tipo di filtro a smusso o a bagliore.

filter.angle

Disponibilità
Flash 8.

Uso
`filter.angle`

Descrizione
Proprietà; un valore float che specifica l'angolo dell'ombra o del colore di evidenziazione, espresso in gradi. I valori accettabili sono compresi tra 0 e 360. Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"bevelFilter"`, `"dropShadowFilter"`, `"gradientBevelFilter"` o `"gradientGlowFilter"`.

Esempio
L'esempio seguente imposta l'angolo su 120 per i filtri a smusso negli oggetti selezionati:

```
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);
```

Consultate anche
`document.setFilterProperty()`

filter.blurX

Disponibilità
Flash 8.

Uso
`filter.blurX`

Descrizione
Proprietà; un valore float che specifica la quantità di sfocatura nella direzione *x*, espressa in pixel. I valori accettabili sono compresi tra 0 e 255. Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"bevelFilter"`, `"blurFilter"`, `"dropShadowFilter"`, `"glowFilter"`, `"gradientBevelFilter"` o `"gradientGlowFilter"`.

Esempio
L'esempio seguente imposta il valore `blurX` su 30 e il valore `blurY` su 20 per i filtri di sfocatura negli oggetti selezionati:

```
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);
```

Consultate anche

`document.setFilterProperty()`, `filter.blurY`

filter.blurY

Disponibilità

Flash 8.

Uso

`filter.blurY`

Descrizione

Proprietà; un valore float che specifica la quantità di sfocatura nella direzione *y*, espressa in pixel. I valori accettabili sono compresi tra 0 e 255. Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"bevelFilter"`, `"blurFilter"`, `"dropShadowFilter"`, `"glowFilter"`, `"gradientBevelFilter"` o `"gradientGlowFilter"`.

Esempio

Consultate `filter.blurX`.

Consultate anche

`document.setFilterProperty()`, `filter.blurX`

filter.brightness

Disponibilità

Flash 8.

Uso

`filter.brightness`

Descrizione

Proprietà; un valore float che specifica la luminosità del filtro. I valori accettabili sono compresi tra -100 e 100. Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"adjustColorFilter"`.

Esempio

L'esempio seguente imposta la luminosità su 30,5 per i filtri di regolazione del colore negli oggetti selezionati:

```
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

Disponibilità

Flash 8.

Uso

`filter.color`

Descrizione

Proprietà; il colore del filtro in uno dei seguenti formati:

- Una stringa nel formato `"#RRGGBB"` o `"#RRGGBBAA"`
- Un numero esadecimale nel formato `0xRRGGBB`
- Un numero intero che rappresenta l'equivalente decimale di un numero esadecimale.

  Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"dropShadowFilter"` o `"glowFilter"`.

Esempio

L'esempio seguente imposta il colore su `"#ff00003e"` per i filtri ombra esterna negli oggetti selezionati:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'dropShadowFilter'){
        myFilters[i].color = '#ff00003e';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

Consultate anche

`document.setFilterProperty()`

filter.contrast

Disponibilità

Flash 8.

Uso

```
filter.contrast
```

Descrizione

Proprietà; un valore float che specifica il valore di contrasto del filtro. I valori accettabili sono compresi tra -100 e 100. Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"adjustColorFilter"`.

Esempio

L'esempio seguente imposta il valore di contrasto su -15,5 per i filtri di regolazione del colore negli oggetti selezionati:

```
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

Disponibilità

Flash 8.

Uso

```
filter.distance
```

Descrizione

Proprietà; un valore float che specifica la distanza tra l'effetto del filtro e un oggetto, espressa in pixel. I valori accettabili sono compresi tra -255 e 255. Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"bevelFilter"`, `"dropShadowFilter"`, `"gradientBevelFilter"` o `"gradientGlowFilter"`.

Esempio

L'esempio seguente imposta la distanza su 10 pixel per i filtri ombra esterna negli oggetti selezionati:

```
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);
```

Consultate anche

`document.setFilterProperty()`

filter.enabled

Disponibilità
Flash CS3 Professional.

Uso
`filter.enabled`

Descrizione
Proprietà; un valore booleano che specifica se il filtro specificato è abilitato (`true`) o disabilitato (`false`).

Esempio
L'esempio seguente disattiva i filtri colore per gli oggetti selezionati:

```
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

Disponibilità
Flash 8.

Uso
`filter.hideObject`

Descrizione
Proprietà; un valore booleano che specifica se l'immagine bitmap di origine è nascosta (`true`) o visualizzata (`false`). Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"dropShadowFilter"`.

Esempio
L'esempio seguente imposta il valore `hideObject` su `true` per i filtri ombra esterna negli oggetti selezionati:

```
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

Disponibilità
Flash 8.

Uso
`filter.highlightColor`

Descrizione
Proprietà; il colore di evidenziazione in uno dei seguenti formati:

- Una stringa nel formato `"#RRGGBB"` o `"#RRGGBBAA"`
- Un numero esadecimale nel formato `0xRRGGBB`
- Un numero intero che rappresenta l'equivalente decimale di un numero esadecimale.

  Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"bevelFilter"`.

Esempio
L'esempio seguente imposta il colore di evidenziazione su `"#ff00003e"` per i filtri a smusso negli oggetti selezionati:

```
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

Disponibilità
Flash 8.

Uso
`filter.hue`

Descrizione
Proprietà; un valore float che specifica la tonalità del filtro. I valori accettabili sono compresi tra -180 e 180. Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"adjustColorFilter"`.

Esempio
L'esempio seguente imposta la tonalità su 120 per i filtri di regolazione del colore negli oggetti selezionati:

```
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

Disponibilità
Flash 8.

Uso
`filter.inner`

Descrizione
Proprietà; un valore booleano che specifica se l'ombra è interna (`true`) o esterna (`false`). Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"dropShadowFilter"` o `"glowFilter"`.

Esempio
L'esempio seguente imposta il valore della proprietà `inner` su `true` per i filtri a bagliore negli oggetti selezionati:

```
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);
```

Consultate anche
`document.setFilterProperty()`

filter.knockout

Disponibilità
Flash 8.

Uso
`filter.knockout`

Descrizione
Proprietà; un valore booleano che specifica se il filtro è di foratura (`true`) o meno (`false`). Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"bevelFilter"`, `"dropShadowFilter"`, `"glowFilter"`, `"gradientBevelFilter"` o `"gradientGlowFilter"`.

Esempio

L'esempio seguente imposta la proprietà `knockout` su `true` per i filtri a bagliore negli oggetti selezionati:

```
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);
```

Consultate anche

`document.setFilterProperty()`

filter.name

Disponibilità

Flash 8.

Uso

```
filter.name
```

Descrizione

Proprietà di sola lettura; una stringa che specifica il tipo di filtro. Il valore di questa proprietà determina quali altre proprietà dell'oggetto Filter sono disponibili. Il valore è uno dei seguenti: `"adjustColorFilter"`, `"bevelFilter"`, `"blurFilter"`, `"dropShadowFilter"`, `"glowFilter"`, `"gradientBevelFilter"` o `"gradientGlowFilter"`.

Esempio

L'esempio seguente visualizza i nomi dei filtri e le posizioni di indice nel pannello Output:

```
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);
```

Consultate anche

`document.getFilters()`, `document.setFilterProperty()`

filter.quality

Disponibilità

Flash 8.

Uso

```
filter.quality
```

Descrizione

Proprietà; una stringa che specifica la qualità della sfocatura. I valori accettabili sono `"low"`, `"medium"` e `"high"` (`"high"` è simile a una sfocatura gaussiana). Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"bevelFilter"`, `"blurFilter"`, `"dropShadowFilter"`, `"glowFilter"`, `"gradientGlowFilter"` o `"gradientBevelFilter"`.

Esempio

L'esempio seguente imposta la qualità della sfocatura su `"medium"` per i filtri a bagliore negli oggetti selezionati:

```
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);
```

Consultate anche

`document.setFilterProperty()`

filter.saturation

Disponibilità

Flash 8.

Uso

```
filter.saturation
```

Descrizione

Proprietà; un valore float che specifica il valore di saturazione del filtro. I valori accettabili sono compresi tra -100 e 100. Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"adjustColorFilter"`.

Esempio

L'esempio seguente imposta il valore di saturazione su -100 (scala di grigi) per i filtri di regolazione del colore negli oggetti selezionati:

```
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);
```

Consultate anche

`document.setFilterProperty()`

filter.shadowColor

Disponibilità
Flash 8.

Uso
`filter.shadowColor`

Descrizione
Proprietà; il colore dell'ombra in uno dei seguenti formati:

- Una stringa nel formato `"#RRGGBB"` o `"#RRGGBBAA"`
- Un numero esadecimale nel formato `0xRRGGBB`
- Un numero intero che rappresenta l'equivalente decimale di un numero esadecimale.

  Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"bevelFilter"`.

Esempio
L'esempio seguente imposta il colore dell'ombra su `"#ff00003e"` per i filtri a smusso negli oggetti selezionati:

```
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);
```

Consultate anche
`document.setFilterProperty()`

filter.strength

Disponibilità
Flash 8.

Uso
`filter.strength`

Descrizione
Proprietà; un numero intero che specifica la percentuale di intensità del filtro. I valori accettabili sono compresi tra 0 e 25.500. Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"bevelFilter"`, `"dropShadowFilter"`, `"glowFilter"`, `"gradientGlowFilter"` o `"gradientBevelFilter"`.

Esempio
L'esempio seguente imposta l'intensità su 50 per i filtri a bagliore negli oggetti selezionati:

```
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);
```

Consultate anche

`document.setFilterProperty()`

filter.type

Disponibilità

Flash 8.

Uso

`filter.type`

Descrizione

Proprietà; una stringa che specifica il tipo di filtro a smusso o a bagliore. I valori accettabili sono `"inner"`, `"outer"` e `"full"`. Questa proprietà viene definita per gli oggetti Filter la cui proprietà `filter.name` ha il valore `"bevelFilter"`, `"gradientGlowFilter"` o `"gradientBevelFilter"`.

Esempio

L'esempio seguente imposta il tipo su `"full"` per tutti i filtri a smusso negli oggetti selezionati:

```
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);
```

Consultate anche

`document.setFilterProperty()`

Capitolo 17: Oggetto Flash (fl)

Disponibilità

Flash MX 2004.

Descrizione

L'oggetto flash rappresenta l'applicazione Flash. Quando fate riferimento a questo oggetto, potete utilizzare `flash` o `fl`. In questa documentazione viene utilizzato `fl` in tutti gli esempi di codice.

Riepilogo dei metodi

I seguenti metodi possono essere utilizzati con l'oggetto Flash:

Metodo	Descrizione
`fl.addEventListener()`	Registra una funzione da chiamare quando viene ricevuto uno specifico evento.
`fl.browseForFileURL()`	Apre una finestra di dialogo di sistema File Apri o File Salva e consente all'utente di specificare un file da aprire o salvare.
`fl.browseForFolderURL()`	Visualizza la finestra di dialogo Sfoglia per cartelle per consentire all'utente di selezionare una cartella.
`fl.clipCopyString()`	Copia la stringa specificata negli Appunti.
`fl.closeAll()`	Chiude tutti i documenti aperti e visualizza una finestra di dialogo Salva con nome per tutti i documenti che non sono stati salvati in precedenza.
`fl.closeAllPlayerDocuments()`	Chiude tutti i file SWF aperti con Controllo > Prova filmato.
`fl.closeDocument()`	Chiude il documento specificato.
`fl.createDocument()`	Apre un nuovo documento e lo seleziona.
`fl.downloadLatestVersion()`	Scarica l'ultima versione di un file attualmente non aperto dal server Version Cue.
`fl.fileExists()`	Verifica se un file è già presente sul disco.
`fl.findDocumentDOM()`	Permette di indirizzare un file specifico tramite il suo identificatore univoco.
`fl.findDocumentIndex()`	Restituisce un array di numeri interi che rappresentano la posizione di un documento nell'array di `fl.documents`.
`fl.findObjectInDocByName()`	Espone gli elementi con nomi di istanza che corrispondono al testo specificato.
`fl.findObjectInDocByType()`	Espone gli elementi del tipo specificato in un documento.
`fl.getAppMemoryInfo()`	Restituisce un numero intero che rappresenta il numero di byte attualmente utilizzati in un'area specificata della memoria di Flash.exe.
`fl.getDocumentDOM()`	Recupera il DOM (Oggetto Document) del documento attualmente attivo.
`fl.isFontInstalled()`	Determina se un carattere specificato è installato.

Metodo	Descrizione
`fl.mapPlayerURL()`	Esegue la mappatura di un URL Unicode con carattere escape a un URL UTF-8 o MBCS.
`fl.openDocument()`	Apre un documento Flash (FLA) da modificare in una nuova finestra di documento Flash e lo rende attivo.
`fl.openScript()`	Apre uno script (JSFL, AS, ASC) o un altro file (XML, TXT) nell'editor di testo di Flash.
`fl.quit()`	Esce da Flash e richiede all'utente di salvare i documenti modificati.
`fl.reloadTools()`	Ricrea il pannello Strumenti mediante il file toolconfig.xml. Utilizzato solo quando si creano strumenti estensibili.
`fl.removeEventListener()`	Annulla la registrazione di una funzione registrata tramite `fl.addEventListener()`.
`fl.resetAS3PackagePaths()`	Azzera l'impostazione del percorso di classe globale nella finestra di dialogo Impostazioni di ActionScript 3.0 ripristinando il valore predefinito.
`fl.resetPackagePaths()`	Azzera l'impostazione del percorso di classe globale nella finestra di dialogo Impostazioni di ActionScript 2.0 ripristinando il valore predefinito.
`fl.revertDocumentToLastVersion()`	Ripristina la versione del documento specificato presente sul server Version Cue.
`fl.runScript()`	Esegue un file JavaScript.
`fl.saveAll()`	Salva tutti i documenti aperti e visualizza una finestra di dialogo Salva con nome per tutti i documenti che non sono stati salvati in precedenza.
`fl.saveAVersionOfDocument()`	Salva una versione del documento specificato sul server Version Cue.
`fl.saveDocument()`	Salva il documento specificato come documento FLA.
`fl.saveDocumentAs()`	Visualizza una finestra di dialogo Salva con nome per il documento specificato.
`fl.selectElement()`	Abilita la selezione o la modifica di un elemento.
`fl.selectTool()`	Seleziona lo strumento specificato nel pannello Strumenti.
`fl.setActiveWindow()`	Imposta la finestra attiva in modo che sia il documento specificato.
`fl.showIdleMessage()`	Consente di disattivare l'avvertenza visualizzata quando l'esecuzione di uno script richiede troppo tempo.
`fl.synchronizeDocumentWithHeadVersion()`	Sincronizza il documento specificato con la più recente versione disponibile sul server Version Cue.
`fl.trace()`	Invia una stringa di testo al pannello Output.

Riepilogo delle proprietà

Le seguenti proprietà possono essere utilizzate con l'oggetto Flash.

Proprietà	Descrizione
`fl.actionsPanel`	Sola lettura; un Oggetto actionsPanel.
`fl.as3PackagePaths`	Una stringa corrispondente all'impostazione del percorso di classe globale nella finestra di dialogo Impostazioni di ActionScript 3.0.
`fl.compilerErrors`	Sola lettura; un Oggetto compilerErrors.
`fl.componentsPanel`	Sola lettura; un Oggetto componentsPanel che rappresenta il pannello Componenti.
`fl.configDirectory`	Sola lettura; una stringa che specifica il percorso completo della cartella Configuration dell'utente locale sotto forma di percorso specifico della piattaforma.
`fl.configURI`	Sola lettura; una stringa che specifica il percorso completo della directory Configuration dell'utente locale nel formato URI file:///.
`fl.contactSensitiveSelection`	Un valore booleano che specifica se la modalità Sensibile al contatto è attiva.
`fl.createNewDocList`	Sola lettura; un array di stringhe che rappresentano i vari tipi di documento che è possibile creare.
`fl.createNewDocListType`	Sola lettura; un array di stringhe che rappresentano le estensioni di file dei vari tipi di documento che è possibile creare.
`fl.createNewTemplateList`	Sola lettura; un array di stringhe che rappresentano i vari tipi di modello che è possibile creare.
`fl.documents`	Sola lettura; un array di oggetti Document (consultate Oggetto Document) che rappresentano i documenti (file FLA) attualmente aperti per la modifica.
`fl.drawingLayer`	Sola lettura; l'Oggetto drawingLayer utilizzato dagli strumenti estensibili quando l'utente desidera disegnare temporaneamente durante un'operazione di trascinamento.
`fl.externalLibraryPath`	Una stringa che contiene un elenco di elementi presenti nel percorso della libreria esterna di ActionScript 3.0 globale, che indica la posizione dei file SWC usati come librerie di runtime condivise.
`fl.flexSDKPath`	Una stringa che specifica il percorso della cartella Flex SDK, che include tra le altre le cartelle bin, framework e lib.
`fl.libraryPath`	Una stringa che contiene un elenco di elementi presenti nel percorso della libreria esterna di ActionScript 3.0 globale, che indica la posizione dei file SWC o delle cartelle SWC contenenti i file SWC.
`fl.Math`	Sola lettura; l'Oggetto Math che fornisce i metodi per le operazioni con le matrici e i punti.
`fl.mruRecentFileList`	Sola lettura; un array dei nomi di file completi presenti nell'elenco dei file utilizzati più di recente gestito dallo strumento di creazione Flash.
`fl.mruRecentFileListType`	Sola lettura; un array dei tipi di file presenti nell'elenco dei file utilizzati più di recente gestito dallo strumento di creazione Flash.
`fl.packagePaths`	Una stringa corrispondente all'impostazione del percorso di classe globale nella finestra di dialogo Impostazioni di ActionScript 2.0.
`fl.objectDrawingMode`	Un numero intero che rappresenta la modalità Disegno oggetto attivata.
`fl.outputPanel`	Sola lettura; fate riferimento all'Oggetto outputPanel.
`fl.presetPanel`	Sola lettura; un Oggetto presetPanel.
`fl.scriptURI`	Sola lettura; una stringa, espressa come URI file:///, che rappresenta il percorso dello script JSFL attualmente in esecuzione.

Proprietà	Descrizione
`fl.sourcePath`	Una stringa che contiene un elenco di elementi presenti nel percorso di origine globale di ActionScript 3.0 che indica la posizione dei file delle classi di ActionScript.
`fl.swfPanels`	Un array degli oggetti swfPanel registrati (consultate Oggetto swfPanel).
`fl.tools`	Sola lettura; un array di oggetti Tools.
`fl.version`	Sola lettura; la stringa lunga corrispondente alla versione dello strumento di creazione Flash, compresa la piattaforma.
`fl.xmlui`	Sola lettura; un Oggetto XMLUI.

fl.actionsPanel

Disponibilità
Flash CS3 Professional.

Uso
```
fl.actionsPanel
```

Descrizione
Proprietà di sola lettura; un oggetto actionsPanel che rappresenta il pannello Azioni attualmente visualizzato. Per informazioni sull'uso di questa proprietà, consultate Oggetto actionsPanel.

fl.addEventListener()

Disponibilità
Flash CS3 Professional.

Uso
```
fl.addEventListener(eventType, callbackFunction)
```

Parametri
`eventType` Una stringa che specifica il tipo di evento da passare a questa funzione di callback. I valori accettabili sono `"documentNew"`, `"documentOpened"`, `"documentClosed"`, `"mouseMove"`, `"documentChanged"`, `"layerChanged"` e `"frameChanged"`.

Il valore `documentChanged` non implica che il contenuto di un documento sia cambiato; significa che un documento diverso è passato in primo piano. In altri termini, `fl.getDocumentDOM()` restituirà un valore diverso da quello restituito prima del verificarsi di questo evento.

`callbackFunction` Il nome della funzione da eseguire ogni volta che si verifica l'evento.

Restituisce
Nulla.

Descrizione
Metodo; registra una funzione da richiamare quando si verifica uno specifico evento.

Utilizzando questo metodo, è necessario ricordare che l'applicazione può bloccarsi o entrare in uno stato di errore se l'evento si verifica frequentemente (come può accadere per `mouseMove`) e l'esecuzione della funzione richiede molto tempo.

Esempio
Nell'esempio seguente viene visualizzato un messaggio nel pannello Output quando si chiude un documento:

```
myFunction = function () {
    fl.trace('document was closed'); }
fl.addEventListener("documentClosed", myFunction);
```

Consultate anche
`fl.removeEventListener()`

fl.as3PackagePaths

Disponibilità
Flash CS3 Professional.

Uso
```
fl.as3PackagePaths
```

Descrizione
Proprietà; una stringa corrispondente all'impostazione del percorso di classe globale nella finestra di dialogo Impostazioni di ActionScript3.0. Gli elementi nella stringa sono separati da punti e virgola. Per visualizzare o modificare le impostazioni dei percorsi di classe di ActionScript 2.0, usate `fl.packagePaths`.

Esempio
L'esempio che segue illustra la modifica delle impostazioni dei percorsi di classe di ActionScript3.0.

```
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
```

Consultate anche
`fl.resetAS3PackagePaths()`

fl.browseForFileURL()

Disponibilità
Flash MX 2004.

Uso
```
fl.browseForFileURL(browseType [, title [, previewArea]])
```

Parametri
`browseType` Una stringa che specifica il tipo di operazione di ricerca file. I valori accettabili sono `"open"`, `"select"` o `"save"`. I valori `"open"` e `"select"` aprono la finestra di dialogo di sistema File Apri. I valori sono presenti per ragioni di compatibilità con Dreamweaver. Il valore `"save"` apre una finestra di dialogo di sistema File Salva.

`title` Una stringa che specifica il titolo della finestra di dialogo File Apri o File Salva. Se si omette questo parametro, viene utilizzato il valore predefinito. Questo parametro è opzionale.

`previewArea` Un parametro opzionale che viene ignorato da Flash e Fireworks e che è presente solo per assicurare la compatibilità con Dreamweaver.

Restituisce
L'URL del file, espresso nel formato file:///URI; restituisce `null` se l'utente esce dalla finestra di dialogo selezionando Annulla.

Descrizione
Metodo; apre una finestra di dialogo di sistema File Apri o File Salva e consente all'utente di specificare un file da aprire o salvare.

Esempio
L'esempio seguente consente all'utente di selezionare un file FLA da aprire, quindi apre il file (il metodo `fl.browseForFileURL()` consente di cercare qualunque tipo di file, tuttavia `fl.openDocument()` è in grado di aprire solo i file FLA).

```
var fileURL = fl.browseForFileURL("open", "Select file");
var doc = fl.openDocument(fileURL);
```

Consultate anche
`fl.browseForFolderURL()`

fl.browseForFolderURL()

Disponibilità
Flash 8.

Uso
```
fl.browseForFolderURL([description])
```

Parametri

`description` Una stringa opzionale che specifica la descrizione della finestra di dialogo Sfoglia per cartelle. Se si omette questo parametro, non viene visualizzata alcuna descrizione nell'area corrispondente.

Restituisce

L'URL della cartella, espresso nel formato file:///URI; restituisce `null` se l'utente esce dalla finestra di dialogo selezionando Annulla.

Descrizione

Metodo; visualizza la finestra di dialogo Sfoglia per cartelle, che consente all'utente di selezionare una cartella.

Nota: *il titolo della finestra di dialogo è sempre Sfoglia per cartelle. Utilizzare il parametro* `description` *per aggiungere dettagli alla descrizione, sotto il titolo, ad esempio "Selezionare una cartella" o "Selezionare il percorso che contiene il profilo che si desidera importare".*

Esempio

L'esempio seguente consente all'utente di selezionare una cartella, quindi visualizza l'elenco dei file presenti al suo interno:

```
var folderURI = fl.browseForFolderURL("Select a folder.");
var folderContents = FLfile.listFolder(folderURI);
```

Consultate anche

`fl.browseForFileURL()`, Oggetto FLfile

fl.clipCopyString()

Disponibilità

Flash CS3 Professional.

Uso

```
fl.clipCopyString(string)
```

Parametri

`string` Una stringa da copiare negli Appunti.

Restituisce

Nulla.

Descrizione

Metodo; copia la stringa specificata negli Appunti.

Per copiare negli Appunti la selezione corrente, usate `document.clipCopy()`.

Esempio

L'esempio seguente copia negli Appunti il percorso del documento corrente:

```
var documentPath = fl.getDocumentDOM().path;
fl.clipCopyString(documentPath);
```

fl.closeAll()

Disponibilità
Flash MX 2004.

Uso
```
fl.closeAll([bPromptToSave])
```

Parametri
`bPromptToSave` Valore booleano opzionale che specifica se deve essere visualizzata la finestra di dialogo Salva per i file di testo eventualmente modificati dopo l'ultimo salvataggio, oppure la finestra di dialogo Salva con nome per i file mai salvati in precedenza. Il valore predefinito è `true`.

Restituisce
Nulla.

Descrizione
Metodo; chiude tutti i file aperti (FLA, SWF, JSFL e così via). Per chiudere tutti i file aperti senza salvare le eventuali modifiche, passate il valore `false` per *bPromptToSave*. Questo metodo non chiude l'applicazione.

Esempio
Il codice seguente chiude tutti i file aperti, chiedendo all'utente se desidera salvare i file nuovi o modificati.

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

Consultate anche
`fl.closeAllPlayerDocuments()`, `fl.closeDocument()`

fl.closeAllPlayerDocuments()

Disponibilità
Flash CS3 Professional.

Uso
```
fl.closeAllPlayerDocuments()
```

Parametri
Nessuno.

Restituisce
Un valore booleano: `true` se sono aperte una o più finestre filmato; `false` in caso contrario.

Descrizione
Metodo; chiude tutti i file SWF aperti con Controllo > Prova filmato.

Esempio
L'esempio che segue chiude tutti i file SWF aperti con Controllo > Prova filmato.

```
fl.closeAllPlayerDocuments();
```

Consultate anche
`fl.closeAll()`, `fl.closeDocument()`

fl.closeDocument()

Disponibilità
Flash MX 2004.

Uso
```
fl.closeDocument(documentObject [, bPromptToSaveChanges])
```

Parametri
`documentObject` Un Oggetto Document. Se *documentObject* fa riferimento al documento attivo, è possibile che la finestra del documento rimanga aperta finché non termina l'esecuzione dello script che richiama questo metodo.

`bPromptToSaveChanges` Un valore booleano. Se *bPromptToSaveChanges* è `false`, l'utente non viene interpellato se il documento contiene modifiche non salvate; pertanto, il file viene chiuso e le modifiche vengono eliminate. Se *bPromptToSaveChanges* è `true` e il documento contiene modifiche non salvate, viene visualizzata una finestra di dialogo standard con la richiesta di conferma. Il valore predefinito è `true`. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; chiude il documento specificato.

Esempio
L'esempio seguente illustra due metodi per chiudere un documento:

```
// 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);
```

Consultate anche
`fl.closeAll()`

fl.compilerErrors

Disponibilità
Flash CS3 Professional.

Uso

```
fl.compilerErrors
```

Descrizione

Proprietà di sola lettura; un oggetto compilerErrors che rappresenta il pannello Errori. Per informazioni sull'uso di questa proprietà, consultate Oggetto compilerErrors.

fl.componentsPanel

Disponibilità

Flash MX 2004.

Uso

```
fl.componentsPanel
```

Descrizione

Proprietà di sola lettura; un Oggetto componentsPanel che rappresenta il pannello Componenti.

Esempio

L'esempio seguente memorizza un oggetto componentsPanel nella variabile `comPanel`:

```
var comPanel = fl.componentsPanel;
```

fl.configDirectory

Disponibilità

Flash MX 2004.

Uso

```
fl.configDirectory
```

Descrizione

Proprietà di sola lettura; una stringa che specifica il percorso completo della directory Configuration dell'utente locale sotto forma di percorso specifico della piattaforma. Per specificare questo percorso nel formato file:///URI, che non è specifico della piattaforma, usate `fl.configURI`.

Esempio

L'esempio seguente visualizza la directory Configuration nel pannello Output:

```
fl.trace("My local configuration directory is " + fl.configDirectory);
```

fl.configURI

Disponibilità
Flash MX 2004.

Uso
`fl.configURI`

Descrizione
Proprietà di sola lettura; una stringa che specifica il percorso completo della directory Configuration dell'utente locale nel formato file:///URI. Consultate anche `fl.configDirectory`.

Esempio
L'esempio seguente esegue uno script specificato. `fl.configURI` consente di specificare la posizione dello script senza conoscere la piattaforma su cui è in esecuzione.

```
// 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

Disponibilità
Flash 8.

Uso
`fl.contactSensitiveSelection`

Descrizione
Un valore booleano che specifica se la modalità Sensibile al contatto è attivata (`true`) o meno (`false`).

Esempio
L'esempio seguente mostra come disattivare la modalità di selezione Sensibile al contatto prima di eseguire una selezione, e come ripristinare il suo valore originale dopo aver effettuato la selezione:

```
var contact = fl.contactSensitiveSelection;
fl.contactSensitiveSelection = false;
// Insert selection code here.
fl.contactSensitiveSelection = contact;
```

fl.createDocument()

Disponibilità
Flash MX 2004.

Uso

```
fl.createDocument([docType])
```

Parametri

`docType` Una stringa che specifica il tipo di documento da creare. I valori accettabili sono `"timeline"`, `"presentation"` e `"application"`. Il valore predefinito è `"timeline"`, con lo stesso effetto della scelta di File > Nuovo > File Flash (ActionScript 3.0). Questo parametro è opzionale.

Restituisce

L'oggetto Document per il documento appena creato, se il metodo ha esito positivo. Se si verifica un errore, il valore è `undefined`.

Descrizione

Metodo; apre un nuovo documento e lo seleziona. I valori per le dimensioni, la risoluzione e il colore sono quelli predefiniti correnti.

Esempio

L'esempio seguente crea vari tipi di documenti:

```
// 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

Disponibilità

Flash MX 2004.

Uso

```
fl.createNewDocList
```

Descrizione

Proprietà di sola lettura; un array di stringhe che rappresentano i vari tipi di documento che è possibile creare.

Esempio

L'esempio seguente visualizza i tipi di documento che potete creare nel pannello Output:

```
fl.trace("Number of choices " + fl.createNewDocList.length);
for (i = 0; i < fl.createNewDocList.length; i++)
    fl.trace("choice: " + fl.createNewDocList[i]);
```

fl.createNewDocListType

Disponibilità
Flash MX 2004.

Uso
```
fl.createNewDocListType
```

Descrizione
Proprietà di sola lettura; un array di stringhe che rappresentano le estensioni di file dei tipi di documento che è possibile creare. Gli elementi dell'array corrispondono direttamente (in base all'indice) agli elementi dell'array `fl.createNewDocList`.

Esempio
L'esempio seguente visualizza le estensioni dei tipi di documento che potete creare nel pannello Output:

```
fl.trace("Number of types " + fl.createNewDocListType.length);
for (i = 0; i < fl.createNewDocListType.length; i++) fl.trace("type: " +
fl.createNewDocListType[i]);
```

fl.createNewTemplateList

Disponibilità
Flash MX 2004.

Uso
```
fl.createNewTemplateList
```

Descrizione
Proprietà di sola lettura; un array di stringhe che rappresentano i vari tipi di modello che è possibile creare.

Esempio
L'esempio seguente visualizza i tipi di modello che potete creare nel pannello Output:

```
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

Disponibilità
Flash MX 2004.

Uso
```
fl.documents
```

Descrizione
Proprietà di sola lettura; un array di oggetti Document (consultate Oggetto Document) che rappresentano i documenti (file FLA) attualmente aperti per la modifica.

Esempio
L'esempio seguente memorizza un array di documenti aperti nella variabile `docs`:

```
var docs = fl.documents;
```

L'esempio seguente visualizza i nomi dei documenti attualmente aperti nel pannello Output:

```
for (doc in fl.documents) {
    fl.trace(fl.documents[doc].name);
}
```

fl.downloadLatestVersion()

Disponibilità
Flash CS3 Professional.

Uso
`fl.downloadLatestVersion(fileURI)`

Parametri
`fileURI` Una stringa, espressa nel formato file:///URI, che specifica il percorso locale del file da scaricare dal server Version Cue. Possono essere scaricati solamente i file attualmente non aperti. Se il file specificato da *fileURI* risulta aperto, il metodo non sortisce alcun effetto.

Restituisce
Un valore booleano: `true` se il file è stato scaricato con successo; `false` in caso contrario.

Descrizione
Metodo; scarica l'ultima versione di un file attualmente non aperto dal server Version Cue. Per scaricare la versione più recente di un file aperto, usate `document.synchronizeWithHeadVersion()`.

Esempio
L'esempio che segue scarica il file myFile.fla dal server Version Cue:

```
fl.downloadLatestVersion("file:///C|/MyFiles/Version Cue/docs/myFile.fla");
```

Consultate anche
`document.synchronizeWithHeadVersion()`, `fl.revertDocumentToLastVersion()`, `fl.saveAVersionOfDocument()`, `fl.synchronizeDocumentWithHeadVersion()`

fl.drawingLayer

Disponibilità
Flash MX 2004.

Uso
`fl.drawingLayer`

Descrizione
Proprietà di sola lettura; l'Oggetto drawingLayer utilizzato dagli strumenti estensibili quando l'utente desidera disegnare temporaneamente durante un'operazione di trascinamento (ad esempio, la creazione di un perimetro di selezione).

Esempio
Consultate `drawingLayer.setColor()`.

fl.externalLibraryPath

Disponibilità
Flash CS4 Professional.

Uso
`fl.externalLibraryPath`

Descrizione
Proprietà; una stringa che contiene un elenco di elementi presenti nel percorso della libreria esterna di ActionScript 3.0 globale, che indica la posizione dei file SWC usati come librerie di runtime condivise. Gli elementi nella stringa sono separati da punti e virgola. Nello strumento di creazione, gli elementi vengono specificati scegliendo Modifica > Preferenze > ActionScript > Impostazioni ActionScript 3.0.

Esempio
L'esempio seguente aggiunge la cartella /SWC_runtime al percorso della libreria esterna di ActionScript 3.0 globale:

```
fl.trace(fl.externalLibraryPath);
fl.externalLibraryPath = "/SWC_runtime;" + fl.externalLibraryPath;
fl.trace(fl.externalLibraryPath);
```

Consultate anche
`fl.flexSDKPath`, `fl.libraryPath`, `fl.sourcePath`, `document.externalLibraryPath`

fl.fileExists()

Disponibilità
Flash MX 2004.

Uso

```
fl.fileExists(fileURI)
```

Parametri

`fileURI` Una stringa, espressa nel formato file:///URI, che contiene il percorso del file.

Restituisce

Un valore booleano: `true` se il file è presente nel disco; `false` in caso contrario.

Descrizione

Metodo; verifica se un file è già presente nel disco.

Esempio

L'esempio seguente visualizza `true` o `false` nel pannello Output per indicare se il file specificato è presente o meno:

```
alert(fl.fileExists("file:///C|/example.fla"));
alert(fl.fileExists("file:///C|/example.jsfl"));
alert(fl.fileExists(""));
```

fl.findDocumentDOM()

Disponibilità

Flash CS3 Professional.

Uso

```
fl.findDocumentDOM(id)
```

Parametri

`id` Un numero intero che rappresenta l'identificatore univoco di un documento.

Restituisce

Un oggetto Document, oppure `null` se non esiste alcun documento con l'*id* specificato.

Descrizione

Metodo; permette di indirizzare uno specifico file tramite il suo identificatore univoco (invece del valore di indice). Questo metodo può essere impiegato assieme a `document.id`.

Esempio

L'esempio che segue illustra come leggere l'ID di un documento, utilizzandolo quindi per indirizzare il documento stesso:

```
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;
```

Consultate anche
`fl.findDocumentIndex()`

fl.findDocumentIndex()

Disponibilità
Flash MX 2004.

Uso
```
fl.findDocumentIndex(name)
```

Parametri
`name` Il nome del documento di cui si desidera trovare l'indice. Il documento deve essere aperto.

Restituisce
Restituisce un array di numeri interi che rappresentano la posizione del *nome* di un documento nell'array di `fl.documents`.

Descrizione
Metodo; restituisce un array di numeri interi che rappresentano la posizione del *nome* di un documento nell'array di `fl.documents`. È possibile aprire più documenti con lo stesso nome, se questi si trovano in cartelle diverse.

Esempio
L'esempio seguente visualizza le informazioni sulla posizione nell'indice dei file aperti denominati test.fla nel pannello Output:

```
var filename = "test.fla"
var docIndex = fl.findDocumentIndex(filename);
for (var index in docIndex)
    fl.trace(filename + " is open at index " + docIndex[index]);
```

Consultate anche
`fl.documents`, `fl.findDocumentDOM()`

fl.findObjectInDocByName()

Disponibilità
Flash CS3 Professional.

Uso
```
fl.findObjectInDocByName(instanceName, document)
```

Parametri
`instanceName` Una stringa che specifica il nome dell'istanza di un elemento nel documento specificato.

`document` L'Oggetto Document in cui deve essere cercato l'elemento specificato.

Restituisce

Un array di oggetti generici. Utilizzate la proprietà `.obj` di ogni elemento dell'array per ottenere l'oggetto. L'oggetto dispone delle seguenti proprietà: `keyframe`, `layer`, `timeline` e `parent`. Queste proprietà possono essere impiegate per accedere alla gerarchia dell'oggetto. Per ulteriori informazioni sulle proprietà e su come accedervi, consultate `fl.findObjectInDocByType()`.

È possibile accedere anche a metodi e proprietà per i valori `layer` e `timeline`; questi equivalgono rispettivamente all'Oggetto Layer e all'Oggetto Timeline.

Descrizione

Metodo; espone gli elementi in un documento con nomi di istanza che corrispondono al testo specificato.

Nota: *in alcuni casi, questo metodo funziona solo quando viene eseguito come comando da un file FLA, non mentre state visualizzando o modificando il file JSFL.*

Esempio

L'esempio che segue ricerca gli elementi denominati `"instance01"` nel documento corrente.

```
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");
}
```

Consultate anche

`fl.findObjectInDocByType()`

fl.findObjectInDocByType()

Disponibilità

Flash CS3 Professional.

Uso

```
fl.findObjectInDocByType(elementType, document)
```

Parametri

`elementType` Una stringa che rappresenta il tipo di elemento da cercare. Per i valori accettabili, consultate `element.elementType`.

`document` L'Oggetto Document in cui deve essere cercato l'elemento specificato.

Restituisce

Un array di oggetti generici. Utilizzate la proprietà `.obj` di ogni elemento dell'array per ottenere l'oggetto Element. Ogni oggetto dispone delle seguenti proprietà: `keyframe`, `layer`, `timeline` e `parent`. Queste proprietà possono essere impiegate per accedere alla gerarchia dell'oggetto.

È possibile accedere anche a metodi e proprietà per i valori `layer` e `timeline`; questi equivalgono rispettivamente all'Oggetto Layer e all'Oggetto Timeline.

L'accesso a queste proprietà è illustrato nel secondo e terzo esempio della sezione relativa agli esempi.

Descrizione

Metodo; espone gli elementi del tipo specificato in un documento.

Nota: *in alcuni casi, questo metodo funziona solo quando viene eseguito come comando da un file FLA, non mentre state visualizzando o modificando il file JSFL.*

Esempio

L'esempio che segue ricerca i campi di testo nel documento corrente, quindi ne modifica il contenuto:

```
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");
}
```

Nell'esempio seguente viene illustrato come accedere alle speciali proprietà dell'oggetto restituito dal metodo:

```
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.
}
```

L'esempio che segue illustra come eseguire il backup del DOM per trovare il nome del livello in cui è stato trovato il campo di testo utilizzando un oggetto resultArray.obj:

```
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");
}
```

Consultate anche

`fl.findObjectInDocByName()`

fl.flexSDKPath

Disponibilità

Flash CS4 Professional.

Uso

```
fl.flexSDKPath
```

Descrizione

Proprietà; una stringa che specifica il percorso della cartella Flex SDK, che include tra le altre le cartelle bin, framework e lib. Nello strumento di creazione, gli elementi vengono specificati scegliendo Modifica > Preferenze > ActionScript > Impostazioni ActionScript 3.0.

Esempio

Il codice seguente visualizza il percorso di Flex SDK nel pannello Output.

```
fl.trace(fl.flexSDKPath);
```

Consultate anche

`fl.externalLibraryPath`, `fl.libraryPath`, `fl.sourcePath`

fl.getAppMemoryInfo()

Disponibilità

Flash 8 (solo Windows).

Uso

```
fl.getAppMemoryInfo(memType)
```

Parametri

`memType` Un numero intero che specifica l'area di utilizzo di memoria in cui eseguire la query. Per un elenco dei valori accettabili, consultate la descrizione seguente.

Restituisce

Un numero intero che rappresenta il numero di byte attualmente utilizzati in un'area specificata della memoria di Flash.exe.

Descrizione

Metodo (solo Windows); restituisce un numero intero che rappresenta il numero di byte attualmente utilizzati in un'area specificata della memoria di Flash.exe. Utilizzate la tabella seguente per determinare il valore da passare come *memType*:

memType	Dati risorsa
0	PAGEFAULTCOUNT
1	PEAKWORKINGSETSIZE
2	WORKINGSETSIZE
3	QUOTAPEAKPAGEDPOOLUSAGE
4	QUOTAPAGEDPOOLUSAGE
5	QUOTAPEAKNONPAGEDPOOLUSAGE
6	QUOTANONPAGEDPOOLUSAGE
7	PAGEFILEUSAGE
8	PEAKPAGEFILEUSAGE

Esempio

L'esempio seguente visualizza il consumo della memoria attualmente in uso:

```
var memsize = fl.getAppMemoryInfo(2);
fl.trace("Flash current memory consumption is " + memsize + " bytes or " + memsize/1024 + "
KB");
```

fl.getDocumentDOM()

Disponibilità

Flash MX 2004.

Uso

```
fl.getDocumentDOM()
```

Parametri

Nessuno.

Restituisce

Un oggetto Document, oppure `null` se non è aperto alcun documento.

Descrizione
Metodo; recupera il DOM (Oggetto Document) del documento attivo (file FLA). Se uno o più documenti sono aperti ma uno di essi non è attivo (ad esempio, è attivo un file JSFL), il metodo recupera il DOM del documento attivo più recente.

Esempio
L'esempio seguente visualizza il nome del documento attivo corrente o più recente nel pannello Output:

```
var currentDoc = fl.getDocumentDOM();
fl.trace(currentDoc.name);
```

fl.isFontInstalled()

Disponibilità
Flash CS4 Professional.

Uso
`fl.isFontInstalled(fontName)`

Parametri
`fontName` Una stringa che specifica il nome del carattere.

Restituisce
Un valore booleano: `true` se il carattere specificato è abilitato; `false` in caso contrario.

Descrizione
Metodo: determina se un carattere specificato è installato.

Esempio
Il codice seguente restituisce "true" nel pannello Output se il carattere Times è installato.

```
fl.trace(fl.isFontInstalled("Times"));
```

fl.libraryPath

Disponibilità
Flash CS4 Professional.

Uso
`fl.libraryPath`

Descrizione
Proprietà; una stringa che contiene un elenco di elementi presenti nel percorso della libreria esterna di ActionScript 3.0 globale, che indica la posizione dei file o delle cartelle SWC contenenti i file SWC. Gli elementi nella stringa sono separati da punti e virgola. Nello strumento di creazione, gli elementi vengono specificati scegliendo Modifica > Preferenze > ActionScript > Impostazioni ActionScript 3.0.

Esempio

L'esempio seguente aggiunge la cartella /SWC al percorso della libreria esterna di ActionScript 3.0 globale:

```
fl.trace(fl.libraryPath);
fl.libraryPath = "/SWC;" + fl.libraryPath;
fl.trace(fl.libraryPath);
```

Consultate anche

`fl.externalLibraryPath`, `fl.flexSDKPath`, `fl.sourcePath`, `document.libraryPath`

fl.mapPlayerURL()

Disponibilità

Flash MX 2004.

Uso

```
fl.mapPlayerURL(URI [, returnMBCS])
```

Parametri

`URI` Una stringa che contiene l'URL Unicode con carattere escape di cui eseguire la mappatura.

`returnMBCS` Un valore booleano che è necessario impostare su `true` affinché restituisca un percorso MBCS con carattere escape. In caso contrario, il metodo restituisce UTF-8. Il valore predefinito è `false`. Questo parametro è opzionale.

Restituisce

Una stringa che corrisponde all'URL convertito.

Descrizione

Metodo; esegue la mappatura di un URL Unicode con carattere escape a un URL UTF-8 o MBCS. Usate questo metodo quando prevedete di utilizzare la stringa in ActionScript per accedere a una risorsa esterna. Deve essere utilizzato se occorre gestire caratteri multibyte.

Esempio

L'esempio seguente converte un URL in UTF-8 per consentire al player di caricarlo:

```
var url = MMExecute( "fl.mapPlayerURL(" + myURL + ", false);" );
mc.loadMovie( url);
```

fl.Math

Disponibilità

Flash MX 2004.

Uso

```
fl.Math
```

Descrizione
Proprietà di sola lettura; l'Oggetto Math fornisce i metodi per le operazioni con le matrici e i punti.

Esempio
L'esempio seguente mostra la matrice di trasformazione dell'oggetto selezionato e il suo inverso:

```
// 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

Disponibilità
Flash MX 2004.

Uso
`fl.mruRecentFileList`

Descrizione
Proprietà di sola lettura; un array dei nomi di file completi presenti nell'elenco dei file utilizzati più di recente gestito dallo strumento di creazione Flash.

Esempio
L'esempio seguente visualizza il numero di file aperti di recente, oltre al nome di ciascun file, nel pannello Output:

```
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

Disponibilità
Flash MX 2004.

Uso
`fl.mruRecentFileListType`

Descrizione
Proprietà di sola lettura; un array dei tipi di file presenti nell'elenco dei file utilizzati più di recente gestito dallo strumento di creazione Flash. L'array corrisponde all'array nella proprietà `fl.mruRecentFileList`.

Esempio

L'esempio seguente visualizza il numero di file aperti di recente, oltre al tipo di ciascun file, nel pannello Output:

```
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

Disponibilità

Flash 8.

Uso

```
fl.objectDrawingMode
```

Descrizione

Proprietà; un valore booleano che specifica se è attiva la modalità di Disegno oggetto (`true`) o la modalità di Disegno unione (`false`).

Esempio

L'esempio seguente cambia lo stato della modalità Disegno oggetto:

```
var toggleMode = fl.objectDrawingMode;
if (toggleMode) {
    fl.objectDrawingMode = false;
} else {
    fl.objectDrawingMode = true;
}
```

fl.openDocument()

Disponibilità

Flash MX 2004.

Uso

```
fl.openDocument(fileURI)
```

Parametri

`fileURI` Una stringa, espressa nel formato file:///URI, che specifica il nome del file da aprire.

Restituisce

L'Oggetto Document per il documento appena aperto, se il metodo ha esito positivo. Se il file non viene trovato, o non è un file FLA valido, viene visualizzato un errore e lo script viene annullato.

Descrizione
Metodo; apre un documento Flash (file FLA) da modificare in una nuova finestra di documento Flash e lo rende attivo. Dal punto di vista dell'utente, l'effetto equivale a selezionare File > Apri e successivamente scegliere un file. Se il file specificato è già aperto, la finestra che contiene il documento viene visualizzata in primo piano. La finestra che contiene il file specificato diventa il documento corrente selezionato.

Esempio
L'esempio seguente apre un file denominato Document.fla memorizzato nella directory principale dell'unità C. Il codice contiene un oggetto Document che rappresenta il documento nella variabile `doc` e imposta il documento come attualmente selezionato. In tal modo, fino a quando non diventa attivo un altro documento, `fl.getDocumentDOM()` fa riferimento a questo documento.

```
var doc = fl.openDocument("file:///c|/Document.fla");
```

fl.openScript()

Disponibilità
Flash MX 2004.

Uso
```
fl.openScript(fileURI)
```

Parametri
`fileURI` Una stringa, espressa nel formato URI file:///, che specifica il percorso del file JSFL, AS, ASC, XML, TXT o di altro tipo che deve essere caricato nell'editor di testo di Flash.

Restituisce
Nulla.

Descrizione
Metodo; apre uno script (JSFL, AS, ASC) o un altro file (XML, TXT) nell'editor di testo di Flash.

Esempio
L'esempio seguente apre un file denominato my_test.jsfl memorizzato nella directory/temp dell'unità C.

```
fl.openScript("file:///c|/temp/my_test.jsfl");
```

fl.outputPanel

Disponibilità
Flash MX 2004.

Uso
```
fl.outputPanel
```

Descrizione
Proprietà di sola lettura; fate riferimento all'Oggetto outputPanel.

Esempio
Consultate Oggetto outputPanel.

fl.packagePaths

Disponibilità
Flash CS3 Professional.

Uso
`fl.packagePaths`

Descrizione
Proprietà; una stringa corrispondente all'impostazione del percorso di classe globale nella finestra di dialogo Impostazioni di ActionScript 2.0. I percorsi delle classi all'interno della stringa sono delimitati mediante punti e virgola (;). Per visualizzare o modificare le impostazioni dei percorsi di classe di ActionScript 3.0, usate `fl.as3PackagePaths`.

Esempio
L'esempio che segue illustra la modifica delle impostazioni dei percorsi di classe di ActionScript 2.0:

```
fl.trace(fl.packagePaths);
// Output (assuming started with default value)
// .;$(LocalData)/Classes
fl.packagePaths="buying;selling";
fl.trace(fl.packagePaths);
// Output
// buying; selling
```

Consultate anche
`fl.resetPackagePaths()`

fl.presetPanel

Disponibilità
Flash CS4 Professional.

Uso
`fl.presetPanel`

Descrizione
Proprietà di sola lettura; un Oggetto presetPanel.

fl.quit()

Disponibilità
Flash MX 2004.

Uso
```
fl.quit([bPromptIfNeeded])
```

Parametri
`bPromptIfNeeded` Il valore booleano `true` (predefinito) se si desidera che all'utente venga richiesto di salvare i documenti modificati. Impostate questo parametro su `false` per evitare che venga visualizzata la richiesta di salvare i documenti modificati. In quest'ultimo caso, tutte le modifiche apportate ai documenti aperti vengono eliminate e l'applicazione viene chiusa immediatamente. Anche se risulta particolarmente utile nelle elaborazioni in batch, è opportuno utilizzare questo metodo con cautela. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; esce da Flash e richiede all'utente di salvare i documenti modificati.

Esempio
L'esempio seguente illustra la chiusura dell'applicazione con e senza la richiesta di salvataggio dei documenti modificati:

```
// 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()

Disponibilità
Flash MX 2004.

Uso
```
fl.reloadEffects()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; ricarica tutti i descrittori di effetti definiti nella cartella Configuration Effects dell'utente. Consente di modificare rapidamente gli script durante lo sviluppo e fornisce un meccanismo per migliorare gli effetti senza riavviare l'applicazione. Questo metodo offre migliori risultati se utilizzato in un comando della cartella Commands.

Esempio
L'esempio seguente è uno script di una riga che è possibile collocare nella cartella Commands. Quando è necessario ricaricare gli effetti, selezionate il menu Comandi ed eseguite lo script.

```
fl.reloadEffects();
```

fl.reloadTools()

Disponibilità
Flash MX 2004.

Uso
```
fl.reloadTools()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; ricrea il pannello Strumenti mediante il file toolconfig.xml. Si utilizza questo metodo solo per la creazione di oggetti estensibili. Utilizzate questo metodo quando dovete caricare nuovamente il pannello Strumenti, ad esempio dopo aver modificato il file JSFL che definisce uno strumento già presente nel pannello.

Esempio
L'esempio seguente è uno script di una riga che è possibile collocare nella cartella Commands. Quando dovete ricaricare il pannello Strumenti, eseguite lo script dal menu Comandi.

```
fl.reloadTools();
```

fl.removeEventListener()

Disponibilità
Flash CS3 Professional.

Uso
```
fl.removeEventListener(eventType)
```

Parametri

`eventType` Una stringa che specifica il tipo di evento da rimuovere da questa funzione di callback. I valori accettabili sono `"documentNew"`, `"documentOpened"`, `"documentClosed"`, `"mouseMove"`, `"documentChanged"`, `"layerChanged"` e `"frameChanged"`.

Restituisce

Il valore booleano `true` se il listener di evento è stato rimosso con successo, oppure `false` se la funzione non è mai stata aggiunta alla lista mediante il metodo `fl.addEventListener()`.

Descrizione

Annulla la registrazione di una funzione registrata tramite `fl.addEventListener()`.

Esempio

L'esempio che segue rimuove il listener di evento associato all'evento `documentClosed`:

```
fl.removeEventListener("documentClosed");
```

Consultate anche

`fl.addEventListener()`

fl.resetAS3PackagePaths()

Disponibilità

Flash CS3 Professional.

Uso

```
fl.resetAS3PackagePaths()
```

Parametri

Nessuno.

Descrizione

Metodo; azzera l'impostazione del percorso di classe globale nella finestra di dialogo Impostazioni di ActionScript3.0 ripristinando il valore predefinito. Per azzerare il percorso di classe globale di ActionScript 2.0, usate `fl.resetPackagePaths()`.

Esempio

L'esempio che segue azzera l'impostazione del percorso di classe globale di ActionScript 3.0 al suo valore predefinito.

```
fl.resetAS3PackagePaths();
```

Consultate anche

`fl.as3PackagePaths`

fl.resetPackagePaths()

Disponibilità
Flash CS3 Professional.

Uso
```
fl.resetPackagePaths()
```

Parametri
Nessuno.

Descrizione
Metodo; azzera l'impostazione del percorso di classe globale nella finestra di dialogo Impostazioni di ActionScript 2.0 ripristinando il valore predefinito. Per azzerare il percorso di classe globale di ActionScript 3.0, usate `fl.resetAS3PackagePaths()`.

Esempio
L'esempio che segue azzera l'impostazione del percorso di classe globale di ActionScript 2.0 al suo valore predefinito.

```
fl.resetPackagePaths();
```

Consultate anche
`fl.packagePaths`

fl.revertDocument()

Disponibilità
Flash MX 2004.

Uso
```
fl.revertDocument(documentObject)
```

Parametri
`documentObject` Un Oggetto Document. Se *documentObject* fa riferimento al documento attivo, è possibile che la finestra del documento non venga ripristinata finché non termina l'esecuzione dello script che richiama questo metodo.

Restituisce
Un valore booleano: `true` se l'operazione di ripristino ha esito positivo; `false` in caso contrario.

Descrizione
Metodo; ripristina l'ultima versione salvata del documento FLA specificato. A differenza dell'opzione di menu File > Ripristina, questo metodo non visualizza una finestra di avvertenza che richiede la conferma dell'operazione. Consultate anche `document.revert()` e `document.canRevert()`.

Per ripristinare la versione del documento specificato presente sul server Version Cue, usate `fl.revertDocumentToLastVersion()`.

Esempio
L'esempio seguente ripristina l'ultima versione salvata del documento FLA corrente; tutte le eventuali modifiche apportate dopo l'ultimo salvataggio vengono perse.

```
fl.revertDocument(fl.getDocumentDOM());
```

fl.revertDocumentToLastVersion()

Disponibilità
Flash CS3 Professional.

Uso
```
fl.revertDocumentToLastVersion(documentObject)
```

Parametri
`documentObject` Un Oggetto Document.

Restituisce
Il valore booleano `true` se il documento viene ripristinato correttamente; `false` in caso contrario.

Descrizione
Metodo; se il file non può essere ripristinato, il metodo ripristina la versione del documento specificato presente sul server Version Cue e visualizza eventuali errori nel pannello Output.

Per ripristinare il documento all'ultima versione salvata localmente, usate `fl.revertDocument()`.

Esempio
L'esempio che segue ripristina il documento corrente all'ultima versione memorizzata sul server Version Cue:

```
fl.revertDocumentToLastVersion(fl.getDocumentDOM());
```

Consultate anche
`document.revertToLastVersion()`, `fl.downloadLatestVersion()`, `fl.saveAVersionOfDocument()`, `fl.synchronizeDocumentWithHeadVersion()`

fl.runScript()

Disponibilità
Flash MX 2004.

Uso
```
fl.runScript(fileURI [, funcName [, arg1, arg2, ...]])
```

Parametri
`fileURI` Una stringa, espressa nel formato file:///URI, che specifica il nome del file di script da eseguire.

`funcName` Una stringa che identifica una funzione da eseguire nel file JSFL specificato in *fileURI*. Questo parametro è opzionale.

`arg` Un parametro opzionale che specifica uno o più argomenti da passare a *funcname*.

Restituisce
Il risultato della funzione sotto forma di stringa, se si specifica *funcName*; in caso contrario, non viene restituito alcun valore.

Descrizione
Metodo; esegue un file JavaScript. Se si specifica una funzione come uno degli argomenti, la funzione viene eseguita insieme all'eventuale codice presente nello script che non sia incluso nella funzione. Il resto del codice dello script viene eseguito prima dell'esecuzione della funzione.

Esempio
Supponete che nella directory principale dell'unità C sia presente un file script denominato testScript.jsfl contenente i seguenti elementi:

```
function testFunct(num, minNum) {
    fl.trace("in testFunct: 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
```

Se eseguite il comando seguente,

```
fl.runScript("file:///C|/testScript.jsfl", "testFunct", 10, 1);
```

nel pannello Output appaiono le informazioni seguenti:

```
in for loop i=0
in for loop i=1
end of for loop
in testFunct: 1st arg: 10 2nd arg: 1
```

È anche possibile richiamare testScript.jsfl senza eseguire alcuna funzione, come indicato di seguito:

```
fl.runScript("file:///C|/testScript.jsfl");
```

Ciò produce il seguente risultato nel pannello Output:

```
in for loop i=0
in for loop i=1
end of for loop
```

fl.saveAll()

Disponibilità
Flash MX 2004.

Uso
`fl.saveAll()`

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; salva tutti i documenti aperti.

Se il file non è mai stato salvato, o se non è stato modificato dall'ultimo salvataggio, non viene salvato. Per rendere possibile il salvataggio di un file che non è mai stato salvato o che non è stato modificato, usate `fl.saveDocumentAs()`.

Esempio
L'esempio che segue salva tutti i documenti aperti salvati precedentemente e modificati dopo l'ultimo salvataggio:

```
fl.saveAll();
```

Consultate anche
`document.save()`, `document.saveAndCompact()`, `fl.saveDocument()`, `fl.saveDocumentAs()`

fl.saveAVersionOfDocument()

Disponibilità
Flash CS3 Professional.

Uso
`fl.saveAVersionOfDocument(document)`

Parametri
`document` Un Oggetto Document.

Restituisce
Il valore booleano `true` se la versione del documento viene salvata correttamente sul server Version Cue; `false` in caso contrario.

Descrizione
Metodo; se il file può essere salvato sul server Version Cue, viene visualizzata una finestra di dialogo che permette all'utente di immettere i propri commenti sulla versione, salva una versione del documento specificato sul server e infine mostra eventuali errori nel pannello Output.

Esempio
L'esempio seguente salva il documento corrente sul server Version Cue:

```
fl.saveAVersionOfDocument(fl.getDocumentDOM());
```

Consultate anche

`document.saveAVersion()`

fl.saveDocument()

Disponibilità

Flash MX 2004.

Uso

```
fl.saveDocument(document [, fileURI])
```

Parametri

`document` Un Oggetto Document che specifica il documento da salvare. Se *document* è `null`, viene salvato il documento attivo.

`fileURI` Una stringa, espressa nel formato file:///URI, che specifica il nome del documento salvato. Se il parametro *fileURI* è `null` oppure omesso, il documento viene salvato con il nome corrente. Questo parametro è opzionale.

Restituisce

Un valore booleano: `true` se l'operazione di salvataggio ha esito positivo; `false` in caso contrario.

Se il file non è mai stato salvato, o se non è stato modificato dall'ultimo salvataggio, non viene salvato e viene restituito il valore `false`. Per rendere possibile il salvataggio di un file che non è mai stato salvato o che non è stato modificato, usate `fl.saveDocumentAs()`.

Descrizione

Metodo; salva il documento specificato come documento FLA.

Esempio

L'esempio seguente salva il documento corrente e i due documenti specificati:

```
// 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"));
```

Consultate anche

`document.save()`, `document.saveAndCompact()`, `fl.saveAll()`, `fl.saveDocumentAs()`

fl.saveDocumentAs()

Disponibilità

Flash MX 2004.

Uso

```
fl.saveDocumentAs(document)
```

Parametri
`document` Un Oggetto Document che specifica il documento da salvare. Se *document* è `null`, viene salvato il documento attivo.

Restituisce
Un valore booleano: `true` se l'operazione di salvataggio con nome ha esito positivo; `false` in caso contrario.

Descrizione
Metodo; visualizza una finestra di dialogo Salva con nome per il documento specificato.

Esempio
L'esempio seguente richiede all'utente di salvare il documento specificato, quindi visualizza un messaggio di avvertimento che informa se il documento è stato salvato o meno:

```
alert(fl.saveDocumentAs(fl.documents[1]));
```

Consultate anche
`document.save()`, `document.saveAndCompact()`, `fl.saveAll()`, `fl.saveDocument()`

fl.scriptURI

Disponibilità
Flash CS3 Professional.

Uso
```
fl.scriptURI
```

Descrizione
Proprietà di sola lettura; una stringa, espressa nel formato file:///URI, che rappresenta il percorso dello script JSFL attualmente in esecuzione Se lo script è stato richiamato da `fl.runScript()`, la proprietà indica il percorso dello script principale immediato. Pertanto, non deve attraversare più chiamate a `fl.runScript()` per trovare il percorso dello script di chiamata originale.

Esempio
L'esempio seguente visualizza il percorso dello script JSFL attualmente eseguito nel pannello Output.

