

NHA TRANG - 2010

MACROMEDIA FLASH 8



ACTIONSRIPT **DICTIONARY**

dunglequang_ndthff@yahoo.com

http://www.vnfx.com/vnfx_assets/asdictionary/Flash/Actionscript_dictionary.htm

Tôi trả!

Cuốn sách này là món quà tôi muốn tặng cho những bạn trẻ nhiệt huyết với FLASH.

Chúc các bạn sẽ có những bài học thật bổ ích với những kiến thức trong cuốn sách này.

Tác giả!

Lê Quang Dũng

Giới thiệu	1
Contents of the dictionary {Các nội dung của từ điển}	3
Accessibility (object :đối tượng) {Sự dễ bị ảnh hưởng}	24
Accessibility.isActive	24
Button (object) : Nút (đối tượng)	25
Button.onDragOver	27
Color (object) : Màu sắc (đối tượng)	27
Color.getTransform	28
CustomActions (object) {Các hành vi tùy chỉnh (đối tượng)}	29
Date (object) : Ngày tháng (đối tượng)	29
delete	34
duplicateMovieClip	35
FCheckBox (component: thành phần)	36
FComboBox (component)	37
FPushButton (component)	40
FRadioButton (component: thành phần)	41
FScrollBar (component)	42
FScrollPane (component)	44
FStyleFormat (object)	45
Constructor (người xây dựng) cho đối tượng FStyleFormat	48
Function (object) : Hàm (đối tượng)	49
Function.prototype	49
fscommand	50
Function	52
getProperty	54
getTimer	54
getURL	55
_global	56
Key (object)	57
LoadVars (object) {Nạp các biến (đối tượng)}	59
_level	61
Math (object)	62
Mouse (object) {Con trỏ chuột (đối tượng)}	64
MovieClip (object)	64
MovieClip._url	69
MovieClip.useHandCursor	70
MovieClip._visible	70
MovieClip._width	70
MovieClip._x	71
MovieClip._xmouse	71

MovieClip._xscale	72
Number (function)	72
Constructor cho Object object	74
Object.registerClass	74
onClipEvent	76
_parent	78
on	78
print	89
random {ngẫu nhiên}	81
removeMovieClip	82
_root	82
scroll	83
Selection (object) {Sự lựa chọn (đối tượng)}	84
Sound (object) {Âm thanh (đối tượng)}	85
Stage (object) {Sân khấu (đối tượng)}	87
startDrag	88
String (function)	89
System.capabilities (object)	90
targetPath	91
tellTarget	92
TextField (object)	93
TextFormat (object) {Định dạng Text (đối tượng)}	97
this	99
toggleHighQuality	100
trace	101
typeof	102
undefined	103
unescape	104
unloadMovie	105
unloadMovieNum	106
updateAfterEvent	107
void	108
while	108
with	109
XML (object) : XML (đối tượng)	113
XML.send	116
XML.status	117
XMLSocket (object) {Ô cắm XML (đối tượng)}	117

Từ điển này mô tả cú pháp và cách sử dụng của yếu tố ActionScript trong Macromedia Flash MX. Để sử dụng các thí dụ trong một script, copy đoạn mã mẫu từ ActionScript Từ điển và dán nó vào trong bảng Actions ở chế độ expert mode.

Từ điển liệt kê tất cả các phần tử ActionScript — operators (các toán tử), keywords (từ khoá), statements (trình bày), actions (hoạt động), properties (thuộc tính), functions (hàm), objects (đối tượng), components (thành phần), và methods (phương pháp). Cho một khái quát về tất cả các mục của Từ điển, hãy xem [Contents of the dictionary](#) (Nội dung của từ điển); những bảng trong phần này là điểm khởi đầu tốt cho việc xem xét những thao tác hoặc cách thức có tính chất tượng trưng mà các đối tượng hoặc các lớp hợp thành chúng bạn không biết.

ActionScript theo tiêu chuẩn ECMA-262 (tiêu chuẩn đặc tả viết bởi hội liên hiệp những người sản xuất máy tính châu Âu: the specification written by the European Computer Manufacturers Association) ngoại trừ một số chú ý khác. Một số phần tử Flash 5 (và trước đó) ActionScript bị phản kháng và thay thế bởi các phần tử ActionScript mới phù hợp với tiêu chuẩn ECMA Standard. Bạn nên sử dụng các phần tử Flash MX mới, mặc dù các phần tử này vẫn được hỗ trợ bởi Flash Player 5.