```
fl.trace(fl.scriptURI);
```

Consultate anche
`fl.runScript()`

fl.selectElement()

Disponibilità
Flash CS3 Professional.

Uso

```
fl.selectElement(elementObject, editMode)
```

Parametri

`elementObject` L'Oggetto Element che intendete selezionare.

`editMode` Un valore booleano che indica se si desidera modificare l'elemento (`true`) o solamente selezionarlo (`false`).

Restituisce

Il valore booleano `true` se l'elemento viene selezionato con successo; `false` in caso contrario.

Descrizione

Metodo; abilita la selezione o la modifica di un elemento. In generale, questo metodo viene impiegato sugli oggetti restituiti da `fl.findObjectInDocByName()` o `fl.findObjectInDocByType()`.

Esempio

L'esempio seguente seleziona un elemento chiamato `"second text field"` (se presente nel documento):

```
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");
    }
```

Consultate anche

`fl.findObjectInDocByName()`, `fl.findObjectInDocByType()`

fl.selectTool()

Disponibilità

Flash CS3 Professional.

Uso

```
fl.selectTool(toolName)
```

Parametri

`toolName` Una stringa che specifica il nome dello strumento da selezionare. Consultate la sezione "Descrizione", di seguito, per informazioni sui valori accettati da questo parametro.

Descrizione

Metodo; seleziona lo strumento specificato nel pannello Strumenti. I valori accettabili per *toolName* sono `"arrow"`, `"bezierSelect"`, `"freeXform"`, `"fillXform"`, `"lasso"`, `"pen"`, `"penplus"`, `"penminus"`, `"penmodify"`, `"text"`, `"line"`, `"rect"`, `"oval"`, `"rectPrimitive"`, `"ovalPrimitive"`, `"polystar"`, `"pencil"`, `"brush"`, `"inkBottle"`, `"bucket"`, `"eyeDropper"`, `"eraser"`, `"hand"` e `"magnifier"`.

In caso di creazione di strumenti personalizzati, i loro nomi possono essere anch'essi passati come parametro *toolName*. L'elenco dei nomi di strumento è disponibile nel file seguente:

- Windows Vista:

  unità di avvio \Utenti*nomeutente*\Impostazioni locali\Dati applicazioni\Adobe\Flash CS3*lingua* \Configuration\Tools\toolConfig.xml

- Windows XP:

  unità di avvio \Documents and Settings*nomeutente*\Impostazioni locali\Dati applicazioni\Adobe\Flash CS3*lingua*\Configuration\Tools\toolConfig.xml

- Mac OS X:

  Macintosh HD/Users/*nomeutente*/Library/Supporto applicazioni/Adobe/Flash CS3/*lingua*/Configuration/Tools\toolConfig.xml

Esempio

L'esempio che segue seleziona lo strumento Penna.

```
fl.selectTool("pen");
```

Consultate anche

Oggetto Tools, Oggetto ToolObj

fl.setActiveWindow()

Disponibilità

Flash MX 2004.

Uso

```
fl.setActiveWindow(document [, bActivateFrame])
```

Parametri

`document` Un Oggetto Document che specifica il documento da selezionare come finestra attiva.

`bActivateFrame` Un parametro opzionale che viene ignorato da Flash e Fireworks e che è presente solo per assicurare la compatibilità con Dreamweaver.

Restituisce

Nulla.

Descrizione
Metodo; imposta la finestra attiva in modo che sia il documento specificato. Questo metodo è supportato anche da Dreamweaver e Fireworks. Se un documento ha più visualizzazioni (create con Finestra > Duplica finestra), viene selezionata la vista attiva più recente.

Esempio
L'esempio seguente mostra due modi per attivare un documento specificato:

```
fl.setActiveWindow(fl.documents[0]);

var theIndex = fl.findDocumentIndex("myFile.fla");
fl.setActiveWindow(fl.documents[theIndex]);
```

fl.showIdleMessage()

Disponibilità
Flash 8.

Uso
```
fl.showIdleMessage(show)
```

Parametri
`show` Un valore booleano che specifica se attivare o disattivare l'avvertenza visualizzata quando l'esecuzione di uno script richiede troppo tempo.

Restituisce
Nulla.

Descrizione
Metodo; consente di disattivare l'avvertenza visualizzata quando l'esecuzione di uno script richiede troppo tempo (passate `false` per *show*). La possibilità di disattivare l'avvertenza può essere utile quando si eseguono delle operazioni in batch il cui completamento richiede molto tempo. Per riattivare l'avvertenza, eseguite nuovamente il comando passando `true` per *show*.

Esempio
L'esempio seguente illustra come disattivare e riattivare l'avvertenza visualizzata quando uno script viene eseguito troppo a lungo:

```
fl.showIdleMessage(false);
var result = timeConsumingFunction();
fl.showIdleMessage(true); ;
var result = timeConsumingFunction();
```

fl.sourcePath

Disponibilità
Flash CS4 Professional.

Uso

```
fl.sourcePath
```

Descrizione

Proprietà; una stringa che contiene un elenco di elementi presenti nel percorso d'origine globale di ActionScript 3.0 che indica la posizione dei file delle classi di ActionScript. Gli elementi nella stringa sono separati da punti e virgola. Nello strumento di creazione, gli elementi vengono specificati scegliendo Modifica > Preferenze > ActionScript > Impostazioni ActionScript 3.0.

Esempio

L'esempio seguente aggiunge la cartella /Classes al percorso d'origine globale di ActionScript 3.0:

```
fl.trace(fl.sourcePath);
fl.sourcePath = "/Classes;" + fl.sourcePath;
fl.trace(fl.sourcePath);
```

Consultate anche

`fl.flexSDKPath`, `fl.externalLibraryPath`, `fl.libraryPath`, `document.sourcePath`

fl.swfPanels

Disponibilità

Flash CS4 Professional.

Uso

```
fl.swfPanels
```

Descrizione

Proprietà di sola lettura; un array degli oggetti swfPanel registrati (consultate Oggetto swfPanel). Un oggetto swfPanel è registrato se è stato aperto almeno una volta.

Una posizione del pannello nell'array rappresenta l'ordine con cui è stato aperto. Se il primo pannello aperto è TraceBitmap e il secondo pannello aperto è AnotherFunction, allora `fl.swfPanels[0]` è l'oggetto swfPanel TraceBitmap, mentre `fl.swfPanels[1]` è l'oggetto swfPanel AnotherFunction e così via.

Esempio

Il codice seguente restituisce il nome e il percorso di tutti i pannelli di Window SWF registrati nel pannello Output:

```
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.synchronizeDocumentWithHeadVersion()

Disponibilità
Flash CS3 Professional.

Uso
`fl.synchronizeDocumentWithHeadVersion(documentObject)`

Parametri
`documentObject` Un Oggetto Document.

Restituisce
Il valore booleano `true` se il file specificato è stato sincronizzato correttamente con il server Version Cue; `false` in caso contrario.

Descrizione
Metodo; sincronizza il documento specificato con la versione più recente presente sul server Version Cue e visualizza eventuali errori nel pannello Output. Questo metodo è identico a `document.synchronizeWithHeadVersion()`.

Esempio
L'esempio seguente sincronizza il documento corrente con il server Version Cue:

```
fl.synchronizeWithHeadVersion(fl.getDocumentDOM());
```

Consultate anche
`fl.downloadLatestVersion()`, `fl.revertDocumentToLastVersion()`, `fl.saveAVersionOfDocument()`

fl.tools

Disponibilità
Flash MX 2004.

Uso
`fl.tools`

Descrizione
Proprietà di sola lettura; un array di oggetti Tools (consultate Oggetto Tools). Si utilizza questa proprietà solo per la creazione di oggetti estensibili.

fl.trace()

Disponibilità
Flash MX 2004.

Uso

```
fl.trace(message)
```

Parametri

`message` Una stringa visualizzata nel pannello Output.

Restituisce

Nulla.

Descrizione

Metodo; invia al pannello Output una stringa di testo, terminata da una nuova riga, e visualizza il pannello Output, se non è già visibile. Questo metodo è identico a `outputPanel.trace()` ed equivale all'istruzione `trace()` di ActionScript.

Per inviare una riga vuota, utilizzate `fl.trace("")` o `fl.trace("\n")`. È possibile utilizzare l'ultimo comando inline, impostando `\n` come parte della stringa *message*.

Esempio

L'esempio seguente visualizza diverse righe di testo nel pannello Output:

```
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

Disponibilità

Flash MX 2004.

Uso

```
fl.version
```

Descrizione

Proprietà di sola lettura; la stringa lunga corrispondente alla versione dello strumento di creazione Flash, compresa la piattaforma.

Esempio

L'esempio seguente visualizza la versione dello strumento di creazione di Flash nel pannello Output:

```
alert(fl.version); // For example, WIN 10,0,0,540
```

fl.xmlui

Disponibilità
Flash MX 2004.

Uso
`fl.xmlui`

Descrizione
Proprietà di sola lettura; un Oggetto XMLUI. Questa proprietà consente di ottenere e impostare le proprietà XMLUI in una finestra di dialogo XMLUI e consente di accettare o annullare la finestra di dialogo a livello di programmazione.

Esempio
Consultate Oggetto XMLUI.

Capitolo 18: Oggetto FLfile

Disponibilità
Flash MX 2004 7.2.

Descrizione
L'oggetto FLfile consente di scrivere estensioni di Flash che accedano, modifichino e rimuovano i file e le cartelle nel file system locale. L'API FLfile viene fornita sotto forma di un'estensione dell'API JavaScript che viene definita *libreria condivisa* e si trova nella cartella seguente:

- Windows Vista:

  unità di avvio \Utenti*nomeutente*\Impostazioni locali\Dati applicazioni\Adobe\Flash CS3*lingua*\Configuration\External Libraries\FLfile.dll

- Windows XP:

  unità di avvio\Documents and Settings*nomeutente*\Impostazioni locali\Dati applicazioni\Adobe\Flash CS3*lingua*\Configuration\External Libraries\FLfile.dll

- Mac OS X:

  Macintosh HD/Users/*nomeutente*/Library/Supporto applicazioni/Adobe/Flash CS3/*lingua*/Configuration/External Libraries/FLfile.dll

Nota: *è importante non confondere le librerie condivise che contengono i simboli dei documenti Flash con le librerie condivise dell'API JavaScript. Sono due tipi di librerie diversi.*

I metodi di FLfile funzionano con i file o le cartelle (directory) sul disco. Pertanto, ciascun metodo richiede uno o più parametri per specificare la posizione di un file o di una cartella. La posizione del file o della cartella è espressa sotto forma di una stringa che ha un formato molto simile all'URL di un sito Web. La stringa è definita file URI (Uniform Resource Identifier) ed è formattata nel modo illustrato di seguito (comprese le virgolette):

```
"file:///drive|/folder 1/folder 2/.../filename"
```

Se, ad esempio, desiderate creare una cartella denominata config nell'unità C e posizionarla nella cartella Programmi/MyApp, utilizzate il comando seguente:

```
FLfile.createFolder("file:///C|/Program Files/MyApp/config");
```

Se successivamente desiderate includere nella stessa cartella un file denominato config.ini, utilizzate il comando seguente:

```
FLfile.write("file:///C|/Program Files/MyApp/config/config.ini", "");
```

Per creare una cartella su un computer Macintosh, potete eseguire il comando seguente:

```
FLfile.createFolder("file:///Macintosh/MyApp/config");
```

Riepilogo dei metodi
I seguenti metodi possono essere utilizzati con l'oggetto FLfile:

Metodo	Descrizione
FLfile.copy()	Copia un file.
FLfile.createFolder()	Crea una o più cartelle.
FLfile.exists()	Determina l'esistenza di un file o di una cartella.
FLfile.getAttributes()	Rileva se un file è scrivibile, di sola lettura, nascosto, visibile o se è una cartella di sistema.
FLfile.getCreationDate()	Specifica il numero dei secondi trascorsi tra il 1° gennaio 1970 e l'ora di creazione del file o della cartella.
FLfile.getCreationDateObj()	Ottiene la data di creazione di un file o di una cartella.
FLfile.getModificationDate()	Specifica il numero dei secondi trascorsi tra il 1° gennaio 1970 e l'ora dell'ultima modifica del file o della cartella.
FLfile.getModificationDateObj()	Ottiene la data dell'ultima modifica di un file o di una cartella.
FLfile.getSize()	Ottiene le dimensioni di un file.
FLfile.listFolder()	Visualizza il contenuto di una cartella.
FLfile.platformPathToURI()	Converte un nome di file in un formato specifico della piattaforma in un URI file:///.
FLfile.read()	Legge il contenuto di un file.
FLfile.remove()	Elimina un file o una cartella.
FLfile.setAttributes()	Rende di sola lettura, nascosto o visibile un file o una cartella.
FLfile.uriToPlatformPath()	Converte un nome di file espresso come URI file:/// in un formato specifico della piattaforma.
FLfile.write()	Crea un file oppure scrive su di esso o vi aggiunge un elemento.

FLfile.copy()

Disponibilità
Flash MX 2004 7.2.

Uso
```
FLfile.copy(fileURI, copyURI)
```

Parametri
`fileURI` Una stringa, espressa come URI file:///, che specifica il file da copiare.

`copyURI` Una stringa, espressa come URI file:///, che specifica la posizione e il nome del file copiato.

Restituisce
Il valore booleano `true` se l'esito è positivo; `false` in caso contrario.

Descrizione
Metodo; copia un file da una posizione a un'altra. Restituisce `false` se *copyURI* esiste già.

Esempio

L'esempio seguente crea una copia di backup di un file di configurazione denominato config.ini e la posiziona nella stessa cartella in cui si trova il file, ma con un nuovo nome:

```
var originalFileURI="file:///C|/Program Files/MyApp/config.ini";
var newFileURI="file:///C|/Program Files/MyApp/config_backup.ini";
FLfile.copy(originalFileURI, newFileURI);
```

La stessa operazione può essere eseguita anche con un solo comando:

```
FLfile.copy("file:///C|:/Program Files/MyApp/config.ini", file:///C|/Program
Files/MyApp/config_backup.ini");
```

FLfile.createFolder()

Disponibilità

Flash MX 2004 7.2.

Uso

```
FLfile.createFolder(folderURI)
```

Parametri

`folderURI` Un URI di tipo cartella che specifica la struttura di cartelle da creare.

Restituisce

Il valore booleano `true` se l'esito è positivo; `false` se *folderURI* esiste già.

Descrizione

Metodo; crea una o più cartelle nella posizione specificata.

È possibile creare più cartelle contemporaneamente. Ad esempio, il comando seguente consente di creare la cartella MyData e la cartella TempData se non esistono già:

```
FLfile.createFolder("file:///c|/MyData/TempData")
```

Esempio

L'esempio seguente crea una cartella e una sottocartella sotto la cartella di configurazione (fl.configURI):

```
fl.trace(FLfile.createFolder(fl.configURI+"folder01/subfolder01"));
```

L'esempio seguente tenta di creare una cartella denominata tempFolder nel livello principale dell'unità C e visualizza una casella di avviso che indica l'esito dell'operazione:

```
var folderURI = "file:///c|/tempFolder";
if (FLfile.createFolder(folderURI)) {
    alert("Created " + folderURI);
}
else {
    alert(folderURI + " already exists");
}
```

Consultate anche

FLfile.remove(), FLfile.write()

FLfile.exists()

Disponibilità

Flash MX 2004 7.2.

Uso

```
FLfile.exists(fileURI)
```

Parametri

fileURI Una stringa, espressa come URI file:///, che specifica il file da verificare.

Restituisce

Il valore booleano `true` se l'esito è positivo; `false` in caso contrario.

Descrizione

Metodo; indica se esiste un file specificato. Se si specifica una cartella e un nome di file, tale cartella deve già esistere. Per creare le cartelle, consultate FLfile.createFolder().

Esempi

L'esempio seguente verifica la presenza di un file denominato mydata.txt nella cartella temp e visualizza una finestra di avviso che indica se il file esiste:

```
var fileURI = "file:///c|/temp/mydata.txt";
if (FLfile.exists(fileURI)) {
    alert( fileURI + " exists.");
}
else {
    alert( fileURI + " does not exist.");
}
```

L'esempio seguente verifica l'esistenza di un file di configurazione richiesto nella cartella MyApplication. Se il file non esiste, viene creato.

```
var configFile = "file:///C|/MyApplication/config.ini";
if (!FLfile.exists(configFile)) {
    FLfile.write(configFile,"");

}
```

Consultate anche

FLfile.write()

FLfile.getAttributes()

Disponibilità
Flash MX 2004 7.2.

Uso
`FLfile.getAttributes(fileOrFolderURI)`

Parametri
`fileOrFolderURI` Una stringa, espressa come URI file:///, che specifica il file o la cartella di cui desiderate recuperare gli attributi.

Restituisce
Una stringa che rappresenta gli attributi del file o della cartella specificati.

i risultati possono essere imprevedibili se il file o la cartella non esistono. È consigliabile utilizzare `FLfile.exists()` prima di utilizzare questo metodo.

Descrizione
Metodo; restituisce una stringa che rappresenta gli attributi del file o della cartella specificati, oppure una stringa vuota se il file non presenta attributi specifici (vale a dire, se non è di sola lettura, nascosto, e così via). È sempre opportuno verificare l'esistenza di un file o di una cartella mediante `FLfile.exists()` prima di utilizzare questo metodo.

I caratteri presenti nella stringa rappresentano gli attributi nel modo seguente:

- `R` — *fileOrFolderURI* è di sola lettura
- `D` — *fileOrFolderURI* è una cartella (directory)
- `H` — *fileOrFolderURI* è nascosto (solo Windows)
- `S` — *fileOrFolderURI* è un file o una cartella di sistema (solo Windows)
- `A` — *fileOrFolderURI* è pronto per essere archiviato (solo Windows)

Ad esempio, se *fileOrFolderURI* è una cartella nascosta, la stringa restituita è`"DH"`.

Esempio
L'esempio seguente ottiene gli attributi del file mydata.txt e visualizza una casella di avviso se il file è di sola lettura.

```
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!");
    }
}
```

Consultate anche
`FLfile.setAttributes()`

FLfile.getCreationDate()

Disponibilità
Flash MX 2004 7.2.

Uso
```
FLfile.getCreationDate(fileOrFolderURI)
```

Parametri
`fileOrFolderURI` Una stringa, espressa come URI file:///, che specifica il file o la cartella di cui desiderate recuperare la data e l'ora di creazione sotto forma di stringa esadecimale.

Restituisce
Una stringa contenente un numero esadecimale che rappresenta il numero di secondi trascorsi tra il 1° gennaio 1970 e l'ora di creazione del file o della cartella, o `"00000000"` se il file o la cartella non esiste.

Descrizione
Metodo; specifica il numero dei secondi trascorsi tra il 1° gennaio 1970 e l'ora di creazione del file o della cartella. Questo metodo è usato principalmente per confrontare le date di creazione o modifica di file e cartelle.

Esempio
L'esempio seguente determina se un file è stato modificato dopo che è stato creato:

```
// 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.");
}
```

Consultate anche
`FLfile.getCreationDateObj()`, `FLfile.getModificationDate()`

FLfile.getCreationDateObj()

Disponibilità
Flash MX 2004 7.2.

Uso
```
FLfile.getCreationDateObj(fileOrFolderURI)
```

Parametri

`fileOrFolderURI` Una stringa, espressa come URI file:///, che specifica il file o la cartella di cui desiderate recuperare la data e l'ora di creazione sotto forma di oggetto JavaScript Date.

Restituisce

Un oggetto JavaScript Date che rappresenta la data e l'ora di creazione del file o della cartella specificati. Se il file non esiste, l'oggetto contiene l'informazione che il file o la cartella sono stati creati alla mezzanotte (GMT) del 31 dicembre 1969.

Descrizione

Metodo; restituisce un oggetto JavaScript Date che rappresenta la data e l'ora di creazione del file o della cartella specificati.

Esempio

L'esempio seguente visualizza (in formato leggibile) la data in cui un file è stato creato nel pannello Output:

```
// Make sure the specified file exists.
var file1Date = FLfile.getCreationDateObj("file:///c|/temp/file1.txt");
fl.trace(file1Date);
```

Consultate anche

`FLfile.getCreationDate()`, `FLfile.getModificationDateObj()`

FLfile.getModificationDate()

Disponibilità

Flash MX 2004 7.2.

Uso

```
FLfile.getModificationDate(fileOrFolderURI)
```

Parametri

`fileOrFolderURI` Una stringa, espressa come URI file:///, che specifica il file di cui desiderate recuperare la data e l'ora di modifica sotto forma di stringa esadecimale.

Restituisce

Una stringa contenente un numero esadecimale che rappresenta il numero di secondi trascorsi tra il 1° gennaio 1970 e l'ora dell'ultima modifica del file o della cartella, o `"00000000"` se il file non esiste.

Descrizione

Metodo; specifica il numero dei secondi trascorsi tra il 1° gennaio 1970 e l'ora di modifica del file o della cartella. Questo metodo è usato principalmente per confrontare le date di creazione o modifica di file e cartelle.

Esempio

L'esempio seguente confronta le date di modifica di due file e determina quale dei due è stato modificato per ultimo:

```
// 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") ;
}
```

Consultate anche

FLfile.getCreationDate(), FLfile.getModificationDateObj()

FLfile.getModificationDateObj()

Disponibilità

Flash MX 2004 7.2.

Uso

```
FLfile.getModificationDateObj(fileOrFolderURI)
```

Parametri

`fileOrFolderURI` Una stringa, espressa come URI file:///, che specifica il file o la cartella di cui desiderate recuperare la data e l'ora di modifica sotto forma di oggetto JavaScript Date.

Restituisce

Un oggetto JavaScript Date che rappresenta la data e l'ora dell'ultima modifica del file o della cartella specificati. Se il file o la cartella non esistono, l'oggetto contiene l'informazione che il file o la cartella sono stati creati alla mezzanotte (GMT) del 31 dicembre 1969.

Descrizione

Metodo; restituisce un oggetto JavaScript Date che rappresenta la data e l'ora dell'ultima modifica del file o della cartella specificati.

Esempio

L'esempio seguente visualizza (in formato leggibile) la data dell'ultima modifica di un file nel pannello Output:

```
// Make sure the specified file exists.
var file1Date = FLfile.getModificationDateObj("file:///c|/temp/file1.txt");
trace(file1Date);
```

Consultate anche

FLfile.getCreationDateObj(), FLfile.getModificationDate()

FLfile.getSize()

Disponibilità
Flash MX 2004 7.2.

Uso
```
FLfile.getSize(fileURI)
```

Parametri
`fileURI` Una stringa, espressa come URI file:///, che specifica il file di cui desiderate recuperare le dimensioni.

Restituisce
Un numero intero che rappresenta le dimensioni espresse in byte del file specificato, oppure 0 se il file non esiste.

Descrizione
Metodo; restituisce un numero intero che rappresenta le dimensioni espresse in byte del file specificato, oppure 0 se il file non esiste. Se il valore restituito è 0, potete utilizzare `FLfile.exists()` per determinare se il file è di zero byte o non esiste.

Questo metodo restituisce valori di dimensione corretti solo per i file che non superano i 2 GB.

Esempio
Nell'esempio seguente, le dimensioni del file mydata.txt vengono memorizzate nella variabile `fileSize`:

```
var URL = "file:///c|/temp/mydata.txt";
var fileSize = FLfile.getSize(URL);
```

FLfile.listFolder()

Disponibilità
Flash MX 2004 7.2.

Uso
```
FLfile.listFolder(folderURI [, filesOrDirectories])
```

Parametri
`folderURI` Una stringa, espressa come URI file:///, che specifica la cartella di cui desiderate recuperare il contenuto. Potete includere un filtro per i caratteri jolly come parte di *folderURI*. I caratteri jolly validi sono*(corrisponde a uno o più caratteri) e ?(corrisponde a un carattere singolo).

`filesOrDirectories` Una stringa opzionale che specifica se restituire solo nomi di file o solo nomi di cartella (directory). Se viene omessa, vengono restituiti sia i nomi di file che i nomi di cartella. I valori accettabili sono `"files"` e `"directories"`.

Restituisce
Un array di stringhe che rappresenta il contenuto della cartella. Se la cartella non esiste o se nessun file o cartella corrisponde ai criteri specificati, restituisce un array vuoto.

Descrizione
Metodo; restituisce un array di stringhe che rappresenta il contenuto della cartella.

Esempi
L'esempio seguente restituisce tre array: Il primo rappresenta tutti i file nella cartella C:\temp, il secondo rappresenta tutte le cartelle nella cartella C:\temp e il terzo rappresenta i file e le cartelle nella cartella C:\temp:

```
var fileURI = "file:///C|/temp/" ;
var folderURI = "file:///C|/temp" ;
var fileList1 = FLfile.listFolder(fileURI, "files"); // files
var fileList2 = FLfile.listFolder(folderURI, "directories"); //folders
var fileList3 = FLfile.listFolder(folderURI); //files and folders
fl.trace("Files: " + fileList1);
fl.trace("");
fl.trace("Folders:  " + fileList2);
fl.trace("");
fl.trace("Files and folders: " + fileList3);
```

L'esempio seguente restituisce un array di tutti i file di testo (.txt) presenti nella cartella temp e visualizza l'elenco in una casella di avviso:

```
var folderURI = "file:///c|/temp";
var fileMask = "*.txt";
var list = FLfile.listFolder(folderURI + "/" + fileMask, "files");
if (list) {
    alert(folderURI + " contains: " + list.join(" "));
}
```

L'esempio seguente utilizza un filtro per i file nella stringa *folderURI* specificata per restituire i nomi di tutti i file eseguibili presenti nella cartella delle applicazioni di Windows:

```
var executables = FLfile.listFolder("file:///C|/WINDOWS/*.exe","files");
alert(executables.join("\n"));
```

FLfile.platformPathToURI()

Disponibilità
Flash CS4 Professional.

Uso
```
FLfile.platformPathToURI(fileName)
```

Parametri
`fileName` Una stringa, espressa in un formato specifico della piattaforma, che specifica il nome di file da convertire.

Restituisce
Una stringa espressa come URI file:///.

Descrizione
Metodo; converte un nome di file in un formato specifico della piattaforma in un URI file:///.

Esempio

Il seguente esempio converte un nome di file da un formato specifico della piattaforma in un URI file:///, che viene passato a `outputPanel.save()`:

```
var myFilename = "C:\\outputPanel.txt";
var myURI=FLfile.platformPathToURI(myFilename);
fl.outputPanel.save(myURI);
```

Consultate anche

`FLfile.uriToPlatformPath()`

FLfile.read()

Disponibilità

Flash MX 2004 7.2.

Uso

```
FLfile.read()
```

Parametri

`fileOrFolderURI` Una stringa, espressa come URI file:///, che specifica il file o la cartella di cui desiderate recuperare gli attributi.

Restituisce

Il contenuto del file specificato sotto forma di stringa, oppure `null` se la lettura non dà esito positivo.

Descrizione

Metodo; restituisce il contenuto del file specificato sotto forma di stringa, oppure `null` se la lettura non dà esito positivo.

Esempi

L'esempio seguente legge il file mydata.txt e, in caso di esito positivo, visualizza una casella di avviso con il contenuto del file:

```
var fileURI = "file:///c|/temp/mydata.txt";
var str = FLfile.read( fileURI);
if (str) {
    alert( fileURL + " contains: " + str);
}
```

L'esempio seguente legge il codice ActionScript da un file di classe e lo memorizza nella variabile `code`:

```
var classFileURI = "file:///C|/MyApplication/TextCarousel.as";
var code = FLfile.read(classFileURI);
```

FLfile.remove()

Disponibilità
Flash MX 2004 7.2.

Uso
```
FLfile.remove(fileOrFolderURI)
```

Parametri
`fileOrFolderURI` Una stringa, espressa come URI file:///, che specifica il file o la cartella che desiderate rimuovere (eliminare).

Restituisce
Il valore booleano `true` se l'esito è positivo; `false` in caso contrario.

Descrizione
Metodo; elimina il file o la cartella specificati. Se la cartella contiene file, anche i file vengono eliminati. I file contrassegnati dall'attributo R (Read only) sono di sola lettura e non possono essere rimossi.

Esempi
L'esempio seguente avverte l'utente se un file esiste, quindi lo elimina se l'utente decide di rimuoverlo:

```
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");
}
```

L'esempio seguente elimina un file di configurazione creato da un'applicazione:

```
if(FLfile.remove("file:///C|/MyApplication/config.ini")) {
    alert("Configuration file deleted");
}
```

L'esempio seguente elimina la cartella Configuration e il relativo contenuto:

```
FLfile.remove("file:///C|/MyApplication/Configuration/");
```

Consultate anche
`FLfile.createFolder()`, `FLfile.getAttributes()`

FLfile.setAttributes()

Disponibilità
Flash MX 2004 7.2.

Uso
```
FLfile.setAttributes(fileURI, strAttrs)
```

Parametri
`fileURI` Una stringa, espressa come URI file:///, che specifica il file di cui desiderate impostare gli attributi.

`strAttrs` Una stringa che specifica i valori degli attributi da impostare. Per conoscere i valori accettabili per *strAttrs*, consultate la sezione "Descrizione" seguente.

Restituisce
Il valore booleano `true` se l'esito è positivo.

Nota: *i risultati possono essere imprevedibili se il file o la cartella non esistono. È consigliabile utilizzare* `FLfile.exists()` *prima di utilizzare questo metodo.*

Descrizione
Metodo; specifica gli attributi a livello di sistema per il file specificato.

I valori seguenti sono validi per *strAttrs*:

- `N` — Nessun attributo specifico (non di sola lettura, né nascosto, e così via)
- `A` — (Archive) Pronto per essere archiviato (solo Windows)
- `R` — (Read-only) Sola lettura (nei sistemi Macintosh, "read-only" significa "bloccato")
- `W` — (Writable) Scrivibile (ha la precedenza su `R`)
- `H` — (Hidden) Nascosto (solo Windows)
- `V` — (Visible) Visibile (ha la precedenza su `H`, solo Windows)

Se includete sia `R` che `W` in *strAttrs*, l'attributo`R` viene ignorato e il file viene impostato come scrivibile. In modo analogo, se passate sia `H` che `V`, l'attributo `H` viene ignorato e il file viene impostato come visibile.

Se desiderate fare in modo che l'attributo archive non venga impostato, utilizzate questo comando con il parametro `N` prima di impostare gli attributi. Non è presente una controparte diretta di `A` che disattiva l'attributo archive.

Esempi
L'esempio seguente imposta il file mydata.txt in modo che sia di sola lettura e nascosto: Non ha alcun effetto sull'attributo archive.

```
var URI = "file:///c|/temp/mydata.txt";
if (FLfile.exists(URI)) {
    FLfile.setAttributes(URI, "RH");
}
```

L'esempio seguente imposta il file mydata.txt in modo che sia di sola lettura e nascosto: Verifica inoltre che non sia impostato l'attributo archive.

```
var URI = "file:///c|/temp/mydata.txt";

if (FLfile.exists(URI)) {
    FLfile.setAttributes(URI, "N");
    FLfile.setAttributes(URI, "RH");
}
```

Consultate anche

FLfile.getAttributes()

FLfile.uriToPlatformPath()

Disponibilità

Flash CS4 Professional.

Uso

```
FLfile.uriToPlatformPath(fileURI)
```

Parametri

`fileURI` Una stringa, espressa come URI file:///, che specifica il nome di file da convertire.

Restituisce

Una stringa che rappresenta un percorso specifico della piattaforma.

Descrizione

Metodo; converte un nome di file espresso come URI file:/// in un formato specifico della piattaforma.

Esempio

L'esempio seguente converte un nome di file espresso come URI file:/// in un formato specifico della piattaforma:

```
var dir =(fl.configDirectory);
var URI = FLfile.platformPathToURI(dir);
fl.trace(URI == fl.configURI); // displays "true"
```

Consultate anche

FLfile.platformPathToURI()

FLfile.write()

Disponibilità

Flash MX 2004 7.2.

Uso

```
FLfile.write(fileURI, textToWrite, [ , strAppendMode])
```

Parametri

`fileURI` Una stringa, espressa come URI file:///, che specifica il file in cui desiderate scrivere.

`textToWrite` Una stringa che rappresenta il testo da inserire nel file.

`strAppendMode` Una stringa opzionale con il valore `"append"`, che specifica di aggiungere *textToWrite* al file esistente. Se viene omessa, *fileURI* viene sovrascritto con *textToWrite*.

Restituisce

Il valore booleano `true` se l'esito è positivo; `false` in caso contrario.

Descrizione

Metodo; scrive la stringa specificata nel file specificato (in formato UTF-8). Se il file specificato non esiste, viene creato. La cartella in cui il file viene inserito deve invece esistere prima dell'utilizzo del metodo. Per creare le cartelle, utilizzate `FLfile.createFolder()`.

Esempio

L'esempio seguente tenta di scrivere la stringa "xxx" nel file mydata.txt e visualizza un messaggio di avviso in caso di esito positivo, quindi, tenta di aggiungere la stringa "aaa" al file e visualizza un secondo messaggio di avviso in caso di esito positivo. Una volta eseguito questo script, il file mydata.txt conterrà solo il testo "xxxaaa".

```
var URI = "file:///c|/temp/mydata.txt";
if (FLfile.write(URI, "xxx")) {
    alert("Wrote xxx to " + URI);
}
if (FLfile.write(URI, "aaa", "append")) {
    alert("Appended aaa to " + fileURI);
}
```

Consultate anche

`FLfile.createFolder()`, `FLfile.exists()`

Capitolo 19: Oggetto folderItem

Ereditarietà Oggetto Item > Oggetto folderItem

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto folderItem è una sottoclasse dell'oggetto Item. Non esistono proprietà o metodi univoci per folderItem. Consultate Oggetto Item.

Capitolo 20: Oggetto fontItem

Ereditarietà Oggetto Item > Oggetto fontItem

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto fontItem è una sottoclasse dell'oggetto Item (consultate Oggetto Item).

Riepilogo delle proprietà
Oltre a quelle dell'oggetto Item, l'oggetto fontItem dispone delle seguenti proprietà:

Proprietà	Descrizione
`fontItem.bitmap`	Specifica se l'elemento Font è bitmap.
`fontItem.bold`	Specifica se l'elemento Font è in grassetto.
`fontItem.embedVariantGlyphs`	Specifica se eseguire l'output di glifi varianti nel carattere durante la pubblicazione di un file SWF.
`fontItem.font`	Specifica il nome di un carattere dispositivo associato all'elemento Font.
`fontItem.isDefineFont4Symbol`	Specifica il formato del carattere di cui viene eseguito l'output durante la pubblicazione di un file SWF.
`fontItem.italic`	Specifica se l'elemento Font è in corsivo.
`fontItem.size`	La dimensione dell'elemento Font in punti.

fontItem.bitmap

Disponibilità
Flash CS4 Professional.

Uso
`fontItem.bitmap`

Descrizione
Proprietà; un valore booleano che specifica se l'elemento Font è bitmap (`true`) o non bitmap (`false`).

Esempio
Se il primo elemento nella libreria è un elemento Font, il codice seguente visualizza `true` nel pannello Output se l'elemento è bitmap e `false` se non lo è:

```
var theItem = fl.getDocumentDOM().library.items[0];
fl.trace("bitmap: "+ theItem.bitmap);
```

fontItem.bold

Disponibilità
Flash CS4 Professional.

Uso
`fontItem.bold`

Descrizione
Proprietà; un valore booleano che specifica se l'elemento Font è in grassetto (`true`) o non lo è (`false`).

Esempio
Se il primo elemento nella libreria è un elemento Font, il codice seguente visualizza `true` nel pannello Output se l'elemento è in grassetto e `false` se non lo è e quindi lo imposta su grassetto.

```
var theItem = fl.getDocumentDOM().library.items[0];
fl.outputPanel.clear();
fl.trace("bold: "+ theItem.bold);
theItem.bold=true;
fl.trace("bold: "+ theItem.bold);
```

fontItem.embedVariantGlyphs

Disponibilità
Flash CS4 Professional.

Uso
`fontItem.embedVariantGlyphs`

Descrizione
Proprietà; un valore booleano che specifica se eseguire (`true`) o meno (`false`) l'output di glifi varianti nel carattere durante la pubblicazione di un file SWF. Impostando questo valore su `true` si incrementa la dimensione del file SWF. Il valore predefinito è `false`.

Alcune lingue sostituiscono dinamicamente i caratteri (glifi) durante la digitazione (ad esempio, Thai, Arabo, Ebraico e Greco). Se in questi tipi di lingue si sta predispondendo o immettendo testo, impostare la proprietà su `true`.

Esempi
Simboli di caratteri compatibili con le API flash.text vengono visualizzati nella libreria e l'utente può gestirli direttamente. Tuttavia, i simboli di caratteri compatibili con le API flash.text.engine (FTE) non vengono visualizzati nella libreria e pertanto devono essere gestiti manualmente. La funzione seguente aggiunge un nuovo carattere alla libreria che può essere utilizzato con le API FTE.

```
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;
        }
    }
}
```

La funzione seguente visualizza tutti i simboli di caratteri nel pannello Output.

```
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);
            }
        }
    }
}
```

Consultate anche

fontItem.isDefineFont4Symbol, text.embedVariantGlyphs

fontItem.font

Disponibilità
Flash CS4 Professional.

Uso
`fontItem.font`

Descrizione
Proprietà; una stringa che specifica il nome di un carattere dispositivo associato all'elemento Font. Se immettete una stringa che non corrisponde a un carattere dispositivo installato, viene visualizzato un messaggio di errore. Per determinare se un carattere è presente nel sistema, potete utilizzare `fl.isFontInstalled()`.

Nota: *quando impostate questo valore, il valore della proprietà risultante potrebbe essere diverso dalla stringa immessa. Consultate l'esempio seguente.*

Esempio
Se il primo elemento nella libreria è un elemento Font, il codice seguente visualizza il nome del carattere dispositivo associato all'elemento Font e quindi lo modifica in Times:

```
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

Disponibilità
Flash CS4 Professional.

Uso
`fontItem.isDefineFont4Symbol`

Descrizione
Proprietà; un valore booleano che specifica il formato del carattere di cui viene eseguito l'output durante la pubblicazione di un file SWF. Se questo valore è `true`, Flash esegue l'output di un carattere che può essere usato con le API flash.text.engine (FTE). Se questo valore è `false`, il carattere non può essere usato con le API flash.text, compresi i campi di testo. Il valore predefinito è `false`.

Esempio
Consultate `fontItem.embedVariantGlyphs`.

fontItem.italic

Disponibilità
Flash CS4 Professional.

Uso
`fontItem.italic`

Descrizione
Proprietà; un valore booleano che specifica se l'elemento Font è in corsivo (`true`) o non lo è (`false`).

Esempio
Se il primo elemento nella libreria è un elemento Font, il codice seguente visualizza `true` nel pannello Output se l'elemento è in corsivo e `false` se non lo è e quindi lo imposta su corsivo.

```
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

Disponibilità
Flash CS4 Professional.

Uso
`fontItem.size`

Descrizione
Proprietà; un numero intero che rappresenta la dimensione dell'elemento Font in punti.

Esempio
Se il primo elemento nella libreria è un elemento Font, il codice seguente visualizza la dimensione dell'elemento in punti nel pannello Output e quindi imposta l'elemento su 24 punti.

```
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);
```

Capitolo 21: Oggetto Frame

Disponibilità

Flash MX 2004.

Descrizione

L'oggetto Frame rappresenta i fotogrammi del livello.

Riepilogo dei metodi

Con l'oggetto Frame è possibile utilizzare i metodi seguenti:

Metodo	Descrizione
`frame.getCustomEase()`	Restituisce un array di oggetti JavaScript, ognuno dei quali dispone di una proprietà *x* e *y*.
`frame.setCustomEase()`	Specifica la curva di Bézier cubica da utilizzare come curva di andamento personalizzata.

Riepilogo delle proprietà

Le seguenti proprietà possono essere utilizzate con l'oggetto Frame:

Proprietà	Descrizione
`frame.actionScript`	Una stringa che rappresenta il codice ActionScript.
`frame.duration`	Di sola lettura; un numero intero che rappresenta il numero di fotogrammi in una sequenza di fotogrammi.
`frame.elements`	Sola lettura; un array di oggetti Element (consultate Oggetto Element).
`frame.hasCustomEase`	Un valore booleano che specifica se il fotogramma ottiene le informazioni sull'andamento dalla curva di andamento personalizzata.
`frame.labelType`	Una stringa che specifica il tipo di nome per l'oggetto Frame.
`frame.motionTweenOrientToPath`	Un valore booleano che specifica se l'elemento interpolato ruota mentre si muove lungo un percorso per mantenere la propria angolazione rispetto a ogni punto sul percorso.
`frame.motionTweenRotate`	Una stringa che specifica il modo in cui ruota l'elemento interpolato.
`frame.motionTweenRotateTimes`	Un numero intero che specifica il numero di rotazioni effettuate dall'elemento interpolato nell'intervallo tra il fotogramma chiave iniziale e quello successivo.
`frame.motionTweenScale`	Un valore booleano che specifica se l'elemento interpolato viene ridimensionato in scala fino a raggiungere le dimensioni dell'oggetto nel fotogramma chiave successivo, aumentando le proprie dimensioni con ogni fotogramma dell'interpolazione (`true`), oppure se non viene ridimensionato in scala (`false`).
`frame.motionTweenSnap`	Un valore booleano che specifica se l'elemento interpolato si aggancia automaticamente al punto più vicino sul livello guida di movimento associato al livello del fotogramma (`true`) oppure no (`false`).
`frame.motionTweenSync`	Un valore booleano; se è impostato su `true`, sincronizza l'animazione dell'oggetto interpolato con la linea temporale principale.

Proprietà	Descrizione
`frame.name`	Una stringa che specifica il nome del fotogramma.
`frame.shapeTweenBlend`	Una stringa che specifica il modo in cui un'interpolazione di forma viene fusa tra la forma nel fotogramma chiave all'inizio dell'interpolazione e la forma nel fotogramma chiave successivo.
`frame.soundEffect`	Una stringa che specifica gli effetti per un suono associato direttamente a un fotogramma (`frame.soundLibraryItem`).
`frame.soundLibraryItem`	Un elemento della libreria (consultate Oggetto SoundItem) utilizzato per creare un suono.
`frame.soundLoop`	Un valore intero che specifica il numero di riproduzioni di un suono associato direttamente a un fotogramma (`frame.soundLibraryItem`).
`frame.soundLoopMode`	Una stringa che specifica se un suono associato direttamente a un fotogramma (`frame.soundLibraryItem`) deve essere riprodotto un numero specificato di volte oppure ripetuto ciclicamente.
`frame.soundName`	Una stringa che specifica il nome di un suono memorizzato nella libreria e associato direttamente a un fotogramma (`frame.soundLibraryItem`).
`frame.soundSync`	Una stringa che specifica il comportamento di sincronizzazione per un suono associato direttamente a un fotogramma (`frame.soundLibraryItem`).
`frame.startFrame`	Di sola lettura; l'indice del primo fotogramma di una sequenza.
`frame.tweenEasing`	Un numero intero che specifica il valore di andamento da applicare all'oggetto interpolato.
`frame.tweenType`	Una stringa che specifica il tipo di interpolazione.
`frame.useSingleEaseCurve`	Un valore booleano che specifica se per le informazioni sull'andamento di tutte le proprietà viene utilizzata una sola curva di andamento personalizzata.

frame.actionScript

Disponibilità

Flash MX 2004.

Uso

```
frame.actionScript
```

Descrizione

Proprietà; una stringa che rappresenta il codice ActionScript. Per inserire un carattere di nuova riga, usate `"\n"`.

Esempio

L'esempio seguente assegna `stop()` all'azione del primo fotogramma del primo livello:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].actionScript = 'stop();';
```

frame.duration

Disponibilità
Flash MX 2004.

Uso
`frame.duration`

Descrizione
Proprietà di sola lettura; un numero intero che rappresenta il numero di fotogrammi in una sequenza di fotogrammi.

Esempio
L'esempio seguente memorizza nella variabile `frameSpan` il numero di fotogrammi di una sequenza che inizia al primo fotogramma del primo livello:

```
var frameSpan = fl.getDocumentDOM().getTimeline().layers[0].frames[0].duration;
```

frame.elements

Disponibilità
Flash MX 2004.

Uso
`frame.elements`

Descrizione
Proprietà di sola lettura; un array di oggetti Element (consultate Oggetto Element). L'ordine degli elementi è quello in cui sono memorizzati nel file FLA. Se sullo stage sono presenti più forme e ognuna di esse non è raggruppata, Flash le considera come un unico elemento. Se ogni forma è raggruppata e pertanto sullo stage sono presenti più gruppi, Flash le considera come elementi separati. In altre parole, Flash considera le forme originarie e non raggruppate come un elemento singolo, a prescindere da quante forme separate sono presenti sullo stage. Se, ad esempio, un fotogramma contiene tre forme originarie e non raggruppate, la proprietà `elements.length` del fotogramma restituisce il valore 1. Per risolvere questo inconveniente, selezionare ogni forma singolarmente, quindi raggrupparle.

Esempio
L'esempio seguente memorizza nella variabile `myElements` un array degli elementi presenti nel primo fotogramma del primo livello:

```
var myElements = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements;
```

frame.getCustomEase()

Disponibilità
Flash 8.

Uso

```
Frame.getCustomEase([property])
```

Parametri

`property` Una stringa opzionale che specifica la proprietà per cui deve essere restituito il valore di andamento personalizzato. I valori accettabili sono `"all"`, `"position"`, `"rotation"`, `"scale"`, `"color"` e `"filters"`. Il valore predefinito è `"all"`.

Restituisce

Restituisce un array di oggetti JavaScript, ognuno dei quali ha una proprietà *x* e *y*.

Descrizione

Metodo; restituisce un array di oggetti che rappresentano i punti di controllo della curva di Bézier cubica che definisce la curva di andamento.

Esempio

L'esempio seguente restituisce il valore di andamento personalizzato della proprietà `position` per il primo fotogramma del primo livello:

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]
var easeArray = theFrame.getCustomEase("position");
```

Consultate anche

`frame.hasCustomEase`, `frame.setCustomEase()`, `frame.useSingleEaseCurve`

frame.hasCustomEase

Disponibilità

Flash 8.

Uso

```
frame.hasCustomEase
```

Descrizione

Proprietà; un valore booleano. Se è `true`, il fotogramma ottiene le informazioni sull'andamento dalla curva di andamento personalizzata. Se è `false`, il fotogramma ottiene le informazioni sull'andamento dal valore di andamento.

Esempio

L'esempio seguente specifica che il primo fotogramma del primo livello deve ottenere le informazioni sull'andamento dal valore di andamento anziché dalla curva di andamento personalizzata:

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]
theFrame.hasCustomEase = false;
```

Consultate anche

`frame.getCustomEase()`, `frame.setCustomEase()`, `frame.useSingleEaseCurve`

frame.labelType

Disponibilità
Flash MX 2004.

Uso
`frame.labelType`

Descrizione
Proprietà; una stringa che specifica il tipo di nome per il fotogramma. I valori accettabili sono `"none"`, `"name"`, `"comment"` e `"anchor"`. Se impostate un'etichetta su `"none"`, la proprietà `frame.name` viene cancellata.

Esempio
L'esempio seguente imposta `"First Frame"` come nome del primo fotogramma del primo livello, quindi imposta `"comment"` come valore dell'etichetta:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].name = 'First Frame';
fl.getDocumentDOM().getTimeline().layers[0].frames[0].labelType = 'comment';
```

frame.motionTweenOrientToPath

Disponibilità
Flash MX 2004.

Uso
`frame.motionTweenOrientToPath`

Descrizione
Proprietà; un valore booleano che specifica se l'elemento interpolato ruota mentre si muove lungo un percorso per mantenere la propria angolazione rispetto a ogni punto sul percorso (`true`) oppure se non ruota (`false`).

Se desiderate specificare un valore per questa proprietà, impostate `frame.motionTweenRotate` su `"none"`.

frame.motionTweenRotate

Disponibilità
Flash MX 2004.

Uso
`frame.motionTweenRotate`

Descrizione
Proprietà; una stringa che specifica il modo in cui ruota l'elemento interpolato. I valori accettabili sono `"none"`, `"auto"`, `"clockwise"` e `"counter-clockwise"`. Il valore `"auto"` indica che l'oggetto ruota nella direzione che richiede il movimento minimo per corrispondere alla rotazione dell'oggetto presente nel fotogramma chiave successivo.

Se desiderate specificare un valore per la proprietà `frame.motionTweenOrientToPath`, utilizzate `"none"`.

Esempio
Consultate `frame.motionTweenRotateTimes`.

frame.motionTweenRotateTimes

Disponibilità
Flash MX 2004.

Uso
```
frame.motionTweenRotateTimes
```

Descrizione
Proprietà; un numero intero che specifica il numero di rotazioni effettuate dall'elemento interpolato nell'intervallo tra il fotogramma chiave iniziale e quello successivo.

Esempio
L'esempio seguente ruota l'elemento presente nel fotogramma in senso antiorario per tre volte prima che raggiunga il fotogramma chiave successivo:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenRotate = "counter-
clockwise";
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenRotateTimes = 3;
```

frame.motionTweenScale

Disponibilità
Flash MX 2004.

Uso
```
frame.motionTweenScale
```

Descrizione
Proprietà; un valore booleano che specifica se l'elemento interpolato viene ridimensionato in scala fino a raggiungere le dimensioni dell'oggetto nel fotogramma chiave successivo, aumentando le proprie dimensioni con ogni fotogramma dell'interpolazione (`true`), oppure se non viene ridimensionato in scala (`false`).

Esempio
L'esempio seguente specifica che l'elemento interpolato deve essere modificato in scala fino a raggiungere le dimensioni dell'oggetto nel fotogramma chiave successivo, aumentando le proprie dimensioni con ogni fotogramma dell'interpolazione.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenScale = true;
```

frame.motionTweenSnap

Disponibilità
Flash MX 2004.

Uso
```
frame.motionTweenSnap
```

Descrizione
Proprietà; un valore booleano che specifica se l'elemento interpolato si aggancia automaticamente al punto più vicino sul livello guida di movimento associato al livello del fotogramma (`true`) oppure no (`false`).

frame.motionTweenSync

Disponibilità
Flash MX 2004.

Uso
```
frame.motionTweenSync
```

Descrizione
Proprietà; un valore booleano. Se è impostato su `true`, sincronizza l'animazione dell'oggetto interpolato con la linea temporale principale.

Esempio
L'esempio seguente specifica che l'oggetto interpolato deve essere sincronizzato con la linea temporale:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenSync = true;
```

frame.name

Disponibilità
Flash MX 2004.

Uso
```
frame.name
```

Descrizione
Proprietà; una stringa che specifica il nome del fotogramma.

Esempio
L'esempio seguente imposta `"First Frame"` come nome del primo fotogramma del primo livello, quindi memorizza il valore `name` nella variabile `frameLabel`:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].name = 'First Frame';
var frameLabel = fl.getDocumentDOM().getTimeline().layers[0].frames[0].name;
```

frame.setCustomEase()

Disponibilità
Flash 8.

Uso
```
frame.setCustomEase(property, easeCurve)
```

Parametri
`property` Una stringa che specifica la proprietà per cui deve essere utilizzata la curva di andamento. I valori accettabili sono `"all"`, `"position"`, `"rotation"`, `"scale"`, `"color"` e `"filters"`.

`easeCurve` Un array di oggetti che definisce la curva di andamento. Ogni elemento dell'array deve essere costituito da un oggetto JavaScript con le proprietà *x* e *y*.

Restituisce
Nulla.

Descrizione
Metodo; specifica un array di coordinate dei punti di controllo e dei punti finali tangenti che descrivono una curva di Bézier cubica da utilizzare come curva di andamento personalizzata. Questo array è determinato dalla posizione orizzontale (ordinale: da sinistra a destra) dei punti di controllo e dei punti finali tangenti.

Esempio
L'esempio seguente imposta una curva di Bézier specificata dall'array `easeCurve` come curva di andamento di tutte le proprietà del primo fotogramma del primo livello:

```
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 );
```

Consultate anche
`frame.getCustomEase()`, `frame.hasCustomEase`, `frame.useSingleEaseCurve`

frame.shapeTweenBlend

Disponibilità
Flash MX 2004.

Uso
`frame.shapeTweenBlend`

Descrizione
Proprietà; una stringa che specifica il modo in cui un'interpolazione di forma viene fusa tra la forma nel fotogramma chiave all'inizio dell'interpolazione e la forma nel fotogramma chiave successivo. I valori accettabili sono `"distributive"` e `"angular"`.

frame.soundEffect

Disponibilità
Flash MX 2004.

Uso
`frame.soundEffect`

Descrizione
Una stringa che specifica gli effetti per un suono associato direttamente a un fotogramma (`frame.soundLibraryItem`). I valori accettabili sono `"none"`, `"left channel"`, `"right channel"`, `"fade left to right"`, `"fade right to left"`, `"fade in"`, `"fade out"` e `"custom"`.

Esempio
L'esempio seguente specifica una dissolvenza in entrata per il suono associato al primo fotogramma:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundEffect = "fade in";
```

frame.soundLibraryItem

Disponibilità
Flash MX 2004.

Uso
`frame.soundLibraryItem`

Descrizione
Proprietà; un elemento della libreria (consultate Oggetto SoundItem) utilizzato per creare un suono. Il suono viene associato direttamente al fotogramma.

Esempio

L'esempio seguente assegna il primo elemento della libreria alla proprietà `soundLibraryItem` del primo fotogramma:

```
// 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

Disponibilità

Flash MX 2004.

Uso

`frame.soundLoop`

Descrizione

Proprietà; un valore intero che specifica il numero di riproduzioni di un suono associato direttamente a un fotogramma (`frame.soundLibraryItem`). Se desiderate specificare un valore per la proprietà, impostate `frame.soundLoopMode` su `"repeat"`.

Esempio

Consultate `frame.soundLoopMode`.

frame.soundLoopMode

Disponibilità

Flash MX 2004.

Uso

`frame.soundLoopMode`

Descrizione

Proprietà; una stringa che specifica se un suono associato direttamente a un fotogramma (`frame.soundLibraryItem`) deve essere riprodotto un numero specificato di volte oppure ripetuto ciclicamente. I valori accettabili sono `"repeat"` e `"loop"`. Per specificare il numero di riproduzioni del suono, impostate un valore per `frame.soundLoop`.

Esempio

L'esempio seguente specifica che un suono deve essere riprodotto due volte:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundLoopMode = "repeat";
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundLoop = 2;
```

frame.soundName

Disponibilità
Flash MX 2004.

Uso
`frame.soundName`

Descrizione
Proprietà; una stringa che specifica il nome di un suono memorizzato nella libreria e associato direttamente a un fotogramma (`frame.soundLibraryItem`).

Esempio
L'esempio seguente imposta la proprietà `soundName` del primo fotogramma su `"song1.mp3"` (il file song1.mp3 deve essere presente nella libreria):

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundName = "song1.mp3";
```

frame.soundSync

Disponibilità
Flash MX 2004.

Uso
`frame.soundSync`

Descrizione
Proprietà: una stringa che specifica il comportamento di sincronizzazione per un suono associato direttamente a un fotogramma (`frame.soundLibraryItem`). I valori accettabili sono `"event"`, `"stop"`, `"start"` e `"stream"`.

Esempio
L'esempio seguente specifica che un suono deve essere riprodotto in streaming:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundSync = 'stream';
```

frame.startFrame

Disponibilità
Flash MX 2004.

Uso
`frame.startFrame`

Descrizione
Proprietà di sola lettura; l'indice del primo fotogramma di una sequenza.

Esempio

Nell'esempio seguente, `stFrame` è l'indice del primo fotogramma di una sequenza. In questo esempio, una sequenza di fotogrammi occupa i sei fotogrammi compresi tra il fotogramma 5 e il fotogramma 10. Pertanto, il valore di `stFrame` per qualunque fotogramma compreso tra il fotogramma 5 e il fotogramma 10 è 4 (è importante ricordare che i valori di indice sono diversi dai valori dei numeri di fotogramma).

```
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

Disponibilità

Flash MX 2004.

Uso

```
frame.tweenEasing
```

Descrizione

Proprietà; un numero intero che specifica il valore di andamento da applicare all'oggetto interpolato. I valori accettabili sono compresi tra -100 e 100. Per avviare l'interpolazione di movimento lentamente e accelerarla verso la fine dell'animazione, immettete un valore compreso tra -1 e -100. Per avviare l'interpolazione di movimento rapidamente e rallentarla verso la fine dell'animazione, immettete un valore positivo compreso tra 1 e 100.

Esempio

L'esempio seguente specifica che il movimento dell'oggetto interpolato deve iniziare abbastanza rapidamente e rallentare verso la fine dell'animazione:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].tweenEasing = 50;
```

frame.tweenType

Disponibilità

Flash MX 2004.

Uso

```
frame.tweenType
```

Descrizione

Proprietà; una stringa che specifica il tipo di interpolazione; i valori accettabili sono `"motion"`, `"shape"` e `"none"`. Il valore `"none"` rimuove l'interpolazione di movimento. Utilizzate il metodo `timeline.createMotionTween()` per creare un'interpolazione di movimento.

Se specificate `"motion"`, l'oggetto nel fotogramma deve essere un simbolo, un campo di testo o un oggetto raggruppato. Viene applicata un'interpolazione tra la sua posizione nel fotogramma chiave e quella indicata nel fotogramma chiave successivo.

Se specificate `"shape"`, l'oggetto nel fotogramma deve essere una forma. Viene effettuata una fusione tra la forma originale nel fotogramma chiave e la forma indicata nel fotogramma chiave successivo.

Esempio

L'esempio seguente specifica che l'oggetto è un'interpolazione di movimento, pertanto deve essere applicata un'interpolazione tra la sua posizione nel fotogramma chiave e quella indicata nel fotogramma chiave successivo.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].tweenType = "motion";
```

frame.useSingleEaseCurve

Disponibilità

Flash 8.

Uso

```
frame.useSingleEaseCurve
```

Descrizione

Proprietà; un valore booleano. Se è `true`, per le informazioni sull'andamento di tutte le proprietà viene utilizzata una sola curva di andamento personalizzata. Se è `false`, ogni proprietà ha una curva di andamento specifica.

Questa proprietà viene ignorata se al fotogramma non è stato applicato un andamento personalizzato.

Esempio

L'esempio seguente specifica che una singola curva di andamento personalizzata deve essere utilizzata per tutte le proprietà del primo fotogramma del primo livello:

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]
theFrame.useSingleEaseCurve = true;
```

Consultate anche

`frame.getCustomEase()`, `frame.hasCustomEase`, `frame.setCustomEase()`

Capitolo 22: Oggetto HalfEdge

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto HalfEdge corrisponde al lato indicato del bordo di un Oggetto Shape. Un bordo ha due mezzi bordi. È possibile seguire i contorni di una forma muovendosi lungo questi mezzi bordi. Ad esempio, se partite da un mezzo bordo, potete tracciare tutti i mezzi bordi attorno al contorno di una forma e ritornare a quello originale.

I mezzi bordi sono disposti in ordine. Un mezzo bordo rappresenta un lato del bordo; l'altro rappresenta l'altro lato.

Riepilogo dei metodi
I seguenti metodi sono disponibili per l'oggetto HalfEdge:

Metodo	Descrizione
`halfEdge.getEdge()`	Ottiene l'Oggetto Edge dell'oggetto HalfEdge.
`halfEdge.getNext()`	Ottiene il mezzo bordo successivo sul contorno corrente.
`halfEdge.getOppositeHalfEdge()`	Ottiene l'oggetto HalfEdge sul lato opposto del bordo.
`halfEdge.getPrev()`	Ottiene l'oggetto HalfEdge precedente sul contorno corrente.
`halfEdge.getVertex()`	Ottiene l'Oggetto Vertex all'estremità dell'oggetto HalfEdge.

Riepilogo delle proprietà
Le seguenti proprietà sono disponibili per l'oggetto HalfEdge:

Proprietà	Descrizione
`halfEdge.id`	Di sola lettura; un identificatore intero univoco per l'oggetto HalfEdge.
`halfEdge.index`	Un numero intero con valore 0 o 1 che specifica l'indice di questo oggetto HalfEdge nel bordo principale.

halfEdge.getEdge()

Disponibilità
Flash MX 2004.

Uso
```
halfEdge.getEdge()
```

Parametri
Nessuno.

Restituisce
Un Oggetto Edge.

Descrizione
Metodo; ottiene l'oggetto Edge dell'oggetto HalfEdge. Consultate Oggetto Edge.

Esempio
L'esempio seguente illustra come si ottengono un bordo e un mezzo bordo per la forma specificata:

```
var shape = fl.getDocumentDOM().selection[0];
var hEdge = shape.edges[0].getHalfEdge(0);
var edge = hEdge.getEdge();
```

halfEdge.getNext()

Disponibilità
Flash MX 2004.

Uso
`halfEdge.getNext()`

Parametri
Nessuno.

Restituisce
Un oggetto HalfEdge.

Descrizione
Metodo; ottiene il mezzo bordo successivo sul contorno corrente.

Nota: *i mezzi bordi hanno una direzione e un ordine di sequenza, mentre i bordi no.*

Esempio
L'esempio seguente memorizza il mezzo bordo successivo per il contorno specificato nella variabile `nextHalfEdge`:

```
var shape = fl.getDocumentDOM().selection[0];
var hEdge = shape.edges[0].getHalfEdge( 0 );
var nextHalfEdge = hEdge.getNext();
```

halfEdge.getOppositeHalfEdge()

Disponibilità
Flash MX 2004.

Uso
`halfEdge.getOppositeHalfEdge()`

Parametri
Nessuno.

Restituisce
Un oggetto HalfEdge.

Descrizione
Metodo; ottiene l'oggetto HalfEdge sul lato opposto del bordo.

Esempio
L'esempio seguente memorizza il mezzo bordo opposto a `hEdge` nella variabile `otherHalfEdge`:

```
var shape = fl.getDocumentDOM().selection[0];
var hEdge = shape.edges[0].getHalfEdge(0);
var otherHalfEdge = hEdge.getOppositeHalfEdge();
```

halfEdge.getPrev()

Disponibilità
Flash MX 2004.

Uso
`halfEdge.getPrev()`

Parametri
Nessuno.

Restituisce
Un oggetto HalfEdge.

Descrizione
Metodo; ottiene l'oggetto HalfEdge precedente sul contorno corrente.

Nota: *i mezzi bordi hanno una direzione e un ordine di sequenza, mentre i bordi no.*

Esempio
L'esempio seguente memorizza il mezzo bordo precedente per il contorno specificato nella variabile `prevHalfEdge`:

```
var shape = fl.getDocumentDOM().selection[0];
var hEdge = shape.edges[0].getHalfEdge( 0 );
var prevHalfEdge = hEdge.getPrev();
```

halfEdge.getVertex()

Disponibilità
Flash MX 2004.

Uso
`halfEdge.getVertex()`

Parametri
Nessuno.

Restituisce
Un Oggetto Vertex

Descrizione
Metodo; ottiene l'oggetto Vertex all'estremità dell'oggetto HalfEdge. Consultate Oggetto Vertex

Esempio
L'esempio seguente memorizza l'oggetto Vertex all'estremità di `hEdge` nella variabile `vertex`:

```
var shape = fl.getDocumentDOM().selection[0];
var edge = shape.edges[0];
var hEdge = edge.getHalfEdge(0);
var vertex = hEdge.getVertex();
```

halfEdge.id

Disponibilità
Flash MX 2004.

Uso
`halfEdge.id`

Descrizione
Proprietà di sola lettura; un identificatore intero univoco per l'oggetto HalfEdge.

Esempio
L'esempio seguente visualizza un identificatore univoco per il mezzo bordo specificato nel pannello Output:

```
var shape = fl.getDocumentDOM().selection[0];
alert(shape.contours[0].getHalfEdge().id);
```

halfEdge.index

Disponibilità
Flash MX 2004.

Uso
`halfEdge.index`

Descrizione

Proprietà di sola lettura; un numero intero con valore 0 o 1 che specifica l'indice di questo oggetto HalfEdge nel bordo principale.

Esempio

L'esempio seguente visualizza un valore di indice per il mezzo bordo specificato nel pannello Output:

```
var shape = fl.getDocumentDOM().selection[0];
var hEdge = shape.edges[0].getHalfEdge(0);
var heIndex = hEdge.index;
```

Capitolo 23: Oggetto Instance

Ereditarietà Oggetto Element > Oggetto Instance

Disponibilità
Flash MX 2004.

Descrizione
Instance è una sottoclasse dell'Oggetto Element.

Riepilogo delle proprietà
Oltre a tutte quelle dell'oggetto Element, l'oggetto Instance dispone delle seguenti proprietà:

Proprietà	Descrizione
`instance.instanceType`	Di sola lettura; una stringa che rappresenta il tipo di istanza.
`instance.libraryItem`	Elemento della libreria utilizzato per creare questa istanza.

instance.instanceType

Disponibilità
Flash MX 2004; valore possibile di `"video"` aggiunto in Flash 8.

Uso
```
instance.instanceType
```

Descrizione
Proprietà di sola lettura; una stringa che rappresenta il tipo di istanza. I valori possibili sono `"symbol"`, `"bitmap"`, `"embedded video"`, `"linked video"`, `"video"` e `"compiled clip"`.

In Flash MX 2004, il valore di `instance.instanceType` per un elemento aggiunto alla libreria mediante `library.addNewItem("video")` è `"embedded_video"`. In Flash 8 e versioni successive, il valore è `"video"`. Consultate `library.addNewItem()`.

Esempio
L'esempio seguente mostra che il tipo di istanza di un clip filmato è `symbol`:

```
// 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

Disponibilità
Flash MX 2004.

Uso
`instance.libraryItem`

Descrizione
Proprietà; un elemento di libreria utilizzato per creare questa istanza. È possibile modificare questa proprietà solo in un altro elemento di libreria dello stesso tipo (in altre parole, non è possibile impostare un'istanza `symbol` in modo che faccia riferimento a una bitmap). Consultate Oggetto library.

Esempio
L'esempio seguente modifica il simbolo selezionato in modo che faccia riferimento al primo elemento nella libreria:

```
fl.getDocumentDOM().selection[0].libraryItem = fl.getDocumentDOM().library.items[0];
```

Capitolo 24: Oggetto Item

Disponibilità

Flash MX 2004.

Descrizione

L'oggetto Item è una classe base astratta. Qualunque elemento presente nella libreria deriva da Item. Consultate anche Oggetto library.

Riepilogo dei metodi

I seguenti metodi sono disponibili per l'oggetto Item:

Metodo	Descrizione
`item.addData()`	Aggiunge i dati specificati a un elemento della libreria.
`item.getData()`	Recupera il valore dei dati specificati.
`item.hasData()`	Determina se l'elemento della libreria contiene i dati indicati.
`item.removeData()`	Rimuove i dati persistenti dall'elemento della libreria.

Riepilogo delle proprietà

Le proprietà seguenti sono disponibili per l'oggetto Item:

Proprietà	Descrizione
`item.itemType`	Di sola lettura; una stringa che specifica il tipo di elemento.
`item.linkageBaseClass`	Una stringa che specifica la classe ActionScript 3.0 che verrà associata al simbolo.
`item.linkageClassName`	Una stringa che specifica la classe ActionScript 2.0 che verrà associata al simbolo.
`item.linkageExportForAS`	Un valore booleano. Se è `true`, l'elemento viene esportato per ActionScript.
`item.linkageExportForRS`	Un valore booleano. Se è `true`, l'elemento viene esportato per la condivisione in runtime.
`item.linkageExportInFirstFrame`	Un valore booleano. Se è `true`, l'elemento viene esportato nel primo fotogramma.
`item.linkageIdentifier`	Una stringa che specifica il nome che verrà utilizzato da Flash per identificare l'elemento quando viene associato al file SWF di destinazione.
`item.linkageImportForRS`	Un valore booleano. Se è `true`, l'elemento viene importato per la condivisione in runtime.
`item.linkageURL`	Una stringa che specifica l'URL del file SWF che contiene l'elemento condiviso.
`item.name`	Una stringa che specifica il nome dell'elemento della libreria. La stringa comprende la struttura delle cartelle.

item.addData()

Disponibilità
Flash MX 2004.

Uso
```
item.addData(name, type, data)
```

Parametri
`name` Una stringa che specifica il nome dei dati.

`type` Una stringa che specifica il tipo dei dati. I tipi validi sono `"integer"`, `"integerArray"`, `"double"`, `"doubleArray"`, `"string"` e `"byteArray"`.

`data` I dati da aggiungere all'elemento della libreria specificato. Il tipo di dati dipende dal valore del parametro type. Ad esempio, se type è "`integer`", il valore dei dati deve essere un numero intero, e così via.

Restituisce
Nulla.

Descrizione
Metodo; aggiunge i dati specificati a un elemento della libreria.

Esempio
L'esempio seguente aggiunge al primo elemento della libreria i dati denominati `myData` con un valore intero pari a 12:

```
fl.getDocumentDOM().library.items[0].addData("myData", "integer", 12);
```

item.getData()

Disponibilità
Flash MX 2004.

Uso
```
item.getData(name)
```

Parametri
`name` Una stringa che specifica il nome dei dati da recuperare.

Restituisce
I dati specificati dal parametro *name*. Il tipo di dati restituiti dipende dal tipo di dati memorizzati.

Descrizione
Metodo; recupera il valore dei dati specificati.

Esempio
L'esempio seguente ottiene il valore dei dati denominati `myData` dal primo elemento della libreria e lo memorizza nella variabile `libData`:

```
var libData = fl.getDocumentDOM().library.items[0].getData("myData");
```

item.hasData()

Disponibilità
Flash MX 2004.

Uso
`item.hasData(name)`

Parametri
`name` Una stringa che specifica il nome dei dati da verificare nell'elemento della libreria.

Restituisce
Un valore booleano: `true` se i dati specificati esistono, `false` in caso contrario.

Descrizione
Metodo; determina se l'elemento della libreria contiene i dati indicati.

Esempio
L'esempio seguente visualizza un messaggio nel pannello Output se il primo elemento della libreria contiene dati denominati `myData`:

```
if (fl.getDocumentDOM().library.items[0].hasData("myData")){
    fl.trace("Yep, it's there!");
}
```

item.itemType

Disponibilità
Flash MX 2004.

Uso
`item.itemType`

Descrizione
Proprietà di sola lettura; una stringa che specifica il tipo di elemento. Il valore è uno dei seguenti: `"undefined"`, `"component"`, `"movie clip"`, `"graphic"`, `"button"`, `"folder"`, `"font"`, `"sound"`, `"bitmap"`, `"compiled clip"`, `"screen"` o `"video"`. Se questa proprietà è `"video"`, potete determinare il tipo di video. Consultate `videoItem.videoType`.

Esempio

L'esempio seguente visualizza il tipo di elemento di libreria specificato nel pannello Output:

```
fl.trace(fl.getDocumentDOM().library.items[0].itemType);
```

item.linkageBaseClass

Disponibilità

Flash CS3 Professional.

Uso

```
item.linkageBaseClass
```

Descrizione

Proprietà; una stringa che specifica la classe ActionScript 3.0 che verrà associata al simbolo. Il valore qui specificato compare nella finestra di dialogo Concatenamento dell'ambiente di creazione, oltre che in altre finestre di dialogo in cui sono presenti i controlli di concatenamento, ad esempio nella finestra di dialogo Proprietà simbolo. Per specificare questo valore per una classe ActionScript 2.0, dovete usare `item.linkageClassName`.

Se la classe di base è quella predefinita per il tipo di simbolo (ad esempio "flash.display.MovieClip" per i clip filmati, "flash.display.SimpleButton" per i pulsanti e così via), questa proprietà è una stringa vuota (""). Analogamente, per specificare un elemento come classe di base predefinita, impostate questo valore su una stringa vuota.

Durante l'impostazione di questo valore, non viene eseguita nessuna delle verifiche eseguite dalla finestra di dialogo Concatenamento; inoltre, se Flash non è in grado di impostare la classe di base sul valore specificato, non viene generato alcun errore. Ad esempio, se questo valore viene impostato nella finestra di dialogo Concatenamento, viene controllato se la classe di base è presente nel percorso di classe del file FLA. Ciò garantisce che ActionScript 3.0 sia selezionato nella scheda Flash della finestra di dialogo Impostazioni pubblicazione e così via. Queste verifiche non vengono eseguite se la proprietà viene impostata tramite uno script.

Esempio

Le righe di codice che seguono mostrano alcuni modi di impiego di questa proprietà:

```
// 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);
```

Consultate anche

`document.docClass`

item.linkageClassName

Disponibilità

Flash MX 2004.

Uso
`item.linkageClassName`

Descrizione
Proprietà; una stringa che specifica la classe ActionScript 2.0 che verrà associata al simbolo. Per specificare questo valore per una classe ActionScript 3.0, dovete usare `item.linkageBaseClass`.

Per definire questa proprietà, le proprietà `item.linkageExportForAS` e/o `item.linkageExportForRS` devono essere impostate su `true`, mentre la proprietà `item.linkageImportForRS` deve essere impostata su `false`.

Esempio
L'esempio seguente specifica che il nome della classe ActionScript 2.0 associata al primo elemento della libreria è `myClass`:

```
fl.getDocumentDOM().library.items[0].linkageClassName = "myClass";
```

item.linkageExportForAS

Disponibilità
Flash MX 2004.

Uso
`item.linkageExportForAS`

Descrizione
Proprietà; un valore booleano. Se la proprietà è `true`, l'elemento viene esportato per ActionScript. Potete inoltre impostare le proprietà `item.linkageExportForRS` e `item.linkageExportInFirstFrame` su `true`.

Se impostate questa proprietà su true, la proprietà `item.linkageImportForRS` deve essere impostata su `false`. Inoltre, dovete specificare un identificatore (`item.linkageIdentifier`) e un URL (`item.linkageURL`).

Esempio
L'esempio seguente imposta questa proprietà per l'elemento di libreria specificato:

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

item.linkageExportForRS

Disponibilità
Flash MX 2004.

Uso
`item.linkageExportForRS`

Descrizione
Proprietà; un valore booleano. Se la proprietà è `true`, l'elemento viene esportato per la condivisione in runtime. Potete inoltre impostare le proprietà `item.linkageExportForAS` e `item.linkageExportInFirstFrame` su `true`.

Se impostate questa proprietà su true, la proprietà `item.linkageImportForRS` deve essere impostata su `false`. Inoltre, dovete specificare un identificatore (`item.linkageIdentifier`) e un URL (`item.linkageURL`).

Esempio
L'esempio seguente imposta questa proprietà per l'elemento di libreria specificato:

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

item.linkageExportInFirstFrame

Disponibilità
Flash MX 2004.

Uso
`item.linkageExportInFirstFrame`

Descrizione
Proprietà; un valore booleano. Se è `true`, l'elemento viene esportato nel primo fotogramma; se è `false`, l'elemento viene esportato nel fotogramma della prima istanza. Se l'elemento non è visualizzato sullo stage, non viene esportato.

Questa proprietà può essere impostata su `true` solo se `item.linkageExportForAS` e/o `item.linkageExportForRS` sono impostate su `true`.

Esempio
L'esempio seguente specifica che l'elemento della libreria specificato viene esportato nel primo fotogramma:

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

item.linkageIdentifier

Disponibilità
Flash MX 2004.

Uso
`item.linkageIdentifier`

Descrizione
Proprietà; una stringa che specifica il nome che verrà usato da Flash per identificare l'elemento quando viene concatenato al file SWF di destinazione. Questa proprietà viene ignorata da Flash se `item.linkageImportForRS`, `item.linkageExportForAS` e `item.linkageExportForRS` sono impostate su `false`. Al contrario, la proprietà deve essere impostata se una qualsiasi delle proprietà elencate in precedenza è impostata su `true`.

Esempio
L'esempio seguente specifica che la stringa `my_mc` viene utilizzata per identificare l'elemento della libreria quando è collegato al file SWF di destinazione in cui lo state esportando:

```
fl.getDocumentDOM().library.items[0].linkageIdentifier = "my_mc";
```

Consultate anche
item.linkageURL

item.linkageImportForRS

Disponibilità
Flash MX 2004.

Uso
```
item.linkageImportForRS
```

Descrizione
Proprietà; valore booleano. Se è `true`, l'elemento viene importato per la condivisione in runtime. Se questa proprietà è impostata su `true`, sia `item.linkageExportForAS` che `item.linkageExportForRS` devono essere impostate su `false`. Inoltre, dovete specificare un identificatore (`item.linkageIdentifier`) e un URL (`item.linkageURL`).

Esempio
L'esempio seguente imposta questa proprietà su `true` per l'elemento di libreria specificato:

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

item.linkageURL

Disponibilità
Flash MX 2004.

Uso
```
item.linkageURL
```

Descrizione
Proprietà; una stringa che specifica l'URL del file SWF che contiene l'elemento condiviso. Questa proprietà viene ignorata da Flash se `item.linkageImportForRS`, `item.linkageExportForAS` e `item.linkageExportForRS` sono impostate su `false`. Al contrario, la proprietà deve essere impostata se una qualsiasi delle proprietà elencate in precedenza è impostata su `true`. È possibile specificare un URL Web o un nome di file in un formato dipendente dalla piattaforma, vale a dire con barra (/) o barra rovesciata (\) a seconda della piattaforma.

Esempio
L'esempio seguente specifica un URL di concatenamento per l'elemento di libreria specificato:

```
fl.getDocumentDOM().library.items[0].linkageURL = "theShareSWF.swf";
```

Consultate anche
item.linkageIdentifier

item.name

Disponibilità
Flash MX 2004.

Uso
`item.name`

Descrizione
Metodo; una stringa che specifica il nome dell'elemento della libreria, che comprende la struttura delle cartelle. Ad esempio, se Simbolo_1 è all'interno di una cartella denominata Cartella_1, la proprietà `name` di Simbolo_1 è `"Cartella_1/Simbolo_1"`.

Esempio
L'esempio seguente visualizza il nome dell'elemento della libreria specificato nel pannello Output:

```
fl.trace(fl.getDocumentDOM().library.items[0].name);
```

item.removeData()

Disponibilità
Flash MX 2004.

Uso
`item.removeData(name)`

Parametri
`name` Specifica il nome dei dati da rimuovere dall'elemento della libreria.

Restituisce
Nulla.

Descrizione
Proprietà; rimuove i dati persistenti dall'elemento della libreria.

Esempio
L'esempio seguente rimuove i dati denominati `myData` dal primo elemento della libreria:

```
fl.getDocumentDOM().library.items[0].removeData("myData");
```

Capitolo 25: Oggetto Layer

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto Layer rappresenta un livello nella linea temporale. La proprietà `timeline.layers` contiene un array di oggetti Layer, a cui potete accedere mediante `fl.getDocumentDOM().getTimeline().layers`.

Riepilogo delle proprietà
Le seguenti proprietà sono disponibili per l'oggetto Layer:

Proprietà	Descrizione
`layer.color`	Una stringa, un valore esadecimale o un numero intero che specifica il colore utilizzato per il contorno del livello.
`layer.frameCount`	Di sola lettura; un numero intero che specifica il numero di fotogrammi nel livello.
`layer.frames`	Di sola lettura; un array di oggetti Frame.
`layer.height`	Un numero intero che specifica l'altezza in percentuale del livello; è equivalente al valore Altezza livello nella finestra di dialogo Proprietà livello.
`layer.layerType`	Una stringa che specifica l'uso corrente del livello; è equivalente all'impostazione Tipo nella finestra di dialogo Proprietà livello.
`layer.locked`	Un valore booleano che specifica lo stato di blocco del livello.
`layer.name`	Una stringa che specifica il nome del livello.
`layer.outline`	Un valore booleano che specifica lo stato dei contorni di tutti gli oggetti presenti nel livello.
`layer.parentLayer`	Un oggetto Layer che rappresenta la cartella contenente il livello, il livello guida o il livello di maschera.
`layer.visible`	Un valore booleano che specifica se gli oggetti del livello sullo stage sono visibili o nascosti.

layer.color

Disponibilità
Flash MX 2004.

Uso
```
layer.color
```

Descrizione
Proprietà; il colore utilizzato per il contorno del livello in uno dei seguenti formati:

- Una stringa nel formato `"#RRGGBB"` o `"#RRGGBBAA"`
- Un numero esadecimale nel formato `0xRRGGBB`

- Un numero intero che rappresenta l'equivalente decimale di un numero esadecimale.

  Questa proprietà è equivalente all'impostazione Colore contorno nella finestra di dialogo Proprietà livello.

Esempio
L'esempio seguente memorizza il valore del primo livello nella variabile `colorValue`:

```
var colorValue = fl.getDocumentDOM().getTimeline().layers[0].color;
```

L'esempio seguente illustra tre modi per impostare il rosso come colore del primo livello:

```
fl.getDocumentDOM().getTimeline().layers[0].color=16711680;
fl.getDocumentDOM().getTimeline().layers[0].color="#ff0000";
fl.getDocumentDOM().getTimeline().layers[0].color=0xFF0000;
```

layer.frameCount

Disponibilità
Flash MX 2004.

Uso
`layer.frameCount`

Descrizione
Proprietà di sola lettura; un numero intero che specifica il numero di fotogrammi nel livello.

Esempio
L'esempio seguente memorizza il numero di fotogrammi del primo livello nella variabile `fcNum`:

```
var fcNum = fl.getDocumentDOM().getTimeline().layers[0].frameCount;
```

layer.frames

Disponibilità
Flash MX 2004.

Uso
`layer.frames`

Descrizione
Proprietà di sola lettura; un array di oggetti Frame (consultate Oggetto Frame).

Esempio
L'esempio seguente imposta la variabile `frameArray` sull'array di oggetti Frame per i fotogrammi del documento corrente:

```
var frameArray = fl.getDocumentDOM().getTimeline().layers[0].frames;
```

Per determinare se un fotogramma è un fotogramma chiave, verificate se la proprietà `frame.startFrame` corrisponde all'indice dell'array, come illustrato nell'esempio seguente:

```
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

Disponibilità
Flash MX 2004.

Uso
`layer.height`

Descrizione
Proprietà; un numero intero che specifica l'altezza in percentuale del livello; è equivalente al valore Altezza livello nella finestra di dialogo Proprietà livello. I valori accettabili rappresentano percentuali dell'altezza predefinita: 100, 200 o 300.

Esempio
L'esempio seguente memorizza il valore percentuale impostato per l'altezza del primo livello:

```
var layerHeight = fl.getDocumentDOM().getTimeline().layers[0].height;
```

L'esempio seguente imposta un'altezza del 300% per il primo livello:

```
fl.getDocumentDOM().getTimeline().layers[0].height = 300;
```

layer.layerType

Disponibilità
Flash MX 2004.

Uso
`layer.layerType`

Descrizione
Proprietà; una stringa che specifica l'uso corrente del livello; è equivalente all'impostazione Tipo nella finestra di dialogo Proprietà livello. I valori accettabili sono `"normal"`, `"guide"`, `"guided"`, `"mask"`, `"masked"` e `"folder"`.

Esempio
L'esempio seguente imposta il primo livello della linea temporale sul tipo `folder`:

```
fl.getDocumentDOM().getTimeline().layers[0].layerType = "folder";
```

layer.locked

Disponibilità
Flash MX 2004.

Uso
`layer.locked`

Descrizione
Proprietà; un valore booleano che specifica lo stato di blocco del livello. Se è impostato su `true`, il livello è bloccato. Il valore predefinito è `false`.

Esempio
L'esempio seguente memorizza il valore booleano corrispondente allo stato del primo livello nella variabile `lockStatus`:

```
var lockStatus = fl.getDocumentDOM().getTimeline().layers[0].locked;
```

L'esempio seguente imposta come sbloccato lo stato del primo livello:

```
fl.getDocumentDOM().getTimeline().layers[0].locked = false;
```

layer.name

Disponibilità
Flash MX 2004.

Uso
`layer.name`

Descrizione
Proprietà; una stringa che specifica il nome del livello.

Esempio
L'esempio seguente imposta `foreground` come nome del primo livello del documento corrente:

```
fl.getDocumentDOM().getTimeline().layers[0].name = "foreground";
```

layer.outline

Disponibilità
Flash MX 2004.

Uso
`layer.outline`

Descrizione
Proprietà; un valore booleano che specifica lo stato dei contorni di tutti gli oggetti presenti nel livello. Se è impostato su `true`, tutti gli oggetti nel livello vengono visualizzati solo mediante i contorni. Se è impostato su `false`, gli oggetti appaiono come sono stati creati.

Esempio
L'esempio seguente specifica che tutti gli oggetti sul primo livello devono essere visualizzati solo mediante i contorni:

```
fl.getDocumentDOM().getTimeline().layers[0].outline = true;
```

layer.parentLayer

Disponibilità
Flash MX 2004.

Uso
`layer.parentLayer`

Descrizione
Proprietà; un oggetto Layer che rappresenta la cartella contenente il livello, il livello guida o il livello di maschera. Il livello principale deve essere una cartella, un livello guida o un livello di maschera che precede il livello, oppure il parametro `parentLayer` del livello precedente o successivo. Quando impostate il parametro `parentLayer` del livello, la posizione del livello nell'elenco non viene modificata; se tentate di impostare il parametro `parentLayer` di un livello su un livello che ne richiederebbe lo spostamento, non ottenete alcun effetto. Per un livello di primo livello viene utilizzato il valore `null`.

Esempio
L'esempio seguente utilizza due livelli della stessa importanza sulla stessa linea temporale. Il primo livello (`layers[0]`) viene convertito in una cartella e successivamente impostato come cartella principale del secondo livello (`layers[1]`). Mediante questa operazione il secondo livello viene spostato all'interno del primo livello.

```
var parLayer = fl.getDocumentDOM().getTimeline().layers[0];
parLayer.layerType = "folder";
fl.getDocumentDOM().getTimeline().layers[1].parentLayer = parLayer;
```

layer.visible

Disponibilità
Flash MX 2004.

Uso
`layer.visible`

Descrizione

Proprietà; un valore booleano che specifica se gli oggetti del livello sullo stage sono visibili o nascosti. Se il valore è `true`, tutti gli oggetti del livello sono visibili; se è `false`, sono nascosti. Il valore predefinito è `true`.

Esempio

L'esempio seguente rende invisibili tutti gli oggetti presenti sul primo livello:

```
fl.getDocumentDOM().getTimeline().layers[0].visible = false;
```

Capitolo 26: Oggetto library

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto library rappresenta il pannello Libreria. È una proprietà dell'oggetto Document (consultate `document.library`) ed è accessibile mediante `fl.getDocumentDOM().library`.

L'oggetto library contiene un array di elementi di diversi tipi, tra cui simboli, bitmap, suoni e video.

Riepilogo dei metodi
I seguenti metodi sono disponibili per l'oggetto library:

Metodo	Descrizione
`library.addItemToDocument()`	Aggiunge nel punto specificato dello stage l'elemento corrente o specificato.
`library.addNewItem()`	Crea un nuovo elemento del tipo specificato nel pannello Libreria e lo imposta come elemento corrente selezionato.
`library.deleteItem()`	Elimina dal pannello Libreria gli elementi correnti o un elemento specificato.
`library.duplicateItem()`	Crea una copia dell'oggetto selezionato o specificato.
`library.editItem()`	Apre l'elemento selezionato o specificato in modalità di modifica.
`library.expandFolder()`	Espande o comprime la cartella selezionata o specificata nella libreria.
`library.findItemIndex()`	Restituisce il valore di indice (a base zero) dell'elemento della libreria.
`library.getItemProperty()`	Ottiene la proprietà dell'elemento selezionato.
`library.getItemType()`	Ottiene il tipo dell'oggetto selezionato o specificato da un percorso di libreria.
`library.getSelectedItems()`	Ottiene l'array di tutti gli elementi correnti selezionati nella libreria.
`library.importEmbeddedSWF()`	Importa un file SWF nella libreria sotto forma di clip compilato.
`library.itemExists()`	Verifica se un elemento specificato è presente nella libreria.
`library.moveToFolder()`	Sposta in una cartella specificata l'elemento di libreria selezionato o specificato.
`library.newFolder()`	Crea nella cartella selezionata una nuova cartella con il nome specificato oppure con un nome predefinito (`"cartella senza nome #"`) se non viene fornito un parametro `folderName`.
`library.renameItem()`	Rinomina l'elemento di libreria selezionato nel pannello Libreria.
`library.selectAll()`	Seleziona o deseleziona tutti gli elementi nella libreria.
`library.selectItem()`	Seleziona un elemento di libreria specificato.
`library.selectNone()`	Deseleziona tutti gli elementi di libreria.
`library.setItemProperty()`	Imposta la proprietà per tutti gli elementi di libreria selezionati (ignorando le cartelle).
`library.updateItem()`	Aggiorna l'elemento specificato.

Riepilogo delle proprietà valide per l'oggetto Library
Le seguenti proprietà sono disponibili per l'oggetto Library:

Proprietà	Descrizione
`library.items`	Un array di oggetti Item presenti nella libreria.

library.addItemToDocument()

Disponibilità
Flash MX 2004.

Uso
```
library.addItemToDocument(position [, namePath])
```

Parametri
`position` Un punto che specifica la posizione *x,y* del centro dell'elemento sullo stage.

`namePath` Una stringa che specifica il nome dell'elemento. Se l'elemento si trova in una cartella, potete specificarne il nome utilizzando la notazione a barra rovesciata. Se *namePath* non è specificato, viene utilizzata la selezione corrente nella libreria. Questo parametro è opzionale.

Restituisce
Un valore booleano: `true` se l'elemento viene aggiunto correttamente al documento; `false` in caso contrario.

Descrizione
Metodo; aggiunge nel punto specificato dello stage l'elemento corrente o specificato.

Esempio
L'esempio seguente aggiunge nel punto (3, 60) dello stage l'elemento corrente selezionato:

```
fl.getDocumentDOM().library.addItemToDocument({x:3, y:60});
```

L'esempio seguente aggiunge nel punto (550, 485) dello stage l'elemento `Symbol1` presente nella cartella folder1 della libreria:

```
fl.getDocumentDOM().library.addItemToDocument({x:550.0, y:485.0}, "folder1/Symbol1");
```

library.addNewItem()

Disponibilità
Flash MX 2004.

Uso
```
library.addNewItem(type [, namePath])
```

Parametri

`type` Una stringa che specifica il tipo di elemento da creare. Gli unici valori accettabili per *type* sono `"video"`, `"movie clip"`, `"button"`, `"graphic"`, `"bitmap"`, `"screen"` e `"folder"` (pertanto non è possibile aggiungere un suono alla libreria mediante questo metodo). Specificare un percorso di cartella equivale a utilizzare library.newFolder()`library.newFolder()` prima di chiamare questo metodo.

`namePath` Una stringa che specifica il nome dell'elemento da aggiungere. Se l'elemento si trova in una cartella, specificatene il nome e il percorso mediante la notazione a barra rovesciata. Questo parametro è opzionale.

Restituisce

Un valore booleano: `true` se l'elemento viene creato correttamente; `false` in caso contrario.

Descrizione

Metodo; crea un nuovo elemento del tipo specificato nel pannello Libreria e lo imposta come elemento corrente selezionato. Per ulteriori informazioni sull'importazione di elementi nella libreria, tra cui elementi quali suoni, consultate `document.importFile()`.

Esempio

L'esempio seguente crea un nuovo elemento pulsante denominato `start` in una nuova cartella denominata `folderTwo`:

```
fl.getDocumentDOM().library.addNewItem("button", "folderTwo/start");
```

library.deleteItem()

Disponibilità

Flash MX 2004.

Uso

```
library.deleteItem([namePath])
```

Parametri

`namePath` Una stringa che specifica il nome dell'elemento da eliminare. Se l'elemento si trova in una cartella, potete specificarne il nome utilizzando la notazione a barra rovesciata. Se passate un nome di cartella, viene eliminata la cartella con tutti i relativi elementi. Se non specificate alcun nome, vengono eliminati gli elementi correnti selezionati. Per eliminare tutti gli elementi nel pannello Libreria, selezionateli tutti prima di utilizzare questo metodo. Questo parametro è opzionale.

Restituisce

Un valore booleano: `true` se gli elementi vengono eliminati correttamente; `false` in caso contrario.

Descrizione

Metodo; elimina dal pannello Libreria gli elementi correnti o un elemento specificato. Questo metodo può avere effetto su più elementi selezionati.

Esempio

L'esempio seguente elimina l'oggetto corrente selezionato:

```
fl.getDocumentDOM().library.deleteItem();
```

L'esempio seguente elimina l'elemento `Symbol_1` dalla cartella `Folder_1` della libreria:

```
fl.getDocumentDOM().library.deleteItem("Folder_1/Symbol_1");
```

library.duplicateItem()

Disponibilità
Flash MX 2004.

Uso
```
library.duplicateItem( [ namePath ] )
```

Parametri
`namePath` Una stringa che specifica il nome dell'elemento da duplicare. Se l'elemento si trova in una cartella, potete specificarne il nome utilizzando la notazione a barra rovesciata. Questo parametro è opzionale.

Restituisce
Un valore booleano: `true` se l'elemento viene duplicato correttamente; `false` in caso contrario. Se è selezionato più di un elemento, viene restituito il valore `false`.

Descrizione
Metodo; crea una copia dell'oggetto selezionato o specificato. Il nuovo elemento ha un nome predefinito (ad esempio `elemento copy`) ed è impostato come elemento corrente selezionato. Se è selezionato più di un elemento, il comando non viene eseguito.

Esempio
L'esempio seguente crea una copia dell'elemento `square` della cartella `test` della libreria:

```
fl.getDocumentDOM().library.duplicateItem("test/square");
```

library.editItem()

Disponibilità
Flash MX 2004.

Uso
```
library.editItem([namePath])
```

Parametri
`namePath` Una stringa che specifica il nome dell'elemento. Se l'elemento si trova in una cartella, potete specificarne il nome utilizzando la notazione a barra rovesciata. Se *namePath* non è specificato, il singolo elemento di libreria selezionato viene aperto in modalità di modifica. Se nella libreria è selezionato più di un elemento o non è selezionato alcun elemento, la prima scena nella linea temporale viene aperta in modalità di modifica. Questo parametro è opzionale.

Restituisce
Un valore booleano: `true` se l'elemento è presente e può essere modificato; `false` in caso contrario.

Descrizione
Metodo; apre l'elemento selezionato o specificato in modalità di modifica.

Esempio
L'esempio seguente apre l'elemento `circle` della cartella `test` della libreria per modificarlo:

```
fl.getDocumentDOM().library.editItem("test/circle");
```

library.expandFolder()

Disponibilità
Flash MX 2004.

Uso
```
library.expandFolder(bExpand [, bRecurseNestedParents [, namePath]])
```

Parametri
`bExpand` Un valore booleano: se è `true`, la cartella viene espansa; se è `false` (impostazione predefinita), la cartella viene compressa.

`bRecurseNestedParents` Un valore booleano: se è `true`, tutte le cartelle all'interno della cartella specificata vengono espanse o compresse, in base al valore di *bExpand*. Il valore predefinito è `false`. Questo parametro è opzionale.

`namePath` Una stringa che specifica il nome e, facoltativamente, il percorso dell'elemento da espandere o comprimere. Se questo parametro non viene specificato, il metodo viene eseguito sulla cartella corrente selezionata. Questo parametro è opzionale.

Restituisce
Un valore booleano: `true` se l'elemento viene espanso o compresso correttamente; `false` in caso contrario o se l'elemento specificato non è una cartella.

Descrizione
Metodo; espande o comprime la cartella selezionata o specificata nella libreria.

Esempio
L'esempio seguente comprime la cartella test della libreria, oltre a tutte le eventuali cartelle presenti all'interno della cartella stessa:

```
fl.getDocumentDOM().library.expandFolder(false, true, "test");
```

library.findItemIndex()

Disponibilità
Flash MX 2004.

Uso

```
library.findItemIndex(namePath)
```

Parametri

`namePath` Una stringa che specifica il nome dell'elemento. Se l'elemento si trova in una cartella, potete specificarne il nome utilizzando la notazione a barra rovesciata.

Restituisce

Un valore intero che rappresenta il valore di indice a base zero dell'elemento.

Descrizione

Metodo; restituisce il valore di indice (a base zero) dell'elemento della libreria. Poiché l'indice della libreria non è strutturato, le cartelle vengono considerate parte dell'indice principale. I percorsi della cartella possono essere utilizzati per specificare un elemento nidificato.

Esempio

L'esempio seguente memorizza nella variabile `sqIndex` il valore di indice a base zero dell'elemento di libreria square che si trova nella cartella test, quindi visualizza il valore di indice in una finestra di dialogo:

```
var sqIndex = fl.getDocumentDOM().library.findItemIndex("test/square");
alert(sqIndex);
```

library.getItemProperty()

Disponibilità

Flash MX 2004.

Uso

```
library.getItemProperty(property)
```

Parametri

`property` Una stringa. Per un elenco dei valori che potete utilizzare come parametri di *proprietà*, consultate la tabella di riepilogo delle proprietà relativa all'Oggetto Item e i riepiloghi delle proprietà delle relative sottoclassi.

Restituisce

Un valore di stringa per la proprietà.

Descrizione

Metodo; ottiene la proprietà dell'elemento selezionato.

Esempio

L'esempio seguente mostra una finestra di dialogo che contiene il valore dell'identificatore di concatenamento per il simbolo quando vi fate riferimento mediante ActionScript o per la condivisione in runtime:

```
alert(fl.getDocumentDOM().library.getItemProperty("linkageIdentifier"));
```

library.getItemType()

Disponibilità
Flash MX 2004.

Uso
`library.getItemType([namePath])`

Parametri
`namePath` Una stringa che specifica il nome dell'elemento. Se l'elemento si trova in una cartella, specificatene il nome e il percorso mediante la notazione a barra rovesciata. Se *namePath* non è specificato, viene fornito il tipo della selezione corrente. Se è selezionato più di un elemento e *namePath* non viene specificato, il comando viene ignorato. Questo parametro è opzionale.

Restituisce
Un valore di stringa che specifica il tipo di oggetto. Per informazioni sui possibili valori restituiti, consultate `item.itemType`.

Descrizione
Metodo; ottiene il tipo dell'oggetto selezionato o specificato da un percorso di libreria.

Esempio
L'esempio seguente mostra una finestra di dialogo che contiene il tipo dell'elemento `Symbol_1` presente nella cartella `Folder_1/Folder_2`:

```
alert(fl.getDocumentDOM().library.getItemType("Folder_1/Folder_2/Symbol_1"));
```

library.getSelectedItems()

Disponibilità
Flash MX 2004.

Parametri
Nessuno.

Restituisce
Un array dei valori di tutti gli elementi correnti selezionati nella libreria.

Descrizione
Metodo; ottiene l'array di tutti gli elementi correnti selezionati nella libreria.

Esempio
L'esempio seguente memorizza nella variabile `selItems` l'array degli elementi di libreria selezionati (in questo caso, si tratta di diversi file audio), quindi imposta su `11 kHz` la proprietà `sampleRate` del primo file audio nell'array:

```
var selItems = fl.getDocumentDOM().library.getSelectedItems();
selItems[0].sampleRate = "11 kHz";
```

library.importEmbeddedSWF()

Disponibilità
Flash MX 2004.

Uso
```
library.importEmbeddedSWF(linkageName, swfData [, libName])
```

Parametri
`linkageName` Una stringa che fornisce il nome del concatenamento SWF del clip filmato radice.

`swfData` Un array di dati SWF binari provenienti da una libreria esterna o da una DLL.

`libName` Una stringa che specifica il nome di libreria per l'elemento creato. Se il nome è già in uso, il metodo crea un nome alternativo. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; importa un file SWF nella libreria sotto forma di clip compilato. A differenza di File > Importa > SWF, questo metodo consente di incorporare nella libreria un file SWF compilato. Questo metodo non corrisponde ad alcuna funzionalità dell'interfaccia e deve essere utilizzato con una libreria o una DLL esterna (consultate "Estensibilità di livello C" a pagina 534).

Il file SWF che state importando deve avere un clip filmato di primo livello che contenga tutto il contenuto. L'identificatore di concatenamento del clip filmato deve essere impostato sullo stesso valore del parametro *linkageName* passato a questo metodo.

Esempio
L'esempio seguente aggiunge alla libreria il file SWF con il valore `MyMovie` specificato per *linkageName*, sotto forma di un clip compilato denominato `Intro`:

```
fl.getDocumentDOM().library.importEmbeddedSWF("MyMovie", swfData, "Intro");
```

library.itemExists()

Disponibilità
Flash MX 2004.

Uso
```
library.itemExists(namePath)
```

Parametri
`namePath` Una stringa che specifica il nome dell'elemento. Se l'elemento si trova in una cartella, specificatene il nome e il percorso mediante la notazione a barra rovesciata.

Restituisce
Un valore booleano: `true` se l'elemento specificato è presente nella libreria; `false` in caso contrario.

Descrizione
Metodo; verifica se un elemento specificato è presente nella libreria.

Esempio
L'esempio seguente visualizza `true` o `false` in una finestra di dialogo, a seconda che l'elemento `Symbol_1` sia presente nella cartella `Folder_1` della libreria:

```
alert(fl.getDocumentDOM().library.itemExists('Folder_1/Symbol_1'));
```

library.items

Disponibilità
Flash MX 2004.

Uso
```
library.items
```

Descrizione
Proprietà; un array di oggetti item presenti nella libreria.

Esempio
L'esempio seguente memorizza nella variabile `itemArray` l'array di tutti gli elementi della libreria:

```
var itemArray = fl.getDocumentDOM().library.items;
```

library.moveToFolder()

Disponibilità
Flash MX 2004.

Uso
```
library.moveToFolder(folderPath [, itemToMove [, bReplace]])
```

Parametri
`folderPath` Una stringa che specifica il percorso della cartella nel formato `"NomeCartella"` o `"NomeCartella/NomeCartella"`. Per spostare un elemento al primo livello, specificare una stringa vuota ("") per *folderPath*.

`itemToMove` Una stringa che specifica il nome dell'elemento da spostare. Se *itemToMove* non viene specificato, gli elementi selezionati vengono spostati. Questo parametro è opzionale.

`bReplace` Un valore booleano. Se esiste un elemento con lo stesso nome, quando specificate `true` per il parametro *bReplace* l'elemento selezionato viene sostituito con l'elemento che state spostando. Se è `false`, il nome dell'elemento rilasciato viene sostituito con un nome univoco. Il valore predefinito è `false`. Questo parametro è opzionale.

Restituisce
Un valore booleano: `true` se l'elemento viene spostato correttamente; `false` in caso contrario.

Descrizione
Metodo; sposta in una cartella specificata l'elemento di libreria selezionato o specificato. Se il parametro *folderPath* è vuoto, gli elementi vengono spostati nel primo livello.

Esempio
L'esempio seguente sposta l'elemento `Symbol_1` nella cartella di libreria `new` e qui sostituisce l'elemento omonimo:

```
fl.getDocumentDOM().library.moveToFolder("new", "Symbol_1", true);
```

library.newFolder()

Disponibilità
Flash MX 2004.

Uso
```
library.newFolder([folderPath])
```

Parametri
`folderPath` Una stringa che specifica il nome della cartella da creare. Se viene specificata sotto forma di un percorso che non esiste, il percorso viene creato. Questo parametro è opzionale.

Restituisce
Un valore booleano: `true` se la cartella viene creata correttamente; `false` in caso contrario.

Descrizione
Metodo; crea nella cartella selezionata una nuova cartella con il nome specificato oppure con un nome predefinito (`"cartella senza nome #"`) se non viene fornito un parametro *folderName*.

Esempio
Nell'esempio seguente vengono create due cartelle della libreria. La seconda cartella è una sottocartella della prima:

```
fl.getDocumentDOM().library.newFolder("first/second");
```

library.renameItem()

Disponibilità
Flash MX 2004.

Uso
```
library.renameItem(name)
```

Parametri
`name` Una stringa che specifica un nuovo nome per l'elemento di libreria.

Restituisce
Il valore booleano `true` se il nome dell'elemento viene modificato correttamente; `false` in caso contrario. Se sono selezionati più elementi, non viene modificato alcun nome e viene restituito il valore `false` (per coerenza con il comportamento dell'interfaccia).

Descrizione
Metodo; rinomina l'elemento di libreria selezionato nel pannello Libreria.

Esempio
L'esempio seguente rinomina l'elemento di libreria selezionato utilizzando il nome `new name`:

```
fl.getDocumentDOM().library.renameItem("new name");
```

library.selectAll()

Disponibilità
Flash MX 2004.

Uso
```
library.selectAll([bSelectAll])
```

Parametri
`bSelectAll` Un valore booleano che specifica se selezionare o deselezionare tutti gli elementi della libreria. Omettere questo parametro oppure utilizzare il valore predefinito `true` per selezionare tutti gli elementi della libreria; se specificate `false`, tutti gli elementi della libreria vengono deselezionati. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; seleziona o deseleziona tutti gli elementi nella libreria.

Esempio
L'esempio seguente seleziona tutti gli elementi della libreria:

```
fl.getDocumentDOM().library.selectAll();
fl.getDocumentDOM().library.selectAll(true);
```

L'esempio seguente deseleziona tutti gli elementi della libreria:

```
fl.getDocumentDOM().library.selectAll(false);
fl.getDocumentDOM().library.selectNone();
```

library.selectItem()

Disponibilità
Flash MX 2004.

Uso
```
library.selectItem(namePath [, bReplaceCurrentSelection [, bSelect]])
```

Parametri
`namePath` Una stringa che specifica il nome dell'elemento. Se l'elemento si trova in una cartella, potete specificarne il nome utilizzando la notazione a barra rovesciata.

`bReplaceCurrentSelection` Un valore booleano che specifica se sostituire la selezione corrente oppure aggiungerle l'elemento. Il valore predefinito è `true` (la selezione corrente viene sostituita). Questo parametro è opzionale.

`bSelect` Un valore booleano che specifica se selezionare o deselezionare un elemento. Il valore predefinito è `true` (l'elemento viene selezionato). Questo parametro è opzionale.

Restituisce
Un valore booleano: `true` se l'elemento specificato esiste, `false` in caso contrario.

Descrizione
Metodo; seleziona un elemento di libreria specificato.

Esempio
L'esempio seguente imposta la selezione corrente nella libreria su `Symbol_1` nella `cartella senza nome 1`:

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1");
```

L'esempio seguente estende la selezione corrente nella libreria affinché includa `Symbol_1` all'interno della `cartella senza nome 1`:

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1", false);
```

L'esempio seguente deseleziona `Symbol_1` all'interno della `cartella senza nome 1` e non modifica gli altri elementi selezionati:

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1", true, false);
```

library.selectNone()

Disponibilità
Flash MX 2004.

Uso
```
library.selectNone()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; deseleziona tutti gli elementi della libreria.

Esempio
L'esempio seguente deseleziona tutti gli elementi della libreria:

```
fl.getDocumentDOM().library.selectNone();
fl.getDocumentDOM().library.selectAll(false);
```

library.setItemProperty()

Disponibilità
Flash MX 2004.

Uso
`library.setItemProperty(property, value)`

Parametri
`property` Una stringa che corrisponde al nome della proprietà da impostare. Per un elenco delle proprietà, consultate la tabella di riepilogo delle proprietà relativa all'Oggetto Item e i riepiloghi delle proprietà delle relative sottoclassi. Per informazioni su quali oggetti sono sottoclassi dell'oggetto Item, consultate "Riepilogo della struttura del DOM" a pagina 12.

`value` Il valore da assegnare alla proprietà specificata.

Restituisce
Nulla.

Descrizione
Metodo; imposta la proprietà per tutti gli elementi di libreria selezionati (ignorando le cartelle).

Esempio
L'esempio seguente assegna il valore button alla proprietà `symbolType` per gli elementi di libreria selezionati. In tal caso, l'elemento deve essere un Oggetto SymbolItem; `symbolType` è una proprietà valida per gli oggetti SymbolItem.

```
fl.getDocumentDOM().library.setItemProperty("symbolType", "button");
```

library.updateItem()

Disponibilità
Flash MX 2004.

Uso

```
library.updateItem([namePath])
```

Parametri

`namePath` Una stringa che specifica il nome dell'elemento. Se l'elemento si trova in una cartella, specificatene il nome e il percorso mediante la notazione a barra rovesciata. Equivale a fare clic con il pulsante destro del mouse su un oggetto e selezionare Aggiorna dal menu visualizzato nell'interfaccia utente. Se non viene fornito alcun nome, la selezione corrente viene aggiornata. Questo parametro è opzionale.

Restituisce

Un valore booleano: `true` se l'elemento viene aggiornato correttamente; `false` in caso contrario.

Descrizione

Metodo; aggiorna l'elemento specificato.

Esempio

L'esempio seguente visualizza una finestra di dialogo che mostra se l'elemento selezionato è aggiornato (`true`) o meno (`false`):

```
alert(fl.getDocumentDOM().library.updateItem());
```

Capitolo 27: Oggetto Math

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto Math è disponibile come proprietà di sola lettura dell'oggetto flash; consultate `fl.Math`. L'oggetto fornisce metodi che eseguono operazioni matematiche comuni.

Riepilogo dei metodi
Per l'oggetto Math sono disponibili i seguenti metodi:

Metodo	Descrizione
`Math.concatMatrix()`	Esegue una concatenazione di matrici e restituisce il risultato.
`Math.invertMatrix()`	Restituisce l'inverso della matrice specificata.
`Math.pointDistance()`	Calcola la distanza tra due punti.

Math.concatMatrix()

Disponibilità
Flash MX 2004.

Uso
```
Math.concatMatrix(mat1, mat2)
```

Parametri
mat1, mat2 Specificano gli oggetti Matrix da concatenare (consultate Oggetto Matrix). Ogni parametro deve essere costituito da un oggetto con i campi `a`, `b`, `c`, `d`, `tx` e `ty`.

Restituisce
Una matrice di oggetti concatenata.

Descrizione
Metodo; esegue una concatenazione di matrici e restituisce il risultato.

Esempio
L'esempio seguente memorizza nella variabile `elt` l'oggetto selezionato, moltiplica la matrice degli oggetti per la matrice delle viste, quindi memorizza il valore nella variabile `mat`:

```
var elt = fl.getDocumentDOM().selection[0];
var mat = fl.Math.concatMatrix( elt.matrix , fl.getDocumentDOM().viewMatrix );
```

Math.invertMatrix()

Disponibilità
Flash MX 2004.

Uso
```
Math.invertMatrix(mat)
```

Parametri
mat Indica l'oggetto Matrix da invertire (consultate Oggetto Matrix). Deve disporre dei campi seguenti: a, b, c, d, tx e ty.

Restituisce
Un oggetto Matrix costituito dall'inverso della matrice originale.

Descrizione
Metodo; restituisce l'inverso della matrice specificata.

Esempio
L'esempio seguente memorizza nella variabile elt l'oggetto selezionato, assegna la matrice alla variabile mat, quindi memorizza l'inverso della matrice nella variabile inv:

```
var elt = fl.getDocumentDOM().selection[0];
var mat = elt.matrix;
var inv = fl.Math.invertMatrix( mat );
```

Math.pointDistance()

Disponibilità
Flash MX 2004.

Uso
```
Math.pointDistance(pt1, pt2)
```

Parametri
pt1, pt2 Specificano i punti tra i quali viene misurata la distanza.

Restituisce
Un valore a virgola mobile che rappresenta la distanza tra i punti.

Descrizione
Metodo; calcola la distanza tra due punti.

Esempio

L'esempio seguente memorizza il valore della distanza tra *pt1* e *pt2* nella variabile `dist`:

```
var pt1 = {x:10, y:20}
var pt2 = {x:100, y:200}
var dist = fl.Math.pointDistance(pt1, pt2);
```

Capitolo 28: Oggetto Matrix

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto Matrix rappresenta una matrice di trasformazione.

Riepilogo delle proprietà
Per l'oggetto Matrix sono disponibili le proprietà seguenti:

Proprietà	Descrizione
`matrix.a`	Un valore a virgola mobile che specifica l'elemento (0,0) della matrice di trasformazione.
`matrix.b`	Un valore a virgola mobile che specifica l'elemento (0,1) della matrice.
`matrix.c`	Un valore a virgola mobile che specifica l'elemento (1,0) della matrice.
`matrix.d`	Un valore a virgola mobile che specifica l'elemento (1,1) della matrice.
`matrix.tx`	Un valore a virgola mobile che specifica la posizione sull'asse *x* del punto di registrazione di un simbolo o del centro di una forma.
`matrix.ty`	Un valore a virgola mobile che specifica la posizione sull'asse *y* del punto di registrazione di un simbolo o del centro di una forma.

matrix.a

Disponibilità
Flash MX 2004.

Uso
`matrix.a`

Descrizione
Proprietà; valore a virgola mobile che specifica l'elemento (0,0) della matrice di trasformazione. Questo valore rappresenta il fattore di scala dell'asse *x* dell'oggetto.

Esempio
Le proprietà `a` e `d` di una matrice rappresentano la modifica in scala. Nell'esempio seguente i valori sono impostati su 2 e 3 rispettivamente per moltiplicare per due la larghezza e per tre l'altezza dell'oggetto selezionato:

```
var mat = fl.getDocumentDOM().selection[0].matrix;
mat.a = 2;
mat.d = 3;
fl.getDocumentDOM().selection[0].matrix = mat;
```

Potete ruotare un oggetto impostando le proprietà di matrice `a`, `b`, `c` e `d` in modo che siano relative l'una all'altra, dove `a = d` e `b = c`. Ad esempio, i valori 0,5, 0,8, -0,8 e 0,5 ruotano l'oggetto di 60°:

```
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;
```

Potete impostare `a = d = 1` e `c = b = 0` per ripristinare la forma originale dell'oggetto.

matrix.b

Disponibilità
Flash MX 2004.

Uso
`matrix.b`

Descrizione
Proprietà; un valore a virgola mobile che specifica l'elemento (0,1) della matrice. Questo valore rappresenta l'inclinazione verticale di una forma, in base alla quale il bordo destro della forma viene spostato lungo l'asse verticale.

Le proprietà `matrix.b` e `matrix.c` di una matrice rappresentano l'inclinazione (consultate `matrix.c`).

Esempio
Nell'esempio seguente potete impostare `b` e `c` rispettivamente su -1 e 0 per inclinare l'oggetto di 45° rispetto all'asse verticale:

```
var mat = fl.getDocumentDOM().selection[0].matrix;
mat.b = -1;
mat.c = 0;
fl.getDocumentDOM().selection[0].matrix = mat;
```

Per inclinare l'oggetto e riportarlo alla forma originale, potete impostare `b` e `c` su 0.

Consultate anche l'esempio `matrix.a`.

matrix.c

Disponibilità
Flash MX 2004.

Uso
`matrix.c`

Descrizione
Proprietà; un valore a virgola mobile che specifica l'elemento (1,0) della matrice. In base a questo valore, l'oggetto viene inclinato spostandone il bordo inferiore lungo l'asse orizzontale.

Le proprietà `matrix.b` e `matrix.c` di una matrice rappresentano l'inclinazione.

Esempio
Consultate anche l'esempio `matrix.b`.

matrix.d

Disponibilità
Flash MX 2004.

Uso
```
matrix.d
```

Descrizione
Proprietà; un valore a virgola mobile che specifica l'elemento (1,1) della matrice. Questo valore rappresenta il fattore di scala dell'asse *y* dell'oggetto.

Esempio
Consultate l'esempio `matrix.a`.

matrix.tx

Disponibilità
Flash MX 2004.

Uso
```
matrix.tx
```

Descrizione
Proprietà; un valore a virgola mobile che specifica la posizione sull'asse *x* del punto di registrazione (detto anche *punto di origine* o *punto zero*) di un simbolo o del centro di una forma. Definisce la traslazione *x* della trasformazione.

Potete spostare un oggetto impostandone le proprietà `matrix.tx` e `matrix.ty` (consultate `matrix.ty`).

Esempio
Nell'esempio seguente `tx` e `ty` vengono impostati su 0 per spostare il punto di registrazione dell'oggetto nel punto 0,0 del documento:

```
var mat = fl.getDocumentDOM().selection[0].matrix;
mat.tx = 0;
mat.ty = 0;
fl.getDocumentDOM().selection[0].matrix = mat;
```

matrix.ty

Disponibilità
Flash MX 2004.

Uso
`matrix.ty`

Descrizione
Proprietà; un valore a virgola mobile che specifica la posizione sull'asse *y* del punto di registrazione di un simbolo o del centro di una forma. Definisce la traslazione *y* della trasformazione.

È possibile spostare un oggetto impostando le proprietà `matrix.tx` e `matrix.ty`.

Esempio
Consultate l'esempio `matrix.tx`.

Capitolo 29: Oggetto outputPanel

Disponibilità
Flash MX 2004.

Descrizione
Questo oggetto rappresenta il pannello Output, che visualizza informazioni utili per la risoluzione dei problemi (ad esempio, errori di sintassi). Per accedere a questo oggetto, utilizzare `fl.outputPanel` oppure `flash.outputPanel`. Consultate `fl.outputPanel`.

Riepilogo dei metodi
L'oggetto outputPanel utilizza i seguenti metodi:

Metodo	Descrizione
`outputPanel.clear()`	Cancella il contenuto del pannello Output.
`outputPanel.save()`	Salva il contenuto del pannello Output in un file di testo locale.
`outputPanel.trace()`	Aggiunge una riga al contenuto del pannello Output, terminato da una nuova riga.

outputPanel.clear()

Disponibilità
Flash MX 2004.