Có hai loại mục từ trong từ điển này:

- ◆ Những mục từ riêng cho các toán tử (operators), từ khoá (keywords), hàm (functions), biến (variables), thuộc tính (properties), phương pháp (methods), và các câu lệnh (statements).
- ◆ Những mục từ đối tượng (Object) và thành phần hợp thành (component), cung cấp thông tin tổng quát về những đối tượng và các component Flash gắn liền.

Sử dụng thông tin trong các mục từ **sample** (ví dụ) để làm sáng tỏ cấu trúc và quy ước sử dụng trong hai kiểu mục từ này.

Sample entry for most ActionScript elements {Mục từ Ví dụ cho hầu hết các phần tử ActionScript}

Mục từ điển Ví dụ (Sample) giải thích các quy ước sử dụng đối với tất cả các phần tử ActionScript mà không phải là các đối tượng hay các components.

Tiêu đề mục từ

Tất cả các mục từ được liệt kê theo bảng chữ cái. Sự sắp xếp theo bảng chữ cái theo những chữ hoa, các từ được gạch dưới,...

Availability {**Có thể dùng được, có hiệu lực**}

Phần này cho biết phiên bản Flash Player nào hỗ trợ phần tử này. Điều này không giống với phiên bản của Flash đã dùng để tạo ra nội dung. Ví dụ, nếu bạn sử dụng công cụ thiết kế Flash MX để tạo nội dung cho Flash Player 5, bạn chỉ có thể sử dụng các phần tử ActionScript có hiệu lực với Flash Player 5.

Usage {**Cách sử dụng**}

Phần này cung cấp cú pháp chính xác cho việc sử dụng phần tử ActionScript trong đoạn mã của bạn. Phần quy định của cú pháp nằm trong *italicized code font*. Các dấu ngoặc vuông ([]) cho biết các tham số tùy chọn.

Parameters {Tham số}

Phần này mô tả bất kỳ tham số nào đã được liệt kê trong cú pháp.

Returns {Trả về}

Phần này cho biết những giá trị nào (nếu có) phần tử trả về.

Description {Phần mô tả}

Phần này nhận dạng kiểu của phần tử (ví dụ, là một toán tử operator, phương pháp method, hàm function, và ...) và sau đó diễn tả cách sử dụng phần tử.

Example {Ví dụ}

Phần này cung cấp một mã ví dụ giải thích cách sử dụng phần tử.

See also {Xem thêm}

Phần này liệt kê các mục từ điển ActionScript có liên quan.

Sample entry for objects and components {Mục từ ví dụ cho các đối tượng và các component}

Mục từ điển ví dụ sau đây giải nghĩa các quy ước đã dùng cho các đối tượng và các component ActionScript gắn kèm. Các đối tượng (object) và các component. Các đối tượng và các component đều được liệt kê theo bảng chữ cái với các phần tử khác trong từ điển. Các component Flash đều được liệt kê như FCheckBox, FComboBox,

Mục từ tiêu đề

The entry title provides the name of the object or component. The object or component name is followed by a paragraph containing general descriptive information.

Bảng tổng quan các method và thuộc tính

Mỗi mục từ đối tượng và component chứa một bảng liệt kê tất cả các method kết hợp. Nếu đối tượng hay component có các thuộc tính (thường là hằng số), những phần tử này được tóm tắt trong một bảng thêm. Tất cả các method và thuộc tính liệt kê trong những bảng này cũng có những mục từ riêng của chúng, theo sau mục từ đối tượng hoặc component.

Constructor

Nếu một đối tượng hoặc component yêu cầu bạn sử dụng một constructor (trình xây dựng) để truy cập các method và thuộc tính của nó, constructor được mô tả trong mỗi mục từ đối tượng hoặc component. Sự mô tả này có tất cả các phần tử chuẩn mực (cú pháp, phần mô tả, ...) của tất cả các mục từ điển khác.

Danh sách các method và property

Các method và thuộc tính của một đối tượng hoặc component đều được liệt kê theo thứ tự bảng chữ cái sau mục từ đối tượng hoặc component.

Contents of the dictionary {Các nội dung của từ điển}

Tất cả các mục từ đều được liệt kê theo bảng chữ cái. Tuy nhiên, một số toán tử và symbol (biểu tượng) đều được trình bày theo thứ tự ASCII. Trong phần bổ sung, các method kết hợp với một đối tượng hoặc component đều được liệt kê theo cùng với tên của đối tượng hoặc component—ví dụ, method `abs` của đối tượng toán `Math` được liệt kê trong `Math.abs`, và method `getValue` của component `FRadioButton` được liệt kê là `FRadioButton.getValue`.

Hai bảng sau đây giúp bạn định vị những phần tử này. Bảng đầu tiên liệt kê các toán tử tương trưng theo thứ tự mà chúng xuất hiện trong từ điển. Bảng thứ hai liệt kê tất cả các phần tử `ActionScript` khác.

Chú ý: Với tính thứ tự và kết hợp của các toán tử, hãy xem phụ lục Appendix A, [Operator Precedence and Associativity](#)

Symbolic operators	
--	-- (decrement)
++	++ (increment)
!	! (logical NOT)
!=	!= (inequality)
!==	!== (strict inequality)
%	% (modulo)
%=	%= (modulo assignment)
&	& (bitwise AND)
&&	&& (short-circuit AND)
&=	&= (bitwise AND assignment)
()	() (parentheses)
-	- (minus)
*	* (multiplication)

<code>*=</code>	<code>*=</code> (multiplication assignment)
<code>,</code>	<code>,</code> (comma)
<code>.</code>	<code>.</code> (dot)
<code>?:</code>	<code>?:</code> (conditional)
<code>/</code>	<code>/</code> (division)
<code>//</code>	<code>//</code> (comment delimiter)
<code>/*</code>	<code>/*</code> (comment delimiter)
<code>/=</code>	<code>/=</code> (division assignment)
<code>[]</code>	<code>[]</code> (array access)
<code>^</code>	<code>^</code> (bitwise XOR)
<code>^=</code>	<code>^=</code> (bitwise XOR assignment)
<code>{}</code>	<code>{}</code> (object initializer)
<code> </code>	<code> </code> (bitwise OR)
<code> </code>	<code> </code> (logical OR)
<code> =</code>	<code> =</code> (bitwise OR assignment)
<code>~</code>	<code>~</code> (bitwise NOT)
<code>+</code>	<code>+</code> (addition)
<code>+=</code>	<code>+=</code> (addition assignment)
<code><</code>	<code><</code> (less than)
<code><<</code>	<code><<</code> (bitwise left shift)
<code><<=</code>	<code><<=</code> (bitwise left shift and assignment)
<code><=</code>	<code><=</code> (less than or equal to)
<code><></code>	<code><></code> (inequality)
<code>=</code>	<code>=</code> (assignment)
<code>-=</code>	<code>-=</code> (subtraction assignment)
<code>==</code>	<code>==</code> (equality)
<code>===</code>	<code>===</code> (strict equality)
<code>></code>	<code>></code> (greater than)
<code>>=</code>	<code>>=</code> (greater than or equal to)
<code>>></code>	<code>>></code> (bitwise right shift)
<code>>>=</code>	<code>>>=</code> (bitwise right shift and assignment)

>>>	>>> (bitwise unsigned right shift)
>>>=	>>>= (bitwise unsigned right shift and assignment)

Bảng sau liệt kê tất cả các phần tử ActionScript không phải là các toán tử tượng trưng

Phần tử ActionScript	Xem mục từ
abs	Math.abs
acos	Math.acos
add	add
addItem	FComboBox.addItem , FListBox.addItem
addItemAt	FComboBox.addItemAt , FListBox.addItem
addListener	FStyleFormat.addListener , Key.addListener , Mouse.addListener , Selection.addListener , Stage.addListener , TextField.addListener
addProperty	Object.addProperty
and	and
align	Stage.align , TextFormat.align
_alpha	MovieClip._alpha , Button._alpha , TextField._alpha
appendChild	XML.appendChild
apply	Function.apply
applyChanges	FStyleFormat.applyChanges
Arguments	arguments (object)
Array	Array (object)
arrow	FStyleFormat.arrow
asfunction	asfunction
asin	Math.asin
atan	Math.atan
atan2	Math.atan2
attachMovie	MovieClip.attachMovie
attachSound	Sound.attachSound