Uso
```
outputPanel.clear()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; cancella il contenuto del pannello Output. Potete utilizzare questo metodo in un'applicazione per le elaborazioni in batch per cancellare un elenco di errori, oppure per salvarli in modo incrementale utilizzando il metodo insieme a `outputPanel.save()`.

Esempio
L'esempio seguente cancella il contenuto corrente del pannello Output:

```
fl.outputPanel.clear();
```

outputPanel.save()

Disponibilità
Flash MX 2004; parametro *bUseSystemEncoding* aggiunto in Flash 8.

Uso
```
outputPanel.save(fileURI [, bAppendToFile [ , bUseSystemEncoding]])
```

Parametri
fileURI Una stringa, espressa come URI file:///, che specifica il file locale in cui deve essere inserito il contenuto del pannello Output.

bAppendToFile Un valore booleano opzionale. Se il valore è `true`, aggiunge il contenuto del pannello Output al file di output; se è `false`, il metodo sovrascrive il file di output, se esiste già. Il valore predefinito è `false`.

bUseSystemEncoding Un valore booleano opzionale. Se il valore è `true`, salva il testo del pannello Output utilizzando la codifica di sistema; se è `false`, salva il testo del pannello Output utilizzando la codifica UTF-8, con i caratteri BOM all'inizio del testo. Il valore predefinito è `false`.

Restituisce
Nulla.

Descrizione
Metodo; salva il contenuto del pannello Output in un file di testo locale, sovrascrivendo il file oppure aggiungendovi il contenuto.

Se *fileURI* non è valido o non è specificato, viene generato un errore.

Questo metodo è particolarmente utile per le elaborazioni in batch. Ad esempio, è possibile creare un file JSFL che compili diversi componenti. Gli errori di compilazione vengono visualizzati nel pannello Output ed è possibile utilizzare questo metodo per salvare gli errori risultanti in un file di testo, che può essere analizzato automaticamente dal sistema di compilazione in uso.

Esempio
L'esempio seguente salva il contenuto del pannello Output nel file batch.log presente nella cartella /tests, sovrascrivendo il file batch.log se esistente:

```
fl.outputPanel.save("file:///c|/tests/batch.log");
```

outputPanel.trace()

Disponibilità
Flash MX 2004.

Uso
```
outputPanel.trace(message)
```

Parametri
message Una stringa che contiene il testo da aggiungere al pannello Output.

Restituisce

Nulla.

Descrizione

Metodo; invia al pannello Output una stringa di testo, terminata da una nuova riga, e visualizza il pannello Output, se non è già visibile. Questo metodo è identico a `fl.trace()` ed equivale all'istruzione `trace()` in ActionScript.

Per inviare una riga vuota, utilizzare `outputPanel.trace("")` o `outputPanel.trace("\n")`. È possibile utilizzare l'ultimo comando inline, impostando `\n` come parte della stringa *message*.

Esempio

L'esempio seguente visualizza diverse righe di testo nel pannello Output:

```
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 +"?");
```

Capitolo 30: Oggetto Oval

Ereditarietà Oggetto Element > Oggetto Shape > Oggetto Oval

Disponibilità
Flash CS3 Professional.

Descrizione
L'oggetto Oval è una forma disegnata utilizzando lo strumento Ovale di base. Per determinare se un elemento è un oggetto Oval, usate `shape.isOvalObject`.

Riepilogo delle proprietà
Oltre a quelle dell'Oggetto Shape, con l'oggetto Oval è possibile utilizzare le seguenti proprietà. Per impostare le proprietà di un oggetto Oval, usate document.setOvalObjectProperty().

Proprietà	Descrizione
`OvalObject.closePath`	Di sola lettura; valore booleano che specifica se la casella di controllo Chiudi tracciato della finestra di ispezione Proprietà è selezionata.
`OvalObject.endAngle`	Di sola lettura; valore float che specifica l'angolo finale dell'oggetto Oval.
`OvalObject.innerRadius`	Di sola lettura; valore float che specifica il raggio interno dell'oggetto Oval come percentuale.
`OvalObject.startAngle`	Di sola lettura; valore float che specifica l'angolo iniziale dell'oggetto Oval.

OvalObject.closePath

Disponibilità
Flash CS3 Professional.

Uso
```
OvalObject.closePath
```

Descrizione
Proprietà di sola lettura; valore booleano che specifica se la casella di controllo Chiudi tracciato della finestra di ispezione Proprietà è selezionata. Se i valori degli angoli iniziale e finale dell'oggetto coincidono, l'impostazione di questa proprietà non ha alcun effetto fino alla modifica dei valori.

Per impostare questo valore, utilizzate `document.setOvalObjectProperty()`.

Esempio
L'esempio che segue deseleziona la proprietà `OvalObject.closePath`:

```
fl.getDocumentDOM().setOvalObjectProperty("closePath",false);
```

Consultate anche

`document.setOvalObjectProperty()`, `shape.isOvalObject`

OvalObject.endAngle

Disponibilità

Flash CS3 Professional.

Uso

```
OvalObject.endAngle
```

Descrizione

Proprietà di sola lettura; valore float che specifica l'angolo finale dell'oggetto Oval. I valori accettabili sono compresi tra 0 e 360.

Per impostare questo valore, utilizzate `document.setOvalObjectProperty()`.

Esempio

L'esempio che segue imposta su 270 l'angolo finale degli oggetti Oval selezionati:

```
fl.getDocumentDOM().setOvalObjectProperty("endAngle",270);
```

Consultate anche

`document.setOvalObjectProperty()`, `OvalObject.startAngle`, `shape.isOvalObject`

OvalObject.innerRadius

Disponibilità

Flash CS3 Professional.

Uso

```
OvalObject.innerRadius
```

Descrizione

Proprietà di sola lettura; valore float che specifica il raggio interno dell'oggetto Oval come percentuale. I valori accettabili sono compresi tra 0 e 99.

Per impostare questo valore, utilizzate `document.setOvalObjectProperty()`.

Esempio

L'esempio che segue imposta sul 50% il raggio interno degli oggetti Oval selezionati:

```
fl.getDocumentDOM().setOvalObjectProperty("innerRadius",50);
```

Consultate anche

`document.setOvalObjectProperty()`, `shape.isOvalObject`

OvalObject.startAngle

Disponibilità

Flash CS3 Professional.

Uso

```
OvalObject.startAngle
```

Descrizione

Proprietà di sola lettura; valore float che specifica l'angolo iniziale dell'oggetto Oval. I valori accettabili sono compresi tra 0 e 360.

Per impostare questo valore, utilizzate `document.setOvalObjectProperty()`.

Esempio

L'esempio che segue imposta su 270 l'angolo iniziale degli oggetti Oval selezionati:

```
fl.getDocumentDOM().setOvalObjectProperty("startAngle",270);
```

Consultate anche

`document.setOvalObjectProperty()`, `OvalObject.endAngle`, `shape.isOvalObject`

Capitolo 31: Oggetto Parameter

Disponibilità
Flash MX 2004.

Descrizione
Il tipo di oggetto Parameter è accessibile dall'array `screen.parameters`, che corrisponde alla finestra di ispezione Proprietà delle schermate nel programma di creazione Flash, o dall'array `componentInstance.parameters`, che corrisponde alla finestra di ispezione Proprietà dei componenti nel programma di creazione Flash.

Riepilogo dei metodi
I seguenti metodi sono disponibili per l'oggetto Parameter:

Metodo	Descrizione
`parameter.insertItem()`	Inserisce un elemento in un elenco, un oggetto o un array.
`parameter.removeItem()`	Rimuove un elemento del tipo di elenco, oggetto o array per un parametro screen o component.

Riepilogo delle proprietà
Le seguenti proprietà sono disponibili per l'oggetto Parameter:

Proprietà	Descrizione
`parameter.category`	Una stringa che specifica la proprietà `category` per il parametro `screen` o `componentInstance`.
`parameter.listIndex`	Un numero intero che specifica il valore dell'elemento selezionato nell'elenco.
`parameter.name`	Di sola lettura; una stringa che specifica il nome del parametro.
`parameter.value`	Corrisponde al campo Valore nella scheda Parametri della finestra di ispezione dei componenti, nella scheda Parametri della finestra di ispezione Proprietà o nella finestra di ispezione Proprietà delle schermate.
`parameter.valueType`	Di sola lettura; una stringa che indica il tipo del parametro screen o component.
`parameter.verbose`	Specifica la posizione in cui il parametro viene visualizzato.

parameter.category

Disponibilità
Flash MX 2004.

Uso
```
parameter.category
```

Descrizione
Proprietà; una stringa che specifica la proprietà `category` per il parametro `screen` o `componentInstance`. Questa proprietà fornisce una modalità alternativa di presentazione per l'elenco di parametri. Questa funzionalità non è disponibile nell'interfaccia utente di Flash.

parameter.insertItem()

Disponibilità
Flash MX 2004.

Uso
```
parameter.insertItem(index, name, value, type)
```

Parametri
`index` Un indice di un numero intero a base zero che indica dove verrà inserito l'elemento nell'elenco, nell'oggetto o nell'array. Se l'indice è 0, l'elemento viene inserito all'inizio dell'elenco. Se l'indice è maggiore delle dimensioni dell'elenco, il nuovo elenco viene inserito alla fine dell'array.

`name` Una stringa che specifica il nome dell'elemento da inserire. Si tratta di un parametro richiesto per i parametri degli oggetti.

`value` Una stringa che specifica il valore dell'elemento da inserire.

`type` Una stringa che specifica il tipo dell'elemento da inserire.

Restituisce
Nulla.

Descrizione
Metodo; inserisce un elemento in un elenco, un oggetto o un array. Se un parametro è un elenco, un oggetto o un array, la proprietà *value* è un array.

Esempio
L'esempio seguente inserisce il valore di `New Value` nel parametro `labelPlacement`:

```
// 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

Disponibilità
Flash MX 2004.

Uso
`parameter.listIndex`

Descrizione
Proprietà; il valore dell'elemento selezionato nell'elenco. Questa proprietà è valida solo se `parameter.valueType` è `"List"`.

Esempio
L'esempio seguente imposta il primo parametro di una diapositiva, `autoKeyNav`. Per impostare il parametro su uno dei valori accettabili (`true`, `false` o `inherit`), `parameter.listIndex` deve essere impostato sull'indice dell'elemento nell'elenco (0 per `true`, 1 per `false`, 2 per `inherit`).

```
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters;
parms[0].listIndex = 1;
```

parameter.name

Disponibilità
Flash MX 2004.

Uso
`parameter.name`

Descrizione
Proprietà di sola lettura; una stringa che specifica il nome del parametro.

Esempio
L'esempio seguente mostra il nome del quinto parametro per il componente selezionato:

```
var parms = fl.getDocumentDOM().selection[0].parameters;
fl.trace("name: " + parms[4].name);
```

L'esempio seguente mostra il nome del quinto parametro per la schermata selezionata:

```
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters; fl.trace("name: " +
parms[4].name);
```

parameter.removeItem()

Disponibilità
Flash MX 2004.

Uso
`parameter.removeItem(index)`

Parametri

index L'indice di un numero intero a base zero dell'elemento da rimuovere dalla proprietà della schermata o del componente.

Restituisce

Nulla.

Descrizione

Metodo; rimuove un elemento del tipo di elenco, oggetto o array per un parametro screen o component.

Esempio

L'esempio seguente rimuove l'elemento associato all'indice 1 dal parametro `labelPlacement` di un componente:

```
// 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);
}
```

L'esempio seguente rimuove l'elemento associato all'indice 1 dal parametro `autoKeyNav` di una schermata:

```
// 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

Disponibilità

Flash MX 2004.

Uso

```
parameter.value
```

Descrizione
Proprietà; corrisponde al campo Valore nella scheda Parametri della finestra di ispezione dei componenti, nella scheda Parametri della finestra di ispezione Proprietà o nella finestra di ispezione Proprietà delle schermate. Il tipo della proprietà `value` è determinato dalla proprietà `valueType` per il parametro (consultate `parameter.valueType`).

parameter.valueType

Disponibilità
Flash MX 2004.

Uso
```
parameter.valueType
```

Descrizione
Proprietà di sola lettura; una stringa che indica il tipo del parametro screen o component. Il tipo può essere uno dei valori seguenti: `"Default"`, `"Array"`, `"Object"`, `"List"`, `"String"`, `"Number"`, `"Boolean"`, `"Font Name"`, `"Color"`, `"Collection"`, `"Web Service URL"` o `"Web Service Operation"`.

Consultate anche
`parameter.value`

parameter.verbose

Disponibilità
Flash MX 2004.

Uso
```
parameter.verbose
```

Descrizione
Proprietà; specifica la posizione in cui il parametro viene visualizzato. Se il valore di questa proprietà è 0 (nonverbose), il parametro è visualizzato solo nella finestra di ispezione dei componenti. Se il valore è 1 (verbose), il parametro viene visualizzato nel pannello di ispezione dei componenti e nella scheda Parametri della finestra di ispezione Proprietà.

Capitolo 32: Oggetto Path

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto Path definisce una sequenza di segmenti di linea (retta, curva o entrambe), utilizzata generalmente quando create strumenti estensibili. L'esempio seguente mostra un'istanza di un oggetto Path restituita dall'oggetto flash:

```
path = fl.drawingLayer.newPath();
```

Consultate anche Oggetto drawingLayer.

Riepilogo dei metodi
I seguenti metodi sono disponibili per l'oggetto Path:

Metodo	Descrizione
`path.addCubicCurve()`	Aggiunge al percorso un segmento di curva Bézier cubica.
`path.addCurve()`	Aggiunge al percorso un segmento di curva Bézier quadratica.
`path.addPoint()`	Aggiunge un punto al percorso.
`path.clear()`	Rimuove tutti i punti dal percorso.
`path.close()`	Aggiunge un punto nella posizione corrispondente al primo punto del percorso ed estende il percorso fino a tale punto, che chiude il percorso.
`path.makeShape()`	Crea una forma sullo stage mediante le impostazioni correnti per il tratto e il riempimento.
`path.newContour()`	Inizia un nuovo contorno nel percorso.

Riepilogo delle proprietà
Le seguenti proprietà sono disponibili per l'oggetto Path:

Proprietà	Descrizione
`path.nPts`	Di sola lettura; un numero intero che rappresenta il numero di punti nel percorso.

path.addCubicCurve()

Disponibilità
Flash MX 2004.

Uso

```
path.addCubicCurve(xAnchor, yAnchor, x2, y2, x3, y3, x4, y4)
```

Parametri
`xAnchor` Un valore a virgola mobile che specifica la posizione *x* del primo punto di controllo.

yAnchor Un valore a virgola mobile che specifica la posizione *y* del primo punto di controllo.

x2 Un valore a virgola mobile che specifica la posizione *x* del secondo punto di controllo.

y2 Un valore a virgola mobile che specifica la posizione *y* del secondo punto di controllo.

x3 Un valore a virgola mobile che specifica la posizione *x* del terzo punto di controllo.

y3 Un valore a virgola mobile che specifica la posizione *y* del terzo punto di controllo.

x4 Un valore a virgola mobile che specifica la posizione *x* del quarto punto di controllo.

y4 Un valore a virgola mobile che specifica la posizione *y* del quarto punto di controllo.

Restituisce
Nulla.

Descrizione
Metodo; aggiunge al percorso un segmento di curva Bézier cubica.

Esempio
L'esempio seguente crea un nuovo percorso, lo memorizza nella variabile `myPath` e assegna la curva al percorso:

```
var myPath = fl.drawingLayer.newPath();
myPath.addCubicCurve(0, 0, 10, 20, 20, 20, 30, 0);
```

path.addCurve()

Disponibilità
Flash MX 2004.

Uso
```
path.addCurve(xAnchor, yAnchor, x2, y2, x3, y3)
```

Parametri
xAnchor Un valore a virgola mobile che specifica la posizione *x* del primo punto di controllo.

yAnchor Un valore a virgola mobile che specifica la posizione *y* del primo punto di controllo.

x2 Un valore a virgola mobile che specifica la posizione *x* del secondo punto di controllo.

y2 Un valore a virgola mobile che specifica la posizione *y* del secondo punto di controllo.

x3 Un valore a virgola mobile che specifica la posizione *x* del terzo punto di controllo.

y3 Un valore a virgola mobile che specifica la posizione *y* del terzo punto di controllo.

Restituisce
Nulla.

Descrizione
Metodo; aggiunge al percorso un segmento di curva Bézier quadratica.

Esempio
L'esempio seguente crea un nuovo percorso, lo memorizza nella variabile `myPath` e assegna la curva al percorso:

```
var myPath = fl.drawingLayer.newPath();
myPath.addCurve(0, 0, 10, 20, 20, 0);
```

path.addPoint()

Disponibilità
Flash MX 2004.

Uso
```
path.addPoint(x, y)
```

Parametri
x Un numero a virgola mobile che specifica la posizione *x* del punto.

y Un numero a virgola mobile che specifica la posizione *y* del punto.

Restituisce
Nulla.

Descrizione
Metodo; aggiunge un punto al percorso.

Esempio
L'esempio seguente crea un nuovo percorso, lo memorizza nella variabile `myPath` e assegna il nuovo punto al percorso:

```
var myPath = fl.drawingLayer.newPath();
myPath.addPoint(10, 100);
```

path.clear()

Disponibilità
Flash MX 2004.

Uso
```
path.clear()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; rimuove tutti i punti dal percorso.

Esempio
L'esempio seguente rimuove tutti i punti da un percorso memorizzato nella variabile `myPath`:

```
var myPath = fl.drawingLayer.newPath();
myPath.clear();
```

path.close()

Disponibilità
Flash MX 2004.

Uso
`path.close()`

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; aggiunge un punto nella posizione corrispondente al primo punto del percorso ed estende il percorso a tale punto, che chiude il percorso. Se il percorso non ha punti, non viene aggiunto alcun punto.

Esempio
L'esempio seguente crea un percorso chiuso:

```
var myPath = fl.drawingLayer.newPath();
myPath.close();
```

path.makeShape()

Disponibilità
Flash MX 2004.

Uso
`path.makeShape([bSupressFill [, bSupressStroke]])`

Parametri
`bSuppressFill` Un valore booleano che, se impostato su `true`, sopprime il riempimento da applicare alla forma. Il valore predefinito è `false`. Questo parametro è opzionale.

`bSupressStroke` Un valore booleano che, se impostato su `true`, sopprime il tratto da applicare alla forma. Il valore predefinito è `false`. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; crea una forma sullo stage mediante le impostazioni correnti per il tratto e il riempimento. Il percorso viene cancellato dopo che è stata creata la forma. Questo metodo prevede due parametri opzionali per la soppressione del riempimento e del tratto dell'oggetto forma risultante. Se omettete questi parametri o li impostate su `false`, vengono utilizzati i valori correnti per il riempimento e il tratto.

Esempio
L'esempio seguente crea una forma con il riempimento corrente e senza tratto:

```
var myPath = fl.drawingLayer.newPath();
myPath.makeShape(false, true);
```

path.newContour()

Disponibilità
Flash MX 2004.

Uso
`path.newContour()`

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; inizia un nuovo contorno nel percorso.

Esempio
L'esempio seguente crea un quadrato vuoto:

```
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

Disponibilità
Flash MX 2004.

Uso
`path.nPts`

Descrizione
Proprietà di sola lettura; un numero intero che rappresenta il numero di punti nel percorso. Un nuovo percorso ha 0 punti.

Esempio
L'esempio seguente utilizza il pannello Output per mostrare il numero di punti nel percorso a cui fa riferimento la variabile `myPath`:

```
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
```

Capitolo 33: Oggetto presetItem

Disponibilità
Flash CS4 Professional.

Descrizione
L'oggetto presetItem rappresenta un elemento (preimpostazione o cartella) nel pannello Preimpostazioni di movimento (Finestra > Preimpostazioni di movimento). L'array di oggetti presetItem è una proprietà dell'oggetto presetPanel (`presetPanel.items`).

Tutte le proprietà dell'oggetto presetItem sono di sola lettura. Per eseguire operazioni come l'eliminazione, la ridenominazione o lo spostamento di elementi, potete usare i metodi dell'Oggetto presetPanel.

Riepilogo delle proprietà
Con l'oggetto presetItem potete usare le proprietà seguenti:

Proprietà	Descrizione
presetItem.isDefault	Specifica se l'elemento viene installato con Flash oppure è un elemento personalizzato creato da voi o da un altro utente.
presetItem.isFolder	Specifica se l'elemento nel pannello Preimpostazioni di movimento è una cartella o una preimpostazione.
presetItem.level	Il livello dell'elemento nella struttura di cartelle del pannello Preimpostazioni di movimento.
presetItem.name	Il nome della preimpostazione o della cartella senza informazioni sul percorso.
presetItem.open	Specifica se una cartella nel pannello Preimpostazioni di movimento è espansa o compressa.
presetItem.path	Il percorso dell'elemento nella struttura di cartelle del pannello Preimpostazioni di movimento e il nome dell'elemento.

presetItem.isDefault

Disponibilità
Flash CS4 Professional.

Uso
```
presetItem.isDefault
```

Descrizione
Proprietà di sola lettura: un valore booleano che specifica se l'elemento viene installato con Flash (`true`) oppure è un elemento personalizzato creato dall'utente o da un altro utente (`false`). Se il valore è `true`, potete considerarlo un elemento di sola lettura che non è possibile spostare, eliminare o con cui non è possibile eseguire operazioni di questo tipo.

Esempio
L'esempio seguente visualizza il contenuto del pannello Preimpostazioni di movimento e indica se un elemento viene installato con Flash:

```
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

Disponibilità
Flash CS4 Professional.

Uso
```
presetItem.isFolder
```

Descrizione
Proprietà di sola lettura: un valore booleano che specifica se l'elemento nel pannello Preimpostazioni di movimento è una cartella (`true`) o una preimpostazione (`false`).

Esempio
L'esempio seguente mostra che il primo elemento nel pannello Preimpostazioni di movimento è una cartella e il secondo è una preimpostazione:

```
var presetItemArray=fl.presetPanel.items;
fl.trace(presetItemArray[0].isFolder);
fl.trace(presetItemArray[1].isFolder);
```

presetItem.level

Disponibilità
Flash CS4 Professional.

Uso
```
presetItem.level
```

Descrizione
Proprietà di sola lettura: un numero intero che specifica il livello dell'elemento nella struttura di cartelle del pannello Preimpostazioni di movimento. Le cartelle Cartella predefinita e Preimpostazioni personalizzate sono al livello 0.

Esempio
L'esempio seguente mostra che il primo elemento nel pannello Preimpostazioni di movimento è al livello 0 e il secondo è al livello 1:

```
var presetItemArray=fl.presetPanel.items;
fl.trace(presetItemArray[0].level);
fl.trace(presetItemArray[1].level);
```

presetItem.name

Disponibilità
Flash CS4 Professional.

Uso
`presetItem.name`

Descrizione
Proprietà di sola lettura: una stringa che rappresenta il nome della preimpostazione o della cartella senza informazioni sul percorso.

Esempio
Consultate `presetItem.path`.

presetItem.open

Disponibilità
Flash CS4 Professional.

Uso
`presetItem.open`

Descrizione
Proprietà di sola lettura: specifica se una cartella nel pannello Preimpostazioni di movimento è espansa (`true`) o compressa (`false`).

Questa proprietà è `true` se l'elemento non è una cartella. Per determinare se un elemento è una cartella o una preimpostazione, utilizzate presetItem.isFolder.

Esempio
L'esempio seguente visualizza informazioni sullo stato di espansione o compressione delle cartelle nel pannello Preimpostazioni di movimento:

```
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

Disponibilità
Flash CS4 Professional.

Uso
`presetItem.path`

Descrizione
Proprietà di sola lettura: una stringa che rappresenta il percorso dell'elemento nella struttura di cartelle del pannello Preimpostazioni di movimento e il nome dell'elemento.

Esempio
L'esempio seguente illustra la differenza tra i valori in `presetItem.name` e `presetItem.path`.

```
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("");
}
```

Capitolo 34: Oggetto presetPanel

Disponibilità

Flash CS4 Professional.

Descrizione

L'oggetto presetPanel rappresenta il pannello Preimpostazioni di movimento (Finestra > Preimpostazioni di movimento). È una proprietà dell'oggetto flash (fl.presetPanel).

Riepilogo dei metodi

Con l'oggetto presetPanel potete utilizzare i metodi seguenti:

Metodo	Descrizione
presetPanel.addNewItem()	Se sullo stage è selezionata una sola interpolazione di movimento, aggiunge il movimento al pannello Preimpostazioni di movimento.
presetPanel.applyPreset()	Applica la preimpostazione specificata o selezionata all'elemento selezionato sullo stage.
presetPanel.deleteFolder()	Elimina la cartella specificata o le sottocartelle dalla struttura di cartelle del pannello Preimpostazioni di movimento.
presetPanel.deleteItem()	Elimina la preimpostazione specificata dal pannello Preimpostazioni di movimento.
presetPanel.expandFolder()	Espande o comprime la cartella o le cartelle selezionate nel pannello Preimpostazioni di movimento.
presetPanel.exportItem()	Esporta la preimpostazione selezionata o specificata in un file XML.
presetPanel.findItemIndex()	Restituisce un numero intero che rappresenta la posizione di indice di un elemento nel pannello Preimpostazioni di movimento.
presetPanel.getSelectedItems()	Restituisce un array di oggetti presetItem corrispondente agli elementi selezionati nel pannello Preimpostazioni di movimento.
presetPanel.importItem()	Aggiunge una preimpostazione al pannello Preimpostazioni di movimento da un file XML specificato.
presetPanel.moveToFolder()	Sposta l'elemento specificato nella cartella specificata.
presetPanel.newFolder()	Crea una cartella nella struttura di cartelle del pannello Preimpostazioni di movimento.
presetPanel.renameItem()	Rinomina la preimpostazione o la cartella selezionata con il nome specificato.
presetPanel.selectItem()	Seleziona o deseleziona un elemento nel pannello Preimpostazioni di movimento.

Riepilogo delle proprietà

Con la proprietà presetPanel potete utilizzare i metodi seguenti:

Proprietà	Descrizione
presetPanel.items	Un array di oggetti presetItem nel pannello Preimpostazioni di movimento.

presetPanel.addNewItem()

Disponibilità
Flash CS4 Professional.

Uso
```
fl.presetPanel.addNewItem( [namePath] );
```

Parametri
`namePath` Una stringa che specifica il percorso e il nome dell'elemento da aggiungere al pannello Preimpostazioni di movimento. Questo parametro è opzionale.

Restituisce
Il valore booleano `true` se l'elemento è stato aggiunto correttamente; `false` in caso contrario.

Descrizione
Metodo; se sullo stage è selezionata una sola interpolazione di movimento, aggiunge il movimento al pannello Preimpostazioni di movimento nella cartella specificata con il nome specificato. Il percorso specificato in *namePath* deve essere presente nel pannello.

Se è già presente una preimpostazione che corrisponde a *namePath*, il metodo non ha alcun effetto e restituisce `false`.

Se non passate un valore per *namePath*, l'elemento viene aggiunto alla cartella Preimpostazioni personalizzate con il nome "Preimpostazione personalizzata *n*", dove *n* viene incrementato ogni volta che aggiungete un elemento in questo modo.

Esempio
Se sullo stage è selezionata una sola interpolazione di movimento, il codice seguente aggiunge una preimpostazione denominata `Bouncing Ball` alla cartella Preimpostazioni personalizzate:

```
fl.presetPanel.addNewItem("Custom Presets/Bouncing Ball");
```

Consultate anche
presetPanel.newFolder()

presetPanel.applyPreset()

Disponibilità
Flash CS4 Professional.

Uso
```
presetPanel.applyPreset( [presetPath] )
```

Parametri
`presetPath` Una stringa che specifica il percorso e il nome della preimpostazione da applicare, come visualizzato nel pannello Preimpostazioni di movimento. Questo parametro è opzionale; se non passate alcun valore, viene applicata la preimpostazione selezionata.

Restituisce
Il valore booleano `true` se la preimpostazione viene applicata correttamente; `false` in caso contrario.

Descrizione
Metodo; applica la preimpostazione specificata o selezionata all'elemento selezionato sullo stage. L'elemento deve essere un'interpolazione di movimento, un simbolo o un elemento che possa essere convertito in un simbolo. Se l'elemento è un'interpolazione di movimento, il movimento corrente viene sostituito con la preimpostazione selezionata senza richiedere la conferma dell'utente.

Questo metodo non riesce nelle situazioni seguenti:

- il percorso che specificate come *presetPath* non esiste;
- non passate alcun valore per *presetPath* e non è selezionata alcuna preimpostazione;
- non passate alcun valore per *presetPath* e sono selezionate più preimpostazioni;
- l'elemento selezionato sullo stage non è un simbolo e non può essere convertito in un simbolo.

Esempio
L'esempio seguente applica la preimpostazione `aDribble` all'elemento selezionato sullo stage:

```
var result = fl.presetPanel.applyPreset("Custom Presets/Bounces/aDribble");
fl.trace(result);
```

presetPanel.deleteFolder()

Disponibilità
Flash CS4 Professional.

Uso
```
presetPanel.deleteFolder( [folderPath])
```

Parametri
`folderPath` Una stringa che specifica la cartella da eliminare dal pannello Preimpostazioni di movimento. Questo parametro è opzionale.

Restituisce
Il valore booleano `true` se la cartella o le cartelle vengono eliminate correttamente; `false` in caso contrario.

Descrizione
Metodo; elimina la cartella specificata o le sottocartelle dalla struttura di cartelle del pannello Preimpostazioni di movimento. Anche tutte le preimpostazioni presenti nelle cartelle vengono eliminate. Non potete eliminare cartelle dalla cartella Preimpostazioni predefinite.

Se non passate un valore per *folderPath*, tutte le cartelle selezionate vengono eliminate.

Nota: *l'eliminazione delle cartelle avviene senza chiedere la conferma dell'utente e l'azione non può essere annullata.*

Esempio
Il codice seguente elimina una cartella denominata `Bouncing` sotto la cartella Preimpostazioni personalizzate; vengono eliminate anche tutte le sottocartelle di `Bouncing`:

```
fl.presetPanel.deleteFolder("Custom Presets/Bouncing");
```

Consultate anche
presetPanel.deleteItem()

presetPanel.deleteItem()

Disponibilità
Flash CS4 Professional.

Uso
```
presetPanel.deleteItem( [namePath] )
```

Parametri
`namePath` Una stringa che specifica il percorso e il nome dell'elemento da eliminare dal pannello Preimpostazioni di movimento. Questo parametro è opzionale.

Restituisce
Il valore booleano `true` se l'elemento o gli elementi vengono eliminati correttamente; `false` in caso contrario.

Descrizione
Metodo; elimina la preimpostazione specificata dal pannello Preimpostazioni di movimento. Se non passate un valore per *namePath*, tutte le preimpostazioni selezionate vengono eliminate. Non potete eliminare elementi dalla cartella Preimpostazioni predefinite.

Nota: *l'eliminazione degli elementi avviene senza chiedere la conferma dell'utente e l'azione non può essere annullata.*

Esempio
Il codice seguente elimina una preimpostazione denominata `aDribble` dalla cartella Preimpostazioni personalizzate:

```
fl.presetPanel.deleteItem("Custom Presets/aDribble");
```

Consultate anche
presetPanel.deleteFolder()

presetPanel.expandFolder()

Disponibilità
Flash CS4 Professional.

Uso
```
presetPanel.expandFolder( [bExpand [, bRecurse [, folderPath] ] ] )
```

Parametri

`bExpand` Un valore booleano che specifica se espandere la cartella (`true`) o comprimerla (`false`). Questo parametro è opzionale; il valore predefinito è `true`.

`bRecurse` Un valore booleano che specifica se espandere o comprimere le sottocartelle della cartella (`true`) o se non eseguire azioni (`false`). Questo parametro è opzionale; il valore predefinito è `false`.

`folderPath` Una stringa che specifica il percorso della cartella da espandere o comprimere. Questo parametro è opzionale.

Restituisce

Il valore booleano `true` se la cartella o le cartelle vengono espanse o compresse correttamente; `false` in caso contrario.

Descrizione

Metodo; espande o comprime la cartella o le cartelle selezionate nel pannello Preimpostazioni di movimento. Per espandere o comprimere cartelle diverse da quelle selezionate, passate un valore per *folderPath*.

Esempio

L'esempio seguente espande la cartella Preimpostazioni personalizzate, ma non le sottocartelle:

```
fl.presetPanel.expandFolder(true, false, "Custom Presets");
```

L'esempio seguente espande la cartella Preimpostazioni personalizzate e tutte le sottocartelle:

```
fl.presetPanel.expandFolder(true, true, "Custom Presets");
```

presetPanel.exportItem()

Disponibilità

Flash CS4 Professional.

Uso

```
presetPanel.exportItem(fileURI [, namePath] )
```

Parametri

`fileURI` Una stringa, espressa come URI file:///, che specifica il percorso e, facoltativamente, un nome per il file esportato. Per ulteriori informazioni, consultate "Descrizione" più avanti.

`namePath` Una stringa che specifica il percorso e il nome dell'elemento da selezionare dal pannello Preimpostazioni di movimento. Questo parametro è opzionale.

Restituisce

Il valore booleano `true` se la preimpostazione è stata esportata correttamente; `false` in caso contrario.

Descrizione

Metodo; esporta la preimpostazione selezionata o specificata in un file XML. È possibile esportare solo le preimpostazioni; il metodo non riesce se tentate di esportare una cartella. Il metodo fallisce inoltre se tentate di sovrascrivere un file su disco.

Se non specificate un nome di file come parte di *fileURI* (ovvero se l'ultimo carattere di *fileURI* è una barra (/)), il file esportato viene salvato con lo stesso nome della preimpostazione esportata. Se non specificate un valore per *namePath*, viene esportata la preimpostazione selezionata. Consultate l'esempio riportato di seguito.

Esempio

L'esempio seguente dimostra i file creati quando diversi parametri vengono passati a questo metodo e informa l'utente della creazione del file specificato. Prima di eseguire l'esempio, selezionate la preimpostazione fly-in-left nella cartella Preimpostazioni predefinite e create la cartella `My Presets` su disco.

```
//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");
```

Consultate anche

`presetPanel.importItem()`

presetPanel.findItemIndex()

Disponibilità

Flash CS4 Professional.

Uso

```
presetPanel.findItemIndex([presetName])
```

Parametri

`presetName` Una stringa che specifica il nome della preimpostazione per cui viene restituito il valore di indice. Questo parametro è opzionale.

Restituisce

Un numero intero che rappresenta l'indice della preimpostazione specificata nell'array `presetPanel.items`. Se non passate un valore per *presetName*, viene restituito l'indice della preimpostazione selezionata. Questo metodo restituisce -1 nelle situazioni seguenti:

- non passate alcun valore per *presetName* e non è selezionata alcuna preimpostazione;
- non passate alcun valore per *presetName* e sono selezionate più preimpostazioni;
- passate un valore per *presetName* che non corrisponde a un elemento nel pannello.

Descrizione

Metodo; restituisce un numero intero che rappresenta la posizione di indice di un elemento nel pannello Preimpostazioni di movimento.

Esempio

Il codice seguente visualizza il valore di indice e il nome completo del percorso della preimpostazione selezionata:

```
// 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()

Disponibilità

Flash CS4 Professional.

Uso

```
presetPanel.getSelectedItems()
```

Parametri

Nessuno.

Restituisce

Un array di oggetti presetItem.

Descrizione

Metodo; restituisce un array di oggetti presetItem corrispondente agli elementi selezionati nel pannello Preimpostazioni di movimento (consultate Oggetto presetItem). Ogni elemento dell'array rappresenta una cartella o una preimpostazione.

Esempio

Il codice seguente visualizza il nome completo del percorso degli elementi selezionati nel pannello Preimpostazioni di movimento:

```
var itemArray = fl.presetPanel.getSelectedItems();
var length =  itemArray.length
for (x=0; x<length; x++) {
    fl.trace(itemArray[x].path);
}
```

Consultate anche

presetPanel.items

presetPanel.importItem()

Disponibilità

Flash CS4 Professional.

Uso

```
presetPanel.importItem(fileURI [,namePath ])
```

Parametri

`fileURI` Una stringa, espressa come URI file:///, che specifica il file XML da importare come preimpostazione nel pannello Preimpostazioni di movimento.

`namePath` Una stringa che specifica in quale cartella inserire il file importato e il nome da assegnare. Questo parametro è opzionale.

Restituisce

Il valore booleano `true` se il file viene importato correttamente; `false` in caso contrario.

Descrizione

Metodo; aggiunge una preimpostazione al pannello Preimpostazioni di movimento da un file XML specificato. Il percorso specificato in *namePath* deve essere presente nel pannello.

Per creare file XML che possano essere importati, utilizzate `presetPanel.exportItem()`.

Se non passate un valore per *namePath*, la preimpostazione importata viene inserita nella cartella Preimpostazioni personalizzate e le viene assegnato lo stesso nome del file importato (senza estensione XML).

Esempio

L'esempio seguente importa una preimpostazione nella cartella Preimpostazioni personalizzate/Pulse e la denomina `fastPulse`.

```
fl.presetPanel.importItem("file:///C|/My Presets/thePulsePreset.xml", "Custom
Presets/Pulse/fastPulse");
```

Consultate anche

`presetPanel.exportItem()`

presetPanel.items

Disponibilità

Flash CS4 Professional.

Uso

```
presetPanel.items
```

Descrizione

Proprietà; un array di oggetti presetItem nel pannello Preimpostazioni di movimento (consultate Oggetto presetItem). Ogni elemento dell'array rappresenta una cartella o una preimpostazione.

Esempio

Il codice seguente visualizza il nome completo del percorso degli elementi nel pannello Preimpostazioni di movimento:

```
var itemArray = fl.presetPanel.items;
var length =  itemArray.length
for (x=0; x<length; x++) {
    fl.trace(itemArray[x].path);
}
```

Consultate anche
presetPanel.getSelectedItems()

presetPanel.moveToFolder()

Disponibilità
Flash CS4 Professional.

Uso
```
presetPanel.moveToFolder(folderPath [, namePath] )
```

Parametri
`folderPath` Una stringa che specifica il percorso della cartella nel pannello Preimpostazioni di movimento in cui vengono spostati gli elementi o l'elemento.

`namePath` Una stringa che specifica il percorso e il nome dell'elemento da spostare. Questo parametro è opzionale.

Restituisce
Il valore booleano `true` se gli elementi vengono spostati correttamente; `false` in caso contrario.

Descrizione
Metodo; sposta l'elemento specificato nella cartella specificata.

Se passate una stringa vuota ("") per *folderPath*, gli elementi vengono spostati nella cartella Preimpostazioni personalizzate. Se non passate alcun valore per *namePath*, vengono spostati gli elementi selezionati.

Non potete spostare elementi dalla cartella Preimpostazioni predefinite o all'interno di essa.

Esempio
Nell'esempio seguente gli elementi selezionati vengono spostati nella cartella Custom Presets/Bouncing e quindi la preimpostazione Fast Bounce viene spostata nella stessa cartella:

```
fl.presetPanel.moveToFolder("Custom Presets/Bouncing");
fl.presetPanel.moveToFolder("Custom Presets/Bouncing" , "Custom Presets/Fast Bounce");
```

presetPanel.newFolder()

Disponibilità
Flash CS4 Professional.

Uso
```
presetPanel.newFolder( [folderPath] )
```

Parametri
`folderPath` Una stringa che specifica dove aggiungere una nuova cartella nel pannello Preimpostazioni di movimento e il nome della nuova cartella. Questo parametro è opzionale.

Restituisce
Il valore booleano `true` se la cartella viene aggiunta correttamente; `false` in caso contrario.

Descrizione
Metodo; crea una cartella nella struttura di cartelle del pannello Preimpostazioni di movimento. Con questo metodo potete creare un solo nuovo livello di cartelle. Se pertanto per *folderPath* passate "Custom Presets/My First Folder/My Second Folder", "Custom Presets/My First Folder" deve essere già presente nella struttura di cartelle.

Se non passate alcun valore per *folderPath*, una cartella denominata "Untitled folder *n*" viene creata al primo livello sotto "Custom Presets", dove *n* viene incrementato ogni volta che una cartella viene aggiunta in questo modo.

Nota: *non potete aggiungere cartelle nella cartella Custom Presets.*

Esempio
Il codice seguente aggiunge una cartella denominata `Bouncing` nella cartella Custom Presets:

```
fl.presetPanel.newFolder("Custom Presets/Bouncing");
```

Consultate anche
`presetPanel.addNewItem()`

presetPanel.renameItem()

Disponibilità
Flash CS4 Professional.

Uso
```
presetPanel.renameItem(newName)
```

Parametri
`newName` Una stringa che specifica il nuovo nome per la preimpostazione o la cartella.

Restituisce
Il valore booleano `true` se la preimpostazione o la cartella viene rinominata correttamente; `false` in caso contrario.

Descrizione
Metodo; rinomina la preimpostazione o la cartella selezionata con il nome specificato. Il metodo ha esito positivo solo se nella cartella Preimpostazioni personalizzate viene selezionata una sola preimpostazione o cartella. Questo metodo non riesce nelle situazioni seguenti:

- non è selezionato alcun elemento;
- sono selezionati più elementi;
- l'elemento selezionato si trova nella cartella Preimpostazioni predefinite.
- nello stesso percorso dell'elemento selezionato è presente un elemento denominato *newName*.

Esempio
L'esempio seguente rinomina la preimpostazione selezionata nella cartella Preimpostazioni personalizzate in `Bounce Faster`.

```
var renamed = fl.presetPanel.renameItem("Bounce Faster");
fl.trace(renamed);
```

presetPanel.selectItem()

Disponibilità
Flash CS4 Professional.

Uso
```
presetPanel.selectItem(namePath [, bReplaceCurrentSelection [, bSelect] ])
```

Parametri
`namePath` Una stringa che specifica il percorso e il nome dell'elemento da selezionare dal pannello Preimpostazioni di movimento.

`bReplaceCurrentSelection` Un valore booleano che specifica se l'elemento specificato sostituisce le selezioni correnti (`true`) o viene aggiunto alla selezione corrente (`false`). Questo parametro è opzionale; il valore predefinito è `true`.

`bSelect` Un valore booleano che specifica se selezionare l'elemento (`true`) o deselezionarlo (`false`). Questo parametro è opzionale; il valore predefinito è `true`. Se passate `false` per *bSelect*, il valore di *bReplaceCurrentSelection* viene ignorato.

Restituisce
Il valore booleano `true` se l'elemento è stato selezionato o deselezionato correttamente; `false` in caso contrario.

Descrizione
Metodo; seleziona o deseleziona un elemento nel pannello Preimpostazioni di movimento, sostituendo facoltativamente eventuali elementi selezionati.

Esempio
Il codice seguente aggiunge la preimpostazione `fly-in-blur-right` alle preimpostazioni selezionate (se presenti) nel pannello Preimpostazioni di movimento:

```
fl.presetPanel.selectItem("Default Presets/fly-in-blur-right", false);
```

Capitolo 35: Oggetto Rectangle

Ereditarietà Oggetto Element > Oggetto Shape > Oggetto Rectangle

Disponibilità
Flash CS3 Professional.

Descrizione
L'oggetto Rectangle è una forma disegnata utilizzando lo strumento Rettangolo di base. Per determinare se un elemento è un oggetto Rectangle, utilizzate `shape.isRectangleObject`.

Riepilogo delle proprietà
Oltre a quelle dell'Oggetto Shape, con l'oggetto Rectangle potete usare le seguenti proprietà. Per impostare le proprietà di un oggetto Rectangle, utilizzate `document.setRectangleObjectProperty()`.

Proprietà	Descrizione
`RectangleObject.bottomLeftRadius`	Di sola lettura; valore float che imposta il raggio dell'angolo inferiore sinistro dell'oggetto Rectangle.
`RectangleObject.bottomRightRadius`	Di sola lettura; valore float che imposta il raggio dell'angolo inferiore destro dell'oggetto Rectangle.
`RectangleObject.lockFlag`	Di sola lettura; valore booleano che determina se angoli differenti del rettangolo possono avere valori di raggio differenti.
`RectangleObject.topLeftRadius`	Di sola lettura; valore float che imposta il raggio di tutti gli angoli del rettangolo, oppure del solo angolo superiore sinistro dell'oggetto Rectangle.
`RectangleObject.topRightRadius`	Di sola lettura; valore float che imposta il raggio dell'angolo superiore destro dell'oggetto Rectangle.

RectangleObject.bottomLeftRadius

Disponibilità
Flash CS3 Professional.

Uso
```
RectangleObject.bottomLeftRadius
```

Descrizione
Proprietà di sola lettura; valore float che imposta il raggio dell'angolo inferiore sinistro dell'oggetto Rectangle. Se `RectangleObject.lockFlag` è `true`, il tentativo di impostare questo valore non riesce.

Per impostare questo valore, utilizzate `document.setRectangleObjectProperty()`.

Consultate anche

document.setRectangleObjectProperty(), RectangleObject.bottomRightRadius, RectangleObject.lockFlag, RectangleObject.topLeftRadius, RectangleObject.topRightRadius

RectangleObject.bottomRightRadius

Disponibilità

Flash CS3 Professional.

Uso

```
RectangleObject.bottomRightRadius
```

Descrizione

Proprietà di sola lettura; valore float che imposta il raggio dell'angolo inferiore destro dell'oggetto Rectangle. Se RectangleObject.lockFlag è true, il tentativo di impostare questo valore non riesce.

Per impostare questo valore, utilizzate document.setRectangleObjectProperty().

Consultate anche

document.setRectangleObjectProperty(), RectangleObject.bottomLeftRadius, RectangleObject.lockFlag, RectangleObject.topLeftRadius, RectangleObject.topRightRadius

RectangleObject.lockFlag

Disponibilità

Flash CS3 Professional.

Uso

```
RectangleObject.lockFlag
```

Descrizione

Proprietà di sola lettura; valore booleano che determina se angoli differenti del rettangolo possono avere valori di raggio differenti. Se il valore è `true`, tutti gli angoli assumono il valore assegnato a `RectangleObject.topLeftRadius`. Se `false`, il raggio di ciascun angolo può essere impostato in modo indipendente.

Per impostare questo valore, utilizzate document.setRectangleObjectProperty().

Consultate anche

document.setRectangleObjectProperty(), RectangleObject.bottomLeftRadius, RectangleObject.bottomRightRadius, RectangleObject.topLeftRadius, RectangleObject.topRightRadius

RectangleObject.topLeftRadius

Disponibilità
Flash CS3 Professional.

Uso
`RectangleObject.topLeftRadius`

Descrizione
Proprietà di sola lettura; valore float che imposta il raggio di tutti gli angoli del rettangolo (se `RectangleObject.lockFlag` è `true`), oppure il raggio del solo angolo superiore sinistro (se `RectangleObject.lockFlag` è `false`).

Per impostare questo valore, utilizzate `document.setRectangleObjectProperty()`.

Consultate anche
`document.setRectangleObjectProperty()`, `RectangleObject.bottomLeftRadius`, `RectangleObject.bottomRightRadius`, `RectangleObject.lockFlag`, `RectangleObject.topRightRadius`

RectangleObject.topRightRadius

Disponibilità
Flash CS3 Professional.

Uso
`RectangleObject.topRightRadius`

Descrizione
Proprietà di sola lettura; valore float che imposta il raggio dell'angolo superiore destro dell'oggetto Rectangle. Se `RectangleObject.lockFlag` è `true`, il tentativo di impostare questo valore non riesce.

Per impostare questo valore, utilizzate `document.setRectangleObjectProperty()`.

Consultate anche
`document.setRectangleObjectProperty()`, `RectangleObject.bottomLeftRadius`, `RectangleObject.bottomRightRadius`, `RectangleObject.lockFlag`, `RectangleObject.topLeftRadius`

Capitolo 36: Oggetto Screen

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto Screen rappresenta una schermata singola in una diapositiva o in un form. Contiene le proprietà relative alla diapositiva o al form. Per accedere all'array di tutti gli oggetti Screen nel documento, utilizzate il codice seguente:

```
fl.getDocumentDOM().screenOutline.screens
```

Riepilogo delle proprietà
L'oggetto Screen ha le proprietà seguenti:

Proprietà	Descrizione
`screen.accName`	Una stringa equivalente al campo Nome del pannello Accessibilità.
`screen.childScreens`	Di sola lettura; l'array delle schermate secondarie della schermata. L'array è vuoto se non sono presenti schermate secondarie.
`screen.description`	Una stringa equivalente al campo Descrizione del pannello Accessibilità.
`screen.forceSimple`	Un valore booleano che abilita e disabilita l'accessibilità degli elementi secondari dell'oggetto.
`screen.hidden`	Un valore booleano che specifica se una schermata è visibile.
`screen.instanceName`	Di sola lettura; una stringa che rappresenta il nome di istanza utilizzato per accedere all'oggetto mediante ActionScript.
`screen.name`	Di sola lettura; una stringa che specifica il nome della schermata.
`screen.nextScreen`	Di sola lettura; un oggetto che rappresenta la successiva schermata di pari livello nell'array `childScreens` dell'elemento principale.
`screen.parameters`	Di sola lettura; un array di proprietà ActionScript 2.0 a cui è possibile accedere dalla finestra di ispezione Proprietà delle schermate.
`screen.parentScreen`	Di sola lettura; un oggetto che rappresenta la schermata principale.
`screen.prevScreen`	Di sola lettura; un oggetto che rappresenta la precedente schermata di pari livello nell'array `childScreens` dell'elemento principale.
`screen.silent`	Un valore booleano che specifica se l'oggetto è accessibile.
`screen.tabIndex`	Equivalente al campo Indice tabulazione del pannello Accessibilità.
`screen.timeline`	Di sola lettura; l'oggetto della linea temporale della schermata. Consultate Oggetto Timeline.

screen.accName

Disponibilità
Flash MX 2004.

Uso
`screen.accName`

Descrizione
Proprietà; una stringa equivalente al campo Nome del pannello Accessibilità. Gli screen reader identificano gli oggetti pronunciandone il nome.

Esempio
L'esempio seguente memorizza il valore del nome dell'oggetto nella variabile `theName`:

```
var theName = fl.getDocumentDOM().screenOutline.screens[1].accName;
```

L'esempio seguente imposta `"Home Button"` come nome dell'oggetto:

```
fl.getDocumentDOM().screenOutline.screens[1].accName = 'Home Button';
```

screen.childScreens

Disponibilità
Flash MX 2004.

Uso
`screen.childScreens`

Descrizione
Proprietà di sola lettura; l'array delle schermate secondarie della schermata. L'array è vuoto se non sono presenti schermate secondarie.

Esempio
L'esempio seguente verifica se il documento corrente è una diapositiva o un form e, in caso affermativo, memorizza l'array delle schermate secondarie nella variabile `myChildren` e ne visualizza i nomi nel pannello Output:

```
var myChildren = new Array();
if(fl.getDocumentDOM().allowScreens) {
    var myParent = fl.getDocumentDOM().screenOutline.rootScreen.name
    for (i in fl.getDocumentDOM().screenOutline.rootScreen.childScreens) {
        myChildren.push(" "+fl.getDocumentDOM().screenOutline.rootScreen.childScreens[i].name);
    }
    fl.trace(" The child screens of "+myParent+" are "+myChildren+". ");
}
```

screen.description

Disponibilità
Flash MX 2004.

Uso
`screen.description`

Descrizione
Proprietà; una stringa equivalente al campo Descrizione del pannello Accessibilità. La descrizione viene letta dallo screen reader.

Esempio
L'esempio seguente ottiene la descrizione della schermata e la memorizza nella variabile `theDescription`:

```
var theDescription = fl.getDocumentDOM().screenOutline.screens[1].description;
```

L'esempio seguente imposta la descrizione della schermata su `Home Screen`:

```
fl.getDocumentDOM().screenOutline.screens[1].description = "Home Screen";
```

screen.forceSimple

Disponibilità
Flash MX 2004.

Uso
`screen.forceSimple`

Descrizione
Proprietà; un valore booleano che abilita o disabilita l'accessibilità degli elementi secondari dell'oggetto. È equivalente alla logica inversa dell'impostazione Rendi accessibili gli oggetti secondari del pannello Accessibilità. In altre parole, un valore `true` per `forceSimple` equivale all'opzione Rendi accessibili gli oggetti secondari deselezionata, mentre un valore `false`equivale alla stessa opzione selezionata.

Esempio
L'esempio seguente memorizza il valore di `forceSimple` nella variabile `areChildrenAccessible` (un valore `false` indica che gli elementi secondari dell'oggetto sono accessibili):

```
var areChildrenAccessible = fl.getDocumentDOM().screenOutline.screens[1].forceSimple
```

L'esempio seguente rende accessibili gli elementi secondari dell'oggetto:

```
fl.getDocumentDOM().screenOutline.screens[1].forceSimple = false;
```

screen.hidden

Disponibilità
Flash MX 2004.

Uso
`screen.hidden`

Descrizione
Proprietà; un valore booleano che specifica se la schermata è visibile. Una schermata con la proprietà `hidden` impostata su `true` non è visibile in altre schermate.

Esempio

L'esempio seguente verifica se la prima schermata nel contorno è nascosta e modifica di conseguenza la visibilità della schermata. Quindi, un messaggio nel pannello Output mostra lo stato della visibilità della schermata precedente alla modifica:

```
if (fl.getDocumentDOM().screenOutline.screens[0].hidden) {
    fl.getDocumentDOM().screenOutline.setScreenProperty("hidden", false);
    fl.trace(fl.getDocumentDOM().screenOutline.screens[0].name+" had its 'hidden' property
set to 'false'");
}
else {
    fl.getDocumentDOM().screenOutline.setScreenProperty("hidden", true);
    fl.trace(fl.getDocumentDOM().screenOutline.screens[0].name+" had its 'hidden' property
set to 'true'");
}
```

screen.instanceName

Disponibilità

Flash MX 2004.

Uso

```
screen.instanceName
```

Descrizione

Proprietà di sola lettura; una stringa che rappresenta il nome di istanza utilizzato per accedere all'oggetto mediante ActionScript.

Esempio

L'esempio seguente verifica se il documento corrente consente l'uso delle schermate (poiché è una diapositiva o un form). Quindi, assegna il valore `instanceName` della prima schermata secondaria dell'array alla variabile `myInstanceName` e apre il pannello Output per mostrare il nome di istanza della schermata:

```
var myChildren = new Array();
if(fl.getDocumentDOM().allowScreens) {
    var myInstanceName =
fl.getDocumentDOM().screenOutline.rootScreen.childScreens[0].instanceName;
    fl.trace(" The instanceName is "+myInstanceName+". ");
}
```

screen.name

Disponibilità

Flash MX 2004.

Uso

```
screen.name
```

Descrizione
Proprietà di sola lettura; una stringa che rappresenta il nome della schermata.

Esempio
L'esempio seguente verifica se il documento corrente consente l'uso delle schermate (poiché è una diapositiva o un form). Quindi, assegna il valore `name` della prima schermata secondaria dell'array alla variabile `myName` e apre il pannello Output per mostrare il nome della schermata:

```
var myChildren = new Array();
if(fl.getDocumentDOM().allowScreens) {
    var myName = fl.getDocumentDOM().screenOutline.rootScreen.childScreens[0].name;
    fl.trace("The name of the screen is "+myName+". ");
}
```

screen.nextScreen

Disponibilità
Flash MX 2004.

Uso
```
screen.nextScreen
```

Descrizione
Proprietà di sola lettura; un oggetto che rappresenta la schermata successiva di pari livello nell'array `childScreens` dell'elemento principale. In altre parole, `screen.nextScreen` viene trovato passando alla schermata successiva verso il basso nell'array delle schermate secondarie. Consultate `screen.prevScreen`.

Se non è presente una schermata di pari livello, il valore è `null`.

Esempio
L'esempio seguente verifica se il documento corrente è una diapositiva o un form e, in caso affermativo, recupera e mostra la sequenza delle schermate nel pannello Output:

```
if(fl.getDocumentDOM().allowScreens) {
    var myCurrent = fl.getDocumentDOM().screenOutline.rootScreen.childScreens[0].name;
    var myNext = fl.getDocumentDOM().screenOutline.rootScreen.childScreens[0].nextScreen.name;
    fl.trace(" The next screen to "+myCurrent+" is "+myNext+". ");
}
```

screen.parameters

Disponibilità
Flash MX 2004.

Uso
```
screen.parameters
```

Descrizione
Proprietà di sola lettura; un array di proprietà ActionScript 2.0 a cui è possibile accedere dalla finestra di ispezione Proprietà delle schermate.

Esempio
L'esempio seguente memorizza i parametri della seconda schermata nel contorno nella variabile `parms`, quindi assegna il valore `some value` alla prima proprietà:

```
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters;
parms[0].value = "some value";
```

Consultate anche
Oggetto Parameter

screen.parentScreen

Disponibilità
Flash MX 2004.

Uso
`screen.parentScreen`

Descrizione
Proprietà di sola lettura; un oggetto che rappresenta la schermata principale. Se `parentScreen` è `null`, la schermata è di primo livello.

Esempio
L'esempio seguente memorizza i valori delle proprietà `childScreens` e `parentScreen` nelle variabili, quindi li visualizza nel pannello Output insieme alle rispettive relazioni gerarchiche:

```
if(fl.getDocumentDOM().allowScreens) {
    var myCurrent = fl.getDocumentDOM().screenOutline.rootScreen.childScreens[1].name;
    var myParent =
fl.getDocumentDOM().screenOutline.rootScreen.childScreens[1].parentScreen.name;
    fl.trace(" The parent screen to "+myCurrent+" is "+myParent+". ");
}
```

screen.prevScreen

Disponibilità
Flash MX 2004.

Uso
`screen.prevScreen`

Descrizione
Proprietà di sola lettura; un oggetto che rappresenta la schermata precedente di pari livello nell'array `childScreens` dell'elemento principale. Se non è presente una schermata di pari livello, il valore è `null`. Consultate anche `screen.nextScreen`.

Esempio
L'esempio seguente verifica se il documento corrente è una diapositiva o un form e, in caso affermativo, recupera e mostra la sequenza delle schermate nel pannello Output:

```
if(fl.getDocumentDOM().allowScreens) {
    var myCurrent = fl.getDocumentDOM().screenOutline.rootScreen.childScreens[1].name;
    var myNext = fl.getDocumentDOM().screenOutline.rootScreen.childScreens[1].prevScreen.name;
    fl.trace(" The previous screen to "+myCurrent+" is "+myNext+". ");
}
```

screen.silent

Disponibilità
Flash MX 2004.

Uso
`screen.silent`

Descrizione
Proprietà; un valore booleano che specifica se l'oggetto è accessibile. È equivalente alla logica inversa dell'impostazione Rendi accessibile l'oggetto del pannello Accessibilità. In altre parole, un valore `true` per `silent` equivale all'opzione Rendi accessibili gli oggetti secondari deselezionata, mentre un valore `false`equivale alla stessa opzione selezionata.

Esempio
L'esempio seguente recupera il valore `silent` dell'oggetto (il valore `false` indica che l'oggetto è accessibile):

```
var isSilent = fl.getDocumentDOM().screenOutline.screens[1].silent;
```

L'esempio seguente imposta l'oggetto come accessibile:

```
fl.getDocumentDOM().screenOutline.screens[1].silent = false;
```

screen.tabIndex

Disponibilità
Flash MX 2004.

Uso
`screen.tabIndex`

Descrizione
Proprietà; equivalente al campo Indice tabulazione del pannello Accessibilità. Consente di determinare l'ordine di accesso agli oggetti quando l'utente preme il tasto Tab.