attributes	XML.attributes
autosize	TextField.autoSize
background	FStyleFormat.background , TextField.background
backgroundColor	TextField.backgroundColor
backgroundDisabled	FStyleFormat.backgroundDisabled
BACKSPACE	Key.BACKSPACE
beginFill	MovieClip.beginFill
beginGradientFill	MovieClip.beginGradientFill
blockIndent	TextFormat.blockIndent
bold	TextFormat.bold
Boolean	Boolean (function) , Boolean (object)
border	TextField.border
borderColor	TextField.borderColor
bottomScroll	TextField.bottomScroll
break	break
bullet	TextFormat.bullet
Button	Button (object)
call	call , Function.call
call function	call function
callee	arguments.callee
caller	arguments.caller
capabilities	System.capabilities (object)
CAPSLOCK	Key.CAPSLOCK
case	case
ceil	Math.ceil
charAt	String.charAt
charCodeAt	String.charCodeAt
check	FStyleFormat.check
childNodes	XML.childNodes

chr	chr
clear	MovieClip.clear
clearInterval	clearInterval
cloneNode	XML.cloneNode
close	XMLSocket.close
Color	Color (object), TextFormat.color
concat	Array.concat, String.concat
connect	XMLSocket.connect
constructor	Array (object), Boolean (object), Color (object), Date (object), Number (function), Object (object), Sound (object), String (object), XML (object), XMLSocket (object)
contentType	LoadVars.contentType, XML.contentType
ccntinue	continue
CONTROL	Key.CONTROL
cos	Math.cos
createElement	XML.createElement
createEmptyMovieClip	MovieClip.createEmptyMovieClip
createTextField	MovieClip.createTextField
createTextNode	XML.createTextNode
_currentframe	MovieClip._currentframe
curveTo	MovieClip.curveTo
Date	Date (object)
darkshadow	FStyleFormat.darkshadow
default	default
delete	delete
DELETEKEY	Key.DELETEKEY
docTypeDecl	XML.docTypeDecl
do while	do while
DOWN	Key.DOWN
_droptarget	MovieClip._droptarget
duplicateMovieClip	duplicateMovieClip, MovieClip.duplicateMovieClip

duration	Sound.duration
E	Math.E
#endinitclip	#endinitclip
else	else
else if	else if
embedFonts	TextField.embedFonts
enabled	Button.enabled, MovieClip.enabled
END	Key.END
endFill	MovieClip.endFill
ENTER	Key.ENTER
eq	eq (equal-string specific)
escape (function)	escape
ESCAPE (constant)	Key.ESCAPE
eval	eval
evaluate	evaluate
exp	Math.exp
face	FStyleFormat.face
false	false
FCheckBox	FCheckBox (component)
FComboBox	FListBox (component)
firstChild	XML.firstChild
FListBox	FListBox (component)
floor	Math.floor
focusEnabled	MovieClip.focusEnabled
_focusrect	_focusrect, Button._focusrect, TextField._focusrect, MovieClip._focusrect
font	TextFormat.font
for	for
for..in	for..in
foregroundDisabled	FStyleFormat.foregroundDisabled

FPushButton	FPushButton (component)
FRadioButton	FPushButton (component)
_framesloaded	MovieClip._framesloaded
fromCharCode	String.fromCharCode
fscommand	fscommand
FScrollBar	FScrollBar (component)
FScrollPane	FScrollPane (component)
FStyleFormat	FStyleFormat (object)
function	function, Function (object)
ge	ge (greater than or equal to-string specific)
get	CustomActions.get
getAscii	Key.getAscii
getBeginIndex	Selection.getBeginIndex
getBounds	MovieClip.getBounds
getBytesLoaded	LoadVars.getBytesLoaded, MovieClip.getBytesLoaded, Sound.getBytesLoaded, XML.getBytesLoaded
getBytesTotal	LoadVars.getBytesTotal, MovieClip.getBytesTotal, Sound.getBytesTotal, XML.getBytesTotal
getCaretIndex	Selection.getCaretIndex
getCode	Key.getCode
getData	FRadioButton.getData
getDate	Date.getDate
getDay	Date.getDay
getDepth	Button.getDepth, MovieClip.getDepth, TextField.getDepth
getEnabled	FCheckBox.getEnabled, FComboBox.getEnabled, FListBox.getEnabled, FPushButton.getEnabled, FRadioButton.getEnabled, FScrollBar.getEnabled
getEndIndex	Selection.getEndIndex
getFocus	Selection.getFocus
getFontList	TextField.getFontList
getFullYear	Date.getFullYear

getHours	Date.getHours
getItemAt	FComboBox.getItemAt, FListBox.addItemAt
getLabel	FCheckBox.getLabel, FPushButton.getLabel, FRadioButton.getLabel
getLength	FComboBox.getLength, FListBox.getLength
getMilliseconds	Date.getMilliseconds
getMinutes	Date.getMinutes
getMonth	Date.getMonth
getNewTextFormat	TextField.getNewTextFormat
getPan	Sound.getPan
getPaneHeight	FScrollPane.getPaneHeight
getPaneWidth	FScrollPane.getPaneWidth
getProperty	getProperty
getRowCount	FComboBox.getRowCount, FListBox.getRowCount
getRGB	Color.getRGB
getScrollContent	FScrollPane.getScrollContent
getScrollPosition	FComboBox.getScrollPosition, FListBox.getScrollPosition, FScrollBar.getScrollPosition, FScrollPane.getScrollPosition
getSeconds	Date.getSeconds
getSelectedIndex	FComboBox.getSelectedIndex, FListBox.getSelectedIndex
getSelectedIndices	FListBox.getSelectedIndices
getSelectedItem	FComboBox.getSelectedItem, FListBox.getSelectedItem
getSelectedItems	FListBox.getSelectedItem
getSelectMultiple	FListBox.getSelectMultiple
getState	FRadioButton.getState
getTextExtent	TextFormat.getTextExtent
getTextFormat	TextField.getTextFormat
getTime	Date.getTime
getTimer	getTimer
getTimezoneOffset	Date.getTimezoneOffset

getTransform	Color.getTransform, Sound.getTransform
getURL	getURL, MovieClip.getURL
getUTCDate	Date.getUTCDate
getUTCDay	Date.getUTCDay
getUTCFullYear	Date.getUTCFullYear
getUTCHours	Date.getUTCHours
getUTCMilliseconds	Date.getUTCMilliseconds
getUTCMinutes	Date.getUTCMinutes
getUTCMonth	Date.getUTCMonth
getUTCSeconds	Date.getUTCSeconds
getValue	FCheckBox.getValue, FComboBox.getValue, FListBox.getValue, FRadioButton.getValue
getVersion	getVersion
getVolume	Sound.getVolume
getYear	Date.getYear
_global	_global
globalStyleFormat	globalStyleFormat
globalToLocal	MovieClip.globalToLocal
goto	gotoAndPlay, gotoAndStop
gotoAndPlay	gotoAndPlay, MovieClip.gotoAndPlay
gotoAndStop	gotoAndStop, MovieClip.gotoAndStop
gt	gt (greater than -string specific)
hasAccessibility	System.capabilities.hasAccessibility
hasAudio	System.capabilities.hasAudio
hasAudioEncoder	System.capabilities.hasMP3
hasMP3	System.capabilities.hasMP3
hasVideoEncoder	System.capabilities.language
hasChildNodes	XML.hasChildNodes
height	Stage.height
_height	MovieClip._height, TextField._height, Button._height

hide	Mouse.hide
highlight	FStyleFormat.highlight
highlight3D	FStyleFormat.highlight3D
_highquality	_highquality, Button._highquality, MovieClip._highquality, TextField._highquality
hitArea	MovieClip.hitArea
hitTest	MovieClip.hitTest
HOME	Key.HOME
hscroll	TextField.hscroll
html	TextField.html
htmlText	TextField.htmlText
if	if
ifFrameLoaded	ifFrameLoaded
ignoreWhite	XML.ignoreWhite
#include	#include
indent	TextFormat.indent
indexOf	String.indexOf
#initclip	#initclip
INSERT	Key.INSERT
insertBefore	XML.insertBefore
install	CustomActions.install
instanceof	instanceof
int	int
isActive	Accessibility.isActive
isDown	Key.isDown
isFinite	isFinite
isNaN	isNaN
isToggled	Key.isToggled
italic	TextFormat.italic
join	Array.join