Esempio

L'esempio seguente ottiene l'indice di tabulazione dell'oggetto:

```
var theTabIndex = fl.getDocumentDOM().screenOutline.screens[1].tabIndex;
```

L'esempio seguente imposta l'indice di tabulazione dell'oggetto su 1:

```
fl.getDocumentDOM().screenOutline.screens[1].tabIndex = 1;
```

screen.timeline

Disponibilità

Flash MX 2004.

Uso

```
screen.timeline
```

Descrizione

Proprietà di sola lettura; l'Oggetto Timeline per la schermata.

Esempio

L'esempio seguente ottiene la proprietà `screenOutline` della diapositiva corrente, assegna a `myArray` l'array delle proprietà `timeline` della prima schermata e visualizza le proprietà nel pannello Output:

```
myArray = new Array();
if(fl.getDocumentDOM().screenOutline) {
    for(i in fl.getDocumentDOM().screenOutline.screens[0].timeline) {
    myArray.push(" "+i+" : "+fl.getDocumentDOM().screenOutline.screens[0].timeline[i]+" ") ;
    }
    fl.trace("Here are the properties of the screen named "+
fl.getDocumentDOM().screenOutline.screens[0].name+": "+myArray);
}
```

Capitolo 37: Oggetto ScreenOutline

Disponibilità

Flash MX 2004.

Descrizione

L'oggetto ScreenOutline rappresenta il gruppo di schermate in una diapositiva o in un form. È accessibile mediante `fl.getDocumentDOM().screenOutline`.

L'oggetto ScreenOutline esiste solo se il documento è una diapositiva o un form, pertanto, prima di accedere alla proprietà, utilizzate `document.allowScreens()` per verificare che sia presente un documento Screens, come mostrato nell'esempio seguente:

```
if(fl.getDocumentDOM().allowScreens) {
var myName = fl.getDocumentDOM().screenOutline.rootScreen.childScreens[0].name;
fl.trace("The name of the screen is " + myName + ". ");
}
```

Riepilogo dei metodi

Con l'oggetto ScreenOutline è possibile utilizzare i metodi seguenti:

Metodo	Descrizione
`screenOutline.copyScreenFromFile()`	Preleva da un documento specificato tutte le schermate, o una schermata specifica con i relativi elementi secondari, e le inserisce nella schermata selezionata.
`screenOutline.deleteScreen()`	Elimina le schermate selezionate oppure una schermata specifica, insieme ai relativi elementi secondari.
`screenOutline.duplicateScreen()`	Duplica le schermate selezionate oppure una schermata specifica.
`screenOutline.getSelectedScreens()`	Restituisce un array degli oggetti Screen selezionati nella struttura delle schermate.
`screenOutline.insertNestedScreen()`	Inserisce una schermata nidificata di un tipo specifico in una posizione particolare della struttura delle schermate.
`screenOutline.insertScreen()`	Inserisce una nuova schermata vuota di un tipo specifico in una posizione particolare del documento.
`screenOutline.moveScreen()`	Sposta la schermata specificata in relazione al valore del parametro ***`referenceScreen`***: prima, dopo o come primo o ultimo elemento secondario.
`screenOutline.renameScreen()`	Sostituisce il nome specificato di una schermata con un nuovo nome.
`screenOutline.setCurrentScreen()`	Imposta la schermata specificata come selezione corrente nella struttura delle schermate.
`screenOutline.setScreenProperty()`	Imposta il valore specificato per la proprietà specificata delle schermate selezionate.
`screenOutline.setSelectedScreens()`	Seleziona le schermate specificate nel pannello Contorno schermata.

Riepilogo delle proprietà

Con l'oggetto ScreenOutline è possibile utilizzare le proprietà seguenti:

Proprietà	Descrizione
screenOutline.currentScreen	Un Oggetto Screen; la schermata attualmente selezionata.
screenOutline.rootScreen	Di sola lettura; la prima schermata nella struttura delle schermate.
screenOutline.screens	Sola lettura; l'array di oggetti Screen di primo livello contenuti nel documento (consultate Oggetto Screen).

screenOutline.copyScreenFromFile()

Disponibilità
Flash MX 2004.

Uso
```
screenOutline.copyScreenFromFile(fileURI [, screenName])
```

Parametri
fileURI Una stringa, espressa come URI file:///, che specifica un nome per il file di creazione che contiene le schermate da copiare nel documento.

screenName Il nome della schermata da copiare. Se è presente il parametro *screenName*, vengono copiati la schermata e i relativi elementi secondari. Se il parametro *screenName* non è specificato, viene copiato l'intero documento. Questo parametro è opzionale.

Restituisce
Nulla. Se il file non viene trovato o non è un file FLA valido, oppure se non viene trovata la schermata specificata, viene generato un errore e lo script viene annullato.

Descrizione
Metodo; preleva da un documento specificato tutte le schermate, o una schermata specifica con i relativi elementi secondari, e le inserisce nella schermata selezionata. Se è selezionata più di una schermata, le schermate vengono inserite nell'ultima schermata selezionata, come elementi di pari livello.

Esempio
L'esempio seguente copia la schermata `slide1` dal file myTarget.fla presente sul desktop al documento corrente (sostituite *userName* con il vostro nome utente):

```
fl.getDocumentDOM().screenOutline.copyScreenFromFile("file:///C|/Documents and
Settings/userName/Desktop/myTarget.fla", "slide1");
```

screenOutline.currentScreen

Disponibilità
Flash MX 2004.

Uso
```
screenOutline.currentScreen
```

Descrizione
Proprietà; un oggetto Screen, la schermata selezionata (consultate Oggetto Screen).

Esempio
L'esempio seguente memorizza nella variabile `myScreen` l'oggetto `currentScreen`, quindi visualizza il nome della schermata nel pannello Output:

```
var myScreen = fl.getDocumentDOM().screenOutline.currentScreen;
fl.trace(myScreen.name);
```

screenOutline.deleteScreen()

Disponibilità
Flash MX 2004.

Uso
`screenOutline.deleteScreen([screenName])`

Parametri
`screenName` Una stringa che specifica il nome della schermata da eliminare. Se non passate un valore per *screenName*, vengono eliminate le schermate selezionate e i relativi elementi secondari. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; elimina le schermate selezionate oppure una schermata specifica, insieme ai relativi elementi secondari.

Esempio
L'esempio seguente elimina la schermata denominata `apple` e tutti i relativi elementi secondari:

```
fl.getDocumentDOM().screenOutline.deleteScreen("apple");
```

screenOutline.duplicateScreen()

Disponibilità
Flash MX 2004.

Uso
`screenOutline.duplicateScreen([screenName])`

Parametri
`screenName` Un valore di stringa che specifica il nome della schermata da duplicare. Se non passate un valore per *screenName*, vengono duplicate le schermate selezionate. Questo parametro è opzionale.

Restituisce
Un valore booleano: `true` se la schermata viene duplicata correttamente; `false` in caso contrario.

Descrizione
Metodo; duplica le schermate selezionate oppure una schermata specifica. Alle schermate duplicate viene assegnato un nome predefinito aggiungendo al nome originale la desinenza `_copy` (ad esempio, Screen_copy, Screen_copy2, e così via). Se duplicate più schermate, queste vengono collocate direttamente sotto la schermata selezionata, che è quella di livello più basso nella gerarchia della struttura delle schermate.

Esempio
L'esempio seguente duplica una schermata denominata apple:

```
fl.getDocumentDOM().screenOutline.duplicateScreen("apple");
```

screenOutline.getSelectedScreens()

Disponibilità
Flash MX 2004.

Uso
```
screenOutline.getSelectedScreens()
```

Parametri
Nessuno.

Restituisce
Un array degli oggetti Screen selezionati (consultate Oggetto Screen).

Descrizione
Metodo; restituisce un array degli oggetti Screen selezionati nella struttura delle schermate.

Esempio
L'esempio seguente memorizza gli oggetti Screen selezionati nella variabile `myArray` e visualizza i nomi delle schermate nel pannello Output:

```
var myArray = fl.getDocumentDOM().screenOutline.getSelectedScreens();
for (var i in myArray) {
    fl.trace(myArray[i].name)
}
```

screenOutline.insertNestedScreen()

Disponibilità
Flash MX 2004.

Uso

```
screenOutline.insertNestedScreen([name [, referenceScreen [, screenTypeName]]])
```

Parametri

name Una stringa che indica il nome della nuova schermata da inserire. Un nome vuoto inserisce una schermata con un nome predefinito, ad esempio Diapositiva*n* o Form*n* (dove *n* è il primo numero univoco disponibile). Questo parametro è opzionale.

referenceScreen Una stringa che indica il nome della schermata in cui la nuova schermata viene inserita come elemento secondario. Se questo parametro non viene specificato, la nuova schermata viene inserita come elemento secondario nella schermata attualmente selezionata. Questo parametro è opzionale.

screenTypeName Una stringa che specifica il tipo di schermata da aggiungere alla nuova schermata nidificata. Vengono impostati il tipo e il nome di classe per la schermata. I valori accettabili sono `"Form"` e `"Slide"`. Questo parametro è opzionale. Se questo parametro non viene specificato, il tipo viene ereditato dalla schermata principale.

Restituisce

Un Oggetto Screen.

Descrizione

Metodo; inserisce una schermata nidificata di un tipo specifico in una posizione particolare della struttura delle schermate.

Esempio

L'esempio seguente inserisce la diapositiva `slide2` come elemento secondario della diapositiva `slide1`:

```
fl.getDocumentDOM().screenOutline.insertNestedScreen("slide2", "slide1", "Slide");
```

screenOutline.insertScreen()

Disponibilità

Flash MX 2004.

Uso

```
screenOutline.insertScreen([name [, referenceScreen [, screenTypeName]]])
```

Parametri

name Una stringa che indica il nome della nuova schermata da inserire. Un nome vuoto inserisce una schermata con un nome predefinito, ad esempio Diapositiva*n* o Form*n* (dove *n* è il primo numero univoco disponibile). Questo parametro è opzionale.

referenceScreen Una stringa che indica il nome della schermata precedente a quella nuova. Se questo parametro viene omesso, la nuova schermata viene inserita dopo quella selezionata. Se il parametro *referenceScreen* identifica una schermata secondaria, la nuova schermata è di pari livello della schermata secondaria ed è una schermata secondaria della stessa schermata principale. Questo parametro è opzionale.

screenTypeName Una stringa che specifica il tipo di schermata da aggiungere a quella nuova. Vengono impostati il tipo e il nome di classe per la schermata. I valori accettabili sono `"Form"` e `"Slide"`. Questo parametro è opzionale.

Restituisce
Un Oggetto Screen.

Descrizione
Metodo; inserisce una nuova schermata vuota di un tipo specifico in una posizione particolare del documento.

Esempio
L'esempio seguente inserisce il form `slide2` dopo la schermata `slide1`:

```
fl.getDocumentDOM().screenOutline.insertScreen("slide2","slide1","Form");
```

L'esempio seguente inserisce la diapositiva `slide4` dopo la schermata `slide3`:

```
fl.getDocumentDOM().screenOutline.insertScreen("slide4","slide3","Slide");
```

screenOutline.moveScreen()

Disponibilità
Flash MX 2004.

Uso
```
screenOutline.moveScreen(screenToMove, referenceScreen, position)
```

Parametri
screenToMove Una stringa che rappresenta il nome della schermata da spostare.

referenceScreen Una stringa che specifica la schermata vicino alla quale viene collocato il parametro *screenToMove*.

position Una stringa che specifica dove spostare la schermata in relazione a *referenceScreen*. I valori accettabili sono `"before"`, `"after"`,`"firstChild"` e `"lastChild"`.

Restituisce
Un valore booleano: `true` se lo spostamento avviene correttamente, `false` in caso contrario.

Descrizione
Metodo; sposta la schermata specificata in relazione al valore del parametro *referenceScreen*: prima, dopo o come primo o ultimo elemento secondario.

Esempio
L'esempio seguente sposta la schermata `slide1` in modo che sia il primo elemento secondario di `slide2`:

```
fl.getDocumentDOM().screenOutline.moveScreen("slide1", "slide2", "firstChild");
```

screenOutline.renameScreen()

Disponibilità
Flash MX 2004.

Uso

```
screenOutline.renameScreen(newScreenName [, oldScreenName[, bDisplayError]])
```

Parametri

newScreenName Una stringa che specifica il nuovo nome della schermata.

oldScreenName Una stringa che specifica il nome della schermata esistente da modificare. Se non viene specificata, viene modificato il nome della schermata selezionata. Questo parametro è opzionale.

bDisplayError Un valore booleano che, se impostato su `true`, visualizza un messaggio di errore (ad esempio, se esiste già una schermata che ha lo stesso nome del valore passato a *newScreenName*). Il valore predefinito è `false`.

Restituisce

Un valore booleano: `true` se la ridenominazione avviene correttamente, `false` in caso contrario.

Descrizione

Metodo; sostituisce il nome specificato di una schermata con un nuovo nome.

Esempio

L'esempio seguente sostituisce il nome di `slide1` con `Intro`:

```
fl.getDocumentDOM().screenOutline.renameScreen("Intro", "slide1");
```

screenOutline.rootScreen

Disponibilità

Flash MX 2004.

Uso

```
screenOutline.rootScreen
```

Descrizione

Proprietà di sola lettura; la prima schermata nella struttura delle schermate. È possibile utilizzare `screenOutline.rootScreen` come scelta rapida per `screenOutline.screens[0]`.

Esempio

L'esempio seguente visualizza il nome del primo elemento secondario della prima schermata nella struttura delle schermate:

```
var n = fl.getDocumentDOM().screenOutline.rootScreen.childScreens[0].name;
fl.trace(n);
```

screenOutline.screens

Disponibilità

Flash MX 2004.

Uso

```
screenOutline.screens
```

Descrizione

Proprietà di sola lettura; l'array di oggetti Screen di primo livello contenuti nel documento (consultate Oggetto Screen).

Esempio

L'esempio seguente memorizza nella variabile `myArray` l'array di oggetti Screen e ne visualizza i nomi nel pannello Output:

```
var myArray = new Array();
if(fl.getDocumentDOM().allowScreens) {
for(var i in fl.getDocumentDOM().screenOutline.screens) {
    myArray.push(" "+fl.getDocumentDOM().screenOutline.screens[i].name);
}
fl.trace(2"The screens array contains objects whose names are: "+myArray+". ");
}
```

screenOutline.setCurrentScreen()

Disponibilità

Flash MX 2004.

Uso

```
screenOutline.setCurrentScreen(name)
```

Parametri

name Una stringa che specifica il nome della schermata che deve diventare la schermata selezionata. Se la schermata è un elemento secondario di un'altra schermata, non è necessario indicare un percorso o una gerarchia.

Restituisce

Nulla.

Descrizione

Metodo; imposta la schermata specificata come selezione corrente nella struttura delle schermate.

Esempio

L'esempio seguente imposta come schermata corrente la schermata denominata `ChildOfSlide_1`:

```
fl.getDocumentDOM().screenOutline.setCurrentScreen("ChildOfSlide_1");
```

screenOutline.setScreenProperty()

Disponibilità
Flash MX 2004.

Uso
```
screenOutline.setScreenProperty(property, value)
```

Parametri
proprietà Una stringa che specifica la proprietà da impostare.

value Il nuovo valore della proprietà. Il tipo di valore dipende dalla proprietà che state impostando.

Le proprietà disponibili sono `screenOutline.currentScreen`, `screenOutline.rootScreen` e `screenOutline.screens`.

Restituisce
Nulla.

Descrizione
Metodo; imposta la proprietà specificata sul valore specificato delle schermate selezionate.

Esempio
L'esempio seguente rende visibili le schermate selezionate nascoste.

```
fl.getDocumentDOM().screenOutline.setScreenProperty("hidden", false);
```

screenOutline.setSelectedScreens()

Disponibilità
Flash MX 2004.

Uso
```
screenOutline.setSelectedScreens(selection [, bReplaceCurrentSelection])
```

Parametri
selection Un array dei nomi delle schermate da selezionare nella struttura delle schermate.

bReplaceCurrentSelection Un valore booleano che, se impostato su `true`, consente di deselezionare la selezione corrente. Il valore predefinito è `true`. Se è `false`, la selezione corrente viene estesa per includere le schermate specificate. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione

Metodo; seleziona le schermate specificate nella struttura delle schermate. Se vengono specificate più schermate, la schermata dell'array di selezione con l'ultimo valore di indice diventa attiva sullo stage.

Esempio

L'esempio seguente deseleziona tutte le schermate selezionate, quindi deseleziona le schermate `slide1`, `slide2`, `slide3` e `slide4` nella struttura delle schermate:

```
myArray = new Array("slide1", "slide2", "slide3", "slide4");
fl.getDocumentDOM().screenOutline.setSelectedScreens(myArray, true);
```

Capitolo 38: Oggetto Shape

Ereditarietà Oggetto Element > Oggetto Shape

Disponibilità

Flash MX 2004.

Descrizione

L'oggetto Shape è una sottoclasse dell'oggetto Element. L'oggetto Shape fornisce un controllo più preciso rispetto alle API di disegno quando dovete manipolare o creare forme geometriche sullo stage. Questo controllo è necessario affinché gli script siano in grado di creare alcuni utili effetti e altri comandi di disegno (consultate Oggetto Element).

Per funzionare correttamente, tutti i metodi e le proprietà Shape che modificano una forma o una qualunque delle sue parti subordinate devono essere collocati tra le chiamate a `shape.beginEdit()` e a `shape.endEdit()`.

Riepilogo dei metodi

Oltre a quelli dell'oggetto Element, con l'oggetto Shape potete utilizzare i seguenti metodi.

Metodo	Descrizione
shape.getCubicSegmentPoints()	Restituisce un array di punti che definisce una curva cubica.
`shape.getCubicSegmentPoints()`	Definisce l'inizio di una sessione di modifica.
`shape.deleteEdge()`	Elimina il bordo specificato.
`shape.endEdit()`	Definisce la fine di una sessione di modifica per la forma.

Riepilogo delle proprietà

Oltre a quelle dell'oggetto Element, per l'oggetto Shape sono disponibili le proprietà seguenti:

Proprietà	Descrizione
`shape.contours`	Sola lettura; un array degli oggetti Contour della forma (consultate Oggetto Contour).
`shape.edges`	Sola lettura; un array di oggetti Edge (consultate Oggetto Edge).
`shape.isDrawingObject`	Sola lettura; se è true, la forma è un oggetto Drawing.
`shape.isGroup`	Sola lettura; se è true, la forma è un gruppo.
`shape.isOvalObject`	Sola lettura; se true, la forma è un oggetto Oval di base (creato mediante lo strumento Ovale).
`shape.isRectangleObject`	Sola lettura; se true, la forma è un oggetto Rectangle di base (creato mediante lo strumento Rettangolo).
`shape.members`	Un array di oggetti nel gruppo selezionato.
`shape.numCubicSegments`	Sola lettura; il numero di segmenti cubici nella forma.
`shape.vertices`	Sola lettura; un array di oggetti Vertex (consultate Oggetto Vertex).

shape.beginEdit()

Disponibilità
Flash MX 2004.

Uso
```
shape.beginEdit()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; definisce l'inizio di una sessione di modifica. È necessario utilizzare questo metodo prima di eseguire qualunque comando che modifichi l'oggetto Shape o una o più delle sue parti subordinate.

Esempio
L'esempio seguente rimuove dalla forma selezionata il primo bordo dell'array di bordi:

```
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit();
shape.deleteEdge(0);
shape.endEdit();
```

shape.contours

Disponibilità
Flash MX 2004.

Uso
```
shape.contours
```

Descrizione
Proprietà di sola lettura; un array degli oggetti Contour della forma (consultate Oggetto Contour).

Esempio
L'esempio seguente memorizza il primo contorno dell'array di contorni nella variabile *c*, quindi memorizza l'Oggetto HalfEdge del contorno nella variabile `he`:

```
var c = fl.getDocumentDOM().selection[0].contours[0];
var he = c.getHalfEdge();
```

shape.deleteEdge()

Disponibilità
Flash MX 2004.

Uso
```
shape.deleteEdge(index)
```

Parametri
index Un indice a base zero che specifica il bordo da eliminare dall'array shape.edges. Questo metodo modifica la lunghezza dell'array `shape.edges`.

Restituisce
Nulla.

Descrizione
Metodo; elimina il bordo specificato. Dovete chiamare `shape.beginEdit()` prima di utilizzare questo metodo.

Esempio
L'esempio seguente rimuove dalla forma selezionata il primo bordo dell'array di bordi:

```
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit();
shape.deleteEdge(0);
shape.endEdit();
```

shape.edges

Disponibilità
Flash MX 2004.

Uso
```
shape.edges
```

Descrizione
Proprietà di sola lettura; un array di oggetti Edge (consultate Oggetto Edge).

shape.endEdit()

Disponibilità
Flash MX 2004.

Uso
```
shape.endEdit()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; definisce la fine di una sessione di modifica per la forma. Tutte le modifiche apportate all'oggetto Shape o a qualunque sua parte subordinata vengono applicate alla forma. Dovete utilizzare questo metodo dopo l'esecuzione di qualunque comando che modifichi l'oggetto Shape o qualche sua parte subordinata.

Esempio
L'esempio seguente rimuove dalla forma selezionata il primo bordo dell'array di bordi:

```
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit();
shape.deleteEdge(0);
shape.endEdit();
```

shape.getCubicSegmentPoints()

Disponibilità
Flash CS4 Professional.

Uso
```
shape.getCubicSegmentPoints(cubicSegmentIndex)
```

Parametri
`cubicSegmentIndex` Un numero intero che specifica il segmento cubico per i punti restituiti.

Restituisce
Un array di punti che definisce una curva cubica per l'oggetto Edge che corrisponde al valore *cubicSegmentIndex* specificato (consultate `edge.cubicSegmentIndex`).

Descrizione
Metodo; restituisce un array di punti che definisce una curva cubica.

Esempio
L'esempio seguente visualizza i valori *x* e *y* per ogni punto della curva cubica del primo bordo della selezione:

```
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

Disponibilità
Flash 8.

Uso
`shape.isDrawingObject`

Descrizione
Di sola lettura; se è `true`, la forma è un oggetto Drawing.

Esempio
L'esempio seguente memorizza nella variabile `sel` il primo oggetto selezionato, quindi utilizza le proprietà `element.elementType` e `shape.isDrawingObject` per verificare se l'elemento selezionato è un oggetto Drawing:

```
var sel = fl.getDocumentDOM().selection[0];
var shapeDrawingObject = (sel.elementType == "shape") && sel.isDrawingObject;
fl.trace(shapeDrawingObject);
```

Consultate anche
`document.crop()`, `document.deleteEnvelope()`, `document.intersect()`, `document.punch()`, `document.union()`, `shape.isGroup`

shape.isGroup

Disponibilità
Flash MX 2004.

Uso
`shape.isGroup`

Descrizione
Proprietà di sola lettura; se è `true`, la forma è un gruppo. Un gruppo può contenere diversi tipi di elementi, tra cui elementi di testo e simboli. Il gruppo tuttavia è considerato una forma e potete utilizzare la proprietà `shape.isGroup` indipendentemente dal tipo di elementi contenuti nel gruppo.

Esempio
L'esempio seguente memorizza nella variabile `sel` il primo oggetto selezionato, quindi utilizza le proprietà `element.elementType` e `shape.isGroup` per determinare se l'elemento selezionato è un gruppo:

```
var sel = fl.getDocumentDOM().selection[0];
var shapeGroup = (sel.elementType == "shape") && sel.isGroup;
fl.trace(shapeGroup);
```

Consultate anche
`shape.isDrawingObject`

shape.isOvalObject

Disponibilità
Flash CS3 Professional.

Uso
`shape.isOvalObject`

Descrizione
Proprietà di sola lettura; se `true`, la forma è un oggetto Oval di base (creato mediante lo strumento Ovale di base).

Esempio
Nell'esempio seguente viene visualizzato `"true"` se il primo elemento selezionato è un oggetto Oval di base, oppure `"false"` in caso contrario:

```
var sel = fl.getDocumentDOM().selection[0];
fl.trace(sel.isOvalObject);
```

Consultate anche
`shape.isRectangleObject`

shape.isRectangleObject

Disponibilità
Flash CS3 Professional.

Uso
`shape.isRectangleObject`

Descrizione
Proprietà di sola lettura; se `true`, la forma è un oggetto Rectangle di base (creato mediante lo strumento Rettangolo di base).

Esempio
Nell'esempio seguente viene visualizzato "`true`" se il primo elemento selezionato è un oggetto Rectangle di base, oppure "`false`" in caso contrario:

```
var sel = fl.getDocumentDOM().selection[0];
fl.trace(sel.isRectangleObject);
```

Consultate anche
`shape.isOvalObject`

shape.members

Disponibilità
Flash CS4 Professional.

Uso
`shape.members`

Descrizione
Proprietà di sola lettura; un array di oggetti nel gruppo selezionato. Questa proprietà è disponibile solo se il valore della proprietà `shape.isGroup` è `true`. Le forme raw del gruppo non sono incluse nell'array `shape.members`.

Se ad esempio il gruppo contiene tre oggetti Drawing e tre forme raw, l'array `shape.members` contiene tre voci, una per ogni oggetto Drawing. Se il gruppo contiene solo forme raw, l'array è vuoto.

Esempio
Il codice seguente visualizza il numero di segmenti cubici di ogni oggetto Drawing nel gruppo selezionato:

```
var shapesArray = fl.getDocumentDOM().selection[0].members;
for (i=0; i<shapesArray.length; i++) {
    fl.trace(shapesArray[i].numCubicSegments);
}
```

Consultate anche
`shape.isGroup`

shape.numCubicSegments

Disponibilità
Flash CS4 Professional.

Uso
`shape.numCubicSegments`

Descrizione
Proprietà di sola lettura; il numero di segmenti cubici nella forma.

Esempio
Se è selezionato un quadrato o un rettangolo, il codice seguente visualizza "4" nel pannello Output:

```
var theShape = fl.getDocumentDOM().selection[0];
fl.trace(theShape.numCubicSegments);
```

shape.vertices

Disponibilità
Flash MX 2004.

Uso
`shape.vertices`

Descrizione
Proprietà di sola lettura; un array di oggetti Vertex (consultate Oggetto Vertex).

Esempio
L'esempio seguente memorizza nella variabile `someShape` il primo oggetto selezionato, quindi mostra il numero dei vertici dell'oggetto nel pannello Output:

```
var someShape = fl.getDocumentDOM().selection[0];
fl.trace("The shape has " + someShape.vertices.length + " vertices.");
```

Capitolo 39: Oggetto SoundItem

Ereditarietà Oggetto Item > Oggetto SoundItem

Disponibilità

Flash MX 2004.

Descrizione

L'oggetto SoundItem è una sottoclasse dell'oggetto Item. Rappresenta un elemento della libreria utilizzato per creare un suono. Consultate anche `frame.soundLibraryItem` e Oggetto Item.

Riepilogo dei metodi

Oltre a quelli dell'oggetto Item, l'oggetto SoundItem è dotato del seguente metodo:

Proprietà	Descrizione
`soundItem.exportToFile()`	Esporta l'elemento specificato in un file QuickTime in Macintosh o in un file WAV o QT in Windows.

Riepilogo delle proprietà

Oltre a quelle dell'oggetto Item, l'oggetto SoundItem dispone delle seguenti proprietà:

Proprietà	Descrizione
`soundItem.bitRate`	Una stringa che specifica la velocità di trasferimento di un suono presente nella libreria. Disponibile solo per la compressione MP3.
`soundItem.bits`	Una stringa che specifica il valore dei bit di un suono con compressione ADPCM presente nella libreria.
`soundItem.compressionType`	Una stringa che specifica il tipo di compressione di un suono presente nella libreria.
`soundItem.convertStereoToMono`	Un valore booleano disponibile solo per i tipi di compressione MP3 e Raw.
`soundItem.fileLastModifiedDate`	Sola lettura; una stringa contenente un numero esadecimale che rappresenta il numero di secondi trascorsi tra il 1° gennaio 1970 e la data della modifica del file originale (su disco) nel momento in cui il file è stato importato nella libreria.
`soundItem.originalCompressionType`	Sola lettura: una stringa che specifica se l'elemento specificato è stato importato come file MP3.
`soundItem.quality`	Una stringa che specifica la qualità di riproduzione di un suono presente nella libreria. Disponibile solo per la compressione MP3.
`soundItem.sampleRate`	Una stringa che specifica la frequenza di campionamento per il clip audio.
`soundItem.sourceFileExists`	Sola lettura; un valore booleano che specifica se il file importato nella libreria esiste ancora nel percorso da cui è stato importato.

Proprietà	Descrizione
soundItem.sourceFileIsCurrent	Sola lettura; un valore booleano che specifica se la data di modifica del file dell'elemento nella libreria è uguale alla data di modifica sul disco del file importato.
soundItem.sourceFilePath	Sola lettura; una stringa, espressa come URI file:///, che rappresenta il percorso e il nome del file importato nella libreria.
soundItem.useImportedMP3Quality	Un valore booleano; se è `true`, tutte le altre proprietà vengono ignorate e viene utilizzata la qualità MP3 importata.

soundItem.bitRate

Disponibilità
Flash MX 2004.

Uso
```
soundItem.bitRate
```

Descrizione
Proprietà; una stringa che specifica la velocità di trasferimento di un suono presente nella libreria. Questa proprietà è disponibile solo per la compressione MP3. I valori accettabili sono `"8kbps"`, `"16kbps"`, `"20kbps"`, `"24kbps"`, `"32kbps"`, `"48kbps"`, `"56kbps"`, `"64kbps"`, `"80kbps"`, `"112kbps"`, `"128kbps"` e `"160kbps"`. L'audio stereo esportato a 8 Kbps o 16 Kbps viene convertito in mono. La proprietà è `undefined` per gli altri tipi di compressione.

Se desiderate specificare un valore per la proprietà, impostate `soundItem.useImportedMP3Quality` su `false`.

Esempio
L'esempio seguente visualizza il valore `bitRate` nel pannello Output se l'elemento specificato nella libreria ha un tipo di compressione MP3:

```
alert(fl.getDocumentDOM().library.items[0].bitRate);
```

Consultate anche
`soundItem.compressionType`, `soundItem.convertStereoToMono`

soundItem.bits

Disponibilità
Flash MX 2004.

Uso
```
soundItem.bits
```

Descrizione
Proprietà; una stringa che specifica il valore dei bit di un suono con compressione ADPCM presente nella libreria. I valori accettabili sono `"2 bit"`, `"3 bit"`, `"4 bit"` e `"5 bit"`.

Se desiderate specificare un valore per la proprietà, impostate `soundItem.useImportedMP3Quality` su `false`.

Esempio
L'esempio seguente visualizza il valore dei bit nel pannello Output se l'elemento selezionato nella libreria ha un tipo di compressione ADPCM:

```
alert(fl.getDocumentDOM().library.items[0].bits);
```

Consultate anche
`soundItem.compressionType`

soundItem.compressionType

Disponibilità
Flash MX 2004.

Uso
```
soundItem.compressionType
```

Descrizione
Proprietà; una stringa che specifica il tipo di compressione di un suono presente nella libreria. I valori accettabili sono `"Default"`, `"ADPCM"`, `"MP3"`, `"Raw"` e `"Speech"`.

Se desiderate specificare un valore per la proprietà, impostate `soundItem.useImportedMP3Quality` su `false`.

Esempio
L'esempio seguente imposta su `Raw` il tipo di compressione di un elemento nella libreria:

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

L'esempio seguente modifica in `Speech` il tipo di compressione degli elementi di libreria selezionati:

```
fl.getDocumentDOM().library.getSelectedItems().compressionType = "Speech";
```

Consultate anche
`soundItem.originalCompressionType`

soundItem.convertStereoToMono

Disponibilità
Flash MX 2004.

Uso
```
soundItem.convertStereoToMono
```

Descrizione
Proprietà; un valore booleano disponibile solo per i tipi di compressione MP3 e Nessuna. Se impostate questo valore su `true`, un suono stereo viene convertito in mono, mentre `false` lo mantiene stereo. Per il tipo di compressione MP3, se `soundItem.bitRate` è inferiore a 20 Kbps, questa proprietà viene ignorata e impostata forzatamente su `true` (consultate `soundItem.bitRate`).

Se desiderate specificare un valore per la proprietà, impostate `soundItem.useImportedMP3Quality` su `false`.

Esempio
L'esempio seguente converte in mono un elemento della libreria, solo se l'elemento ha una compressione di tipo MP3 o Raw:

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

Consultate anche
`soundItem.compressionType`

soundItem.exportToFile()

Disponibilità
Flash CS4 Professional.

Uso
`soundItem.exportToFile(fileURI)`

Parametri
`fileURI` Una stringa, espressa come URI file:///, che specifica il percorso e il nome del file esportato.

Restituisce
Il valore booleano `true` se il file è stato esportato correttamente; `false` in caso contrario.

Descrizione
Metodo; esporta l'elemento specificato in un file QuickTime in Macintosh o in un file WAV o QT in Windows. I file QuickTime o QT esportati contengono solo audio; il video non viene esportato. Le impostazioni di esportazione sono basate sull'elemento da esportare.

Esempio
Se il primo elemento nella libreria è un suono, il codice seguente lo esporta come file WAV:

```
var soundFileURL = "file:///C|/out.wav";
var libItem = fl.getDocumentDOM().library.items[0];
libItem.exportToFile(soundFileURL);
```

soundItem.fileLastModifiedDate

Disponibilità
Flash CS4 Professional.

Uso
```
soundItem.fileLastModifiedDate
```

Descrizione
Proprietà di sola lettura: una stringa contenente un numero esadecimale che rappresenta il numero di secondi trascorsi tra il 1 gennaio 1970 e la data della modifica del file originale (su disco) nel momento in cui il file è stato importato nella libreria. Se il file non esiste più, il valore corrisponde a "00000000".

Esempio
Se il primo elemento nella libreria è un suono, il codice seguente visualizza un numero esadecimale come descritto in precedenza.

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

Consultate anche
soundItem.sourceFileExists, soundItem.sourceFileIsCurrent, soundItem.sourceFilePath, FLfile.getModificationDate()

soundItem.originalCompressionType

Disponibilità
Flash CS4 Professional.

Uso
```
soundItem.originalCompressionType
```

Descrizione
Proprietà di sola lettura: una stringa che specifica se l'elemento specificato è stato importato come file MP3. I valori possibili per la proprietà sono "RAW" e "MP3".

Esempio
Se il primo elemento nella libreria è un suono, il codice seguente visualizza "MP3" se il file è stato importato nella libreria come MP3. oppure "RAW" in caso contrario:

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

Consultate anche
soundItem.compressionType

soundItem.quality

Disponibilità
Flash MX 2004.

Uso
`soundItem.quality`

Descrizione
Proprietà; una stringa che specifica la qualità di riproduzione di un suono presente nella libreria. Questa proprietà è disponibile solo per il tipo di compressione MP3. I valori accettabili sono `"Fast"`, `"Medium"` e `"Best"`.

Se desiderate specificare un valore per la proprietà, impostate `soundItem.useImportedMP3Quality` su `false`.

Esempio
L'esempio seguente imposta su `Best` la qualità di riproduzione di un elemento della libreria, se l'elemento ha una compressione di tipo MP3:

```
fl.getDocumentDOM().library.items[0].quality = "Best";
```

Consultate anche
`soundItem.compressionType`

soundItem.sampleRate

Disponibilità
Flash MX 2004.

Uso
`soundItem.sampleRate`

Descrizione
Proprietà; una stringa che specifica la frequenza di campionamento per il clip audio. Questa proprietà è disponibile solo per i tipi di compressione ADPCM, Raw e Speech. I valori accettabili sono `"5 kHz"`, `"11 kHz"`, `"22 kHz"` e `"44 kHz"`.

Se desiderate specificare un valore per la proprietà, impostate `soundItem.useImportedMP3Quality` su `false`.

Esempio
L'esempio seguente imposta su 5 kHz la frequenza di campionamento di un elemento della libreria, se l'elemento ha una compressione di tipo ADPCM, Raw o Speech:

```
fl.getDocumentDOM().library.items[0].sampleRate = "5 kHz";
```

Consultate anche
`soundItem.compressionType`

soundItem.sourceFileExists

Disponibilità
Flash CS4 Professional.

Uso
`soundItem.sourceFileExists`

Descrizione
Proprietà di sola lettura: il value booleano `true` se il file importato nella libreria esiste ancora nel percorso da cui è stato importato; `false` in caso contrario.

Esempio
Se il primo elemento nella libreria è un suono, il codice seguente visualizza "true" se il file importato nella libreria esiste ancora.

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

Consultate anche
`soundItem.sourceFileIsCurrent`, `soundItem.sourceFilePath`

soundItem.sourceFileIsCurrent

Disponibilità
Flash CS4 Professional.

Uso
`soundItem.sourceFileIsCurrent`

Descrizione
Proprietà di sola lettura: il value booleano `true` se la data di modifica del file dell'elemento della libreria corrisponde alla data di modifica su disco del file importato; `false` in caso contrario.

Esempio
Se il primo elemento nella libreria è un suono, il codice seguente visualizza "true" se il file importato non è stato modificato da quando è stato importato.

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

Consultate anche
`soundItem.fileLastModifiedDate`, `soundItem.sourceFilePath`

soundItem.sourceFilePath

Disponibilità
Flash CS4 Professional.

Uso
`soundItem.sourceFilePath`

Descrizione
Proprietà di sola lettura; una stringa, espressa come URI file:///, che rappresenta il percorso e il nome del file importato nella libreria.

Esempio
L'esempio seguente visualizza il nome e il percorso dei file di origine degli elementi della libreria di tipo `"sound"`:

```
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);
    }
}
```

Consultate anche
`soundItem.sourceFileExists`

soundItem.useImportedMP3Quality

Disponibilità
Flash MX 2004.

Uso
`soundItem.useImportedMP3Quality`

Descrizione
Proprietà; un valore booleano. Se è `true`, tutte le altre proprietà vengono ignorate e viene utilizzata la qualità MP3 importata.

Esempio
L'esempio seguente imposta un elemento della libreria in modo che utilizzi la qualità MP3 importata:

```
fl.getDocumentDOM().library.items[0].useImportedMP3Quality = true;
```

Consultate anche
`soundItem.compressionType`

Capitolo 40: Oggetto Stroke

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto Stroke contiene tutte le impostazioni relative a un tratto, comprese quelle personalizzate. Rappresenta le informazioni contenute nella finestra di ispezione Proprietà. Se utilizzate l'oggetto Stroke insieme al metodo `document.setCustomStroke()`, potete modificare le impostazioni del tratto per il pannello Strumenti, la finestra di ispezione Proprietà e la selezione corrente. Potete inoltre ottenere tutte le stesse impostazioni anche utilizzando il metodo `document.getCustomStroke()`.

Questo oggetto dispone sempre delle seguenti quattro proprietà: `style`, `thickness`, `color` e `breakAtCorners`. In Flash CS3 l'uso della proprietà `breakAtCorners` è sconsigliato. Al suo posto utilizzate `stroke.joinType`. Potete impostare altre proprietà, a seconda del valore della proprietà `stroke.style`.

Riepilogo delle proprietà
Le seguenti proprietà sono disponibili per l'oggetto Stroke:

Proprietà	Descrizione
`stroke.breakAtCorners`	Un valore booleano; equivale all'impostazione Angoli acuti della finestra di dialogo Stile tratto personalizzato.
`stroke.capType`	Una stringa che specifica il tipo di estremità di un tratto.
`stroke.color`	Una stringa, un valore esadecimale o un numero intero che rappresenta il colore del tratto.
`stroke.curve`	Una stringa che specifica il tipo di puntinatura del tratto.
`stroke.dash1`	Un numero intero che specifica la lunghezza della parte piena di una linea tratteggiata.
`stroke.dash2`	Un numero intero che specifica la lunghezza della parte vuota di una linea tratteggiata.
`stroke.density`	Una stringa che specifica la densità di una linea schizzata.
`stroke.dotSize`	Una stringa che specifica le dimensioni del punto di una linea schizzata.
`stroke.dotSpace`	Un numero intero che specifica lo spazio tra i punti di una linea punteggiata.
`stroke.hatchThickness`	Una stringa che specifica lo spessore di una linea puntinata.
`stroke.jiggle`	Una stringa che specifica l'increspatura di una linea puntinata.
`stroke.joinType`	Una stringa che specifica il tipo di giunzione di un tratto.
`stroke.length`	Una stringa che specifica la lunghezza di una linea puntinata.
`stroke.miterLimit`	Un valore float che specifica l'angolazione oltre la quale la punta della giunzione ad angolo viene troncata di un segmento.
`stroke.pattern`	Una stringa che specifica il motivo di una linea dentellata.
`stroke.rotate`	Una stringa che specifica la rotazione di una linea puntinata.
`stroke.scaleType`	Una stringa che specifica il tipo di modifica in scala da applicare al tratto.
`stroke.shapeFill`	Un Oggetto Fill che rappresenta l'impostazione di riempimento del tratto.

Proprietà	Descrizione
stroke.space	Una stringa che specifica la spaziatura di una linea puntinata.
stroke.strokeHinting	Un valore booleano che specifica se per il tratto sono impostati i suggerimenti per il tratto.
stroke.style	Una stringa che descrive lo stile del tratto.
stroke.thickness	Un numero intero che specifica la dimensione del tratto.
stroke.variation	Una stringa che specifica la variazione di una linea schizzata.
stroke.waveHeight	Una stringa che specifica l'altezza dell'onda di una linea dentellata.
stroke.waveLength	Una stringa che specifica la lunghezza dell'onda di una linea dentellata.

stroke.breakAtCorners

Disponibilità

Flash MX 2004. Utilizzo sconsigliato in Flash CS3. Al suo posto utilizzate `stroke.joinType`.

Uso

```
stroke.breakAtCorners
```

Descrizione

Proprietà; un valore booleano. Equivale all'impostazione Angoli acuti della finestra di dialogo Stile tratto personalizzato.

Esempio

L'esempio seguente imposta la proprietà `breakAtCorners` su `true`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.breakAtCorners = true;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.capType

Disponibilità

Flash 8.

Uso

```
stroke.capType
```

Descrizione

Proprietà; una stringa che specifica il tipo di estremità di un tratto. I valori accettabili sono `"none"`, `"round"` e `"square"`.

Esempio

L'esempio seguente imposta l'estremità del tratto su `round`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.capType = "round";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.color

Disponibilità

Flash MX 2004. In Flash 8 e versioni successive, l'utilizzo di questa proprietà è sconsigliato. Al suo posto utilizzate `stroke.shapeFill.color`.

Uso

```
stroke.color
```

Descrizione

Proprietà; il colore del tratto in uno dei seguenti formati:

- Una stringa nel formato `"#RRGGBB"` o `"#RRGGBBAA"`
- Un numero esadecimale nel formato `0xRRGGBB`
- Un numero intero che rappresenta l'equivalente decimale di un numero esadecimale.

Esempio

L'esempio seguente imposta il colore del tratto:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.color = "#000000";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

Consultate anche

`stroke.shapeFill`

stroke.curve

Disponibilità

Flash MX 2004.

Uso

```
stroke.curve
```

Descrizione

Proprietà; una stringa che specifica il tipo di puntinatura del tratto. Questa proprietà può essere impostata solo se la proprietà `stroke.style` è impostata su `"hatched"` (consultate `stroke.style`). I valori accettabili sono `"straight"`, `"slight curve"`, `"medium curve"` e `"very curved"`.

Esempio

L'esempio seguente imposta, tra le altre, la proprietà curve per un tratto con stile `hatched`:

```
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

Disponibilità
Flash MX 2004.

Uso
`stroke.dash1`

Descrizione
Proprietà; un numero intero che specifica la lunghezza della parte piena di una linea tratteggiata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `dashed` (consultate `stroke.style`).

Esempio
L'esempio seguente imposta le proprietà `dash1` e `dash2` per lo stile di tratto `dashed`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "dashed";
myStroke.dash1 = 1;
myStroke.dash2 = 2;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.dash2

Disponibilità
Flash MX 2004.

Uso
`stroke.dash2`

Descrizione
Proprietà; un numero intero che specifica la lunghezza della parte vuota di una linea tratteggiata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `dashed` (consultate `stroke.style`).

Esempio
Consultate `stroke.dash1`.

stroke.density

Disponibilità
Flash MX 2004.

Uso
`stroke.density`

Descrizione
Proprietà; una stringa che specifica la densità di una linea schizzata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `stipple` (consultate `stroke.style`). I valori accettabili sono `"very dense"`, `"dense"`, `"sparse"` e `"very sparse"`.

Esempio
L'esempio seguente imposta la densità su `sparse` per lo stile del tratto `stipple`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "stipple";
myStroke.dotSpace= 3;
myStroke.variation = "random sizes";
myStroke.density = "sparse";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.dotSize

Disponibilità
Flash MX 2004.

Uso
`stroke.dotSize`

Descrizione
Proprietà; una stringa che specifica le dimensioni del punto di una linea schizzata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `stipple` (consultate `stroke.style`). I valori accettabili sono `"tiny"`, `"small"`, `"medium"` e `"large"`.

L'esempio seguente imposta la proprietà `dotSize` su `tiny` per lo stile del tratto `stipple`:

```
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

Disponibilità
Flash MX 2004.

Uso
`stroke.dotSpace`

Descrizione
Proprietà; un numero intero che specifica lo spazio tra i punti di una linea punteggiata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `dotted`. Consultate `stroke.style`.

Esempio
L'esempio seguente imposta la proprietà `dotSpace` su 3 per lo stile di tratto `dotted`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "dotted";
myStroke.dotSpace= 3;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.hatchThickness

Disponibilità
Flash MX 2004.

Uso
`stroke.hatchThickness`

Descrizione
Proprietà; una stringa che specifica lo spessore di una linea puntinata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `hatched` (consultate `stroke.style`). I valori accettabili sono `"hairline"`, `"thin"`, `"medium"` e `"thick"`.

Esempio
L'esempio seguente imposta la proprietà `hatchThickness` su `thin` per lo stile del tratto `hatched`:

```
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

Disponibilità
Flash MX 2004.

Uso
`stroke.jiggle`

Descrizione
Proprietà; una stringa che specifica l'increspatura di una linea puntinata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `hatched` (consultate `stroke.style`). I valori accettabili sono `"none"`, `"bounce"`, `"loose"` e `"wild"`.

Esempio
L'esempio seguente imposta la proprietà `jiggle` su `wild` per lo stile del tratto `hatched`:

```
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

Disponibilità
Flash 8.

Uso
`stroke.joinType`

Descrizione
Proprietà; una stringa che specifica il tipo di giunzione del tratto. I valori accettabili sono `"miter"`, `"round"` e `"bevel"`.

Consultate anche
`stroke.capType`

stroke.length

Disponibilità
Flash MX 2004.

Uso

```
stroke.length
```

Descrizione

Proprietà; una stringa che specifica la lunghezza di una linea puntinata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `hatched` (consultate `stroke.style`). I valori accettabili sono `"equal"`, `"slight"`, `"variation"`, `"medium variation"` e `"random"`.

Esempio

L'esempio seguente imposta la proprietà `length` su `slight` per lo stile del tratto `hatched`:

```
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.miterLimit

Disponibilità

Flash 8.

Uso

```
stroke.miterLimit
```

Descrizione

Proprietà; un valore float che specifica l'angolazione oltre la quale la punta della giunzione ad angolo viene troncata di un segmento. In altre parole, la giunzione ad angolo viene troncata solo se ha un'angolazione superiore al valore di `miterLimit`.

Esempio

L'esempio seguente imposta su 3 il limite della giunzione ad angolo del tratto. Se l'angolazione è superiore a 3, la giunzione ad angolo viene troncata.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.miterLimit = 3;
var myStroke = fl.getDocumentDOM().setCustomStroke();
```

stroke.pattern

Disponibilità

Flash MX 2004.

Uso

```
stroke.pattern
```

Descrizione

Proprietà; una stringa che specifica il motivo di una linea dentellata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `ragged` (consultate `stroke.style`). I valori validi sono `"solid"`, `"simple"`, `"random"`, `"dotted"`, `"random dotted"`, `"triple dotted"` e `"random triple dotted"`.

Esempio

L'esempio seguente imposta la proprietà `pattern` su `random` per lo stile del tratto `ragged`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "ragged";
myStroke.pattern = "random";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.rotate

Disponibilità

Flash MX 2004.

Uso

```
stroke.rotate
```

Descrizione

Proprietà; una stringa che specifica la rotazione di una linea puntinata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `hatched` (consultate `stroke.style`). I valori accettabili sono `"none"`, `"slight"`, `"medium"` e `"free"`.

Esempio

L'esempio seguente imposta la proprietà `rotate` su `free` per lo stile del tratto `hatched`:

```
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

Disponibilità

Flash 8.

Uso
`stroke.scaleType`

Descrizione
Proprietà; una stringa che specifica il tipo di modifica in scala da applicare al tratto. I valori accettabili sono `"normal"`, `"horizontal"`, `"vertical"` e `"none"`.

Esempio
L'esempio seguente imposta il tipo di modifica in scala del tratto su `horizontal`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.scaleType = "horizontal";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.shapeFill

Disponibilità
Flash 8.

Uso
`stroke.shapeFill`

Descrizione
Proprietà; un Oggetto Fill che rappresenta l'impostazione di riempimento del tratto.

Esempio
L'esempio seguente specifica le impostazioni del riempimento e le applica al tratto:

```
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

Disponibilità
Flash MX 2004.

Uso
`stroke.space`

Descrizione
Proprietà; una stringa che specifica lo spazio di una linea puntinata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `hatched` (consultate `stroke.style`). I valori accettabili sono `"very close"`, `"close"`, `"distant"` e `"very distant"`.

Esempio
L'esempio seguente imposta la proprietà `space` su `close` per lo stile del tratto `hatched`:

```
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

Disponibilità
Flash 8.

Uso
`stroke.strokeHinting`

Descrizione
Proprietà; un valore booleano che specifica se per il tratto è impostato il suggerimento tratto.

Esempio
L'esempio seguente definisce il suggerimento tratto per il tratto:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.strokeHinting = true;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.style

Disponibilità
Flash MX 2004.

Uso
`stroke.style`

Descrizione
Proprietà; una stringa che descrive lo stile del tratto. I valori accettabili sono `"noStroke"`,`"solid"`, `"dashed"`, `"dotted"`, `"ragged"`, `"stipple"` e `"hatched"`. Alcuni di essi richiedono l'impostazione di alcune proprietà aggiuntive dell'oggetto Stroke, come descritto nell'elenco seguente:

- Se il valore è `"solid"` o `"noStroke"`, non sono richieste altre proprietà.
- Se il valore è `dashed`, sono presenti due ulteriori proprietà: `dash1` e `dash2`.
- Se il valore è `dotted`, è richiesta una proprietà aggiuntiva: `dotSpace`.

- Se il valore è `ragged`, sono presenti tre ulteriori proprietà: `pattern`, `waveHeight` e `waveLength`.
- Se il valore è `"stipple"`, sono presenti tre ulteriori proprietà: `dotSize`, `variation` e `density`.
- Se il valore è `"hatched"`, sono presenti sei ulteriori proprietà: `hatchThickness`, `space`, `jiggle`, `rotate`, `curve` e `length`.

Esempio
L'esempio seguente imposta lo stile del tratto su `ragged`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "ragged";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.thickness

Disponibilità
Flash MX 2004.

Uso
`stroke.thickness`

Descrizione
Proprietà; un numero intero che specifica la dimensione del tratto.

Esempio
L'esempio seguente imposta le proprietà `thickness` del tratto su `2`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.thickness = 2;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.variation

Disponibilità
Flash MX 2004.

Uso
`stroke.variation`

Descrizione
Proprietà; una stringa che specifica la variazione di una linea schizzata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `stipple` (consultate `stroke.style`). I valori accettabili sono `"one size"`, `"small variation"`, `"varied sizes"` e `"random sizes"`.

Esempio
L'esempio seguente imposta la proprietà variation su `random sizes` per lo stile del tratto `stipple`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "stipple";
myStroke.dotSpace= 3;
myStroke.variation = "random sizes";
myStroke.density = "sparse";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.waveHeight

Disponibilità
Flash MX 2004.

Uso
`stroke.waveHeight`

Descrizione
Proprietà; una stringa che specifica l'altezza dell'onda di una linea dentellata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `ragged` (consultate `stroke.style`). I valori accettabili sono `"flat"`, `"wavy"`, `"very wavy"` e `"wild"`.

Esempio
L'esempio seguente imposta la proprietà `waveHeight` su `flat` per lo stile del tratto `ragged`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "ragged";
myStroke.pattern = "random";
myStroke.waveHeight = "flat";
myStroke.waveLength = "short";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.waveLength

Disponibilità
Flash MX 2004.

Uso
`stroke.waveLength`

Descrizione
Proprietà; una stringa che specifica la lunghezza dell'onda di una linea dentellata. Questa proprietà è disponibile solo se la proprietà `stroke.style` è impostata su `ragged` (consultate `stroke.style`). I valori accettabili sono `"very short"`, `"short"`, `"medium"` e `"long"`.

Esempio
L'esempio seguente imposta la proprietà `waveLength` su `short` per lo stile del tratto `ragged`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "ragged";
myStroke.pattern = "random";
myStroke.waveHeight = 'flat";
myStroke.waveLength = "short";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

Capitolo 41: Oggetto swfPanel

Disponibilità
Flash CS4 Professional.

Descrizione
L'oggetto swfPanel rappresenta un pannello Finestra SWF. I pannelli Finestra SWF sono file SWF che implementano le applicazioni che potete eseguire dall'ambiente di creazione Flash; sono disponibili dal menu Finestra > Altri pannelli. Per impostazione predefinita, i pannelli Finestra SWF sono memorizzati in una sottocartella della cartella Configuration (consultate "Salvataggio di file JSFL" a pagina 2). Ad esempio, in Windows XP la cartella si trova in *unità di avvio*\Documents and Settings*utente*\Impostazioni locali\Dati applicazioni\Adobe\FlashCS4*lingua*\Configuration\WindowSWF. È disponibile un pannello Finestra SWF di esempio; consultate "Pannello Ricalco bitmap di esempio" a pagina 14. L'array di pannelli Finestra SWF è memorizzato nella proprietà `fl.swfPanels`.

Riepilogo dei metodi
Con l'oggetto swfPanel potete utilizzare il metodo seguente:

Metodo	Descrizione
`swfPanel.call()`	Può essere utilizzato con i metodi ActionScript `ExternalInterface.addCallback()` e `MMExecute()` per comunicare con il pannello SWF dall'ambiente di creazione.

Riepilogo delle proprietà
Con l'oggetto swfPanel potete utilizzare le proprietà seguenti:

Proprietà	Descrizione
swfPanel.name	Sola lettura; una stringa che rappresenta il nome del pannello Finestra SWF specificato.
swfPanel.path	Sola lettura; una stringa che rappresenta il percorso del file SWF utilizzato nel pannello Finestra SWF specificato.

swfPanel.call()

Disponibilità
Flash CS4 Professional.

Uso
```
swfPanel.call(request)
```

Parametri
`request` Parametri da passare alla funzione (consultate "Descrizione" ed "Esempio" di seguito).

Restituisce
`null` o una stringa restituita dalla chiamata alla funzione. Il risultato della funzione può essere una stringa vuota.

Descrizione

Metodo; può essere utilizzato con i metodi ActionScript `ExternalInterface.addCallback()` e `MMExecute()` per comunicare con il pannello SWF dall'ambiente di creazione.

Esempio

Il seguente esempio illustra come usare il codice ActionScript e JavaScript per creare un pannello Finestra SWF e comunicare con esso dall'ambiente di creazione.

1 Create un file FLA ActionScript 3.0, impostate il relativo colore su grigio medio, quindi impostate le dimensioni su 400 pixel di larghezza e 250 pixel di altezza.

2 Inserite una casella di testo dinamica al centro dello stage, impostate il relativo nome di istanza su `myTextField` e digitate la parola "Status" nella casella di testo.

3 Impostate le altre proprietà della casella di testo in modo analogo alle seguenti:

- Allineato al centro
- 355 pixel di larghezza e 46 pixel di altezza
- Carattere Times New Roman, 28 punti, rosso

4 Aggiungere il seguente codice ActionScript:

```
// 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 Salvare il file con nome fileStatus.fla e pubblicare il file SWF con le impostazioni di pubblicazione predefinite.

6 Chiudere Flash.

7 Copiate il file fileStatus.swf nella cartella WindowSWF, una sottocartella della cartella Configuration (consultate "Salvataggio di file JSFL" a pagina 2). Ad esempio, in Windows XP la cartella si trova in *unità di avvio*\Documents and Settings*utente*\Impostazioni locali\Dati applicazioni\Adobe\ FlashCS4*lingua*\Configuration\WindowSWF.

8 Avviate Flash.

9 Create un file JSFL con il seguente codice:

```
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 Salvate il file JSFL con il nome fileOp.jsfl nella stessa directory del file SWF.

11 Scegliete Finestra > Altri pannelli > fileStatus.

Ora, mentre create, aprite e chiudete file FLA, il pannello fileStatus visualizza un messaggio che indica l'azione eseguita.

swfPanel.name

Disponibilità
Flash CS4 Professional.

Uso

```
swfPanel.name
```

Descrizione
Proprietà di sola lettura; una stringa che rappresenta il nome del pannello Finestra SWF specificato.

Esempio
Il codice seguente visualizza il nome del primo pannello Finestra SWF registrato nel pannello Output:

```
fl.trace(fl.swfPanels[0].name);
```

Consultate anche

swfPanel.path, fl.swfPanels

swfPanel.path

Disponibilità

Flash CS4 Professional.

Uso

```
swfPanel.path
```

Descrizione

Proprietà di sola lettura; una stringa che rappresenta il percorso del file SWF utilizzato nel pannello Finestra SWF specificato.

Esempio

Il codice seguente visualizza il percorso del file SWF utilizzato nel primo pannello Finestra SWF registrato nel pannello Output:

```
fl.trace(fl.swfPanels[0].path);
```

Consultate anche

swfPanel.name, fl.swfPanels

Capitolo 42: Oggetto SymbolInstance

Ereditarietà Oggetto Element > Oggetto Instance > Oggetto SymbolInstance

Disponibilità
Flash MX 2004.

Descrizione
SymbolInstance è una sottoclasse dell'oggetto Instance e rappresenta un simbolo in un fotogramma (consultate Oggetto Instance).

Riepilogo delle proprietà
Oltre a quelle dell'oggetto Instance, l'oggetto SymbolInstance dispone delle seguenti proprietà.