Key	Key (object)
language	System.capabilities.language
lastChild	XML.lastChild
lastIndexOf	String.lastIndexOf
le	le (less than or equal to – string specific)
leading	TextFormat.leading
LEFT	Key.LEFT
leftMargin	TextFormat.leftMargin
length	arguments.length, Array.length, String.length, Sound.loadSound, TextField.length
level	_level
lineStyle	MovieClip.lineStyle
lineTo	MovieClip.lineTo
list	CustomActions.uninstall
LN2	Math.LN2
LN10	Math.LN10
load	XML.load, LoadVars.load
loaded	XML.loaded, LoadVars.loaded
loadMovie	loadMovie, MovieClip.loadMovie
loadMovieNum	loadMovieNum
loadScrollContent	FScrollPane.loadScrollContent
loadSound	Sound.loadSound
loadVariables	loadVariables, MovieClip.loadVariables
loadVariablesNum	loadVariablesNum
LoadVars	LoadVars (object)
localToGlobal	MovieClip.localToGlobal
log	Math.log
LOG2E	Math.LOG2E
LOG10E	Math.LOG10E
lt	lt (less than – string specific)

manufacturer	<code>System.capabilities.manufacturer</code>
Math	<code>Math</code> (object)
max	<code>Math.max</code>
maxChars	<code>TextField.maxChars</code>
maxhscroll	<code>TextField.maxhscroll</code>
maxscroll	<code>maxscroll</code> , <code>TextField.maxscroll</code>
MAX_VALUE	<code>Number.MAX_VALUE</code>
mbchr	<code>mbchr</code>
mblength	<code>mblength</code>
mbord	<code>mbord</code>
mbsubstring	<code>mbsubstring</code>
method	<code>method</code>
min	<code>Math.min</code>
MIN_VALUE	<code>Number.MIN_VALUE</code>
Mouse	<code>Mouse</code> (object)
moveTo	<code>MovieClip.moveTo</code>
MovieClip	<code>MovieClip</code> (object)
multiline	<code>TextField.multiline</code>
_name	<code>MovieClip._name</code> , <code>TextField._name</code> , <code>Button._name</code>
NaN	<code>NaN</code> , <code>Number.NaN</code>
ne	<code>ne</code> (not equal – string specific)
NEGATIVE_INFINITY	<code>Number.NEGATIVE_INFINITY</code>
new (operator)	<code>new</code>
newline	<code>newline</code>
nextFrame	<code>nextFrame</code> , <code>MovieClip.nextFrame</code>
nextScene	<code>nextScene</code>
nextSibling	<code>XML.nextSibling</code>
nodeName	<code>XML.nodeName</code>
nodeType	<code>XML.nodeType</code>
nodeValue	<code>XML.nodeValue</code>

not	not
null	null
Number	Number (function), Number (object)
Object	Object (object)
on	on
onClipEvent	onClipEvent
onClose	XMLSocket.onClose
onChanged	TextField.onChanged
onConnect	XMLSocket.onConnect
onData	XML.onData, XMLSocket.onData
onDragOut	Button.onDragOut, MovieClip.onDragOut
onDragOver	Button.onDragOver, MovieClip.onDragOver
onEnterFrame	MovieClip.onEnterFrame
onKeyDown	Button.onKeyDown, Key.onKeyDown, MovieClip.onKeyDown
onKeyUp	Button.onKeyUp, Key.onKeyUp, MovieClip.onKeyUp
onKillFocus	Button.onKillFocus, MovieClip.onKillFocus, TextField.onKillFocus
onLoad	LoadVars.onLoad, MovieClip.onLoad, Sound.onLoad, XML.onLoad
onMouseDown	Mouse.onMouseDown, MovieClip.onMouseDown
onMouseMove	Mouse.onMouseMove, MovieClip.onMouseMove
onMouseUp	Mouse.onMouseUp, MovieClip.onMouseUp
onPress	Button.onPress, MovieClip.onPress
onRelease	Button.onRelease, MovieClip.onRelease
onReleaseOutside	Button.onReleaseOutside, MovieClip.onReleaseOutside
onResize	Stage.onResize
onRollOut	Button.onRollOut
onRollOver	Button.onRollOver
onScroller	TextField.onScroller
onSetFocus	Button.onSetFocus, MovieClip.onSetFocus,