Proprietà	Descrizione
`symbolInstance.accName`	Una stringa equivalente al campo Nome del pannello Accessibilità.
`symbolInstance.actionScript`	Una stringa che specifica le azioni assegnate al simbolo.
`symbolInstance.blendMode`	Una stringa che specifica il metodo di fusione da applicare a un simbolo di clip filmato.
`symbolInstance.buttonTracking`	Una stringa che, solo per i simboli di pulsante, imposta la stessa proprietà del menu a comparsa Traccia come pulsante o Traccia come voce menu nella finestra di ispezione Proprietà.
`symbolInstance.cacheAsBitmap`	Un valore booleano che specifica se il caching bitmap in runtime è attivato.
`symbolInstance.colorAlphaAmount`	Un numero intero che fa parte della trasformazione del colore dell'istanza e che specifica le impostazioni Alfa della finestra di dialogo Effetto avanzato; equivale a utilizzare l'impostazione Colore > Avanzato nella finestra di ispezione Proprietà e a regolare i controlli presenti sul lato destro della finestra di dialogo.
`symbolInstance.colorAlphaPercent`	Un numero intero che specifica parte della trasformazione del colore dell'istanza; equivale a utilizzare l'impostazione Colore > Avanzato nella finestra di ispezione Proprietà dell'istanza (i controlli percentuali presenti sul lato sinistro della finestra di dialogo).
`symbolInstance.colorBlueAmount`	Un numero intero che fa parte della trasformazione del colore per l'istanza; equivale a utilizzare l'impostazione Colore > Avanzato nella finestra di ispezione Proprietà dell'istanza.
`symbolInstance.colorBluePercent`	Un numero intero che rappresenta parte della trasformazione del colore dell'istanza; equivale a utilizzare l'impostazione Colore > Avanzato nella finestra di ispezione Proprietà dell'istanza (i controlli percentuali presenti sul lato sinistro della finestra di dialogo).
`symbolInstance.colorGreenAmount`	Un numero intero che fa parte della trasformazione del colore per l'istanza; equivale a utilizzare l'impostazione Colore > Avanzato nella finestra di ispezione Proprietà dell'istanza. I valori accettabili sono quelli compresi tra -255 e 255.
`symbolInstance.colorGreenPercent`	Parte della trasformazione del colore per l'istanza; equivale a utilizzare l'impostazione Colore > Avanzato nella finestra di ispezione Proprietà dell'istanza (i controlli percentuali sul lato sinistro della finestra di dialogo).

Proprietà	Descrizione
symbolInstance.colorMode	Una stringa che specifica la modalità di colore identificata nel menu a comparsa Colore della finestra di ispezione Proprietà del simbolo.
symbolInstance.colorRedAmount	Un numero intero che fa parte della trasformazione del colore per l'istanza; equivale a utilizzare l'impostazione Colore > Avanzato nella finestra di ispezione Proprietà dell'istanza.
symbolInstance.colorRedPercent	Parte della trasformazione del colore per l'istanza; equivale a utilizzare l'impostazione Colore > Avanzato nella finestra di ispezione Proprietà dell'istanza (i controlli percentuali sul lato sinistro della finestra di dialogo).
symbolInstance.description	Una stringa equivalente al campo Descrizione del pannello Accessibilità.
symbolInstance.filters	Un array di oggetti Filter (consultate Oggetto Filter).
symbolInstance.firstFrame	Un numero intero a base zero che specifica il primo fotogramma che compare nella linea temporale dell'elemento grafico.
symbolInstance.forceSimple	Un valore booleano che abilita o disabilita l'accessibilità degli elementi secondari dell'oggetto; equivalente alla logica inversa dell'impostazione Rendi accessibili gli oggetti secondari del pannello Accessibilità.
symbolInstance.loop	Una stringa che imposta per i simboli grafici la stessa proprietà del menu a comparsa Ciclo della finestra di ispezione Proprietà.
symbolInstance.shortcut	Una stringa che equivale al tasto di scelta rapida associato al simbolo; equivale al campo Tasto di scelta rapida del pannello Accessibilità.
symbolInstance.silent	Un valore booleano che abilita o disabilita l'accessibilità dell'oggetto; equivalente alla logica inversa dell'impostazione Rendi accessibile l'oggetto del pannello Accessibilità.
symbolInstance.symbolType	Una stringa che specifica il tipo di simbolo; equivalente al valore dell'opzione Comportamento nelle finestre di dialogo Crea un nuovo simbolo e Converti in simbolo.
symbolInstance.tabIndex	Un numero intero equivalente al campo Indice tabulazione del pannello Accessibilità.

symbolInstance.accName

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.accName
```

Descrizione
Proprietà; una stringa equivalente al campo Nome del pannello Accessibilità. Gli screen reader identificano gli oggetti pronunciandone il nome. Questa proprietà non è disponibile per i simboli grafici.

Esempio
L'esempio seguente memorizza il valore del nome del pannello Accessibilità dell'oggetto nella variabile `theName`:

```
var theName = fl.getDocumentDOM().selection[0].accName;
```

L'esempio seguente imposta `Home Button` come valore per il nome del pannello Accessibilità dell'oggetto:

```
fl.getDocumentDOM().selection[0].accName = "Home Button";
```

symbolInstance.actionScript

Disponibilità
Flash MX 2004.

Uso
`symbolInstance.actionScript`

Descrizione
Proprietà; una stringa che specifica le azioni assegnate al simbolo. Si applica solo alle istanze dei clip filmato e dei pulsanti. Per le istanze dei simboli grafici, viene restituito il valore `undefined`.

Esempio
L'esempio seguente assegna un'azione `onClipEvent` al primo elemento del primo fotogramma sul primo livello della linea temporale:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].actionScript
    = "onClipEvent(enterFrame) {trace('movie clip enterFrame');}";
```

symbolInstance.blendMode

Disponibilità
Flash 8.

Uso
`symbolInstance.blendMode`

Descrizione
Proprietà; una stringa che specifica il metodo di fusione da applicare a un simbolo di clip filmato. I valori accettabili sono `"normal"`, `"layer"`, `"multiply"`, `"screen"`, `"overlay"`, `"hardlight"`, `"lighten"`, `"darken"`, `"difference"`, `"add"`, `"subtract"`, `"invert"`, `"alpha"` e `"erase"`.

Esempio
L'esempio seguente imposta su `add` il metodo di fusione per il primo simbolo di clip filmato nel primo fotogramma del primo livello:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].blendMode = "add";
```

Consultate anche
`document.setBlendMode()`

symbolInstance.buttonTracking

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.buttonTracking
```

Descrizione
Proprietà; una stringa che, solo per i simboli di pulsante, imposta la stessa proprietà del menu a comparsa Traccia come pulsante o Traccia come voce menu nella finestra di ispezione Proprietà. Per gli altri tipi di simbolo, la proprietà viene ignorata. I valori accettabili sono `"button"` e `"menu"`.

Esempio
L'esempio seguente imposta il primo simbolo nel primo fotogramma sul primo livello nella linea temporale su Traccia come voce menu, a condizione che il simbolo sia un pulsante:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].buttonTracking = "menu";
```

symbolInstance.cacheAsBitmap

Disponibilità
Flash 8.

Uso
```
symbolInstance.cacheAsBitmap
```

Descrizione
Proprietà; un valore booleano che specifica se il caching bitmap in runtime è attivato.

Esempio
L'esempio seguente attiva il caching bitmap in runtime per il primo elemento nel primo fotogramma del primo livello:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].cacheAsBitmap = true;
```

symbolInstance.colorAlphaAmount

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.colorAlphaAmount
```

Descrizione
Proprietà; un numero intero che fa parte della trasformazione di colore per l'istanza e che specifica le impostazioni alfa nella finestra di dialogo Effetto avanzato. Equivale a utilizzare l'impostazione Colore > Avanzato nella finestra di ispezione Proprietà dell'istanza e a regolare i controlli presenti sul lato destro della finestra di dialogo. Questo valore riduce o aumenta i valori della tinta e i valori alfa in base a una quantità costante. Viene aggiunto al valore corrente. Questa proprietà è utile soprattutto se utilizzata con. `symbolInstance.colorAlphaPercent`. I valori accettabili sono quelli compresi tra -255 e 255.

Esempio
L'esempio seguente sottrae 100 dall'impostazione alfa dell'istanza di simbolo selezionata:

```
fl.getDocumentDOM().selection[0].colorAlphaAmount = -100;
```

symbolInstance.colorAlphaPercent

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.colorAlphaPercent
```

Descrizione
Proprietà; un numero intero che specifica parte della trasformazione del colore per l'istanza. Equivale a utilizzare l'impostazione Colore > Avanzato nella finestra di ispezione Proprietà dell'istanza (i controlli percentuali presenti sul lato sinistro della finestra di dialogo). Questo valore modifica i valori della tinta e i valori alfa in base a una percentuale specificata. I valori accettabili sono quelli compresi tra -100 e 100. Consultate anche `symbolInstance.colorAlphaAmount`.

Esempio
L'esempio seguente imposta su 80 la proprietà `colorAlphaPercent` dell'istanza di simbolo selezionata:

```
fl.getDocumentDOM().selection[0].colorAlphaPercent = 80;
```

symbolInstance.colorBlueAmount

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.colorBlueAmount
```

Descrizione
Proprietà; un numero intero che fa parte della trasformazione del colore per l'istanza. Equivale a utilizzare l'impostazione Colore > Avanzato della finestra di ispezione Proprietà dell'istanza. I valori accettabili sono quelli compresi tra -255 e 255.

symbolInstance.colorBluePercent

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.colorBluePercent
```

Descrizione
Proprietà; un numero intero che fa parte della trasformazione del colore per l'istanza. Equivale a utilizzare l'impostazione Colore > Avanzato nella finestra di ispezione Proprietà dell'istanza (i controlli percentuali presenti sul lato sinistro della finestra di dialogo). Questo valore imposta i valori del blu su una percentuale specificata. I valori accettabili sono quelli compresi tra -100 e 100.

Esempio
L'esempio seguente imposta su 80 la proprietà `colorBluePercent` dell'istanza di simbolo selezionata:

```
fl.getDocumentDOM().selection[0].colorBluePercent = 80;
```

symbolInstance.colorGreenAmount

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.colorGreenAmount
```

Descrizione
Proprietà; un numero intero che fa parte della trasformazione del colore per l'istanza. Equivale a utilizzare l'impostazione Colore > Avanzato della finestra di ispezione Proprietà dell'istanza. I valori accettabili sono quelli compresi tra -255 e 255.

symbolInstance.colorGreenPercent

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.colorGreenPercent
```

Descrizione
Proprietà; parte della trasformazione del colore per l'istanza. Equivale a utilizzare l'impostazione Colore > Avanzato nella finestra di ispezione Proprietà dell'istanza (i controlli percentuali presenti sul lato sinistro della finestra di dialogo). Questo valore imposta i valori del verde su una percentuale specificata. I valori accettabili sono quelli compresi tra -100 e 100.

Esempio
L'esempio seguente imposta su 70 la proprietà `colorGreenPercent` dell'istanza di simbolo selezionata:

```
fl.getDocumentDOM().selection[0].colorGreenPercent = 70;
```

symbolInstance.colorMode

Disponibilità
Flash MX 2004.

Uso
`symbolInstance.colorMode`

Descrizione
Proprietà; una stringa che specifica la modalità di colore identificata nel menu a comparsa Colore della finestra di ispezione Proprietà del simbolo. I valori accettabili sono `"none"`, `"brightness"`, `"tint"`, `"alpha"` e `"advanced"`.

Esempio
L'esempio seguente imposta su `alpha` la proprietà `colorMode` del primo elemento nel primo fotogramma del primo livello della linea temporale:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].colorMode = "alpha";
```

symbolInstance.colorRedAmount

Disponibilità
Flash MX 2004.

Uso
`symbolInstance.colorRedAmount`

Descrizione
Proprietà; un numero intero che fa parte della trasformazione del colore per l'istanza. Equivale a utilizzare l'impostazione Colore > Avanzato della finestra di ispezione Proprietà dell'istanza. I valori accettabili sono quelli compresi tra -255 e 255.

Esempio
L'esempio seguente imposta su 255 la proprietà `colorRedAmount` dell'istanza di simbolo selezionata:

```
fl.getDocumentDOM().selection[0].colorRedAmount = 255;
```

symbolInstance.colorRedPercent

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.colorRedPercent
```

Descrizione
Proprietà; parte della trasformazione del colore per l'istanza. Equivale a utilizzare l'impostazione Colore > Avanzato nella finestra di ispezione Proprietà dell'istanza (i controlli percentuali presenti sul lato sinistro della finestra di dialogo). Questo valore imposta i valori del rosso su una percentuale specificata. I valori accettabili sono quelli compresi tra -100 e 100.

Esempio
L'esempio seguente imposta su 10 la proprietà `colorRedPercent` dell'istanza di simbolo selezionata:

```
fl.getDocumentDOM().selection[0].colorRedPercent = 10;
```

symbolInstance.description

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.description
```

Descrizione
Proprietà; una stringa equivalente al campo Descrizione del pannello Accessibilità. La descrizione viene letta dallo screen reader. Questa proprietà non è disponibile per i simboli grafici.

Esempio
L'esempio seguente memorizza nella variabile `theDescription` il valore della descrizione del pannello Accessibilità dell'oggetto:

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

L'esempio seguente imposta la frase `Click the home button to go to home` come descrizione del pannello Accessibilità dell'oggetto:

```
fl.getDocumentDOM().selection[0].description= "Click the home button to go to home";
```

symbolInstance.filters

Disponibilità
Flash 8.

Uso
```
symbolInstance.filters
```

Descrizione

Proprietà; un array di oggetti Filter (consultate Oggetto Filter). Per modificare le proprietà Filter, non potete scrivere direttamente nell'array. Dovete invece recuperare l'array, impostare le singole proprietà e quindi impostare l'array in modo che rifletta le nuove proprietà.

Esempio

L'esempio seguente traccia il nome del filtro nella posizione di indice 0. Se si tratta di un filtro Bagliore, la proprietà `blurX` viene impostata su 100 e il nuovo valore viene scritto nell'array filters.

```
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

Disponibilità

Flash MX 2004.

Uso

```
symbolInstance.firstFrame
```

Descrizione

Proprietà; un numero intero a base zero che specifica il primo fotogramma che compare nella linea temporale dell'elemento grafico. Questa proprietà si applica solo ai simboli grafici e imposta la stessa proprietà del campo Primo della finestra di ispezione Proprietà. Per gli altri tipi di simbolo, la proprietà è `undefined`.

Esempio

L'esempio seguente specifica che il fotogramma 10 deve essere il primo che compare nella linea temporale dell'elemento specificato:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].firstFrame = 10;
```

symbolInstance.forceSimple

Disponibilità

Flash MX 2004.

Uso

```
symbolInstance.forceSimple
```

Descrizione
Proprietà; un valore booleano che abilita e disabilita l'accessibilità degli elementi secondari dell'oggetto. È equivalente alla logica inversa dell'impostazione Rendi accessibili gli oggetti secondari del pannello Accessibilità. Ad esempio, un valore `true` per `forceSimple` equivale all'opzione Rendi accessibili gli oggetti secondari deselezionata. Un valore `false` per `forceSimple` equivale alla stessa opzione selezionata.

Questa proprietà è disponibile solo per gli oggetti MovieClip.

Esempio
L'esempio seguente verifica se gli elementi secondari dell'oggetto sono accessibili; se viene restituito il valore `false` significa che gli oggetti secondari sono accessibili:

```
var areChildrenAccessible = fl.getDocumentDOM().selection[0].forceSimple;
```

L'esempio seguente rende accessibili gli elementi secondari dell'oggetto:

```
fl.getDocumentDOM().selection[0].forceSimple = false;
```

symbolInstance.loop

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.loop
```

Descrizione
Proprietà: una stringa che imposta per i simboli grafici la stessa proprietà del menu a comparsa Ciclo della finestra di ispezione Proprietà. Per gli altri tipi di simbolo, la proprietà è `undefined`. I valori accettabili per impostare l'animazione dell'elemento grafico sono `"loop"`, `"play once"` e `"single frame"`.

Esempio
L'esempio seguente imposta il primo simbolo nel primo fotogramma sul primo livello nella linea temporale su `Fotogramma singolo` (visualizza un fotogramma specificato nella linea temporale dell'elemento grafico), a condizione che il simbolo sia un elemento grafico:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].loop = 'single frame';
```

symbolInstance.shortcut

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.shortcut
```

Descrizione
Proprietà; una stringa equivalente al tasto di scelta rapida associato al simbolo. Equivale al campo Tasto di scelta rapida del pannello Accessibilità. Il tasto viene letto dallo screen reader. Questa proprietà non è disponibile per i simboli grafici.

Esempio
L'esempio seguente memorizza nella variabile `theShortcut` il valore del tasto di scelta rapida dell'oggetto:

```
var theShortcut = fl.getDocumentDOM().selection[0].shortcut;
```

L'esempio seguente imposta il tasto di scelta rapida dell'oggetto su `Ctrl+i`:

```
fl.getDocumentDOM().selection[0].shortcut = "Ctrl+i";
```

symbolInstance.silent

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.silent
```

Descrizione
Proprietà; un valore booleano che abilita e disabilita l'accessibilità dell'oggetto. È equivalente alla logica inversa dell'impostazione Rendi accessibile l'oggetto del pannello Accessibilità. Ad esempio, un valore `true` per `silent` equivale all'opzione Rendi accessibili gli oggetti secondari deselezionata. Un valore `false` per `silent` equivale alla stessa opzione selezionata.

Questa proprietà non è disponibile per gli oggetti grafici.

Esempio
L'esempio seguente verifica se l'oggetto è accessibile; se viene restituito il valore `false` significa che l'oggetto è accessibile:

```
var isSilent = fl.getDocumentDOM().selection[0].silent;
```

L'esempio seguente imposta l'oggetto come accessibile:

```
fl.getDocumentDOM().selection[0].silent = false;
```

symbolInstance.symbolType

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.symbolType
```

Descrizione
Proprietà; una stringa che specifica il tipo di simbolo. Equivale al valore dell'opzione Comportamento nelle finestre di dialogo Crea un nuovo simbolo e Converti in simbolo. I valori accettabili sono `"button"`, `"movie clip"` e `"graphic"`.

Esempio
L'esempio seguente imposta il primo simbolo sul primo fotogramma del primo livello nella linea temporale del documento corrente in modo che si comporti come un simbolo grafico:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].symbolType = "graphic";
```

symbolInstance.tabIndex

Disponibilità
Flash MX 2004.

Uso
```
symbolInstance.tabIndex
```

Descrizione
Proprietà; un numero intero equivalente al campo Indice tabulazione del pannello Accessibilità. Crea l'ordine di tabulazione in base al quale viene eseguito l'accesso agli oggetti quando l'utente preme il tasto Tab. Questa proprietà non è disponibile per i simboli grafici.

Esempio
L'esempio seguente imposta su 3 la proprietà `tabIndex` dell'oggetto `mySymbol` e visualizza il valore nel pannello Output:

```
var mySymbol = fl.getDocumentDOM().selection[0];
mySymbol.tabIndex = 3;
fl.trace(mySymbol.tabIndex);
```

Capitolo 43: Oggetto SymbolItem

Ereditarietà Oggetto Item > Oggetto SymbolItem

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto SymbolItem è una sottoclasse dell'Oggetto Item.

Riepilogo dei metodi
Oltre a quelli dell'oggetto Item, con l'oggetto SymbolItem potete utilizzare i seguenti metodi.

Metodo	Descrizione
`symbolItem.convertToCompiledClip()`	Converte un simbolo presente nella libreria in un clip filmato compilato.
`symbolItem.exportSWC()`	Esporta il simbolo in un file SWC.
`symbolItem.exportSWF()`	Esporta il simbolo in un file SWF.

Riepilogo delle proprietà
Oltre a quelle dell'oggetto Item, l'oggetto SymbolItem dispone delle seguenti proprietà.

Proprietà	Descrizione
`symbolItem.scalingGrid`	Un valore booleano che specifica se la modifica in scala a 9 porzioni è attivata per l'elemento.
`symbolItem.scalingGridRect`	Un oggetto Rectangle che specifica le posizioni delle quattro guide delle 9 porzioni.
`symbolItem.sourceAutoUpdate`	Un valore booleano che specifica se l'elemento viene aggiornato quando viene pubblicato il file FLA.
`symbolItem.sourceFilePath`	Una stringa che specifica il percorso del file FLA di origine come URI file:///.
`symbolItem.sourceLibraryName`	Una stringa che specifica il nome dell'elemento presente nella libreria dei file di origine.
`symbolItem.symbolType`	Una stringa che specifica il tipo di simbolo.
`symbolItem.timeline`	Sola lettura; un Oggetto Timeline.

symbolItem.convertToCompiledClip()

Disponibilità
Flash MX 2004.

Uso
```
symbolItem.convertToCompiledClip()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; converte un simbolo presente nella libreria in un clip filmato compilato.

Esempio
L'esempio seguente converte un elemento presente nella libreria in un clip filmato compilato:

```
fl.getDocumentDOM().library.items[3].convertToCompiledClip();
```

symbolItem.exportSWC()

Disponibilità
Flash MX 2004.

Uso
```
symbolItem.exportSWC(outputURI)
```

Parametri
`outputURI` Una stringa, espressa come URI file:///, che specifica il file SWC in cui il metodo esporta il simbolo. Il parametro *outputURI* deve fare riferimento a un file locale. Se *outputURI* non esiste, non viene creata una cartella.

Restituisce
Nulla.

Descrizione
Metodo; esporta il simbolo in un file SWC.

Esempio
L'esempio seguente esporta un elemento della libreria nel file SWC denominato `mySymbol.swc` presente nella cartella tests:

```
fl.getDocumentDOM.library.selectItem("mySymbol");
var currentSelection = fl.getDocumentDOM().library.getSelectedItems();
currentSelection[0].exportSWC("file:///Macintosh HD/SWCDirectory/mySymbol.swc");
```

symbolItem.exportSWF()

Disponibilità
Flash MX 2004.

Uso

```
symbolItem.exportSWF(outputURI)
```

Parametri

`outputURI` Una stringa, espressa come URI file:///, che specifica il file SWF in cui il metodo esporta il simbolo. Il parametro *outputURI* deve fare riferimento a un file locale. Se *outputURI* non esiste, non viene creata una cartella.

Restituisce

Nulla.

Descrizione

Metodo; esporta il simbolo in un file SWF.

Esempio

L'esempio seguente esporta un elemento della libreria nel file `my.swf` presente nella cartella tests:

```
fl.getDocumentDOM().library.items[0].exportSWF("file:///c|/tests/my.swf");
```

symbolItem.scalingGrid

Disponibilità

Flash 8.

Uso

```
symbolItem.scalingGrid
```

Descrizione

Proprietà; un valore booleano che specifica se la modifica in scala a 9 porzioni è attivata per l'elemento.

Esempio

L'esempio seguente attiva la modifica in scala a 9 porzioni per un elemento della libreria:

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

Consultate anche

symbolItem.scalingGridRect

symbolItem.scalingGridRect

Disponibilità

Flash 8.

Uso

```
symbolItem.scalingGridRect
```

Descrizione
Proprietà; un oggetto Rectangle che specifica le posizioni delle quattro guide delle 9 porzioni. Per informazioni sul formato del rettangolo, consultate `document.addNewRectangle()`.

Esempio
L'esempio seguente specifica le posizioni delle guide delle 9 porzioni:

```
fl.getDocumentDOM().library.items[0].scalingGridRect = {left:338, top:237, right:3859,
bottom:713};
```

Consultate anche
`symbolItem.scalingGrid`

symbolItem.sourceAutoUpdate

Disponibilità
Flash MX 2004.

Uso
```
symbolItem.sourceAutoUpdate
```

Descrizione
Proprietà; un valore booleano che specifica se l'elemento viene aggiornato quando viene pubblicato il file FLA. Il valore predefinito è `false`. Questa proprietà viene utilizzata per i simboli delle librerie condivise.

Esempio
L'esempio seguente imposta la proprietà `sourceAutoUpdate` per un elemento di libreria:

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

symbolItem.sourceFilePath

Disponibilità
Flash MX 2004.

Uso
```
symbolItem.sourceFilePath
```

Descrizione
Proprietà; una stringa che specifica il percorso del file FLA di origine come URI file:///. Il percorso deve essere assoluto, non relativo. Questa proprietà viene utilizzata per i simboli delle librerie condivise.

Esempio
L'esempio seguente mostra il valore della proprietà `sourceFilePath` nel pannello Output:

```
fl.trace(fl.getDocumentDOM().library.items[0].sourceFilePath);
```

symbolItem.sourceLibraryName

Disponibilità
Flash MX 2004.

Uso
`symbolItem.sourceLibraryName`

Descrizione
Proprietà; una stringa che specifica il nome dell'elemento presente nella libreria dei file di origine. Questa proprietà viene utilizzata per i simboli delle librerie condivise.

Esempio
L'esempio seguente mostra il valore della proprietà `sourceLibraryName` nel pannello Output:

```
fl.trace(fl.getDocumentDOM().library.items[0].sourceLibraryName);
```

symbolItem.symbolType

Disponibilità
Flash MX 2004.

Uso
`symbolItem.symbolType`

Descrizione
Proprietà; una stringa che specifica il tipo di simbolo. I valori accettabili sono `"movie clip"`, `"button"` e `"graphic"`.

Esempio
L'esempio seguente mostra il valore corrente della proprietà `symbolType`, la trasforma in `button` e la mostra nuovamente:

```
alert(fl.getDocumentDOM().library.items[0].symbolType);
fl.getDocumentDOM().library.items[0].symbolType = "button";
alert(fl.getDocumentDOM().library.items[0].symbolType);
```

symbolItem.timeline

Disponibilità
Flash MX 2004.

Uso
`symbolItem.timeline`

Descrizione

Proprietà di sola lettura; un Oggetto Timeline.

Esempio

L'esempio seguente ottiene e mostra il numero di livelli contenuti nel clip filmato selezionato nella libreria:

```
var tl = fl.getDocumentDOM().library.getSelectedItems()[0].timeline;
alert(tl.layerCount);
```

Capitolo 44: Oggetto Text

Ereditarietà Oggetto Element > Oggetto Text

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto Text rappresenta un elemento di testo singolo in un documento. Tutte le proprietà del testo agiscono sull'intero blocco di testo.

Per impostare le proprietà di un testo all'interno del campo di testo, consultate il riepilogo delle proprietà per l'Oggetto TextAttrs. Per modificare le proprietà di una selezione all'interno di un campo di testo, potete utilizzare `document.setElementTextAttr()` e specificare un intervallo di testo, oppure utilizzare la selezione corrente.

Per impostare proprietà generiche del campo di testo selezionato, utilizzate `document.setElementProperty()`. L'esempio seguente imposta su 50 il valore `x` del punto di registrazione del campo di testo selezionato:

```
fl.getDocumentDOM().setElementProperty("x", 50);
```

Riepilogo dei metodi
Oltre a quelli dell'oggetto Element, per l'oggetto Text sono disponibili i metodi seguenti:

Metodo	Descrizione
`text.getTextAttr()`	Recupera l'attributo specificato per il testo identificato dai parametri opzionali ***startIndex*** ed ***endIndex***.
`text.getTextString()`	Recupera l'intervallo di testo specificato.
`text.setTextAttr()`	Imposta l'attributo specificato associato al testo identificato dai parametri ***startIndex*** ed ***endIndex***.
`text.setTextString()`	Modifica la stringa di testo all'interno dell'oggetto Text.

Riepilogo delle proprietà
Oltre a quelle dell'oggetto Element, per l'oggetto Text sono disponibili le proprietà seguenti:

Proprietà	Descrizione
`text.accName`	Una stringa equivalente al campo Nome del pannello Accessibilità.
`text.antiAliasSharpness`	Un valore float che specifica la precisione di antialiasing del testo.
`text.antiAliasThickness`	Un valore float che specifica lo spessore di antialiasing del testo.
`text.autoExpand`	Un valore booleano che controlla l'espansione della larghezza di delimitazione dei campi di testo statici o la larghezza e l'altezza di delimitazione del testo dinamico o di input.
`text.border`	Un valore booleano che controlla se il bordo attorno al testo dinamico o di input è visibile (`true`) o nascosto (`false`).
`text.description`	Una stringa equivalente al campo Descrizione del pannello Accessibilità.
`text.embeddedCharacters`	Una stringa che specifica i caratteri da incorporare. Equivale a immettere il testo nella finestra di dialogo Incorporamento caratteri.

Proprietà	Descrizione
text.embedRanges	Una stringa composta da numeri interi delimitati che corrispondono agli elementi selezionabili nella finestra di dialogo Incorporamento caratteri.
text.embedVariantGlyphs	Una valore booleano che specifica se abilitare l'incorporamento di glifi varianti.
text.fontRenderingMode	Una stringa che specifica la modalità di rendering del testo.
text.length	Di sola lettura; un numero intero che rappresenta il numero di caratteri presenti nell'oggetto Text.
text.lineType	Una stringa che imposta il tipo di riga su "single line", "multiline", "multiline no wrap" o "password".
text.maxCharacters	Un numero intero che specifica il numero massimo di caratteri che l'utente può immettere nell'oggetto Text.
text.orientation	Una stringa che specifica l'orientamento del campo di testo.
text.renderAsHTML	Un valore booleano che controlla se il testo viene disegnato come HTML e se i tag HTML incorporati vengono interpretati.
text.scrollable	Un valore booleano che controlla se è possibile (true) o meno (false) scorrere il testo.
text.selectable	Un valore booleano che controlla se il testo può essere selezionato (true) oppure no (false). Il testo di input è sempre selezionabile.
text.selectionEnd	Un numero intero a base zero che specifica l'offset alla fine di una sottoselezione di testo.
text.selectionStart	Un numero intero a base zero che specifica l'offset all'inizio di una sottoselezione di testo.
text.shortcut	Una stringa equivalente al campo Tasto di scelta rapida del pannello Accessibilità.
text.silent	Un valore booleano che specifica se l'oggetto è accessibile.
text.tabIndex	Un numero intero equivalente al campo Indice tabulazione del pannello Accessibilità.
text.textRuns	Di sola lettura; un array di oggetti TextRun.
text.textType	Una stringa che specifica il tipo di campo di testo. I valori accettabili sono "static", "dynamic" e "input".
text.useDeviceFonts	Un valore booleano. Se specificate true, il testo viene disegnato utilizzando i caratteri del dispositivo.
text.variableName	Una stringa che include il contenuto dell'oggetto Text.

text.accName

Disponibilità
Flash MX 2004.

Uso
```
text.accName
```

Descrizione
Proprietà; una stringa equivalente al campo Nome del pannello Accessibilità. Gli screen reader identificano gli oggetti pronunciandone il nome. Questa proprietà non può essere utilizzata con il testo dinamico.

Esempio

L'esempio seguente recupera il nome dell'oggetto:

```
var doc = fl.getDocumentDOM();
var theName = doc.selection[0].accName;
```

L'esempio seguente imposta il nome dell'oggetto selezionato:

```
fl.getDocumentDOM().selection[0].accName = "Home Button";
```

text.antiAliasSharpness

Disponibilità

Flash 8.

Uso

`text.antiAliasSharpness`

Descrizione

Proprietà; un valore float che specifica la precisione di antialiasing del testo. Questa proprietà controlla la precisione del testo disegnato; a un valore più alto corrisponde una precisione (o nitidezza) maggiore. Il valore 0 specifica una nitidezza normale. Questa proprietà è disponibile solo se `text.fontRenderingMode` è impostata su `customThicknessSharpness`.

Esempio

Consultate `text.fontRenderingMode`.

Consultate anche

`text.antiAliasThickness`, `text.fontRenderingMode`

text.antiAliasThickness

Disponibilità

Flash 8.

Uso

`text.antiAliasThickness`

Descrizione

Proprietà; un valore float che specifica lo spessore di antialiasing del testo. Questa proprietà controlla lo spessore del testo disegnato; a un valore più alto corrisponde uno spessore maggiore. Il valore 0 specifica uno spessore normale. Questa proprietà è disponibile solo se `text.fontRenderingMode` è impostata su `customThicknessSharpness`.

Esempio

Consultate `text.fontRenderingMode`.

Consultate anche

`text.antiAliasSharpness`, `text.fontRenderingMode`

text.autoExpand

Disponibilità

Flash MX 2004.

Uso

```
text.autoExpand
```

Descrizione

Proprietà; un valore booleano. Per i campi di testo statici, il valore `true` fa in modo che la larghezza di delimitazione venga espansa per mostrare tutto il testo. Per i campi di testo dinamici o di input, il valore `true` fa in modo che la larghezza e l'altezza di delimitazione vengano espanse per mostrare tutto il testo.

Esempio

L'esempio seguente imposta la proprietà `autoExpand` sul valore `true`:

```
fl.getDocumentDOM().selection[0].autoExpand = true;
```

text.border

Disponibilità

Flash MX 2004.

Uso

```
text.border
```

Descrizione

Proprietà; un valore booleano. Se specificate `true`, viene visualizzato un bordo attorno al testo.

Esempio

L'esempio seguente imposta la proprietà `border` sul valore `true`:

```
fl.getDocumentDOM().selection[0].border = true;
```

text.description

Disponibilità

Flash MX 2004.

Uso

```
text.description
```

Descrizione
Proprietà; una stringa equivalente al campo Descrizione del pannello Accessibilità. La descrizione viene letta dallo screen reader.

Esempio
L'esempio seguente recupera la descrizione dell'oggetto:

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

L'esempio seguente imposta la descrizione dell'oggetto:

```
var doc = fl.getDocumentDOM();
doc.selection[0].description= "Enter your name here";
```

text.embeddedCharacters

Disponibilità
Flash MX 2004.

Uso
`text.embeddedCharacters`

Descrizione
Proprietà; una stringa che specifica i caratteri da incorporare. Equivale a immettere il testo nella finestra di dialogo Incorporamento caratteri.

Questa proprietà funziona solo con testo dinamico o di input e genera un avviso se viene utilizzata con altri tipi di testo.

Esempio
L'esempio seguente imposta la proprietà `embeddedCharacters` su `abc`:

```
fl.getDocumentDOM().selection[0].embeddedCharacters = "abc";
```

text.embedRanges

Disponibilità
Flash MX 2004.

Uso
`text.embedRanges`

Descrizione
Proprietà; una stringa composta da numeri interi delimitati che corrispondono agli elementi selezionabili nella finestra di dialogo Incorporamento caratteri. Questa proprietà funziona solo con testo dinamico o di input; viene ignorata se viene utilizzata con il testo statico.

Nota: *questa proprietà corrisponde al file XML nella cartella Configuration/Font Embedding.*

Esempio

L'esempio seguente imposta la proprietà `embedRanges` su `"1|3|7"`:

```
var doc = fl.getDocumentDOM();
doc.selection[0].embedRanges = "1|3|7";
```

L'esempio seguente reimposta la proprietà:

```
var doc = fl.getDocumentDOM();
doc.selection[0].embedRanges = "";
```

text.embedVariantGlyphs

Disponibilità

Flash CS4 Professional.

Uso

```
text.embedVariantGlyphs
```

Descrizione

Proprietà; un valore booleano che specifica se abilitare (`true`) o meno (`false`) l'incorporamento di glifi varianti. Questa proprietà funziona solo con testo dinamico o di input; viene ignorata se viene utilizzata con il testo statico. Il valore predefinito è `false`.

Esempio

L'esempio seguente abilita l'incorporamento di glifi varianti nell'oggetto Text selezionato:

```
fl.getDocumentDOM().selection[0].embedVariantGlyphs = true;
```

Consultate anche

`fontItem.embedVariantGlyphs`

text.fontRenderingMode

Disponibilità

Flash 8.

Uso

```
text.fontRenderingMode
```

Descrizione

Proprietà; una stringa che specifica la modalità di rendering del testo. Questa proprietà ha effetto sulla visualizzazione del testo nello stage e in Flash Player. I valori accettabili sono elencati nella tabella seguente:

Valore della proprietà	Modalità di rendering del testo
`device`	Esegue il rendering del testo con i caratteri del dispositivo.
`bitmap`	Esegue il rendering del testo sottoposto ad aliasing sotto forma di bitmap, oppure come se venisse usato un pixel font.
`standard`	Esegue il rendering del testo mediante il metodo standard di antialiasing utilizzato da Flash MX 2004. È l'impostazione ottimale per testo animato, di dimensioni molto grandi o inclinato.
`advanced`	Esegue il rendering del testo mediante la tecnologia avanzata di antialiasing dei caratteri implementata in Flash 8, che produce un antialiasing migliore e garantisce una maggiore leggibilità, soprattutto nel caso di testo di dimensioni molto piccole.
`customThicknessSharpness`	Consente di specificare impostazioni personalizzate per la precisione e lo spessore del testo quando si utilizza la tecnologia di antialiasing avanzato per il rendering dei caratteri implementata in Flash 8.

Esempio

L'esempio seguente illustra come utilizzare il valore `customThicknessSharpness` per specificare la precisione e lo spessore del testo:

```
fl.getDocumentDOM().setElementProperty("fontRenderingMode", "customThicknessSharpness");
fl.getDocumentDOM().setElementProperty("antiAliasSharpness", 400);
fl.getDocumentDOM().setElementProperty("antiAliasThickness", -200);
```

Consultate anche

`text.antiAliasSharpness`, `text.antiAliasThickness`

text.getTextAttr()

Disponibilità

Flash MX 2004.

Uso

```
text.getTextAttr(attrName [, startIndex [, endIndex]])
```

Parametri

`attrName` Una stringa che specifica il nome della proprietà dell'oggetto TextAttrs da restituire. Per un elenco dei valori possibili per *`attrName`*, consultate il riepilogo delle proprietà per l'Oggetto TextAttrs.

`startIndex` Un numero intero che corrisponde all'indice del primo carattere. Questo parametro è opzionale.

`endIndex` Un numero intero che specifica la fine dell'intervallo di testo compreso tra *startIndex* e *endIndex* (escluso). Questo parametro è opzionale.

Restituisce

Il valore dell'attributo specificato nel parametro *attrName*.

Descrizione
Metodo; recupera l'attributo specificato dal parametro *attrName* per il testo identificato dai parametri opzionali *startIndex* ed *endIndex*. Se l'attributo non è coerente per l'intervallo specificato, viene restituito il valore `undefined`. Se si omettono i parametri opzionali *startIndex* ed *endIndex*, il metodo utilizza l'intero intervallo di testo. Se specificate solo *startIndex*, l'intervallo utilizzato è un carattere singolo nella posizione indicata. Se specificate sia *startIndex* che *endIndex*, l'intervallo è compreso tra *startIndex* ed *endIndex* escluso.

Esempio
L'esempio seguente ottiene la dimensione del carattere del campo di testo selezionato e la visualizza:

```
var TheTextSize = fl.getDocumentDOM().selection[0].getTextAttr("size");
fl.trace(TheTextSize);
```

L'esempio seguente ottiene il colore di riempimento del testo per il campo di testo selezionato:

```
var TheFill = fl.getDocumentDOM().selection[0].getTextAttr("fillColor");
fl.trace(TheFill);
```

L'esempio seguente ottiene le dimensioni del terzo carattere:

```
var Char3 = fl.getDocumentDOM().selection[0].getTextAttr("size", 2);
fl.trace(Char3);
```

L'esempio seguente ottiene il colore del campo di testo selezionato dal terzo all'ottavo carattere:

```
fl.getDocumentDOM().selection[0].getTextAttr("fillColor", 2, 8);
```

text.getTextString()

Disponibilità
Flash MX 2004.

Uso
```
text.getTextString([startIndex [, endIndex]])
```

Parametri
`startIndex` Un numero intero che specifica l'indice (a base zero) del primo carattere. Questo parametro è opzionale.

`endIndex` Un numero intero che specifica la fine dell'intervallo di testo compreso tra *startIndex* e *endIndex* (escluso). Questo parametro è opzionale.

Restituisce
Una stringa del testo nell'intervallo specificato.

Descrizione
Metodo; recupera l'intervallo di testo specificato. Se si omettono i parametri opzionali *startIndex* ed *endIndex*, viene restituita la stringa di testo intera. Se specificate solo *startIndex*, il metodo restituisce la stringa che inizia in corrispondenza dell'indice e che termina alla fine del campo. Se specificate sia *startIndex* che *endIndex*, il metodo restituisce la stringa compresa tra *startIndex* ed *endIndex* escluso.

Esempio
L'esempio seguente ottiene i caratteri compresi tra il quinto carattere e la fine del campo di testo selezionato:

```
var myText = fl.getDocumentDOM().selection[0].getTextString(4);
fl.trace(myText);
```

L'esempio seguente ottiene i caratteri dal quarto al nono a partire dal campo di testo selezionato:

```
var myText = fl.getDocumentDOM().selection[0].getTextString(3, 9);
fl.trace(myText);
```

text.length

Disponibilità
Flash MX 2004.

Uso
`text.length`

Descrizione
Proprietà di sola lettura; un numero intero che rappresenta il numero di caratteri presenti nell'oggetto Text.

Esempio
L'esempio seguente restituisce il numero di caratteri presenti nel campo di testo:

```
var textLength = fl.getDocumentDOM().selection[0].length;
```

text.lineType

Disponibilità
Flash MX 2004.

Uso
`text.lineType`

Descrizione
Proprietà; una stringa che imposta il tipo di riga. I valori accettabili sono `"single line"`, `"multiline"`, `"multiline no wrap"` e `"password"`.

Questa proprietà funziona solo con testo dinamico o di input e genera un avviso se viene utilizzata con testo statico. Il valore `"password"` funziona solo con il testo di input.

Esempio
L'esempio seguente imposta la proprietà `lineType` sul valore `multiline no wrap`:

```
fl.getDocumentDOM().selection[0].lineType = "multiline no wrap";
```

text.maxCharacters

Disponibilità
Flash MX 2004.

Uso
`text.maxCharacters`

Descrizione
Proprietà; un numero intero che specifica il numero massimo di caratteri che l'utente può immettere nell'oggetto Text.

Questa proprietà funziona solo con il testo di input; se viene utilizzata con altri tipi di testo, genera un avviso.

Esempio
L'esempio seguente imposta su 30 il valore della proprietà `maxCharacters`:

```
fl.getDocumentDOM().selection[0].maxCharacters = 30;
```

text.orientation

Disponibilità
Flash MX 2004.

Uso
`text.orientation`

Descrizione
Proprietà; una stringa che specifica l'orientamento del campo di testo. I valori accettabili sono `"horizontal"`, `"vertical left to right"` e `"vertical right to left"`.

Questa proprietà funziona solo con testo statico e genera un avviso se viene utilizzata con altri tipi di testo.

Esempio
L'esempio seguente imposta la proprietà dell'orientamento sul valore `vertical right to left`:

```
fl.getDocumentDOM().selection[0].orientation = "vertical right to left";
```

text.renderAsHTML

Disponibilità
Flash MX 2004.

Uso
`text.renderAsHTML`

Descrizione
Proprietà; un valore booleano. Se il valore è `true`, il testo viene disegnato come HTML e i tag HTML incorporati vengono interpretati.

Questa proprietà funziona solo con testo dinamico o di input e genera un avviso se viene utilizzata con altri tipi di testo.

Esempio
L'esempio seguente imposta la proprietà `renderAsHTML` su `true`:

```
fl.getDocumentDOM().selection[0].renderAsHTML = true;
```

text.scrollable

Disponibilità
Flash MX 2004.

Uso
```
text.scrollable
```

Descrizione
Proprietà; un valore booleano. Se il valore è `true`, è possibile scorrere il testo.

Questa proprietà funziona solo con testo dinamico o di input e genera un avviso se viene utilizzata con testo statico.

Esempio
L'esempio seguente imposta la proprietà `scrollable` su `false`:

```
fl.getDocumentDOM().selection[0].scrollable = false;
```

text.selectable

Disponibilità
Flash MX 2004.

Uso
```
text.selectable
```

Descrizione
Proprietà; un valore booleano. Se il valore è `true`, il testo è selezionabile.

Il testo di input è sempre selezionabile. Viene generato un avviso se la proprietà è impostata su `false` e viene utilizzata con il testo di input.

Esempio
L'esempio seguente imposta la proprietà `selectable` su `true`:

```
fl.getDocumentDOM().selection[0].selectable = true;
```

text.selectionEnd

Disponibilità
Flash MX 2004.

Uso
`text.selectionEnd`

Descrizione
Proprietà; un numero intero a base zero che specifica la fine di una sottoselezione di testo. Per ulteriori informazioni, consultate `text.selectionStart`.

text.selectionStart

Disponibilità
Flash MX 2004.

Uso
`text.selectionStart`

Descrizione
Proprietà; un numero intero a base zero che specifica l'inizio di una sottoselezione di testo. Potete utilizzare questa proprietà con `text.selectionEnd` per selezionare un intervallo di caratteri. Vengono selezionati i caratteri fino a `text.selectionEnd` escluso. Consultate `text.selectionEnd`.

- Se è presente un punto di inserimento oppure non è presente alcuna selezione, `text.selectionEnd` è uguale a `text.selectionStart`.
- Se `text.selectionStart` è impostato su un valore superiore a `text.selectionEnd`, `text.selectionEnd` è impostato su `text.selectionStart` e non è selezionato alcun testo.

Esempio
L'esempio seguente imposta il sesto carattere come inizio della sottoselezione di testo:

```
fl.getDocumentDOM().selection[0].selectionStart = 5;
```

L'esempio seguente seleziona i caratteri `Barbara` da un campo di testo che contiene il testo `My name is Barbara` e li formatta in grassetto verde:

```
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()

Disponibilità
Flash MX 2004.

Uso
```
text.setTextAttr(attrName, attrValue [, startIndex [, endIndex]])
```

Parametri
`attrName` Una stringa che specifica il nome della proprietà dell'oggetto TextAttrs da modificare.

`attrValue` Il valore della proprietà dell'oggetto TextAttrs.

Per un elenco dei valori possibili per *`attrName`* e *`attrValue`*, consultate il riepilogo delle proprietà per l'Oggetto TextAttrs.

`startIndex` Un numero intero che corrisponde all'indice (a base zero) del primo carattere nell'array. Questo parametro è opzionale.

`endIndex` Un numero intero che specifica l'indice del punto finale nella stringa di testo selezionata compresa tra *startIndex* ed *endIndex* escluso. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; imposta il valore specificato da *attrValue* per l'attributo specificato dal parametro *attrName* associato al testo identificato dai parametri *startIndex* ed *endIndex*. Questo metodo può essere utilizzato per modificare gli attributi di testo che possono coprire elementi TextRun (consultate Oggetto TextRun) o che sono porzioni di elementi TextRun esistenti. Se utilizzate il metodo, è possibile che vengano modificati la posizione e il numero degli elementi TextRun all'interno dell'array `text.textRuns` di questo oggetto (consultate text.textRuns).

Se si omettono i parametri opzionali, il metodo utilizza l'intero intervallo di caratteri dell'oggetto Text. Se specificate solo *startIndex*, l'intervallo è un carattere singolo nella posizione indicata. Se specificate sia *startIndex* che *endIndex*, l'intervallo è compreso tra *startIndex* e il carattere in corrispondenza di *endIndex* escluso.

Esempio
L'esempio seguente imposta il corsivo per il testo selezionato:

```
fl.getDocumentDOM().selection[0].setTextAttr("italic", true);
```

L'esempio seguente imposta su 10 le dimensioni del terzo carattere:

```
fl.getDocumentDOM().selection[0].setTextAttr("size", 10, 2);
```

L'esempio seguente imposta il rosso come colore dei caratteri dal terzo all'ottavo nel testo selezionato:

```
fl.getDocumentDOM().selection[0].setTextAttr("fillColor", 0xff0000, 2, 8);
```

text.setTextString()

Disponibilità
Flash MX 2004.

Uso
```
text.setTextString(text [, startIndex [, endIndex]])
```

Parametri
`text` Una stringa costituita dai caratteri da inserire nell'oggetto Text.

`startIndex` Un numero intero che specifica l'indice (a base zero) del carattere della stringa in cui verrà inserito il testo. Questo parametro è opzionale.

`endIndex` Un numero intero che specifica l'indice del punto finale nella stringa di testo selezionata. Il nuovo testo sovrascrive il testo compreso tra *startIndex* ed *endIndex* escluso. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Proprietà; modifica la stringa di testo all'interno dell'oggetto Text. Se si omettono i parametri opzionali, viene sostituito l'intero oggetto Text. Se specificate solo *startIndex*, la stringa specificata viene inserita in corrispondenza di *startIndex.* Se specificateno sia *startIndex* che *endIndex*, la stringa specificata sostituisce il segmento di testo compreso tra *startIndex* ed *endIndex* escluso.

Esempio
L'esempio seguente assegna la stringa `this is a string` al campo di testo selezionato:

```
fl.getDocumentDOM().selection[0].setTextString("this is a string");
```

L'esempio seguente inserisce la stringa `abc` a partire dal quinto carattere del campo di testo selezionato:

```
fl.getDocumentDOM().selection[0].setTextString("01234567890");
fl.getDocumentDOM().selection[0].setTextString("abc", 4);
// text field is now "0123abc4567890"
```

L'esempio seguente sostituisce il testo compreso tra il terzo e l'ottavo carattere della stringa di testo selezionata con la stringa `abcdefghij`. I caratteri compresi tra *startIndex* ed *endIndex* vengono sovrascritti. I caratteri che cominciano con *endIndex* seguono la stringa inserita.

```
fl.getDocumentDOM().selection[0].setTextString("01234567890");
fl.getDocumentDOM().selection[0].setTextString("abcdefghij", 2, 8);
// text field is now "01abcdefghij890"
```

text.shortcut

Disponibilità
Flash MX 2004.

Uso
`text.shortcut`

Descrizione
Proprietà; una stringa equivalente al campo Tasto di scelta rapida del pannello Accessibilità. Il tasto di scelta rapida viene letto dallo screen reader. Questa proprietà non può essere utilizzata con il testo dinamico.

Esempio
L'esempio seguente ottiene il tasto di scelta rapida dell'oggetto selezionato e mostra il valore:

```
var theShortcut = fl.getDocumentDOM().selection[0].shortcut;
fl.trace(theShortcut);
```

L'esempio seguente imposta il tasto di scelta rapida dell'oggetto selezionato:

```
fl.getDocumentDOM().selection[0].shortcut = "Ctrl+i";
```

text.silent

Disponibilità
Flash MX 2004.

Uso
`text.silent`

Descrizione
Proprietà; un valore booleano che specifica se l'oggetto è accessibile. È equivalente alla logica inversa dell'impostazione Rendi accessibile l'oggetto del pannello Accessibilità. In altre parole, se `silent` è `true`, l'opzione Rendi accessibile l'oggetto è deselezionata. Se è `false`, l'opzione è selezionata.

Esempio
L'esempio seguente determina se l'oggetto è accessibile (se viene restituito il valore `false` significa che l'oggetto è accessibile):

```
var isSilent = fl.getDocumentDOM().selection[0].silent;
```

L'esempio seguente imposta l'oggetto come accessibile:

```
fl.getDocumentDOM().selection[0].silent = false;
```

text.tabIndex

Disponibilità
Flash MX 2004.

Uso
`text.tabIndex`

Descrizione
Proprietà; un numero intero equivalente al campo Indice tabulazione del pannello Accessibilità. Consente di determinare l'ordine di accesso agli oggetti quando l'utente preme il tasto Tab.

Esempio
L'esempio seguente ottiene il valore di `tabIndex` per l'oggetto selezionato:

```
var theTabIndex = fl.getDocumentDOM().selection[0].tabIndex;
```

L'esempio seguente imposta il valore di `tabIndex` dell'oggetto selezionato:

```
fl.getDocumentDOM().selection[0].tabIndex = 1;
```

text.textRuns

Disponibilità
Flash MX 2004.

Uso
`text.textRuns`

Descrizione
Proprietà di sola lettura; un array di oggetti TextRun (consultate Oggetto TextRun).

Esempio
L'esempio seguente memorizza il valore della proprietà `textRuns` nella variabile `myTextRuns`:

```
var myTextRuns = fl.getDocumentDOM().selection[0].textRuns;
```

text.textType

Disponibilità
Flash MX 2004.

Uso
`text.textType`

Descrizione
Proprietà; una stringa che specifica il tipo di campo di testo. I valori accettabili sono `"static"`, `"dynamic"` e `"input"`.

Esempio
L'esempio seguente imposta la proprietà `textType` su `input`:

```
fl.getDocumentDOM().selection[0].textType = "input";
```

text.useDeviceFonts

Disponibilità
Flash MX 2004.

Uso
`text.useDeviceFonts`

Descrizione
Proprietà; un valore booleano. Se specificate `true`, il testo viene disegnato utilizzando i caratteri del dispositivo.

Esempio
L'esempio seguente fa in modo che il testo venga disegnato mediante i caratteri del dispositivo:

```
fl.getDocumentDOM().selection[0].useDeviceFonts = true;
```

text.variableName

Disponibilità
Flash MX 2004.

Uso
`text.variableName`

Descrizione
Proprietà; una stringa che contiene il nome della variabile associata all'oggetto Text. Questa proprietà funziona solo con testo dinamico o di input e genera un avviso se viene utilizzata con altri tipi di testo.

Questa proprietà è supportata solo in ActionScript 1.0 e in ActionScript 2.0.

Esempio
L'esempio seguente imposta su `firstName` il nome della variabile del campo di testo selezionato:

```
fl.getDocumentDOM().selection[0].variableName = "firstName";
```

Capitolo 45: Oggetto TextAttrs

Disponibilità

Flash MX 2004.

Descrizione

L'oggetto TextAttrs contiene tutte le proprietà di testo che è possibile applicare a una sottoselezione. Questo oggetto è una proprietà dell'oggetto TextRun (`textRun.textAttrs`).

Riepilogo delle proprietà

Le seguenti proprietà sono disponibili per l'oggetto TextAttrs:

Proprietà	Descrizione
`textAttrs.aliasText`	Un valore booleano che specifica che il testo deve essere disegnato utilizzando un metodo ottimizzato per aumentare la leggibilità del testo di piccole dimensioni.
`textAttrs.alignment`	Una stringa che specifica l'allineamento del paragrafo. I valori accettabili sono `"left"`, `"center"`, `"right"` e `"justify"`.
`textAttrs.autoKern`	Un valore booleano che determina se le informazioni sulla crenatura delle coppie nei caratteri vengono utilizzate (`true`) o ignorate (`false`) quando si applica la crenatura al testo.
`textAttrs.bold`	Un valore booleano. Se specificate `true`, il testo viene visualizzato utilizzando la versione in grassetto del carattere.
`textAttrs.characterPosition`	Una stringa che determina la linea di base del testo.
`textAttrs.characterSpacing`	L'utilizzo è sconsigliato. Al suo posto, utilizzate `textAttrs.letterSpacing`. Un numero intero che rappresenta lo spazio tra i caratteri.
`textAttrs.face`	Una stringa che rappresenta il nome del carattere, ad esempio `"Arial"`.
`textAttrs.fillColor`	Una stringa, un valore esadecimale o un numero intero che rappresenta il colore di riempimento.
`textAttrs.indent`	Un numero intero che specifica il rientro del paragrafo.
`textAttrs.italic`	Un valore booleano. Se specificate `true`, il testo viene visualizzato utilizzando la versione in corsivo del carattere.
`textAttrs.leftMargin`	Un numero intero che specifica il margine sinistro del paragrafo.
`textAttrs.letterSpacing`	Un numero intero che rappresenta lo spazio tra i caratteri.
`textAttrs.lineSpacing`	Un numero intero che specifica l'interlinea del paragrafo.
`textAttrs.rightMargin`	Un numero intero che specifica il margine destro del paragrafo.
`textAttrs.rotation`	Un valore booleano. Se specificate `true`, i caratteri del testo vengono ruotati di 90°. Il valore predefinito è `false`.
`textAttrs.size`	Un numero intero che specifica le dimensioni del carattere.
`textAttrs.target`	Una stringa che rappresenta la proprietà `target` del campo di testo.
`textAttrs.url`	Una stringa che rappresenta la proprietà `URL` del campo di testo.

textAttrs.aliasText

Disponibilità
Flash MX 2004.

Uso
```
textAttrs.aliasText
```

Descrizione
Proprietà; un valore booleano che specifica che il testo deve essere disegnato utilizzando un metodo ottimizzato per aumentare la leggibilità del testo di piccole dimensioni.

Esempio
L'esempio seguente imposta la proprietà `aliasText` su `true` per tutto il testo presente nel campo di testo selezionato:

```
fl.getDocumentDOM().setElementTextAttr('aliasText', true);
```

textAttrs.alignment

Disponibilità
Flash MX 2004.

Uso
```
textAttrs.alignment
```

Descrizione
Proprietà; una stringa che specifica l'allineamento del paragrafo. I valori accettabili sono `"left"`, `"center"`, `"right"` e `"justify"`.

Esempio
L'esempio seguente giustifica i paragrafi che contengono i caratteri compresi tra l'indice 0 e l'indice 3 escluso. Questa impostazione può influire sui caratteri che fanno parte dello stesso paragrafo ma non sono compresi nell'intervallo specificato.

```
fl.getDocumentDOM().setTextSelection(0, 3);
fl.getDocumentDOM().setElementTextAttr("alignment", "justify");
```

textAttrs.autoKern

Disponibilità
Flash MX 2004.

Uso
```
textAttrs.autoKern
```

Descrizione
Proprietà; un valore booleano che determina se le informazioni sulla crenatura delle coppie nei caratteri vengono utilizzate (`true`) o ignorate (`false`) quando si applica la crenatura al testo.

Esempio
L'esempio seguente seleziona i caratteri compresi tra l'indice 2 e l'indice 6 escluso e imposta la proprietà `autoKern` su `true`:

```
fl.getDocumentDOM().setTextSelection(3, 6);
fl.getDocumentDOM().setElementTextAttr('autoKern', true);
```

textAttrs.bold

Disponibilità
Flash MX 2004.

Uso
```
textAttrs.bold
```

Descrizione
Proprietà; un valore booleano. Se specificate `true`, il testo viene visualizzato utilizzando la versione in grassetto del carattere.

Esempio
L'esempio seguente seleziona il primo carattere dell'oggetto Text selezionato e imposta la proprietà `bold` su `true`:

```
fl.getDocumentDOM().setTextSelection(0, 1);
fl.getDocumentDOM().setElementTextAttr('bold', true);
```

textAttrs.characterPosition

Disponibilità
Flash MX 2004.

Uso
```
textAttrs.characterPosition
```

Descrizione
Proprietà; una stringa che determina la linea di base del testo. I valori accettabili sono `"normal"`, `"subscript"` e `"superscript"`. Questa proprietà funziona solo con il testo statico.

Esempio
L'esempio seguente seleziona i caratteri compresi tra l'indice 2 e l'indice 6 escluso del testo selezionato e imposta la proprietà `characterPosition` su `subscript`:

```
fl.getDocumentDOM().setTextSelection(2, 6);
fl.getDocumentDOM().setElementTextAttr("characterPosition", "subscript");
```

textAttrs.characterSpacing

Disponibilità
Flash MX 2004. L'utilizzo è sconsigliato in Flash 8. Al suo posto utilizzate `textAttrs.letterSpacing`.

Uso
```
textAttrs.characterSpacing
```

Descrizione
Proprietà; un numero intero che rappresenta lo spazio tra i caratteri. I valori accettabili sono compresi tra -60 e 60.

Questa proprietà funziona solo con il testo statico e genera un avviso se viene utilizzata con altri tipi di testo.

Esempio
L'esempio seguente imposta su 10 la spaziatura dei caratteri del testo selezionato:

```
fl.getDocumentDOM().setElementTextAttr("characterSpacing", 10);
```

textAttrs.face

Disponibilità
Flash MX 2004.

Uso
```
textAttrs.face
```

Descrizione
Proprietà; una stringa che rappresenta il nome del carattere, ad esempio `"Arial"`.

Esempio
L'esempio seguente imposta su `Arial` i caratteri compresi tra l'indice 2 e l'indice 8 escluso nel campo di testo selezionato:

```
fl.getDocumentDOM().selection[0].setTextAttr("face", "Arial", 2, 8);
```

textAttrs.fillColor

Disponibilità
Flash MX 2004.

Uso
```
textAttrs.fillColor
```

Descrizione
Proprietà; il colore di riempimento in uno dei seguenti formati:

- Una stringa nel formato `"#RRGGBB"` o `"#RRGGBBAA"`
- Un numero esadecimale nel formato `0xRRGGBB`
- Un numero intero che rappresenta l'equivalente decimale di un numero esadecimale.

Esempio
L'esempio seguente imposta il rosso come colore per i caratteri compresi tra l'indice 2 e l'indice 8 escluso nel campo di testo selezionato:

```
fl.getDocumentDOM().selection[0].setTextAttr("fillColor", 0xff0000, 2, 8);
```

textAttrs.indent

Disponibilità
Flash MX 2004.

Uso
```
textAttrs.indent
```

Descrizione
Proprietà; un numero intero che specifica il rientro del paragrafo. I valori accettabili sono compresi tra -720 e 720.

Esempio
L'esempio seguente imposta su 100 il rientro per i caratteri compresi tra l'indice 2 e l'indice 8 escluso nel campo di testo selezionato. Questa impostazione può influire sui caratteri che fanno parte dello stesso paragrafo ma non sono compresi nell'intervallo specificato.