	<code>Selection.onSetFocus, TextField.onSetFocus</code>
<code>onSort</code>	<code>Array.pop</code>
<code>onSoundComplete</code>	<code>Sound.onSoundComplete</code>
<code>onUnload</code>	<code>MovieClip.onUnload</code>
<code>onXML</code>	<code>XMLSocket.onXML</code>
<code>or (logical OR)</code>	<code>or</code>
<code>ord</code>	<code>ord</code>
<code>os</code>	<code>System.capabilities.os</code>
<code>_parent</code>	<code>_parent, Button._parent, MovieClip._parent, TextField._parent</code>
<code>parentNode</code>	<code>XML.parentNode</code>
<code>parseFloat</code>	<code>parseFloat</code>
<code>parseInt</code>	<code>parseInt</code>
<code>parseXML</code>	<code>XML.parseXML</code>
<code>password</code>	<code>TextField.password</code>
<code>PGDN</code>	<code>Key.PGDN</code>
<code>PGUP</code>	<code>Key.PGUP</code>
<code>PI</code>	<code>Math.PI</code>
<code>pixelAspectRatio</code>	<code>System.capabilities.pixelAspectRatio</code>
<code>play</code>	<code>play, MovieClip.play</code>
<code>pop</code>	<code>Array.pop</code>
<code>position</code>	<code>Sound.position</code>
<code>POSITIVE_INFINITY</code>	<code>Number.POSITIVE_INFINITY</code>
<code>pow</code>	<code>Math.pow</code>
<code>prevFrame</code>	<code>prevFrame, MovieClip.prevFrame</code>
<code>previousSibling</code>	<code>XML.previousSibling</code>
<code>prevScene</code>	<code>prevScene</code>
<code>print</code>	<code>print</code>
<code>printAsBitmap</code>	<code>printAsBitmap</code>
<code>printAsBitmapNum</code>	<code>printAsBitmapNum</code>

printNum	printNum
__proto__	Object.__proto__
push	Array.push
_quality	_quality, TextField._quality, Button._quality
radioDot	FStyleFormat.radioDot
random	random, Math.random
refreshPane	FScrollPane.refreshPane
registerClass	Object.registerClass
registerSkinElement	FCheckBox.registerSkinElement, FComboBox.registerSkinElement, FListBox.registerSkinElement, FPushButton.registerSkinElement, FRadioButton.registerSkinElement, FScrollPane.registerSkinElement, FScrollPane.registerSkinElement
removeAll	FComboBox.removeAll, FListBox.removeAll
removeItemAt	FComboBox.removeItemAt, FListBox.removeItemAt
removeListener	FStyleFormat.removeListener, Key.removeListener, Mouse.removeListener, Selection.removeListener, Stage.removeListener, TextField.removeListener
removeMovieClip	removeMovieClip, MovieClip.removeMovieClip
removeNode	XML.removeNode
removeTextField	TextField.removeTextField
replaceItemAt	FComboBox.replaceItemAt, FListBox.replaceItemAt
replaceSel	TextField.replaceSel
resolutionX	Capabilities.screenResolutionX
resolutionY	Capabilities.screenResolutionY
restrict	TextField.restrict
return	return
reverse	Array.reverse
RIGHT	Key.RIGHT
rightMargin	TextFormat.rightMargin

<code>_root</code>	<code>_root</code>
<code>_rotation</code>	<code>MovieClip._rotation, Button._rotation, TextField._rotation</code>
<code>round</code>	<code>Math.round</code>
<code>scaleMode</code>	<code>Stage.scaleMode</code>
<code>screenColor</code>	<code>System.capabilities.screenColor</code>
<code>screenDPI</code>	<code>System.capabilities.screenDPI</code>
<code>screenResolution.x</code>	<code>System.capabilities.screenResolution.x</code>
<code>screenResolution.y</code>	<code>System.capabilities.screenResolution.y</code>
<code>scroll</code>	<code>scroll, TextField.scroll</code>
<code>scrollTrack</code>	<code>FStyleFormat.scrollTrack</code>
<code>selectable</code>	<code>TextField.selectable</code>
<code>selection</code>	<code>FStyleFormat.selection</code>
<code>Selection</code>	<code>Selection (object)</code>
<code>selectionDisabled</code>	<code>FStyleFormat.selectionDisabled</code>
<code>selectionUnfocused</code>	<code>FStyleFormat.selectionUnfocused</code>
<code>send</code>	<code>LoadVars.send, XML.send, XMLSocket.send</code>
<code>sendAndLoad</code>	<code>LoadVars.sendAndLoad, XML.sendAndLoad</code>
<code>set variable</code>	<code>set variable</code>
<code>setAutoHideScrollBar</code>	<code>FListBox.setAutoHideScrollBar</code>
<code>setChangeHandler</code>	<code>FCheckBox.setChangeHandler, FComboBox.setChangeHandler, FListBox.setChangeHandler, FRadioButton.setChangeHandler, FScrollBar.setChangeHandler</code>
<code>setClickHandler</code>	<code>FPushButton.setClickHandler</code>
<code>setData</code>	<code>FRadioButton.setData</code>
<code>setDataProvider</code>	<code>FComboBox.setDataProvider, FListBox.setDataProvider</code>
<code>setDate</code>	<code>Date.setDate</code>
<code>setDragContent</code>	<code>FScrollPane.setDragContent</code>
<code>setEditable</code>	<code>FComboBox.setEditable</code>
<code>setEnabled</code>	<code>FCheckBox.setEnabled, FComboBox.setEnabled, FListBox.setEnabled, FPushButton.setEnabled, FRadioButton.setEnabled, FScrollBar.setEnabled</code>