```
fl.getDocumentDOM().selection[0].setTextAttr("indent", 100, 2, 8);
```

textAttrs.italic

Disponibilità
Flash MX 2004.

Uso
```
textAttrs.italic
```

Descrizione
Proprietà; un valore booleano. Se specificate `true`, il testo viene visualizzato utilizzando la versione in corsivo del carattere.

Esempio
L'esempio seguente imposta il corsivo per il testo selezionato:

```
fl.getDocumentDOM().selection[0].setTextAttr("italic", true);
```

textAttrs.leftMargin

Disponibilità
Flash MX 2004.

Uso
`textAttrs.leftMargin`

Descrizione
Proprietà; un numero intero che specifica il margine sinistro del paragrafo. I valori accettabili sono quelli compresi tra 0 e 720.

Esempio
L'esempio seguente imposta su 100 la proprietà `leftMargin` per i caratteri compresi tra l'indice 2 e l'indice 8 escluso nel campo di testo selezionato. Questa impostazione può influire sui caratteri che fanno parte dello stesso paragrafo ma non sono compresi nell'intervallo specificato.

```
fl.getDocumentDOM().selection[0].setTextAttr("leftMargin", 100, 2, 8);
```

textAttrs.letterSpacing

Disponibilità
Flash 8.

Uso
`textAttrs.letterSpacing`

Descrizione
Proprietà; un numero intero che rappresenta lo spazio tra i caratteri. I valori accettabili sono compresi tra -60 e 60.

Questa proprietà funziona solo con il testo statico e genera un avviso se viene utilizzata con altri tipi di testo.

Esempio
Il codice seguente seleziona i caratteri compresi tra l'indice 0 e l'indice 10 escluso e imposta la spaziatura dei caratteri su 60:

```
fl.getDocumentDOM().setTextSelection(0, 10);
fl.getDocumentDOM().setElementTextAttr("letterSpacing", 60);
```

textAttrs.lineSpacing

Disponibilità
Flash MX 2004.

Uso
`textAttrs.lineSpacing`

Descrizione
Proprietà; un numero intero che specifica la spaziatura tra le righe (*interlinea*) del paragrafo. I valori accettabili sono quelli compresi tra -360 e 720.

Esempio
L'esempio seguente imposta su 100 la proprietà `lineSpacing` del campo di testo selezionato:

```
fl.getDocumentDOM().selection[0].setTextAttr("lineSpacing", 100);
```

textAttrs.rightMargin

Disponibilità
Flash MX 2004.

Uso
`textAttrs.rightMargin`

Descrizione
Proprietà; un numero intero che specifica il margine destro del paragrafo. I valori accettabili sono quelli compresi tra 0 e 720.

Esempio
L'esempio seguente imposta su 100 la proprietà `rightMargin` per i caratteri compresi tra l'indice 2 e l'indice 8 escluso nel campo di testo selezionato. Questa impostazione può influire sui caratteri che fanno parte dello stesso paragrafo ma non sono compresi nell'intervallo specificato.

```
fl.getDocumentDOM().selection[0].setTextAttr("rightMargin", 100, 2, 8);
```

textAttrs.rotation

Disponibilità
Flash MX 2004.

Uso
`textAttrs.rotation`

Descrizione
Proprietà; un valore booleano. Se specificate `true`, i caratteri del testo vengono ruotati di 90°. Il valore predefinito è `false`. Questa proprietà funziona solo con il testo statico con orientamento verticale e genera un avviso se viene utilizzata con altri tipi di testo.

Esempio
L'esempio seguente imposta su `true` la rotazione del campo di testo selezionato:

```
fl.getDocumentDOM().setElementTextAttr("rotation", true);
```

textAttrs.size

Disponibilità
Flash MX 2004.

Uso
`textAttrs.size`

Descrizione
Proprietà; un numero intero che specifica le dimensioni del carattere.

Esempio
L'esempio seguente recupera la dimensione del carattere in corrispondenza dell'indice 3 e visualizza il risultato nel pannello Output:

```
fl.outputPanel.trace(fl.getDocumentDOM().selection[0].getTextAttr("size", 2));
```

textAttrs.target

Disponibilità
Flash MX 2004.

Uso
`textAttrs.target`

Descrizione
Proprietà; una stringa che rappresenta la proprietà `target` del campo di testo. Questa proprietà funziona solo con il testo statico.

Esempio
L'esempio seguente ottiene la proprietà `target` del campo di testo presente nel primo fotogramma sul primo livello della scena corrente e la visualizza nel pannello Output:

```
fl.outputPanel.trace(fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].getTe
xtAttr("target"));
```

textAttrs.url

Disponibilità
Flash MX 2004.

Uso

```
textAttrs.url
```

Descrizione

Proprietà; una stringa che rappresenta la proprietà `URL` del campo di testo. Questa proprietà funziona solo con il testo statico.

Esempio

L'esempio seguente imposta l'URL del campo di testo selezionato su `http://www.adobe.com`:

```
fl.getDocumentDOM().setElementTextAttr("url", "http://www.adobe.com");
```

Capitolo 46: Oggetto TextRun

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto TextRun rappresenta una sequenza di caratteri che dispongono di attributi che corrispondono a tutte le proprietà dell'Oggetto TextAttrs. Questo oggetto è una proprietà dell'oggetto Text (`text.textRuns`).

Riepilogo delle proprietà
Oltre a quelle dell'oggetto Text, con l'oggetto TextRun è possibile utilizzare le seguenti proprietà:

Proprietà	Descrizione
`textRun.characters`	Una stringa che rappresenta il testo contenuto nell'oggetto TextRun.
`textRun.textAttrs`	L'oggetto TextAttrs che contiene gli attributi della sequenza di testo.

textRun.textAttrs

Disponibilità
Flash MX 2004.

Uso
```
textRun.textAttrs
```

Descrizione
Proprietà; l'Oggetto TextAttrs che contiene gli attributi della sequenza di testo.

Esempio
L'esempio seguente visualizza nel pannello Output le proprietà della prima sequenza di caratteri nel campo di testo selezionato:

```
var curTextAttrs = fl.getDocumentDOM().selection[0].textRuns[0].textAttrs;
for (var prop in curTextAttrs) {
    fl.trace(prop + " = " + curTextAttrs[prop]);
}
```

textRun.characters

Disponibilità
Flash MX 2004.

Uso
```
textRun.characters
```

Descrizione

Proprietà; il testo contenuto nell'oggetto TextRun.

Esempio

L'esempio seguente visualizza nel pannello Output i caratteri che formano la prima sequenza di caratteri nel campo di testo selezionato:

```
fl.trace(fl.getDocumentDOM().selection[0].textRuns[0].characters);
```

Capitolo 47: Oggetto Timeline

Disponibilità

Flash MX 2004.

Descrizione

L'oggetto Timeline rappresenta la linea temporale di Flash, a cui potete accedere mediante `fl.getDocumentDOM().getTimeline()`. Questo metodo restituisce la linea temporale della scena corrente o del simbolo che state modificando.

Quando lavorate con le scene, la linea temporale di ogni scena ha un valore di indice ed è accessibile mediante `fl.getDocumentDOM().timelines[i]`. (In questo esempio, `i` è l'indice del valore della linea temporale).

Quando lavorate con i fotogrammi utilizzando i metodi e le proprietà dell'oggetto Timeline, ricordate che il valore del fotogramma è un indice a base zero (e non il numero di fotogramma nella sequenza di fotogrammi presente nella linea temporale). Quindi, il primo fotogramma ha l'indice di fotogramma 0.

Riepilogo dei metodi

I seguenti metodi sono disponibili per l'oggetto Timeline:

Metodo	Descrizione
`timeline.addMotionGuide()`	Aggiunge un livello guida di movimento sopra il livello corrente e associa il livello corrente al livello guida appena aggiunto.
`timeline.addNewLayer()`	Aggiunge un nuovo livello al documento e lo rende il livello corrente.
`timeline.clearFrames()`	Elimina l'intero contenuto di un fotogramma o di un intervallo di fotogrammi sul livello corrente.
`timeline.clearKeyframes()`	Converte un fotogramma chiave in un fotogramma normale e ne elimina il contenuto sul livello corrente.
`timeline.convertToBlankKeyframes()`	Converte i fotogrammi in fotogrammi chiave vuoti sul livello corrente.
`timeline.convertToKeyframes()`	Converte un intervallo di fotogrammi in fotogrammi chiave (oppure converte la selezione se non sono specificati dei fotogrammi) sul livello corrente.
`timeline.copyFrames()`	Copia negli Appunti un intervallo di fotogrammi del livello corrente.
`timeline.copyMotion()`	Copia il movimento presente sui fotogrammi selezionati, da un'interpolazione di movimento o da un'animazione fotogramma per fotogramma, in modo che possa essere applicato ad altri fotogrammi.
`timeline.copyMotionAsAS3()`	Copia negli Appunti, come codice ActionScript 3.0, il movimento presente sui fotogrammi selezionati, da un'interpolazione di movimento o da un'animazione fotogramma per fotogramma.
`timeline.createMotionTween()`	Imposta la proprietà `frame.tweenType` su `motion` per ogni fotogramma chiave selezionato sul livello corrente e, se necessario, converte il contenuto di ogni fotogramma in un'istanza di un simbolo.
`timeline.cutFrames()`	Taglia un intervallo di fotogrammi sul livello corrente della linea temporale e lo salva negli Appunti.
`timeline.deleteLayer()`	Elimina un livello.

Metodo	Descrizione
`timeline.expandFolder()`	Espande o comprime le cartelle specificate.
`timeline.findLayerIndex()`	Trova un array di indici per i livelli con il nome specificato.
`timeline.getFrameProperty()`	Recupera il valore della proprietà specificato per i fotogrammi selezionati.
`timeline.getGuidelines()`	Restituisce una stringa XML che rappresenta le posizioni attuali delle linee delle guide orizzontali e verticali per una linea temporale (Visualizza > Guide >Mostra guide).
`timeline.getLayerProperty()`	Recupera il valore della proprietà specificato per i livelli selezionati.
`timeline.getSelectedFrames()`	Recupera i fotogrammi selezionati in un array.
`timeline.getSelectedLayers()`	Recupera i valori di indice a base zero dei livelli selezionati.
`timeline.insertBlankKeyframe()`	Inserisce un fotogramma chiave vuoto in corrispondenza dell'indice di fotogramma specificato; se non è stato specificato un indice, inserisce il fotogramma chiave vuoto in corrispondenza dell'indicatore di riproduzione o della selezione.
`timeline.insertFrames()`	Inserisce il numero specificato di fotogrammi in corrispondenza del numero di fotogramma specificato.
`timeline.insertKeyframe()`	Inserisce un fotogramma chiave in corrispondenza del fotogramma specificato.
`timeline.pasteFrames()`	Incolla nei fotogrammi specificati l'intervallo di fotogrammi copiato negli Appunti.
`timeline.pasteMotion()`	Incolla nella linea temporale l'intervallo di fotogrammi di movimento recuperati da `timeline.copyMotion()`.
`timeline.removeFrames()`	Elimina il fotogramma.
`timeline.reorderLayer()`	Sposta il primo livello specificato prima o dopo il secondo livello specificato.
`timeline.reverseFrames()`	Inverte un intervallo di fotogrammi.
`timeline.selectAllFrames()`	Seleziona tutti i fotogrammi presenti nella linea temporale corrente.
`timeline.setFrameProperty()`	Imposta la proprietà dell'oggetto Frame per i fotogrammi selezionati.
`timeline.setGuidelines()`	Sostituisce le linee delle guide della linea temporale con le informazioni specificate.
`timeline.setLayerProperty()`	Imposta sul valore specificato la proprietà specificata su tutti i livelli selezionati.
`timeline.setSelectedFrames()`	Seleziona un intervallo di fotogrammi sul livello corrente oppure imposta i fotogrammi selezionati sul valore dell'array passato in questo metodo.
`timeline.setSelectedLayers()`	Imposta il livello da selezionare; inoltre rende corrente il livello specificato.
`timeline.showLayerMasking()`	Mostra l'effetto maschera del livello durante la fase di creazione, bloccando la maschera e i livelli mascherati.

Riepilogo delle proprietà

Le seguenti proprietà sono disponibili per l'oggetto Timeline:

Proprietà	Descrizione
`timeline.currentFrame`	Un indice a base zero per il fotogramma che si trova nella posizione corrente dell'indicatore di riproduzione.
`timeline.currentLayer`	Un indice a base zero per il livello attivo corrente.
`timeline.frameCount`	Sola lettura; un numero intero che rappresenta il numero di fotogrammi presenti sul livello più lungo della linea temporale.
`timeline.layerCount`	Sola lettura; un numero intero che rappresenta il numero di livelli presenti nella linea temporale specificata.
`timeline.layers`	Sola lettura; un array di oggetti Layer.
`timeline.name`	Una stringa che rappresenta il nome della linea temporale corrente.

timeline.addMotionGuide()

Disponibilità
Flash MX 2004.

Uso
```
timeline.addMotionGuide()
```

Parametri
Nessuno.

Restituisce
Un numero intero che rappresenta l'indice a base zero del livello guida appena aggiunto. Se il livello corrente non è di tipo `"Normale"`, viene restituito il valore -1.

Descrizione
Metodo; aggiunge un livello guida di movimento al di sopra del livello corrente e associa il livello corrente al livello guida appena aggiunto, convertendo il livello corrente in un livello di tipo `"Guidato"`.

Questo metodo funziona solo con i livelli di tipo `"Normale"`. Non ha alcun effetto sui tipi `"Cartella"`, `"Maschera"`, `"Mascherato"`, `"Guida"` o `"Guidato"`.

Esempio
L'esempio seguente aggiunge un livello guida di movimento sopra il livello corrente e converte il livello corrente nel tipo `Guidato`:

```
fl.getDocumentDOM().getTimeline().addMotionGuide();
```

timeline.addNewLayer()

Disponibilità
Flash MX 2004.

Uso

```
timeline.addNewLayer([name] [, layerType [, bAddAbove]])
```

Parametri

`name` Una stringa che specifica il nome del nuovo livello. Se omettete questo parametro, un nuovo nome predefinito viene assegnato al nuovo livello ("Livello n", dove *n* è il numero totale di livelli). Questo parametro è opzionale.

`layerType` Una stringa che specifica il tipo di livello da aggiungere. Se omettete questo parametro, viene creato un livello di tipo "Normale". Questo parametro è opzionale. I valori accettabili sono `"normal"`, `"guide"`, `"guided"`, `"mask"`, `"masked"` e `"folder"`.

`bAddAbove` Un valore booleano che, se impostato su `true` (il valore predefinito), fa in modo che il nuovo livello venga aggiunto sopra quello corrente; se specificate `false`, il livello viene aggiunto sotto quello corrente. Questo parametro è opzionale.

Restituisce

Un valore intero che rappresenta l'indice a base zero del livello appena aggiunto.

Descrizione

Metodo; aggiunge un nuovo livello al documento e lo rende il livello corrente.

Esempio

L'esempio seguente aggiunge un nuovo livello alla linea temporale con un nome predefinito generato automaticamente:

```
fl.getDocumentDOM().getTimeline().addNewLayer();
```

L'esempio seguente aggiunge un nuovo livello di cartella sopra quello corrente e vi assegna il nome `"Folder1"`:

```
fl.getDocumentDOM().getTimeline().addNewLayer("Folder1", "folder", true);
```

timeline.clearFrames()

Disponibilità

Flash MX 2004.

Uso

```
timeline.clearFrames([startFrameIndex [, endFrameIndex]])
```

Parametri

`startFrameIndex` Un numero intero a base zero che definisce l'inizio dell'intervallo di fotogrammi da cancellare. Se omettete *startFrameIndex*, il metodo utilizza la selezione corrente. Questo parametro è opzionale.

`endFrameIndex` Un numero intero a base zero che definisce la fine dell'intervallo di fotogrammi da cancellare. L'intervallo arriva fino a *endFrameIndex* escluso. Se specificate solo *startFrameIndex*, *endFrameIndex* utilizza *startFrameIndex* come valore predefinito. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione
Metodo; elimina l'intero contenuto di un fotogramma o di un intervallo di fotogrammi sul livello corrente.

Esempio
L'esempio seguente cancella i fotogrammi da 6 a 11 escluso (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().clearFrames(5, 10);
```

Il codice seguente cancella il fotogramma 15:

```
fl.getDocumentDOM().getTimeline().clearFrames(14);
```

timeline.clearKeyframes()

Disponibilità
Flash MX 2004.

Uso
```
timeline.clearKeyframes([startFrameIndex [, endFrameIndex]])
```

Parametri
`startFrameIndex` Un numero intero a base zero che definisce l'inizio dell'intervallo di fotogrammi da cancellare. Se omettete *startFrameIndex*, il metodo utilizza la selezione corrente. Questo parametro è opzionale.

`endFrameIndex` Un numero intero a base zero che definisce la fine dell'intervallo di fotogrammi da cancellare. L'intervallo arriva fino a *endFrameIndex* escluso. Se specificate solo *startFrameIndex*, *endFrameIndex* utilizza *startFrameIndex* come valore predefinito. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; converte un fotogramma chiave in un fotogramma normale e ne elimina il contenuto sul livello corrente.

Esempio
L'esempio seguente cancella i fotogrammi chiave presenti nei fotogrammi da 5 a 10 escluso (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().clearKeyframes(4, 9);
```

L'esempio seguente cancella il fotogramma chiave in corrispondenza del fotogramma 15 e lo converte in un fotogramma normale:

```
fl.getDocumentDOM().getTimeline().clearKeyframes(14);
```

timeline.convertToBlankKeyframes()

Disponibilità
Flash MX 2004.

Uso
```
timeline.convertToBlankKeyframes([startFrameIndex [, endFrameIndex]])
```

Parametri
`startFrameIndex` Un indice a base zero che specifica il primo fotogramma da convertire in fotogramma chiave. Se omettete *startFrameIndex*, il metodo converte i fotogrammi selezionati. Questo parametro è opzionale.

`endFrameIndex` Un indice a base zero che specifica il fotogramma in cui termina la conversione in fotogrammi chiave. L'intervallo di fotogrammi da convertire arriva fino a *endFrameIndex* escluso. Se specificate solo *startFrameIndex*, *endFrameIndex* utilizza *startFrameIndex* come valore predefinito. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; converte i fotogrammi in fotogrammi chiave vuoti sul livello corrente.

Esempio
L'esempio seguente converte in fotogrammi chiave vuoti i fotogrammi da 2 a 10 escluso (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().convertToBlankKeyframes(1, 9);
```

L'esempio seguente converte il fotogramma 5 in un fotogramma chiave vuoto:

```
fl.getDocumentDOM().getTimeline().convertToBlankKeyframes(4);
```

timeline.convertToKeyframes()

Disponibilità
Flash MX 2004.

Uso
```
timeline.convertToKeyframes([startFrameIndex [, endFrameIndex]])
```

Parametri
`startFrameIndex` Un indice a base zero che specifica il primo dei fotogrammi da convertire in fotogrammi chiave. Se omettete *startFrameIndex*, il metodo converte i fotogrammi selezionati. Questo parametro è opzionale.

`endFrameIndex` Un indice a base zero che specifica il fotogramma in cui termina la conversione in fotogrammi chiave. L'intervallo di fotogrammi da convertire arriva fino a *endFrameIndex* escluso. Se specificate solo *startFrameIndex*, *endFrameIndex* utilizza *startFrameIndex* come valore predefinito. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; converte un intervallo di fotogrammi in fotogrammi chiave (oppure converte la selezione se non vengono specificati fotogrammi) sul livello corrente.

Esempio
L'esempio seguente converte i fotogrammi selezionati in fotogrammi chiave:

```
fl.getDocumentDOM().getTimeline().convertToKeyframes();
```

L'esempio seguente converte in fotogrammi chiave i fotogrammi da 2 a 10 escluso (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().convertToKeyframes(1, 9);
```

L'esempio seguente converte il fotogramma 5 in un fotogramma chiave:

```
fl.getDocumentDOM().getTimeline().convertToKeyframes(4);
```

timeline.copyFrames()

Disponibilità
Flash MX 2004.

Uso
```
timeline.copyFrames([startFrameIndex [, endFrameIndex]])
```

Parametri
`startFrameIndex` Un numero intero a base zero che definisce l'inizio dell'intervallo di fotogrammi da copiare. Se omettete *startFrameIndex*, il metodo utilizza la selezione corrente. Questo parametro è opzionale.

`endFrameIndex` Un indice a base zero che specifica il fotogramma in cui termina l'operazione di copiatura. L'intervallo di fotogrammi da copiare arriva fino a *endFrameIndex* escluso. Se specificate solo *startFrameIndex*, *endFrameIndex* utilizza *startFrameIndex* come valore predefinito. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; copia negli Appunti un intervallo di fotogrammi presente sul livello corrente.

Esempio
L'esempio seguente copia negli Appunti i fotogrammi selezionati:

```
fl.getDocumentDOM().getTimeline().copyFrames();
```

L'esempio seguente copia negli Appunti i fotogrammi dal 2 al 10 escluso (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().copyFrames(1, 9);
```

L'esempio seguente copia negli Appunti il fotogramma 5:

```
fl.getDocumentDOM().getTimeline().copyFrames(4);
```

timeline.copyMotion()

Disponibilità
Flash CS3 Professional.

Uso

```
timeline.copyMotion()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; copia il movimento sui fotogrammi selezionati da un'interpolazione di movimento o da un'animazione fotogramma per fotogramma. A questo punto potete utilizzare `timeline.pasteMotion()` per applicare il movimento ad altri fotogrammi.

Per copiare il movimento come testo (codice) che possa essere incollato in uno script, consultate `timeline.copyMotionAsAS3()`.

Esempio
L'esempio seguente copia il movimento a partire da uno o più fotogrammi selezionati:

```
fl.getDocumentDOM().getTimeline().copyMotion();
```

Consultate anche
`timeline.copyMotionAsAS3()`, `timeline.pasteMotion()`

timeline.copyMotionAsAS3()

Disponibilità
Flash CS3 Professional.

Uso

```
timeline.copyMotionAsAS3()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; copia negli Appunti, come codice ActionScript 3.0, il movimento presente sui fotogrammi selezionati, da un'interpolazione di movimento o da un'animazione fotogramma per fotogramma. In seguito potete incollare il codice in uno script.

Per copiare il movimento in un formato applicabile ad altri fotogrammi, consultate `timeline.copyMotion()`.

Esempio
L'esempio seguente copia il movimento negli Appunti come codice ActionScript 3.0, a partire da uno o più fotogrammi selezionati:

```
fl.getDocumentDOM().getTimeline().copyMotionAsAS3();
```

Consultate anche
`timeline.copyMotion()`

timeline.createMotionTween()

Disponibilità
Flash MX 2004.

Uso
```
timeline.createMotionTween([startFrameIndex [, endFrameIndex]])
```

Parametri
`startFrameIndex` Un indice a base zero che specifica il fotogramma iniziale di un'interpolazione di movimento. Se omettete *startFrameIndex*, il metodo utilizza la selezione corrente. Questo parametro è opzionale.

`endFrameIndex` Un indice a base zero che specifica il fotogramma in cui termina l'interpolazione di movimento. L'intervallo dei fotogrammi arriva fino a *endFrameIndex* escluso. Se specificate solo *startFrameIndex*, *endFrameIndex* utilizza *startFrameIndex* come valore predefinito. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; imposta la proprietà `frame.tweenType` su `motion` per ogni fotogramma chiave selezionato sul livello corrente e, se necessario, converte il contenuto di ogni fotogramma in un'istanza di un simbolo. Questa proprietà equivale alla voce di menu Crea interpolazione movimento dello strumento di creazione di Flash.

Esempio
L'esempio seguente effettua la conversione di una forma in un'istanza di simbolo grafico dal primo fotogramma al fotogramma 10 escluso e imposta `frame.tweenType` su `motion` (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().createMotionTween(0, 9);
```

timeline.currentFrame

Disponibilità
Flash MX 2004.

Uso
`timeline.currentFrame`

Descrizione
Proprietà; l'indice a base zero per il fotogramma che si trova nella posizione corrente dell'indicatore di riproduzione.

Esempio
L'esempio seguente imposta sul fotogramma 10 l'indicatore di riproduzione della linea temporale corrente (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().currentFrame = 9;
```

L'esempio seguente memorizza il valore della posizione corrente dell'indicatore di riproduzione nella variabile `curFrame`:

```
var curFrame = fl.getDocumentDOM().getTimeline().currentFrame;
```

timeline.currentLayer

Disponibilità
Flash MX 2004.

Uso
`timeline.currentLayer`

Descrizione
Proprietà; un indice a base zero per il livello attivo corrente. Il valore 0 specifica il primo livello, il valore 1 specifica il livello sottostante e così via.

Esempio
L'esempio seguente rende attivo il primo livello:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
```

L'esempio seguente memorizza l'indice del livello attivo nella variabile `curLayer`:

```
var curLayer = fl.getDocumentDOM().getTimeline().currentLayer;
```

timeline.cutFrames()

Disponibilità
Flash MX 2004.

Uso

```
timeline.cutFrames([startFrameIndex [, endFrameIndex]])
```

Parametri

`startFrameIndex` Un numero intero a base zero che specifica l'inizio dell'intervallo di fotogrammi da tagliare. Se omettete *startFrameIndex*, il metodo utilizza la selezione corrente. Questo parametro è opzionale.

`endFrameIndex` Un indice a base zero che specifica il fotogramma in cui termina l'operazione di taglio. L'intervallo dei fotogrammi arriva fino a *endFrameIndex* escluso. Se specificate solo *startFrameIndex*, *endFrameIndex* utilizza *startFrameIndex* come valore predefinito. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione

Metodo; taglia un intervallo di fotogrammi sul livello corrente della linea temporale e lo salva negli Appunti.

Esempio

L'esempio seguente taglia i fotogrammi selezionati dalla linea temporale e li inserisce negli Appunti.

```
fl.getDocumentDOM().getTimeline().cutFrames();
```

L'esempio seguente taglia i fotogrammi da 2 a 10 escluso e li salva negli Appunti (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().cutFrames(1, 9);
```

L'esempio seguente taglia il fotogramma 5 dalla linea temporale e lo inserisce negli Appunti:

```
fl.getDocumentDOM().getTimeline().cutFrames(4);
```

timeline.deleteLayer()

Disponibilità

Flash MX 2004.

Uso

```
timeline.deleteLayer([index])
```

Parametri

`index` Un numero intero a base zero che specifica il livello da eliminare. Se nella linea temporale esiste solo un livello, questo metodo non ha effetto. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione

Metodo; elimina un livello. Se il livello è una cartella, vengono eliminati tutti i livelli contenuti nella cartella. Se non specificate l'indice del livello, i livelli attualmente selezionati vengono eliminati.

Esempio
L'esempio seguente elimina il secondo livello a partire dall'alto:

```
fl.getDocumentDOM().getTimeline().deleteLayer(1);
```

L'esempio seguente elimina i livelli selezionati:

```
fl.getDocumentDOM().getTimeline().deleteLayer();
```

timeline.expandFolder()

Disponibilità
Flash MX 2004.

Uso
```
timeline.expandFolder(bExpand [, bRecurseNestedParents [, index]])
```

Parametri
`bExpand` Un valore booleano in base al quale il metodo espande o comprime la cartella a seconda che il valore sia impostato su `true` o su `false`.

`bRecurseNestedParents` Un valore booleano che, se impostato su `true`, causa l'apertura o la chiusura di tutti i livelli all'interno della cartella specificata, in base al parametro *bExpand*. Questo parametro è opzionale.

`index` Un numero intero a base zero della cartella da espandere o comprimere. Utilizzate -1 per applicarlo a tutti i livelli (dovete inoltre impostare *bRecurseNestedParents* su `true`). Questa proprietà equivale alla voce di menu Espandi tutte le cartelle/Comprimi tutte le cartelle dello strumento di creazione di Flash. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; espande o comprime le cartelle specificate. Se non specificate un livello, il metodo agisce sul livello corrente.

Esempio
Nell'esempio seguente viene usata questa struttura di cartelle:

```
Folder 1 ***
--layer 7
--Folder 2 ****
----Layer 5
```

L'esempio seguente espande solo la cartella 1:

```
fl.getDocumentDOM().getTimeline().currentLayer = 1;
fl.getDocumentDOM().getTimeline().expandFolder(true);
```

L'esempio seguente espande solo la cartella 1 (supponendo che la cartella 2 sia stata compressa quando la cartella 1 è stata compressa l'ultima volta; in caso contrario, la cartella 2 appare espansa):

```
fl.getDocumentDOM().getTimeline().expandFolder(true, false, 0);
```

L'esempio seguente comprime tutte le cartelle della linea temporale corrente:

```
fl.getDocumentDOM().getTimeline().expandFolder(false, true, -1);
```

timeline.findLayerIndex()

Disponibilità
Flash MX 2004.

Uso
`timeline.findLayerIndex(name)`

Parametri
`name` Una stringa che specifica il nome del livello da trovare.

Restituisce
Un array di valori indice per il livello specificato. Se il livello specificato non viene trovato, viene restituito il valore `undefined`.

Descrizione
Metodo; trova un array di indici per i livelli con il nome specificato. L'indice del livello non è strutturato, pertanto le cartelle vengono considerate parte dell'indice principale.

Esempio
L'esempio seguente mostra i valori di indice di tutti i livelli denominati `Layer 7` nel pannello Output:

```
var layerIndex = fl.getDocumentDOM().getTimeline().findLayerIndex("Layer 7");
fl.trace(layerIndex);
```

L'esempio seguente illustra come passare nuovamente a `timeline.setSelectedLayers()` i valori restituiti da questo metodo:

```
var layerIndex = fl.getDocumentDOM().getTimeline().findLayerIndex("Layer 1");
fl.getDocumentDOM().getTimeline().setSelectedLayers(layerIndex[0], true);
```

timeline.frameCount

Disponibilità
Flash MX 2004.

Uso
`timeline.frameCount`

Descrizione
Proprietà di sola lettura; un numero intero che rappresenta il numero di fotogrammi presenti sul livello più lungo della linea temporale.

Esempio

L'esempio seguente utilizza una variabile `countNum` per memorizzare il numero di fotogrammi presenti sul livello più lungo del documento corrente:

```
var countNum = fl.getDocumentDOM().getTimeline().frameCount;
```

timeline.getFrameProperty()

Disponibilità

Flash MX 2004.

Uso

```
timeline.getFrameProperty(property [, startframeIndex [, endFrameIndex]])
```

Parametri

`property` Una stringa che specifica il nome della proprietà di cui si deve ottenere il valore. Consultate il riepilogo delle proprietà relativo all'Oggetto Frame per un elenco completo delle proprietà.

`startFrameIndex` Un indice a base zero che specifica il numero del fotogramma iniziale di cui deve essere ottenuto il valore. Se omettete *startFrameIndex*, il metodo utilizza la selezione corrente. Questo parametro è opzionale.

`endFrameIndex` Un numero intero a base zero che definisce la fine dell'intervallo di fotogrammi da selezionare. L'intervallo arriva fino a *endFrameIndex* escluso. Se specificate solo *startFrameIndex*, *endFrameIndex* utilizza *startFrameIndex* come valore predefinito. Questo parametro è opzionale.

Restituisce

Un valore per la proprietà specificata, oppure `undefined` se tutti i fotogrammi selezionati non hanno lo stesso valore per la proprietà.

Descrizione

Metodo; recupera il valore specificato della proprietà per i fotogrammi selezionati.

Esempio

L'esempio seguente recupera il nome del primo fotogramma presente nel primo livello del documento corrente e ne visualizza il nome nel pannello Output:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
fl.getDocumentDOM().getTimeline().setSelectedFrames(0, 0, true);
var frameName = fl.getDocumentDOM().getTimeline().getFrameProperty("name");
fl.trace(frameName);
```

timeline.getGuidelines()

Disponibilità

Flash CS4 Professional.

Uso

```
timeline.getGuidelines()
```

Parametri
Nessuno.

Restituisce
Una stringa XML.

Descrizione
Metodo: restituisce una stringa XML che rappresenta le posizioni correnti delle linee delle guide orizzontali e verticali per una linea temporale (Visualizza > Guide >Mostra guide). Per applicare queste linee delle guide a una linea temporale, utilizzare `timeline.setGuidelines()`.

Esempio
Se nella prima linea temporale sono presenti alcune linee delle guide, l'esempio seguente le visualizza come una stringa XML nel pannello Output:

```
var currentTimeline = fl.getDocumentDOM().timelines[0];
fl.trace(currentTimeline.getGuidelines());
```

timeline.getLayerProperty()

Disponibilità
Flash MX 2004.

Uso
`timeline.getLayerProperty(property)`

Parametri
`property` Una stringa che specifica il nome della proprietà di cui si deve recuperare il valore. Per un elenco delle proprietà, consultate il riepilogo delle proprietà per l'Oggetto Frame.

Restituisce
Il valore della proprietà specificata. Per determinare il tipo, vengono verificate le proprietà del livello. Se tutti i livelli specificati non hanno lo stesso valore della proprietà, viene restituito il valore `undefined`.

Descrizione
Metodo; recupera il valore specificato della proprietà per i livelli selezionati.

Esempio
L'esempio seguente recupera il nome del primo livello del documento corrente e lo visualizza nel pannello Output:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
var layerName = fl.getDocumentDOM().getTimeline().getLayerProperty("name");
fl.trace(layerName);
```

timeline.getSelectedFrames()

Disponibilità
Flash MX 2004.

Parametri
Nessuno.

Restituisce
Un array che contiene 3*n* numeri interi, dove *n* è il numero di aree selezionate. Il primo numero intero di ogni gruppo è l'indice del livello, il secondo è il fotogramma iniziale della selezione e il terzo è il fotogramma finale dell'intervallo della selezione. Il fotogramma finale non è compreso nella selezione.

Descrizione
Metodo; recupera i fotogrammi selezionati in un array.

Esempio
L'esempio seguente considera il primo livello come corrente e visualizza `0,5,10,0,20,25` nel pannello Output:

```
var timeline = fl.getDocumentDOM().getTimeline();
timeline.setSelectedFrames(5,10);
timeline.setSelectedFrames(20,25,false);
var theSelectedFrames = timeline.getSelectedFrames();
fl.trace(theSelectedFrames);
```

Consultate anche
`timeline.setSelectedFrames()`

timeline.getSelectedLayers()

Disponibilità
Flash MX 2004.

Parametri
Nessuno.

Restituisce
Un array dei valori di indice a base zero dei livelli selezionati.

Descrizione
Metodo; recupera i valori di indice a base zero dei livelli selezionati.

Esempio
L'esempio seguente visualizza `1,0` nel pannello Output:

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(0);
fl.getDocumentDOM().getTimeline().setSelectedLayers(1, false);
var layerArray = fl.getDocumentDOM().getTimeline().getSelectedLayers();
fl.trace(layerArray);
```

Consultate anche

`timeline.setSelectedLayers()`

timeline.insertBlankKeyframe()

Disponibilità

Flash MX 2004.

Uso

```
timeline.insertBlankKeyframe([frameNumIndex])
```

Parametri

`frameNumIndex` Un indice a base zero che specifica il fotogramma in cui inserire il fotogramma chiave. Se omettete *frameNumIndex*, il metodo utilizza il numero corrente dell'indicatore di riproduzione. Questo parametro è opzionale.

Se il fotogramma specificato o selezionato è un fotogramma standard, il fotogramma chiave viene inserito in tale fotogramma. Ad esempio, se è presente una sequenza di 10 fotogrammi numerati da 1 a 10 e selezionate il fotogramma 5, questo metodo trasforma il fotogramma 5 in un fotogramma chiave vuoto e la lunghezza della sequenza di fotogrammi rimane di 10 fotogrammi. Se il fotogramma 5 è selezionato ed è un fotogramma chiave adiacente a un fotogramma standard, questo metodo inserisce un fotogramma chiave vuoto in corrispondenza del fotogramma 6. Se il fotogramma 5 è un fotogramma chiave e il fotogramma adiacente è già un fotogramma chiave, non viene inserito alcun fotogramma ma l'indicatore di riproduzione si sposta sul fotogramma 6.

Restituisce

Nulla.

Descrizione

Metodo; inserisce un fotogramma chiave vuoto in corrispondenza dell'indice di fotogramma specificato; se non è stato specificato un indice, inserisce il fotogramma chiave vuoto in corrispondenza dell'indicatore di riproduzione o della selezione. Consultate anche `timeline.insertKeyframe()`.

Esempio

L'esempio seguente inserisce un fotogramma chiave vuoto nel fotogramma 20 (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().insertBlankKeyframe(19);
```

L'esempio seguente inserisce un fotogramma chiave vuoto nel fotogramma attualmente selezionato (o in corrispondenza dell'indicatore di riproduzione, se non è selezionato alcun fotogramma):

```
fl.getDocumentDOM().getTimeline().insertBlankKeyframe();
```

timeline.insertFrames()

Disponibilità
Flash MX 2004.

Uso
```
timeline.insertFrames([numFrames [, bAllLayers [, frameNumIndex]]])
```

Parametri
`numFrames` Un numero intero che specifica il numero di fotogrammi da inserire. Se omettete questo parametro, il metodo inserisce i fotogrammi sul livello corrente in corrispondenza della selezione corrente. Questo parametro è opzionale.

`bAllLayers` Un valore booleano che, se impostato su `true` (il valore predefinito), fa in modo che il metodo inserisca in tutti i livelli il numero specificato di fotogrammi nel parametro *numFrames*; se è impostato su `false`, il metodo inserisce i fotogrammi sul livello corrente. Questo parametro è opzionale.

`frameNumIndex` Un indice a base zero che specifica il fotogramma in cui inserire un nuovo fotogramma. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; inserisce il numero specificato di fotogrammi in corrispondenza dell'indice specificato.

Se non viene specificato alcun parametro, il metodo agisce nel modo seguente:

- Se sono selezionati uno o più fotogrammi, il metodo inserisce il numero selezionato di fotogrammi nel primo fotogramma selezionato sul livello corrente. In altre parole, se sono selezionati i fotogrammi da 6 a 10 (per un totale di cinque fotogrammi), il metodo aggiunge cinque fotogrammi nel fotogramma 6 sul livello che contiene i fotogrammi selezionati.
- Se non è selezionato alcun fotogramma, il metodo inserisce un solo fotogramma su tutti i livelli in corrispondenza del fotogramma corrente.

  Se vengono specificati dei parametri, il metodo agisce nel modo seguente:

- Se è specificato solo *numFrames*, inserisce sul livello corrente il numero specificato di fotogrammi in corrispondenza del fotogramma corrente.
- Se *numFrames* è specificato e *bAllLayers* è `true`, inserisce su tutti i livelli il numero specificato di fotogrammi in corrispondenza del fotogramma corrente.
- Se sono specificati tutti e tre i parametri, inserisce il numero specificato di fotogrammi in corrispondenza dell'indice specificato (*frameIndex*); il valore passato per *bAllLayers* determina se i fotogrammi vengono aggiunti solo al livello corrente oppure a tutti i livelli.

  Se il fotogramma specificato o selezionato è di tipo standard, il fotogramma viene inserito in corrispondenza di tale fotogramma. Ad esempio, se è presente una sequenza di 10 fotogrammi numerati da 1 a 10 e selezionate il fotogramma 5 (o passate il valore 4 per *frameIndex*), questo metodo aggiunge un fotogramma in corrispondenza del fotogramma 5 e la lunghezza della sequenza di fotogrammi diventa di 11 fotogrammi. Se il fotogramma 5 è selezionato e si tratta di un fotogramma chiave, questo metodo inserisce un fotogramma in corrispondenza del fotogramma 6 anche se il fotogramma adiacente è un fotogramma chiave.

Esempio
L'esempio seguente inserisce sul livello corrente uno o più fotogrammi (a seconda della selezione) in corrispondenza della selezione corrente:

```
fl.getDocumentDOM().getTimeline().insertFrames();
```

L'esempio seguente inserisce su tutti i livelli cinque fotogrammi in corrispondenza del fotogramma corrente:

```
fl.getDocumentDOM().getTimeline().insertFrames(5);
```

Nota: *se sono presenti più livelli contenenti fotogrammi e selezionate un fotogramma su un livello mediante il comando precedente, i fotogrammi vengono inseriti solo sul livello selezionato. Se sono presenti più livelli ma in nessuno di essi sono selezionati dei fotogrammi, i fotogrammi vengono inseriti su tutti i livelli.*

L'esempio seguente inserisce tre fotogrammi solo sul livello corrente:

```
fl.getDocumentDOM().getTimeline().insertFrames(3, false);
```

L'esempio seguente inserisce quattro fotogrammi su tutti i livelli, a partire dal primo fotogramma:

```
fl.getDocumentDOM().getTimeline().insertFrames(4, true, 0);
```

timeline.insertKeyframe()

Disponibilità
Flash MX 2004.

Uso
```
timeline.insertKeyframe([frameNumIndex])
```

Parametri
`frameNumIndex` Un indice a base zero che specifica l'indice di fotogramma in cui inserire il fotogramma chiave sul livello corrente. Se omettete *frameNumIndex*, il metodo utilizza il numero del fotogramma corrente o dell'indicatore di riproduzione. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; inserisce un fotogramma chiave in corrispondenza del fotogramma specificato. Se omettete questo parametro, il metodo inserisce un fotogramma chiave nella posizione dell'indicatore di riproduzione o della selezione corrente.

Questo metodo funziona in modo simile a `timeline.insertBlankKeyframe()`, con la differenza che nel fotogramma chiave è presente il contenuto del fotogramma che è stato convertito (in altre parole, non è vuoto).

Esempio
L'esempio seguente inserisce un fotogramma chiave in corrispondenza dell'indicatore di riproduzione o della posizione selezionata:

```
fl.getDocumentDOM().getTimeline().insertKeyframe();
```

L'esempio seguente inserisce un fotogramma chiave vuoto nel fotogramma 10 (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().currentLayer = 1;
fl.getDocumentDOM().getTimeline().insertKeyframe(9);
```

timeline.layerCount

Disponibilità
Flash MX 2004.

Uso
`timeline.layerCount`

Descrizione
Proprietà di sola lettura; un numero intero che rappresenta il numero di livelli presenti nella linea temporale specificata.

Esempio
L'esempio seguente ottiene la variabile `NumLayer` per memorizzare il numero di livelli presenti nella scena corrente:

```
var NumLayer = fl.getDocumentDOM().getTimeline().layerCount;
```

timeline.layers

Disponibilità
Flash MX 2004.

Uso
`timeline.layers`

Descrizione
Proprietà di sola lettura; un array di oggetti layer.

Esempio
L'esempio seguente ottiene la variabile `currentLayers` per memorizzare l'array di oggetti layer presenti nel documento corrente:

```
var currentLayers = fl.getDocumentDOM().getTimeline().layers;
```

timeline.name

Disponibilità
Flash MX 2004.

Uso

```
timeline.name
```

Descrizione

Proprietà; una stringa che specifica il nome della linea temporale corrente. Si tratta del nome della scena, della schermata (diapositiva o form) o del simbolo che state modificando.

Esempio

L'esempio seguente recupera il nome della prima scena:

```
var sceneName = fl.getDocumentDOM().timelines[0].name;
```

L'esempio seguente imposta `FirstScene` come nome della prima scena:

```
fl.getDocumentDOM().timelines[0].name = "FirstScene";
```

timeline.pasteFrames()

Disponibilità

Flash MX 2004.

Uso

```
timeline.pasteFrames([startFrameIndex [, endFrameIndex]])
```

Parametri

`startFrameIndex` Un numero intero a base zero che specifica l'inizio dell'intervallo di fotogrammi da incollare. Se omettete *startFrameIndex*, il metodo utilizza la selezione corrente. Questo parametro è opzionale.

`endFrameIndex` Un indice a base zero che specifica il fotogramma in cui si termina di incollare. L'intervallo incolla gli elementi fino a *endFrameIndex* escluso. Se specificate solo *startFrameIndex*, *endFrameIndex* utilizza *startFrameIndex* come valore predefinito. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione

Metodo; incolla nei fotogrammi specificati l'intervallo di fotogrammi copiato negli Appunti.

Esempio

L'esempio seguente incolla i fotogrammi salvati negli Appunti nel fotogramma selezionato o in corrispondenza dell'indicatore di riproduzione:

```
fl.getDocumentDOM().getTimeline().pasteFrames();
```

L'esempio seguente incolla i fotogrammi salvati negli Appunti nei fotogrammi da 2 a 10 escluso (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().pasteFrames(1, 9);
```

L'esempio seguente incolla i fotogrammi salvati negli Appunti a partire dal fotogramma 5:

```
fl.getDocumentDOM().getTimeline().pasteFrames(4);
```

timeline.pasteMotion()

Disponibilità
Flash CS3 Professional.

Uso
```
timeline.pasteMotion()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; incolla nella linea temporale l'intervallo di fotogrammi di movimento recuperati da `timeline.copyMotion()`. Se necessario, i fotogrammi esistenti vengono spostati verso destra per creare spazio per i fotogrammi che vengono incollati.

Esempio
L'esempio seguente incolla il movimento dagli Appunti al fotogramma attualmente selezionato o nella posizione dell'indicatore di riproduzione, spostando il fotogramma a destra di quelli che vengono incollati:

```
fl.getDocumentDOM().getTimeline().pasteMotion();
```

Consultate anche
`timeline.copyMotion()`

timeline.removeFrames()

Disponibilità
Flash MX 2004.

Uso
```
timeline.removeFrames([startFrameIndex [, endFrameIndex]])
```

Parametri
`startFrameIndex` Un indice a base zero che specifica il fotogramma da cui inizia la rimozione dei fotogrammi. Se omettete *startFrameIndex*, il metodo utilizza la selezione corrente; se non è presente alcuna selezione, vengono rimossi tutti i fotogrammi su tutti i livelli in corrispondenza dell'indicatore di riproduzione. Questo parametro è opzionale.

`endFrameIndex` Un indice a base zero che specifica il fotogramma in cui termina la rimozione dei fotogrammi; l'intervallo di fotogrammi arriva fino a *endFrameIndex* escluso. Se specificate solo *startFrameIndex*, *endFrameIndex* utilizza *startFrameIndex* come valore predefinito. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; elimina il fotogramma.

Esempio
L'esempio seguente elimina i fotogrammi da 5 a 10 escluso sul primo livello della scena corrente (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
fl.getDocumentDOM().getTimeline().removeFrames(4, 9);
```

L'esempio seguente elimina il fotogramma 8 sul primo livello della scena corrente:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
fl.getDocumentDOM().getTimeline().removeFrames(7);
```

timeline.reorderLayer()

Disponibilità
Flash MX 2004.

Uso
```
timeline.reorderLayer(layerToMove, layerToPutItBy [, bAddBefore])
```

Parametri
`layerToMove` Un numero intero a base zero che specifica il livello da spostare.

`layerToPutItBy` Un numero intero a base zero che specifica il livello accanto al quale deve essere spostato il livello. Ad esempio, se specificate 1 per *layerToMove* e 0 per *layerToPutItBy*, il secondo livello viene posizionato accanto al primo.

`bAddBefore` Specifica se il livello deve essere spostato prima o dopo *layerToPutItBy*. Se specificate `false`, il livello viene spostato dopo *layerToPutItBy*. Il valore predefinito è `true`. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; sposta il primo livello specificato prima o dopo il secondo livello specificato.

Esempio
L'esempio seguente sposta il livello in corrispondenza dell'indice 2 verso l'alto (sopra il livello in corrispondenza dell'indice 0):

```
fl.getDocumentDOM().getTimeline().reorderLayer(2, 0);
```

L'esempio seguente colloca il livello nella posizione di indice 3 dopo il livello in corrispondenza dell'indice 5:

```
fl.getDocumentDOM().getTimeline().reorderLayer(3, 5, false);
```

timeline.reverseFrames()

Disponibilità
Flash MX 2004.

Uso
```
timeline.reverseFrames([startFrameIndex [, endFrameIndex]])
```

Parametri
`startFrameIndex` Un indice a base zero che specifica il fotogramma da cui inizia l'inversione dei fotogrammi. Se omettete *startFrameIndex*, il metodo utilizza la selezione corrente. Questo parametro è opzionale.

`endFrameIndex` Un indice a base zero che specifica il fotogramma in cui termina l'inversione dei fotogrammi; l'intervallo di fotogrammi arriva fino a *endFrameIndex* escluso. Se specificate solo *startFrameIndex*, *endFrameIndex* utilizza *startFrameIndex* come valore predefinito. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; inverte un intervallo di fotogrammi.

Esempio
L'esempio seguente inverte le posizioni dei fotogrammi selezionati:

```
fl.getDocumentDOM().getTimeline().reverseFrames();
```

L'esempio seguente inverte i fotogrammi da 10 a 15 escluso (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().reverseFrames(9, 14);
```

timeline.selectAllFrames()

Disponibilità
Flash MX 2004.

Uso
```
timeline.selectAllFrames()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; seleziona tutti i fotogrammi presenti nella linea temporale corrente:

Esempio

L'esempio seguente seleziona tutti i fotogrammi presenti nella linea temporale corrente:

```
fl.getDocumentDOM().getTimeline().selectAllFrames();
```

timeline.setFrameProperty()

Disponibilità

Flash MX 2004.

Uso

```
timeline.setFrameProperty(property, value [, startFrameIndex [, endFrameIndex]])
```

Parametri

`property` Una stringa che specifica il nome della proprietà da modificare. Per un elenco completo delle proprietà e dei valori, consultate il riepilogo delle proprietà per l'Oggetto Frame.

Non potete utilizzare questo metodo per impostare i valori delle proprietà di sola lettura, ad esempio `frame.duration` e `frame.elements`.

`value` Specifica il valore su cui impostare la proprietà. Per determinare i valori e il tipo appropriati, consultate il riepilogo delle proprietà per l'Oggetto Frame.

`startFrameIndex` Un indice a base zero che specifica il numero del fotogramma iniziale da modificare. Se omettete *startFrameIndex*, il metodo utilizza la selezione corrente. Questo parametro è opzionale.

`endFrameIndex` Un indice a base zero che specifica il primo fotogramma in corrispondenza del quale interrompere l'operazione. L'intervallo dei fotogrammi arriva fino a *endFrameIndex* escluso. Se specificate solo *startFrameIndex* ma omettete *endFrameIndex*, *endFrameIndex* utilizza *startFrameIndex* come valore predefinito. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione

Metodo; imposta la proprietà dell'oggetto Frame per i fotogrammi selezionati.

Esempio

L'esempio seguente assegna il comando ActionScript `stop()` al primo fotogramma sul primo livello del documento corrente:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
fl.getDocumentDOM().getTimeline().setSelectedFrames(0,0,true);
fl.getDocumentDOM().getTimeline().setFrameProperty("actionScript", "stop();");
```

L'esempio seguente imposta un'interpolazione di movimento dal fotogramma 2 al fotogramma 5 escluso (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
var doc = fl.getDocumentDOM();
doc.getTimeline().setFrameProperty("tweenType","motion",1,4);
```

timeline.setGuidelines()

Disponibilità
Flash CS4 Professional.

Uso
```
timeline.setGuidelines(xmlString)
```

Parametri
`xmlString` Una stringa XML che contiene informazioni sulle linee delle guide da applicare.

Restituisce
Il valore booleano `true` se le guide vengono applicate correttamente; `false` in caso contrario.

Descrizione
Metodo: sostituisce le linee delle guide della linea temporale (Visualizza > Guide > Mostra guide) con le informazioni specificate in *xmlString*. Per recuperare una stringa XML che può essere passata a questo metodo, utilizzate `timeline.getGuidelines()`.

Per visualizzare le linee delle guide appena impostate, potrebbe essere necessario nasconderle e quindi visualizzarle di nuovo.

Esempio
L'esempio seguente applica le linee delle guide da un file FLA a un altro file FLA:

```
var doc0 = fl.documents[0];
var guides0 = doc0.timelines[0].getGuidelines();
var doc1 = fl.documents[1];
doc1.timelines[0].setGuidelines(guides0);
```

timeline.setLayerProperty()

Disponibilità
Flash MX 2004.

Uso
```
timeline.setLayerProperty(property, value [, layersToChange])
```

Parametri
`property` Una stringa che specifica la proprietà da impostare. Per un elenco delle proprietà, consultate "Oggetto Layer" a pagina 314.

`value` Il valore su cui impostare la proprietà. Utilizzate lo stesso tipo di valore utilizzato normalmente per impostare la proprietà nell'oggetto Layer.

`layersToChange` Una stringa che identifica i livelli da modificare. I valori accettabili sono `"selected"`, `"all"` e `"others"`. Se omettete questo parametro, il valore predefinito è `"selected"`. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; imposta su un valore specifico la proprietà specificata per tutti i livelli selezionati.

Esempio
L'esempio seguente rende invisibili i livelli selezionati:

```
fl.getDocumentDOM().getTimeline().setLayerProperty("visible", false);
```

L'esempio seguente imposta su `selLayer` il nome dei livelli selezionati:

```
fl.getDocumentDOM().getTimeline().setLayerProperty("name", "selLayer");
```

timeline.setSelectedFrames()

Disponibilità
Flash MX 2004.

Uso

```
timeline.setSelectedFrames(startFrameIndex, endFrameIndex [, bReplaceCurrentSelection])
timeline.setSelectedFrames(selectionList [, bReplaceCurrentSelection])
```

Parametri
`startFrameIndex` Un indice a base zero che specifica il fotogramma iniziale da impostare.

`endFrameIndex` Un numero intero a base zero che definisce la fine della selezione; *endFrameIndex* corrisponde al fotogramma successivo all'ultimo fotogramma nell'intervallo da selezionare.

`bReplaceCurrentSelection` Un valore booleano che, se impostato su `true`, deseleziona i fotogrammi selezionati prima che vengano selezionati i fotogrammi specificati. Il valore predefinito è `true`.

`selectionList` Un array di tre numeri interi, restituito da `timeline.getSelectedFrames()`.

Restituisce
Nulla.

Descrizione
Metodo; seleziona un intervallo di fotogrammi sul livello corrente oppure imposta i fotogrammi selezionati sul valore dell'array passato in questo metodo.

Esempio
L'esempio seguente illustra due modi per selezionare, nel primo livello, i fotogrammi da 1 a 10 escluso e quindi aggiungere alla selezione corrente sullo stesso livello i fotogrammi da 12 a 15 escluso (ricordate che i valori di indice sono diversi dai valori dei numeri di fotogramma):

```
fl.getDocumentDOM().getTimeline().setSelectedFrames(0, 9);
fl.getDocumentDOM().getTimeline().setSelectedFrames(11, 14, false);
fl.getDocumentDOM().getTimeline().setSelectedFrames([0, 0, 9]);
fl.getDocumentDOM().getTimeline().setSelectedFrames([0, 11, 14], false);
```

L'esempio seguente memorizza l'array dei fotogrammi selezionati nella variabile `savedSelectionList`, quindi utilizza l'array nel codice per selezionare nuovamente tali fotogrammi dopo che la selezione è stata modificata da un comando o dall'intervento dell'utente:

```
var savedSelectionList = fl.getDocumentDOM().getTimeline().getSelectedFrames();
// Do something that changes the selection.
fl.getDocumentDOM().getTimeline().setSelectedFrames(savedSelectionList);
```

Consultate anche

`timeline.getSelectedFrames()`

timeline.setSelectedLayers()

Disponibilità

Flash MX 2004.

Uso

```
timeline.setSelectedLayers(index [, bReplaceCurrentSelection])
```

Parametri

`index` Un numero intero a base zero che specifica il livello da selezionare.

`bReplaceCurrentSelection` Un valore booleano che, se impostato su `true`, determina la sostituzione della selezione corrente; se è impostato su `false`, il metodo estende la selezione. Il valore predefinito è `true`. Questo parametro è opzionale.

Restituisce

Nulla.

Descrizione

Metodo; imposta il livello da selezionare; inoltre rende corrente il livello specificato. Quando si seleziona un livello vengono selezionati anche tutti i fotogrammi che contiene.

Esempio

L'esempio seguente seleziona il primo livello:

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(0);
```

L'esempio seguente aggiunge il livello successivo alla selezione:

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(1, false);
```

Consultate anche

`timeline.getSelectedLayers()`

timeline.showLayerMasking()

Disponibilità
Flash MX 2004.

Uso
```
timeline.showLayerMasking([layer])
```

Parametri
`layer` Un indice a base zero di una maschera o di un livello mascherato che mostra l'effetto maschera durante la creazione. Questo parametro è opzionale.

Restituisce
Nulla.

Descrizione
Metodo; mostra l'effetto maschera del livello durante la fase di creazione bloccando la maschera e i livelli mascherati. Se non è specificato alcun livello, il metodo utilizza il livello corrente. Se utilizzate questo metodo su un livello che non è di tipo Maschera o Mascherato, nel pannello Output viene visualizzato un errore.

Esempio
L'esempio seguente specifica che l'effetto maschera applicato al primo livello deve essere visibile durante la creazione:

```
fl.getDocumentDOM().getTimeline().showLayerMasking(0);
```

Capitolo 48: Oggetto ToolObj

Disponibilità

Flash MX 2004.

Descrizione

L'oggetto ToolObj rappresenta un singolo strumento nel pannello Strumenti. Per accedere all'oggetto ToolObj, utilizzate le proprietà dell'Oggetto Tools: l'array `tools.toolObjs` o `tools.activeTool`.

Riepilogo dei metodi

I seguenti metodi sono disponibili per l'oggetto ToolObj.

Nota: *i metodi seguenti vengono utilizzati solo durante la creazione di oggetti estensibili.*

Metodo	Descrizione
`toolObj.enablePIControl()`	Abilita o disabilita il controllo specificato in una finestra di ispezione Proprietà. Utilizzato solo quando si creano strumenti estensibili.
`toolObj.setIcon()`	Identifica un file PNG da utilizzare come icona di uno strumento nel pannello Strumenti di Flash.
`toolObj.setMenuString()`	Imposta come nome dello strumento la stringa visualizzata nel menu a comparsa.
`toolObj.setOptionsFile()`	Associa un file XML allo strumento.
`toolObj.setPI()`	Imposta una finestra di ispezione Proprietà particolare da utilizzare quando viene attivato lo strumento.
`toolObj.setToolName()`	Assegna un nome allo strumento per la configurazione del pannello Strumenti.
`toolObj.setToolTip()`	Imposta la descrizione comando visualizzata quando si posiziona il mouse sopra l'icona dello strumento.
`toolObj.showPIControl()`	Mostra o nasconde un controllo nella finestra di ispezione Proprietà.
`toolObj.showTransformHandles()`	Viene chiamato nel metodo `configureTool()` di un file JavaScript di uno strumento estensibile per indicare che le maniglie di trasformazione libera devono essere visualizzate quando lo strumento è attivo.

Riepilogo delle proprietà

Le seguenti proprietà sono disponibili per l'oggetto ToolObj:

Proprietà	Descrizione
`toolObj.depth`	Un numero intero che specifica la profondità dello strumento nel menu a comparsa del pannello Strumenti.
`toolObj.iconID`	Un numero intero che specifica l'ID risorsa dello strumento.
`toolObj.position`	Di sola lettura; un numero intero che specifica la posizione dello strumento nel pannello Strumenti.

toolObj.depth

Disponibilità
Flash MX 2004.

Uso
```
toolObj.depth
```

Descrizione
Proprietà di sola lettura; un numero intero che specifica la profondità dello strumento nel menu a comparsa del pannello Strumenti. Si utilizza questa proprietà solo per la creazione di oggetti estensibili.

Esempio
L'esempio seguente specifica che lo strumento ha una profondità pari a 1, ovvero si trova un livello sotto a uno strumento del pannello Strumenti:

```
fl.tools.activeTool.depth = 1;
```

toolObj.enablePIControl()

Disponibilità
Flash MX 2004.

Uso
```
toolObj.enablePIControl(control, bEnable)
```

Parametri
`control` Una stringa che specifica il nome del controllo da abilitare o disabilitare. I valori validi dipendono dalla finestra di ispezione Proprietà richiamata dallo strumento; consultate toolObj.setPI().

La finestra di ispezione Proprietà di una forma contiene i seguenti controlli:

stroke	fill

La finestra di ispezione Proprietà di un testo contiene i seguenti controlli:

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

La finestra di ispezione Proprietà di un filmato contiene i seguenti controlli:

size	publish	background
framerate	player	profile

`bEnable` Un valore booleano che determina se abilitare (`true`) o disabilitare (`false`) il controllo.

Restituisce
Nulla.

Descrizione
Metodo; abilita o disabilita il controllo specificato in una finestra di ispezione Proprietà. Utilizzato solo quando si creano strumenti estensibili.

Esempio
Il comando seguente presente in un file JavaScript di uno strumento estensibile specifica che nella finestra di ispezione Proprietà dello strumento non devono essere visualizzate le opzioni del tratto:

```
theTool.enablePIControl("stroke",false);
```

toolObj.iconID

Disponibilità
Flash MX 2004.

Uso
`toolObj.iconID`

Descrizione
Proprietà di sola lettura; un numero intero con un valore -1. Questa proprietà viene usata solo per creare strumenti estensibili. Un valore `iconID` pari a -1 indica che non verrà eseguito un tentativo di trovare un'icona per lo strumento. Al contrario, l'icona da visualizzare nel pannello Strumenti deve essere specificata nello script dello strumento. Consultate `toolObj.setIcon()`.

Esempio
L'esempio seguente assegna un valore pari a -1 (l'ID dell'icona dello strumento corrente) alla variabile `toolIconID`:

```
var toolIconID = fl.tools.activeTool.iconID
```

toolObj.position

Disponibilità
Flash MX 2004.

Uso
`toolObj.position`

Descrizione
Proprietà di sola lettura; un numero intero che specifica la posizione dello strumento nel pannello Strumenti. Si utilizza questa proprietà solo per la creazione di oggetti estensibili.

Esempio
I comandi seguenti presenti nel metodo `mouseDown()` di un file JavaScript di uno strumento visualizzano la posizione dello strumento nel pannello Strumenti sotto forma di numero intero nel pannello Output:

```
myToolPos = fl.tools.activeTool.position;
fl.trace(myToolPos);
```

toolObj.setIcon()

Disponibilità
Flash MX 2004.

Uso
`toolObj.setIcon(file)`

Parametri
`file` Una stringa che specifica il nome del file PNG da utilizzare come icona. Il file PNG deve essere posizionato nella stessa cartella del file JSFL.

Restituisce
Nulla.

Descrizione
Metodo; identifica un file PNG da utilizzare come icona di uno strumento nel pannello Strumenti. Si utilizza questo metodo solo per la creazione di oggetti estensibili.

Esempio
L'esempio seguente specifica che l'immagine nel file PolyStar.png deve essere utilizzata come icona dello strumento `PolyStar`. Il seguente codice è tratto dal file PolyStar.jsfl (consultate "Strumento di esempio PolyStar" a pagina 14):

```
theTool = fl.tools.activeTool;
theTool.setIcon("PolyStar.png");
```

toolObj.setMenuString()

Disponibilità
Flash MX 2004.

Uso
```
toolObj.setMenuString(menuStr)
```

Parametri
`menuStr` Una stringa che specifica come nome dello strumento il nome visualizzato nel menu a comparsa.

Restituisce
Nulla.

Descrizione
Metodo; imposta come nome dello strumento la stringa visualizzata nel menu a comparsa. Si utilizza questo metodo solo per la creazione di oggetti estensibili.

Esempio
L'esempio seguente specifica che per lo strumento `theTool` deve essere visualizzato il nome "PolyStarTool" nel relativo menu a comparsa. Il seguente codice è tratto dal file PolyStar.jsfl (consultate "Strumento di esempio PolyStar" a pagina 14):

```
theTool = fl.tools.activeTool;
theTool.setMenuString("PolyStar Tool");
```

toolObj.setOptionsFile()

Disponibilità
Flash MX 2004.

Uso
```
toolObj.setOptionsFile(xmlFile)
```

Parametri
`xmlFile` Una stringa che specifica il nome del file XML che contiene la descrizione delle opzioni dello strumento. Il file XML deve essere posizionato nella stessa cartella del file JSFL.

Restituisce
Nulla.

Descrizione
Metodo; associa un file XML allo strumento. Il file specifica le opzioni che devono essere visualizzate in un pannello a scelta obbligatoria richiamato da un pulsante Opzioni nella finestra di ispezione Proprietà. Questo metodo viene in genere utilizzato nella funzione `configureTool()` nel file JSFL. Consultate `configureTool()`.

Ad esempio, il file PolyStar.xml specifica tre opzioni associate allo strumento PolyStar:

```
<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>
```

Esempio

L'esempio seguente specifica che il file PolyStar.xml deve essere associato allo strumento attivo corrente. Il seguente codice è tratto dal file PolyStar.jsfl (consultate “Strumento di esempio PolyStar” a pagina 14):

```
theTool = fl.tools.activeTool;
theTool.setOptionsFile("PolyStar.xml");
```

toolObj.setPI()

Disponibilità

Flash MX 2004.

Uso

```
toolObj.setPI(pi)
```

Parametri

`pi` Una stringa che specifica la finestra di ispezione Proprietà da richiamare per lo strumento.

Restituisce

Nulla.

Descrizione

Metodo; imposta la finestra di ispezione Proprietà da utilizzare quando viene attivato lo strumento. Si utilizza questo metodo solo per la creazione di oggetti estensibili. I valori accettabili sono `"shape"` (il valore predefinito), `"text"` e `"movie"`.

Esempio

L'esempio seguente specifica che, quando viene attivato lo strumento, deve essere utilizzata la finestra di ispezione Proprietà della forma. Il seguente codice è tratto dal file PolyStar.jsfl (consultate "Strumento di esempio PolyStar" a pagina 14):

```
theTool = fl.tools.activeTool;
theTool.setPI("shape");
```

toolObj.setToolName()

Disponibilità

Flash MX 2004.

Uso

```
toolObj.setToolName(name)
```

Parametri

`name` Una stringa che specifica il nome dello strumento.

Restituisce

Nulla.

Descrizione

Metodo; assegna un nome allo strumento per la configurazione del pannello Strumenti. Si utilizza questo metodo solo per la creazione di oggetti estensibili. Il nome viene utilizzato solo dal file di layout XML che viene letto da Flash per creare il pannello Strumenti e non viene visualizzato nell'interfaccia utente del programma.

Esempio

L'esempio seguente assegna il nome `polystar` allo strumento `theTool`. Il seguente codice è tratto dal file PolyStar.jsfl (consultate "Strumento di esempio PolyStar" a pagina 14):

```
theTool = fl.tools.activeTool;
theTool.setToolName("polystar");
```

toolObj.setToolTip()

Disponibilità

Flash MX 2004.

Uso

```
toolObj.setToolTip(toolTip)
```

Parametri

`toolTip` Una stringa che specifica la descrizione comando dello strumento.

Restituisce
Nulla.

Descrizione
Metodo; imposta la descrizione comando che viene visualizzata quando posizionate il mouse sopra l'icona dello strumento. Si utilizza questo metodo solo per la creazione di oggetti estensibili.

Esempio
L'esempio seguente specifica `PolyStar Tool` come descrizione comando dello strumento. Il seguente codice è tratto dal file PolyStar.jsfl (consultate "Strumento di esempio PolyStar" a pagina 14):

```
theTool = fl.tools.activeTool;
theTool.setToolTip("PolyStar Tool");
```

toolObj.showPIControl()

Disponibilità
Flash MX 2004.

Uso
`toolObj.showPIControl(control, bShow)`

Parametri
`control` Una stringa che specifica il nome del controllo da mostrare o nascondere. Si utilizza questo metodo solo per la creazione di oggetti estensibili. I valori validi dipendono dalla finestra di ispezione Proprietà richiamata dallo strumento (consultate toolObj.setPI()`toolObj.setPI()`).

La finestra di ispezione Proprietà di una forma contiene i seguenti controlli:

stroke	fill

La finestra di ispezione Proprietà di un testo contiene i seguenti controlli:

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

La finestra di ispezione Proprietà di un filmato contiene i seguenti controlli:

size	publish	background
framerate	player	profile

`bShow` Un valore booleano che determina se il controllo specificato deve essere visualizzato (`true`) o nascosto (`false`).

Restituisce
Nulla.

Descrizione
Metodo; mostra o nasconde un controllo nella finestra di ispezione Proprietà. Si utilizza questo metodo solo per la creazione di oggetti estensibili.

Esempio
Il comando seguente presente in un file JavaScript di uno strumento estensibile specifica che nella finestra di ispezione Proprietà dello strumento non devono essere visualizzate le opzioni di riempimento:

```
fl.tools.activeTool.showPIControl("fill", false);
```

toolObj.showTransformHandles()

Disponibilità
Flash MX 2004.

Uso

```
toolObj.showTransformHandles(bShow)
```

Parametri
`bShow` Un valore booleano che determina se le maniglie di trasformazione libera dello strumento corrente devono essere visualizzate (`true`) o nascoste (`false`).

Restituisce
Nulla.

Descrizione
Metodo; viene chiamato nel metodo `configureTool()` di un file JavaScript di uno strumento estensibile per indicare che le maniglie di trasformazione libera devono essere visualizzate quando lo strumento è attivo. Si utilizza questo metodo solo per la creazione di oggetti estensibili.

Esempio
Consultate `configureTool()`.

Capitolo 49: Oggetto Tools

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto Tools è accessibile dall'oggetto Flash (`fl.tools`). La proprietà `tools.toolObjs` contiene un array di oggetti ToolObj, mentre la proprietà `tools.activeTool` restituisce l'oggetto ToolObj per lo strumento attivo corrente. Consultate anche Oggetto ToolObj e l'elenco degli strumenti estensibili in "Funzioni e metodi di primo livello" a pagina 15.

Nota: *i metodi e le proprietà seguenti vengono utilizzati solo durante la creazione di oggetti estensibili.*

Riepilogo dei metodi
I seguenti metodi sono disponibili per l'oggetto Tools:

Metodo	Descrizione
`tools.constrainPoint()`	Accetta due punti e restituisce un nuovo punto adattato o *vincolato*.
`tools.getKeyDown()`	Restituisce l'ultimo tasto premuto.
`tools.setCursor()`	Imposta l'aspetto specificato per il puntatore.
`tools.snapPoint()`	Considera un punto come input e restituisce un nuovo punto che può essere adattato o *agganciato* all'oggetto geometrico più vicino.

Riepilogo delle proprietà
Le seguenti proprietà sono disponibili per l'oggetto Tools:

Proprietà	Descrizione
`tools.activeTool`	Di sola lettura; restituisce l'Oggetto ToolObj per lo strumento attivo corrente.
`tools.altIsDown`	Di sola lettura; un valore booleano che indica se il tasto Alt è premuto.
`tools.ctlIsDown`	Di sola lettura; un valore booleano che indica se il tasto Ctrl è premuto.
`tools.mouseIsDown`	Di sola lettura; un valore booleano che indica se il pulsante sinistro del mouse è premuto.
`tools.penDownLoc`	Di sola lettura; un punto che rappresenta la posizione sullo stage dell'ultimo evento corrispondente al pulsante del mouse premuto.
`tools.penLoc`	Di sola lettura; una stringa che rappresenta la posizione corrente del mouse.
`tools.shiftIsDown`	Di sola lettura; un valore booleano che indica se il tasto Maiusc è premuto.
`tools.toolObjs`	Di sola lettura; un array di oggetti ToolObj.

tools.activeTool

Disponibilità
Flash MX 2004.

Uso

```
tools.activeTool
```

Descrizione

Proprietà di sola lettura; restituisce l'Oggetto ToolObj per lo strumento attivo corrente.

Esempio

L'esempio seguente salva un oggetto che rappresenta lo strumento attivo corrente nella variabile `theTool`:

```
var theTool = fl.tools.activeTool;
```

tools.altIsDown

Disponibilità

Flash MX 2004.

Uso

```
tools.altIsDown
```

Descrizione

Proprietà di sola lettura; un valore booleano che indica se il tasto Alt è premuto. Il valore è `true` se il tasto è premuto; in caso contrario è `false`.

Esempio

L'esempio seguente determina se il tasto Alt è premuto:

```
var isAltDown = fl.tools.altIsDown;
```

tools.constrainPoint()

Disponibilità

Flash MX 2004.

Uso

```
tools.constrainPoint(pt1, pt2)
```

Parametri

`pt1, pt2` Punti che specificano il punto corrispondente al clic iniziale e il punto in cui viene trascinata la selezione.

Restituisce

Un nuovo punto adattato o vincolato.

Descrizione
Metodo; accetta due punti e restituisce un nuovo punto adattato o vincolato. Se quando eseguite il comando premete il tasto Maiusc, il punto restituito segue un vincolo di 45º (utile ad esempio per una linea con una freccia all'estremità) o a fare in modo che un oggetto mantenga le proprie proporzioni originali (ad esempio, quando create un quadrato perfetto mediante lo strumento Rettangolo).

Esempio
L'esempio seguente restituisce un punto vincolato:

```
pt2 = fl.tools.constrainPoint(pt1, tempPt);
```

tools.ctlIsDown

Disponibilità
Flash MX 2004.

Uso
```
tools.ctlIsDown
```

Descrizione
Proprietà di sola lettura; il valore booleano `true` se il tasto Ctrl è premuto; `false` in caso contrario.

Esempio
L'esempio seguente determina se il tasto Ctrl è premuto:

```
var isCtrldown = fl.tools.ctrlIsDown;
```

tools.getKeyDown()

Disponibilità
Flash MX 2004.

Uso
```
tools.getKeyDown()
```

Parametri
Nessuno.

Restituisce
Il valore intero del tasto.

Descrizione
Metodo; restituisce l'ultimo tasto premuto.

Esempio

L'esempio seguente visualizza il valore intero dell'ultimo tasto premuto:

```
var theKey = fl.tools.getKeyDown();
fl.trace(theKey);
```

tools.mouseIsDown

Disponibilità

Flash MX 2004.

Uso

```
tools.mouseIsDown
```

Descrizione

Proprietà di sola lettura; il valore booleano `true` se il pulsante sinistro del mouse è premuto; `false` in caso contrario.

Esempio

L'esempio seguente determina se il pulsante sinistro del mouse è premuto.

```
var isMouseDown = fl.tools.mouseIsDown;
```

tools.penDownLoc

Disponibilità

Flash MX 2004.

Uso

```
tools.penDownLoc
```

Descrizione

Proprietà di sola lettura; un punto che rappresenta la posizione sullo stage dell'ultimo evento corrispondente al pulsante del mouse premuto. La proprietà `tools.penDownLoc` include due proprietà, *x* e *y*, corrispondenti alla posizione *x*, *y* del puntatore del mouse.

Esempio

L'esempio seguente determina la posizione dell'ultimo evento di pressione del pulsante del mouse nello stage e visualizza i valori *x* e *y* nel pannello Output:

```
var pt1 = fl.tools.penDownLoc;
fl.trace("x,y location of last mouseDown event was " + pt1.x + ", " + pt1.y)
```

Consultate anche

`tools.penLoc`

tools.penLoc

Disponibilità
Flash MX 2004.

Uso
```
tools.penLoc
```

Descrizione
Proprietà di sola lettura; un punto che rappresenta la posizione corrente del puntatore del mouse. La proprietà `tools.penLoc` include due proprietà, *x* e *y*, corrispondenti alla posizione *x,y* del puntatore del mouse.

Esempio
L'esempio seguente determina la posizione corrente del mouse:

```
var tempPt = fl.tools.penLoc;
```

Consultate anche
`tools.penDownLoc`

tools.setCursor()

Disponibilità
Flash MX 2004.

Uso
```
tools.setCursor(cursor)
```

Parametri
`cursor` Un numero intero che definisce l'aspetto del puntatore, come descritto nell'elenco seguente:

- 0 = puntatore a croce (+)
- 1 = freccia nera
- 2 = freccia bianca
- 3 = freccia a quattro direzioni
- 4 = freccia a due direzioni orizzontali
- 5 = freccia a due direzioni verticali
- 6 = X
- 7 = cursore mano

Restituisce
Nulla.

Descrizione
Metodo; imposta l'aspetto specificato per il puntatore.

Esempio
L'esempio seguente visualizza un puntatore a forma di freccia nera:

```
fl.tools.setCursor(1);
```

tools.shiftIsDown

Disponibilità
Flash MX 2004.

Uso
```
tools.shiftIsDown
```

Descrizione
Proprietà di sola lettura; il valore booleano `true` se il tasto Maiusc è premuto; `false` in caso contrario.

Esempio
L'esempio seguente determina se il tasto Maiusc è premuto.

```
var isShiftDown = fl.tools.shiftIsDown;
```

tools.snapPoint()

Disponibilità
Flash MX 2004.

Uso
```
tools.snapPoint(pt)
```

Parametri
`pt` Specifica la posizione del punto per cui desiderate venga restituito un punto di aggancio.

Restituisce
Un nuovo punto che può essere adattato o agganciato all'oggetto geometrico più vicino.

Descrizione
Metodo; considera un punto come input e restituisce un nuovo punto che può essere adattato o *agganciato* all'oggetto geometrico più vicino. Se l'aggancio è disabilitato nel menu Visualizza dell'interfaccia utente di Flash, il punto restituito è quello originale.

Esempio
L'esempio seguente restituisce un nuovo punto che può essere agganciato all'oggetto geometrico più vicino.

```
var theSnapPoint = fl.tools.snapPoint(pt1);
```

tools.toolObjs

Disponibilità
Flash MX 2004.

Uso
```
tools.toolObjs
```

Descrizione
Proprietà di sola lettura; un array di oggetti ToolObj (consultate Oggetto ToolObj).

Capitolo 50: Oggetto Vertex

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto Vertex è la parte della struttura di dati di una forma che contiene i dati relativi alle coordinate.

Riepilogo dei metodi
Con l'oggetto Vertex è possibile utilizzare i metodi seguenti:

Metodo	Descrizione
`vertex.getHalfEdge()`	Ottiene un Oggetto HalfEdge che condivide il vertice.
`vertex.setLocation()`	Imposta la posizione del vertice.

Riepilogo delle proprietà
Le seguenti proprietà sono disponibili per l'oggetto Vertex:

Proprietà	Descrizione
`vertex.x`	Di sola lettura; la posizione *x* del vertice, espressa in pixel.
`vertex.y`	Di sola lettura; la posizione *y* del vertice, espressa in pixel.

vertex.getHalfEdge()

Disponibilità
Flash MX 2004.

Uso
```
vertex.getHalfEdge()
```

Parametri
Nessuno.

Restituisce
Un Oggetto HalfEdge.

Descrizione
Metodo; ottiene un Oggetto HalfEdge che condivide il vertice.

Esempio
L'esempio seguente mostra la procedura per ottenere gli altri mezzi bordi che condividono lo stesso vertice:

```
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()

Disponibilità
Flash MX 2004.

Uso
```
vertex.setLocation(x, y)
```

Parametri
x Un valore a virgola mobile che specifica la coordinata *x*, espressa in pixel, del punto in cui deve essere collocato il vertice.

y Un valore a virgola mobile che specifica la coordinata *y*, espressa in pixel, del punto in cui deve essere collocato il vertice.

Restituisce
Nulla.

Descrizione
Metodo; imposta la posizione del vertice. Dovete chiamare `shape.beginEdit()` prima di utilizzare questo metodo.

Esempio
L'esempio seguente imposta il vertice sul punto di origine:

```
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

Disponibilità
Flash MX 2004.

Uso
```
vertex.x
```

Descrizione
Proprietà di sola lettura; posizione *x* del vertice, espressa in pixel.

Esempio
L'esempio seguente visualizza la posizione dei valori *x* e *y* del vertice nel pannello Output:

```
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

Disponibilità
Flash MX 2004.

Uso
`vertex.y`

Descrizione
Di sola lettura; la posizione *y* del vertice, espressa in pixel.

Esempio
Consultate `vertex.x`.

Capitolo 51: Oggetto VideoItem

Ereditarietà Oggetto Item > Oggetto VideoItem

Disponibilità
Flash MX 2004.

Descrizione
L'oggetto VideoItem è una sottoclasse dell'Oggetto Item.

Riepilogo dei metodi
Oltre a quelli dell'oggetto Item, l'oggetto VideoItem è dotato del seguente metodo:

Proprietà	Descrizione
`videoItem.exportToFLV()`	Esporta l'elemento specificato in un file FLV.

Riepilogo delle proprietà
Oltre a quelle dell'oggetto Item, con l'oggetto VideoItem potete utilizzare le seguenti proprietà:

Proprietà	Descrizione
`videoItem.fileLastModifiedDate`	Sola lettura; una stringa contenente un numero esadecimale che rappresenta il numero di secondi trascorsi tra il 1° gennaio 1970 e la data della modifica del file originale (su disco) nel momento in cui il file è stato importato nella libreria.
`videoItem.sourceFileExists`	Sola lettura; un valore booleano che specifica se il file importato nella libreria esiste ancora nel percorso da cui è stato importato.
`videoItem.sourceFileIsCurrent`	Sola lettura; un valore booleano che specifica se la data di modifica del file dell'elemento nella libreria è uguale alla data di modifica (sul disco) del file importato.
`videoItem.sourceFilePath`	Sola lettura; una stringa che specifica il percorso dell'elemento video.
`videoItem.videoType`	Sola lettura; una stringa che specifica il tipo di video rappresentato dall'elemento.

videoItem.exportToFLV()

Disponibilità
Flash CS4 Professional.

Uso
```
videoItem.exportToFLV(fileURI)
```

Parametri
`fileURI` Una stringa, espressa come URI file:///, che specifica il percorso e il nome del file esportato.

Restituisce
Il valore booleano `true` se il file è stato esportato correttamente; `false` in caso contrario.

Descrizione
Metodo; esporta l'elemento specificato in un file FLV.

Esempio
Se il primo elemento nella libreria è un video, il codice seguente lo esporta come file FLV:

```
var videoFileURL = "file:///C|/out.flv";
var libItem = fl.getDocumentDOM().library.items[0];
libItem.exportToFLV(videoFileURL);
```

videoItem.fileLastModifiedDate

Disponibilità
Flash CS4 Professional.

Uso
`videoItem.fileLastModifiedDate`

Descrizione
Proprietà di sola lettura: una stringa contenente un numero esadecimale che rappresenta il numero di secondi trascorsi tra il 1 gennaio 1970 e la data della modifica del file originale (su disco) nel momento in cui il file è stato importato nella libreria. Se il file non esiste più, il valore corrisponde a "00000000".

Esempio
Se il primo elemento nella libreria è un video, il codice seguente visualizza un numero esadecimale come descritto in precedenza.

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

Consultate anche
`videoItem.sourceFileExists`, `videoItem.sourceFileIsCurrent`, `videoItem.sourceFilePath`, `FLfile.getModificationDate()`

videoItem.sourceFileExists

Disponibilità
Flash CS4 Professional.

Uso
`videoItem.sourceFileExists`

Descrizione
Proprietà di sola lettura: il value booleano `true` se il file importato nella libreria esiste ancora nel percorso da cui è stato importato; `false` in caso contrario.

Esempio
Se il primo elemento nella libreria è un video, il codice seguente visualizza "true" se il file importato nella libreria esiste ancora.

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

Consultate anche
`videoItem.sourceFileIsCurrent`, `videoItem.sourceFilePath`

videoItem.sourceFileIsCurrent

Disponibilità
Flash CS4 Professional.

Uso
```
videoItem.sourceFileIsCurrent
```

Descrizione
Proprietà di sola lettura: il value booleano `true` se la data di modifica del file dell'elemento della libreria corrisponde alla data di modifica (su disco) del file importato; `false` in caso contrario.

Esempio
Se il primo elemento nella libreria è un video, il codice seguente visualizza "true" se il file importato non è stato modificato da quando è stato importato.

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

Consultate anche
`videoItem.fileLastModifiedDate`, `videoItem.sourceFilePath`

videoItem.sourceFilePath

Disponibilità
Flash 8.

Uso
```
videoItem.sourceFilePath
```

Descrizione
Proprietà di sola lettura; una stringa, espressa nel formato file:///URI, che specifica il percorso dell'elemento video.

Esempio

L'esempio seguente visualizza il percorso e il nome e il percorso del file di origine degli elementi della libreria di tipo `video`:

```
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);
    }
}
```

Consultate anche

`videoItem.sourceFileExists`

videoItem.videoType

Disponibilità

Flash 8.

Uso

```
videoItem.videoType
```

Descrizione

Proprietà di sola lettura; una stringa che specifica il tipo di video rappresentato dall'elemento. I valori possibili sono `"embeddedvideo"`, `"linkedvideo"` e `"video"`.

Esempio

L'esempio seguente visualizza il nome e il tipo degli elementi della libreria di tipo `video`:

```
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);
}
}
```

Capitolo 52: Oggetto XMLUI

Disponibilità
Flash MX 2004.

Descrizione
Flash 8 supporta le finestre di dialogo personalizzate scritte con un sottoinsieme del linguaggio di interfaccia utente XML (XUL). Una finestra di dialogo dell'interfaccia utente XML (XMLUI) può essere utilizzata da diverse funzioni di Flash, ad esempio i comandi e i comportamenti, per fornire un'interfaccia utente per le funzioni create mediante l'estensibilità. L'oggetto XMLUI consente di ottenere e impostare le proprietà di una finestra di dialogo XMLUI e di uscire dalla finestra facendo clic su OK o su Annulla. I metodi XMLUI possono essere utilizzati nelle funzioni di callback, quali i gestori `oncommand` dei pulsanti.

Potete scrivere un file dialog.xml e richiamarlo dall'API JavaScript utilizzando il metodo `document.xmlPanel()`. Per recuperare un oggetto che rappresenta la finestra di dialogo XMLUI corrente, utilizzate `fl.xmlui`.

Riepilogo dei metodi
I metodi seguenti sono disponibili per l'oggetto XMLUI:

Metodo	Descrizione
`xmlui.accept()`	Chiude la finestra di dialogo XMLUI corrente accettando l'operazione.
`xmlui.cancel()`	Chiude la finestra di dialogo XMLUI corrente annullando l'operazione.
`xmlui.get()`	Recupera il valore della proprietà specificata della finestra di dialogo XMLUI corrente.
`xmlui.getControlItemElement()`	Restituisce l'elemento corrente per il controllo specificato.
`xmlui.getEnabled()`	Restituisce un valore booleano che specifica se il controllo è abilitato o disabilitato (visualizzato in grigio).
`xmlui.getVisible()`	Restituisce un valore booleano che specifica se il controllo è visibile o nascosto.
`xmlui.set()`	Modifica il valore della proprietà specificata della finestra di dialogo XMLUI corrente.
`xmlui.setControlItemElement()`	Imposta etichetta e valore dell'elemento corrente.
`xmlui.setControlItemElements()`	Imposta le coppie label/value dell'elemento corrente.
`xmlui.setEnabled()`	Abilita o disabilita un controllo.
`xmlui.setVisible()`	Mostra o nasconde un controllo.

xmlui.accept()

Disponibilità
Flash MX 2004.

Uso
```
xmlui.accept()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; chiude la finestra di dialogo XMLUI corrente accettando l'operazione, azione che equivale a quando l'utente fa clic sul pulsante OK.

Consultate anche
`fl.xmlui`, `document.xmlPanel()`, `xmlui.cancel()`

xmlui.cancel()

Disponibilità
Flash MX 2004.

Uso
```
xmlui.cancel()
```

Parametri
Nessuno.

Restituisce
Nulla.

Descrizione
Metodo; chiude la finestra di dialogo XMLUI corrente annullando l'operazione, azione che equivale a quando l'utente fa clic sul pulsante Annulla.

Consultate anche
`fl.xmlui`, `document.xmlPanel()`, `xmlui.accept()`

xmlui.get()

Disponibilità
Flash MX 2004.

Uso
```
xmlui.get(controlPropertyName)
```

Parametri

`controlPropertyName` Una stringa che specifica il nome della proprietà XMLUI di cui desiderate recuperare il valore.

Restituisce

Una stringa che rappresenta il valore della proprietà specificata. Nei casi in cui è possibile prevedere un valore booleano `true` o `false`, restituisce la stringa `"true"` o `"false"`.

Descrizione

Metodo; recupera il valore della proprietà specificata per la finestra di dialogo XMLUI corrente.

Esempio

L'esempio seguente restituisce il valore della proprietà `URL`:

```
fl.xmlui.get("URL");
```

Consultate anche

`fl.xmlui`, `document.xmlPanel()`, `xmlui.getControlItemElement()`, `xmlui.set()`

xmlui.getControlItemElement()

Disponibilità

Flash 8.

Uso

```
xmlui.getControlItemElement(controlPropertyName)
```

Parametri

`controlPropertyName` Una stringa che specifica la proprietà di cui desiderate recuperare l'elemento del controllo.

Restituisce

Un oggetto che rappresenta l'elemento corrente per il controllo specificato da *controlPropertyName*.

Descrizione

Metodo; restituisce l'etichetta e il valore della riga selezionata in un controllo ListBox o ComboBox per il controllo specificato da *controlPropertyName*.

Esempio

L'esempio seguente restituisce l'etichetta e il valore della riga selezionata per il controllo `myListBox`:

```
var elem = new Object();
elem = fl.xmlui.getControlItemElement("myListBox");
fl.trace("label = " + elem.label + " value = " + elem.value);
```

Consultate anche

`fl.xmlui`, `document.xmlPanel()`, `xmlui.get()`, `xmlui.setControlItemElement()`, `xmlui.setControlItemElements()`

xmlui.getEnabled()

Disponibilità
Flash 8.

Uso
`xmlui.getEnabled(controlID)`

Parametri
`controlID` Una stringa che specifica l'attributo ID del controllo di cui desiderate recuperare lo stato.

Restituisce
Il valore booleano `true` se il controllo è abilitato; `false` in caso contrario.

Descrizione
Metodo; restituisce un valore booleano che specifica se il controllo è abilitato o disabilitato.

Esempio
L'esempio seguente restituisce un valore che indica se il controllo con l'attributo ID `myListBox` è abilitato:

```
var isEnabled = fl.xmlui.getEnabled("myListBox");
fl.trace(isEnabled);
```

Consultate anche
`fl.xmlui`, `document.xmlPanel()`, `xmlui.setEnabled()`

xmlui.getVisible()

Disponibilità
Flash 8.

Uso
`xmlui.getVisible(controlID)`

Parametri
`controlID` Una stringa che specifica l'attributo ID del controllo di cui desiderate recuperare lo stato relativo alla visibilità.

Restituisce
Un valore booleano `true` se il controllo è visibile, o `false` se è invisibile (nascosto).

Descrizione
Metodo; restituisce un valore booleano che specifica se il controllo è visibile o nascosto.

Esempio
L'esempio seguente restituisce un valore che indica se il controllo con l'attributo ID `myListBox` è visibile:

```
var isVisible = fl.xmlui.getVisible("myListBox");
fl.trace(isVisible);
```

Consultate anche

xmlui.setVisible()

xmlui.set()

Disponibilità

Flash MX 2004.

Uso

```
xmlui.set(controlPropertyName, value)
```

Parametri

controlPropertyName Una stringa che specifica il nome della proprietà XMLUI da modificare.

value Una stringa che specifica il valore su cui impostare la proprietà XMLUI.

Restituisce

Nulla.

Descrizione

Metodo; modifica il valore della proprietà specificata per la finestra di dialogo XMLUI corrente.

Esempio

L'esempio seguente imposta il valore della proprietà URL su `www.adobe.com`:

```
fl.xmlui.set("URL", "www.adobe.com");
```

Consultate anche

fl.xmlui, document.xmlPanel(), xmlui.get(), xmlui.setControlItemElement(), xmlui.setControlItemElements()

xmlui.setControlItemElement()

Disponibilità

Flash 8.

Uso

```
xmlui.setControlItemElement(controlPropertyName, elementItem)
```

Parametri

controlPropertyName Una stringa che specifica l'elemento del controllo da impostare.

`elementItem` Un oggetto JavaScript con una proprietà di stringa denominata `label` e una proprietà di stringa opzionale denominata `value`. Se la proprietà `value` non esiste, viene creata e a essa viene assegnato lo stesso valore di `label`.

Restituisce
Nulla.

Descrizione
Metodo; imposta l'etichetta e il valore della riga selezionata nel controllo ListBox o ComboBox specificato da *controlPropertyName*.

Esempio
L'esempio seguente imposta l'etichetta e il valore dell'elemento corrente della proprietà del controllo `PhoneNumber`:

```
var elem = new Object();
elem.label = "Fax";
elem.value = "707-555-5555";
fl.xmlui.setControlItemElement("PhoneNumber",elem);
```

Consultate anche
`fl.xmlui`, `document.xmlPanel()`, `xmlui.getControlItemElement()`, `xmlui.set()`, `xmlui.setControlItemElements()`

xmlui.setControlItemElements()

Disponibilità
Flash 8.

Uso

```
xmlui.setControlItemElements(controlID, elementItemArray)
```

Parametri
`controlID` Una stringa che specifica l'attributo ID del controllo che desiderate impostare.

`elementItemArray` Un array di oggetti JavaScript, in cui ogni oggetto ha una proprietà di stringa denominata `label` e una proprietà di stringa opzionale denominata `value`. Se la proprietà `value` non esiste, viene creata e a essa viene assegnato lo stesso valore di `label`.

Restituisce
Nulla.

Descrizione
Metodo; cancella i valori del controllo ListBox o ComboBox specificato da *controlID* e sostituisce le voci di elenco o di menu con le coppie `label`,`value` specificate da *elementItemArray*.

Esempio
L'esempio seguente imposta l'etichetta e il valore degli elementi del controllo con l'attributo ID `myControlID` sulle coppie `label`,`value` specificate:

```
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);
```

Consultate anche

xmlui.getControlItemElement(), xmlui.set(), xmlui.setControlItemElement()

xmlui.setEnabled()

Disponibilità

Flash 8.

Uso

```
xmlui.setEnabled(controlID, enable)
```

Parametri

controlID Una stringa che specifica l'attributo ID del controllo che desiderate abilitare o disabilitare.

enable Il valore booleano `true` se desiderate abilitare il controllo; `false` se desiderate disabilitarlo.

Restituisce

Nulla.

Descrizione

Metodo; abilita o disabilita un controllo.

Esempio

L'esempio seguente disabilita il controllo con l'attributo ID `myControl`:

```
fl.xmlui.setEnabled("myControl", false);
```

Consultate anche

xmlui.getEnabled()

xmlui.setVisible()

Disponibilità

Flash 8.

Uso

```
xmlui.setVisible(controlID, visible)
```

Parametri

`controlID` Una stringa che specifica l'attributo ID del controllo che desiderate visualizzare o nascondere.

`visible` Il valore booleano `true` se desiderate visualizzare il controllo; `false` se desiderate nasconderlo.

Restituisce

Nulla.

Descrizione

Metodo; mostra o nasconde un controllo.

Esempio

L'esempio seguente nasconde il controllo con l'attributo ID `myControl`:

```
fl.xmlui.setVisible("myControl", false);
```

Consultate anche

`xmlui.getVisible()`

Capitolo 53: Estensibilità di livello C

In questo capitolo è descritto il meccanismo di estensibilità di livello C, che consente di implementare i file di estensibilità di Adobe Flash CS4 Professional mediante una combinazione di JavaScript e di codice C personalizzato. Non è stata introdotta nessuna modifica al meccanismo in questa versione di Flash.

Informazioni sull'estensibilità

Per implementare l'estensibilità, l'utente definisce le funzioni nel linguaggio C e le raccoglie in una DLL (Dynamic Linked Library) o in una libreria condivisa, quindi salva la libreria nella directory appropriata. Infine, le funzioni vengono chiamate da JavaScript mediante l'API JavaScript di Adobe Flash.

Ad esempio, in questo modo è possibile definire una funzione che esegua dei calcoli complessi in modo più efficace di JavaScript, ottenendo un miglioramento delle prestazioni, oppure creare strumenti o effetti più avanzati.

Il meccanismo di estensibilità è un sottoinsieme dell'API di Adobe Dreamweaver CS3. Se si ha una certa esperienza con tale API, è possibile riconoscere le funzioni nell'API dell'estensibilità di livello C. Tuttavia, l'API in questione differisce da quella di Dreamweaver nei modi seguenti:

- Non contiene tutti i comandi presenti nell'API di Dreamweaver.
- Tutte le dichiarazioni di tipo `wchar_t` e `char` presenti nell'API di Dreamweaver sono qui implementate come dichiarazioni `unsignedshort`, al fine di supportare Unicode quando vengono passate le stringhe.
- La funzione `JSVal JS_BytesToValue()` di questa API non fa parte dell'API di Dreamweaver.
- Il percorso in cui è necessario salvare i file DLL o della libreria condivisa è diverso (consultate “Integrazione delle funzioni C” a pagina 534).

Integrazione delle funzioni C

L'estensibilità di livello C consente di implementare i file di estensibilità di Flash mediante una combinazione di JavaScript e codice C. Il processo di implementazione di questa funzione è riassunto nei punti seguenti:

1 Definire le funzioni mediante il linguaggio C o C++.

2 Raggrupparle in un file DLL (Windows) o in una libreria condivisa (Macintosh).

3 Salvate il file DLL o della libreria nel percorso appropriato:

- Windows Vista:

  unità di avvio \Utenti*nomeutente*\Impostazioni locali\Dati applicazioni\Adobe\Flash CS3*lingua*\Configuration\External Libraries

- Windows XP:

  unità di avvio\Documents and Settings*nomeutente*\Impostazioni locali\Dati applicazioni\Adobe\Flash CS3*lingua*\Configuration\External Libraries

- Mac OS X:

Macintosh HD/Users/*nomeutente*/Library/Supporto applicazioni/Adobe/Flash CS3/*lingua*/Configuration/External Libraries

4 Create un file JSFL che chiami le funzioni.

5 Eseguite il file JSFL dal menu Comandi nell'ambiente di creazione Flash.

Per ulteriori informazioni, consultate "Esempio di implementazione DLL" a pagina 538.

Estensibilità di livello C e interprete JavaScript

Il codice C presente nella DLL o nella libreria condivisa interagisce con l'API JavaScript di Flash in tre momenti diversi:

- All'avvio, per registrare le funzioni della libreria.
- Quando viene chiamata la funzione C, per decomprimere gli argomenti che vengono passati da JavaScript a C.
- Prima che la funzione C restituisca un risultato, per comprimere il valore restituito.

Per eseguire queste operazioni, l'interprete definisce diversi tipi di dati ed espone un'API. Le definizioni dei tipi di dati e delle funzioni presenti in questa sezione sono contenute nel file mm_jsapi.h. Per un corretto funzionamento della libreria, includere il file mm_jsapi.h sopra ogni file nella libreria, con la riga seguente:

```
#include "mm_jsapi.h"
```

Quando si include il file mm_jsapi.h, viene a sua volta incluso il file mm_jsapi_environment.h, che definisce la struttura MM_Environment.

Per ottenere una copia del file mm_jsapi.h, estraetelo dal file ZIP o SIT di esempio (consultate "Esempio di implementazione DLL" a pagina 538), oppure copiate il codice seguente in un file, quindi assegnate al file il nome mm_jsapi.h:

```
#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_ */
```

Esempio di implementazione DLL

Questa sezione illustra come creare una semplice implementazione DLL. Per verificare come funziona il processo senza dover costruire personalmente la DLL, potete installare i file DLL di esempio forniti nel file Samples.zip, che si trovano nella cartella ExtendingFlash/dllSampleComputeSum. (Per informazioni su come scaricare il file Samples.zip file, consultate “Esempi di implementazione” a pagina 13.) Estraete i file di esempio dal file dllSampleComputeSum.dmg o dllSampleComputeSum.zip, quindi effettuate le seguenti operazioni:

- Salvate il file Sample.jsfl nella directory Configuration/Commands (consultate “Salvataggio di file JSFL” a pagina 2).
- Salvate il file Sample.dll nella directory Configuration/External Libraries (consultate “Integrazione delle funzioni C” a pagina 534).
- Nell'ambiente di creazione Flash, selezionate Comandi > Sample. L'istruzione trace contenuta nel file JSFL invia al pannello Output i risultati della funzione definita in Sample.dll.

Nel resto di questa sezione viene descritto lo sviluppo dell'esempio. In questo caso, la DLL contiene una sola funzione, che aggiunge due numeri. Il codice C è indicato nell'esempio seguente:

```
// 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;
}
```

Dopo aver scritto questo codice, create il file della DLL o della libreria condivisa e salvatelo nella directory Configuration/External Libraries appropriata (consultate "Integrazione delle funzioni C" a pagina 534). Quindi, create un file JSFL con il codice seguente e salvatelo nella directory Configuration/Commands (consultate "Salvataggio di file JSFL" a pagina 2).

```
// JSFL file to run C function defined above.
var a = 5;
var b = 10;
var sum = Sample.computeSum(a, b);
fl.trace("The sum of " + a + " and " + b + " is " + sum );
```

Per eseguire la funzione definita nella DLL, selezionate Comandi > Sample nell'ambiente di creazione Flash.

Tipi di dati

L'interprete JavaScript definisce i tipi di dati descritti in questa sezione.

typedef struct JSContext JSContext

Un puntatore a questo tipo di dati opaco passa alla funzione di livello C. Alcune funzioni nell'API accettano questo puntatore come argomento.

typedef struct JSObject JSObject

Un puntatore a questo tipo di dati opaco passa alla funzione di livello C. Questo tipo di dati rappresenta un oggetto, che può essere un oggetto array o di altro tipo.

typedef struct jsval jsval

Una struttura di dati opaca che può contenere un numero intero oppure un puntatore a un valore float, a una stringa o a un oggetto. Alcune funzioni nell'API sono in grado di leggere i valori degli argomenti o delle funzioni leggendo il contenuto di una struttura `jsval`, mentre altre possono essere utilizzate per scrivere il valore restituito dalla funzione utilizzando una struttura `jsval`.

typedef enum { JS_FALSE = 0, JS_TRUE = 1 } JSBool

Un tipo di dati semplice che memorizza un valore booleano.

API di livello C

L'API di estensibilità di livello C è composta dall'indicatore della funzione `JSBool (*JSNative)` e dalle funzioni seguenti:

- `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)

Descrizione

Metodo; descrive le implementazioni di livello C delle funzioni JavaScript nelle situazioni seguenti:

- Il puntatore *cx* è un puntatore a una struttura `JSContext` opaca, che deve essere passata ad alcune delle funzioni nell'API JavaScript. Questa variabile contiene il contesto per l'esecuzione dell'interprete.
- Il puntatore *obj* è un puntatore all'oggetto nel cui contesto viene eseguito lo script. Mentre lo script è in esecuzione, la parola chiave `this` è uguale a questo oggetto.
- Il valore intero *argc* è il numero di argomenti passati alla funzione.
- Il puntatore *argv* è un puntatore a un array di strutture `jsval`. La lunghezza dell'array è di `argc` elementi.
- Il puntatore *rval* è un puntatore a una singola struttura `jsval`. Il valore restituito dalla funzione deve essere scritto in `*rval`.

La funzione restituisce `JS_TRUE` se ha esito positivo e `JS_FALSE` se ha esito negativo. Se la funzione restituisce `JS_FALSE`, viene interrotta l'esecuzione dello script corrente e viene visualizzato un messaggio di errore.

JSBool JS_DefineFunction()

Uso

```
JSBool JS_DefineFunction(unsigned short *name, JSNative call, unsigned int nargs)
```

Descrizione

Metodo; registra una funzione di livello C con l'interprete JavaScript in Flash. Dopo che la funzione `JS_DefineFunction()` ha registrato la funzione di livello C specificata dall'utente nell'argomento *call*, è possibile chiamarla in uno script JavaScript facendo riferimento a essa con il nome specificato nell'argomento *name*. L'argomento *name* fa distinzione tra maiuscole e minuscole.

Generalmente questa funzione viene chiamata dalla funzione `MM_Init()`, a propria volta chiamata da Flash all'avvio.

Argomenti

`unsigned short *`*name*, `JSNative`*call*, `unsigned int` *nargs*

- L'argomento *name* è il nome della funzione come viene esposta a JavaScript.
- L'argomento *call* è un puntatore a una funzione di livello C. La funzione deve restituire un valore `JSBool`, che indica l'esito positivo o negativo.
- L'argomento *nargs* è il numero di argomenti previsti dalla funzione.

Restituisce

Un valore booleano: `JS_TRUE` indica un esito positivo; `JS_FALSE` indica un esito negativo.

unsigned short *JS_ValueToString()

Uso

```
unsigned short *JS_ValueToString(JSContext *cx, jsval v, unsigned int *pLength)
```

Descrizione
Metodo; estrae un argomento di una funzione da una struttura `jsval`, lo converte (se possibile) in una stringa, quindi ripassa il valore convertito al chiamante.

Nota: *non modificate il puntatore buffer restituito per evitare di danneggiare le strutture di dati dell'interprete JavaScript. Per modificare la stringa, copiate i caratteri in un altro buffer e creare una nuova stringa JavaScript.*

Argomenti
`JSContext` *`*cx`*, `jsval` *`v`*, `unsigned int` *`*pLength`*

- L'argomento *cx* è il puntatore opaco `JSContext` che viene passato alla funzione JavaScript.
- L'argomento *v* è la struttura `jsval` da cui deve essere estratta la stringa.
- L'argomento *pLength* è un puntatore a un numero intero senza segno. La funzione imposta `*plength` su un valore uguale alla lunghezza della stringa espressa in byte.

Restituisce
Un puntatore che punta a una stringa terminata da un byte nullo in caso di esito positivo o a un valore `null` in caso di esito negativo. La routine che effettua la chiamata non deve liberare questa stringa quando termina l'operazione.

JSBool JS_ValueToInteger()

Uso
```
JSBool JS_ValueToInteger(JSContext *cx, jsval v, long *lp);
```

Descrizione
Metodo; estrae un argomento di una funzione da una struttura `jsval`, lo converte (se possibile) in un numero intero, quindi ripassa il valore convertito al chiamante.

Argomenti
`JSContext` *`*cx`*, `jsval` *`v`*, `long` *`*lp`*

- L'argomento *cx* è il puntatore opaco `JSContext` che viene passato alla funzione JavaScript.
- L'argomento *v* è la struttura `jsval` da cui deve essere estratto l'intero.
- L'argomento *lp* è un puntatore a un numero intero a 4 byte. Questa funzione memorizza il valore convertito in `*lp`.

Restituisce
Un valore booleano: `JS_TRUE` indica un esito positivo; `JS_FALSE` indica un esito negativo.

JSBool JS_ValueToDouble()

Uso
```
JSBool JS_ValueToDouble(JSContext *cx, jsval v, double *dp);
```

Descrizione
Metodo; estrae un argomento di una funzione da una struttura `jsval`, lo converte (se possibile) in un doppio, quindi ripassa il valore convertito al chiamante.

Argomenti

`JSContext *cx, jsval v, double *dp`

- L'argomento *cx* è il puntatore opaco `JSContext` che è stato passato alla funzione JavaScript.
- L'argomento *v* è la struttura `jsval` da cui deve essere estratto il doppio.
- L'argomento *dp* è un puntatore a un doppio a 8 byte. Questa funzione memorizza il valore convertito in `*dp`.

Restituisce

Un valore booleano: `JS_TRUE` indica un esito positivo; `JS_FALSE` indica un esito negativo.

JSBool JS_ValueToBoolean()

Uso

```
JSBool JS_ValueToBoolean(JSContext *cx, jsval v, JSBool *bp);
```

Descrizione

Metodo; estrae un argomento di una funzione da una struttura `jsval`, lo converte (se possibile) in un valore booleano, quindi ripassa il valore convertito al chiamante.

Argomenti

`JSContext *cx, jsval v, JSBool *bp`

- L'argomento *cx* è il puntatore opaco `JSContext` che viene passato alla funzione JavaScript.
- L'argomento *v* è la struttura `jsval` da cui deve essere estratto il valore booleano.
- L'argomento *bp* è un puntatore a un valore booleano `JSBool`. Questa funzione memorizza il valore convertito in `bp`.

Restituisce

Un valore booleano: `JS_TRUE` indica un esito positivo; `JS_FALSE` indica un esito negativo.

JSBool JS_ValueToObject()

Uso

```
JSBool JS_ValueToObject(JSContext *cx, jsval v, JSObject **op);
```

Descrizione

Metodo; estrae un argomento di una funzione da una struttura `jsval`, lo converte (se possibile) in un oggetto, quindi ripassa il valore convertito al chiamante. Se l'oggetto è un array, utilizzare `JS_GetArrayLength()` e `JS_GetElement()` per leggerne il contenuto.

Argomenti

`JSContext *cx, jsval v, JSObject **op`

- L'argomento *cx* è il puntatore opaco `JSContext` che viene passato alla funzione JavaScript.
- L'argomento *v* è la struttura `jsval` da cui deve essere estratto l'oggetto.
- L'argomento *op* è un puntatore a un puntatore `JSObject`. Questa funzione memorizza il valore convertito in `*op`.

Restituisce
Un valore booleano: `JS_TRUE` indica un esito positivo; `JS_FALSE` indica un esito negativo.

JSBool JS_StringToValue()

Uso
```
JSBool JS_StringToValue(JSContext *cx, unsigned short *bytes, uint sz, jsval *vp);
```

Descrizione
Metodo; memorizza il valore restituito da una stringa in una struttura `jsval`. Alloca un nuovo oggetto stringa JavaScript.

Argomenti
`JSContext` **cx*, `unsigned short` **bytes*, `size_t` *sz*, `jsval` **vp*

- L'argomento *cx* è il puntatore opaco `JSContext` che viene passato alla funzione JavaScript.
- L'argomento *bytes* è la stringa da memorizzare nella struttura `jsval`. I dati della stringa vengono copiati per consentire al chiamante di liberare la stringa quando non è necessaria. Se le dimensioni della stringa non sono specificate (consultate l'argomento *sz*), la stringa deve essere terminata da un byte nullo.
- L'argomento *sz* corrisponde alle dimensioni della stringa, espresse in byte. Se *sz* è uguale a 0, la lunghezza della stringa terminata da un byte nullo viene calcolata automaticamente.
- L'argomento *vp* è un puntatore alla struttura `jsval` in cui deve essere copiato il contenuto della stringa.

Restituisce
Un valore booleano: `JS_TRUE` indica un esito positivo; `JS_FALSE` indica un esito negativo.

JSBool JS_DoubleToValue()

Uso
```
JSBool JS_DoubleToValue(JSContext *cx, double dv, jsval *vp);
```

Descrizione
Metodo; memorizza il valore restituito da un numero a virgola mobile in una struttura `jsval`.

Argomenti
`JSContext` **cx*, `double` *dv*, `jsval` **vp*

- L'argomento *cx* è il puntatore opaco `JSContext` che viene passato alla funzione JavaScript.
- L'argomento *dv* è un numero a virgola mobile a 8 byte.
- L'argomento *vp* è un puntatore alla struttura `jsval` in cui deve essere copiato il contenuto del doppio.

Restituisce
Un valore booleano: `JS_TRUE` indica un esito positivo; `JS_FALSE` indica un esito negativo.

JSVal JS_BooleanToValue()

Uso

```
jsval JS_BooleanToValue(JSBool bv);
```

Descrizione

Metodo; memorizza un valore booleano restituito in una struttura `jsval`.

Argomenti

`JSBool` *bv*

- L'argomento *bv* è un valore booleano: `JS_TRUE` indica un esito positivo; `JS_FALSE` indica un esito negativo.

Restituisce

Una struttura `JSVal` che contiene il valore booleano che viene passato alla funzione sotto forma di argomento.

JSVal JS_BytesToValue()

Uso

```
JSBool JS_BytesToValue(JSContext *cx, unsigned short *bytes, uint sz, jsval *vp);
```

Descrizione

Metodo; converte i byte in un valore JavaScript.

Argomenti

`JSContext` **cx*, `unsignedshort`**bytes*, `uint`*sz*, `jsval` **vp*

- L'argomento *cx* è il contesto JavaScript.
- L'argomento *bytes* è la stringa di byte da convertire in un oggetto JavaScript.
- L'argomento *sz* è il numero di byte da convertire.
- L'argomento *vp* è il valore JavaScript.

Restituisce

Un valore booleano: `JS_TRUE` indica un esito positivo; `JS_FALSE` indica un esito negativo.

JSVal JS_IntegerToValue()

Uso

```
jsval JS_IntegerToValue(long lv);
```

Descrizione

Metodo; converte il valore long di un numero intero nella struttura `JSVal`.

Argomenti

lv

L'argomento *lv* è il valore long dell'intero che si desidera convertire in una struttura `jsval`.

Restituisce
Una struttura `JSVal` che contiene l'intero che viene passato alla funzione sotto forma di argomento.

JSVal JS_ObjectToValue()

Uso
```
jsval JS_ObjectToValue(JSObject *obj);
```

Descrizione
Metodo; memorizza il valore restituito da un oggetto in una struttura `JSVal`. Utilizzate `JS_NewArrayObject()` per creare un oggetto array; utilizzate `JS_SetElement()` per definirne il contenuto.

Argomenti
`JSObject` *`*obj`*

L'argomento *obj* è un puntatore all'oggetto `JSObject` che si desidera convertire in una struttura `JSVal`.

Restituisce
Una struttura `JSVal` che contiene l'oggetto che è stato passato alla funzione sotto forma di argomento.

unsigned short *JS_ObjectType()

Uso
```
unsigned short *JS_ObjectType(JSObject *obj);
```

Descrizione
Metodo; dato un riferimento a un oggetto, restituisce il nome di classe dell'oggetto. Ad esempio, se l'oggetto è un oggetto DOM, la funzione restituisce "`Document`". Se l'oggetto è un nodo nel documento, la funzione restituisce "`Element`". Per un oggetto array, la funzione restituisce "`Array`".

Nota: *non modificare il puntatore buffer restituito per evitare di danneggiare le strutture di dati dell'interprete JavaScript.*

Argomenti
`JSObject` *`*obj`*

Generalmente questo argomento viene passato e convertito mediante la funzione `JS_ValueToObject()`.

Restituisce
Un puntatore a una stringa terminata da un byte nullo. Il chiamante non deve liberare questa stringa quando termina l'operazione.

JSObject *JS_NewArrayObject()

Uso
```
JSObject *JS_NewArrayObject(JSContext *cx, unsigned int length [, jsval *v])
```

Descrizione
Metodo; crea un nuovo oggetto che contiene un array di `JSVal`.

Argomenti
`JSContext *`*cx*, `unsigned int` *length*, `jsval *`*v*

- L'argomento *cx* è il puntatore opaco `JSContext` che viene passato alla funzione JavaScript.
- L'argomento *length* è il numero di elementi che l'array è in grado di contenere.
- L'argomento *v* è un puntatore opzionale ai `jsval` da memorizzare nell'array. Se il valore restituito è diverso da `null`, *v* è un array che contiene degli elementi *length*. Se il valore restituito è `null`, il contenuto iniziale dell'oggetto array non è definito e può essere impostato mediante la funzione `JS_SetElement()`.

Restituisce
Un puntatore a un nuovo oggetto array oppure, in caso di errore, il valore `null`.

long JS_GetArrayLength()

Uso
```
long JS_GetArrayLength(JSContext *cx, JSObject *obj)
```

Descrizione
Metodo; dato un puntatore a un oggetto array, ottiene il numero di elementi nell'array.

Argomenti
`JSContext *`*cx*, `JSObject*`*obj*

- L'argomento *cx* è il puntatore opaco `JSContext` che viene passato alla funzione JavaScript.
- L'argomento *obj* è un puntatore a un oggetto array.

Restituisce
Il numero di elementi nell'array oppure, in caso di errore, il valore -1.

JSBool JS_GetElement()

Uso
```
JSBool JS_GetElement(JSContext *cx, JSObject *obj, jsint idx, jsval *vp)
```

Descrizione
Metodo; legge un singolo elemento di un oggetto array.

Argomenti
`JSContext *`*cx*, `JSObject *`*obj*, `jsint` *idx*, `jsval *`*vp*

- L'argomento *cx* è il puntatore opaco `JSContext` che viene passato alla funzione JavaScript.
- L'argomento *obj* è un puntatore a un oggetto array.

- L'argomento *idx* è l'indice di un numero intero nell'array. Il primo elemento è l'indice `0`, mentre l'ultimo è l'indice (`length 1`-).
- L'argomento *vp* è un puntatore a un `jsval` in cui deve essere copiato il contenuto della struttura `jsval` nell'array.

Restituisce
Un valore booleano: `JS_TRUE` indica un esito positivo; `JS_FALSE` indica un esito negativo.

JSBool JS_SetElement()

Uso
```
JSBool JS_SetElement(JSContext *cx, JSObject *obj, jsint idx, jsval *vp)
```

Descrizione
Metodo; scrive un singolo elemento di un oggetto array.

Argomenti
`JSContext` *`*cx`*, `JSObject` *`*obj`*, `jsint` *`idx`*, `jsval` *`*vp`*

- L'argomento *cx* è il puntatore opaco `JSContext` che viene passato alla funzione JavaScript.
- L'argomento *obj* è un puntatore a un oggetto array.
- L'argomento *idx* è l'indice di un numero intero nell'array. Il primo elemento è l'indice `0`, mentre l'ultimo è l'indice (`length 1`-).
- L'argomento *vp* è un puntatore a una struttura `jsval` il cui contenuto deve essere copiato nel `jsval` nell'array.

Restituisce
Un valore booleano: `JS_TRUE` indica un esito positivo; `JS_FALSE` indica un esito negativo.

JSBool JS_ExecuteScript()

Uso
```
JS_ExecuteScript (JSContext *cx, JSObject *obj, unsigned short *script, unsigned int sz, jsval
*rval)
```

Descrizione
Metodo; compila ed esegue una stringa JavaScript. Se lo script restituisce un valore, ritorna in `*rval`.

Argomenti
`JSContext` *`*cx`*, `JSObject` *`*obj`*, `unsigned short` *`*script`*, `unsigned int`*`sz`*, `jsval` *`*rval`*

- L'argomento *cx* è il puntatore opaco `JSContext` che viene passato alla funzione JavaScript.
- L'argomento *obj* è un puntatore all'oggetto nel cui contesto viene eseguito lo script. Mentre lo script è in esecuzione, la parola chiave `this` è uguale a questo oggetto. Di solito si tratta del puntatore `JSObject` che viene passato alla funzione JavaScript.
- L'argomento *script* è una stringa che contiene del codice JavaScript. Se le dimensioni della stringa non sono specificate (consultate l'argomento *sz*), la stringa deve essere terminata da un byte nullo.

- L'argomento *sz* corrisponde alle dimensioni della stringa, espresse in byte. Se *sz* è uguale a 0, la lunghezza della stringa terminata da un byte nullo viene calcolata automaticamente.
- L'argomento *rval* è un puntatore a una singola struttura `jsval`. Il valore restituito dalla funzione viene memorizzato in `*rval`.

Restituisce

Un valore booleano: `JS_TRUE` indica un esito positivo; `JS_FALSE` indica un esito negativo.