setFocus	<code>Selection.setFocus</code>
setFullYear	<code>Date.setFullYear</code>
setGroupName	<code>FRadioButton.setGroupName</code>
setHorizontal	<code>FScrollBar.setHorizontal</code>
setHours	<code>Date.setHours</code>
setHScroll	<code>FScrollPane.setHScroll</code>
setInterval	<code>setInterval</code>
setItemSymbol	<code>FComboBox.setItemSymbol, FListBox.setItemSymbol</code>
setLabel	<code>FCheckBox.setLabel, FRadioButton.setLabel, FPushButton.setLabel</code>
setLabelPlacement	<code>FCheckBox.setLabelPlacement, FRadioButton.setLabelPlacement</code>
setLargeScroll	<code>FScrollBar.setLargeScroll</code>
setMask	<code>MovieClip.setMask</code>
setMilliseconds	<code>Date.setMilliseconds</code>
setMinutes	<code>Date.setMinutes</code>
setMonth	<code>Date.setMonth</code>
setNewTextFormat	<code>TextField.setNewTextFormat</code>
setPan	<code>Sound.setPan</code>
setProperty	<code>setProperty</code>
setRGB	<code>Color.setRGB</code>
setRowCount	<code>FComboBox.setRowCount, FListBox.setRowCount</code>
setScrollContent	<code>FScrollBar.setSize, FScrollPane.setScrollContent</code>
setScrollPosition	<code>FListBox.setScrollPosition, FScrollBar.setScrollPosition, FScrollPane.setScrollPosition</code>
setScrollProperties	<code>FScrollBar.setScrollProperties</code>
setScrollTarget	<code>FScrollBar.setScrollTarget</code>
setSeconds	<code>Date.setSeconds</code>
setSelectedIndex	<code>FComboBox.setSelectedIndex, FListBox.setSelectedIndex</code>
setSelectedIndices	<code>FListBox.setSelectedIndices</code>
setSelection	<code>Selection.setSelection</code>

setSelectMultiple	FListBox.setSelectMultiple
setSize	FCheckBox.setSize, FComboBox.setSize, FListBox.setSize, FPushButton.setSize, FScrollBar.setSize, FScrollPane.setSize
setSmallScroll	FScrollBar.setSmallScroll
setState	FRadioButton.setState
setStyleProperty	FCheckBox.setStyleProperty, FComboBox.setStyleProperty, FListBox.setStyleProperty, FPushButton.setStyleProperty, FRadioButton.setStyleProperty, FScrollBar.setStyleProperty, FScrollPane.setStyleProperty
setTextFormat	TextField.setTextFormat
setTime	Date.setTime
setTransform	Color.setTransform, Sound.setTransform
setUTCDate	Date.setUTCDate
setUTCFullYear	Date.setUTCFullYear
setUTCHours	Date.setUTCHours
setUTCMilliseconds	Date.setUTCMilliseconds
setUTCMinutes	Date.setUTCMinutes
setUTCMonth	Date.setUTCMonth
setUTCSeconds	Date.setUTCSeconds
setValue	FCheckBox.setValue, FComboBox.setValue, FRadioButton.setValue
setVolume	Sound.setVolume
setVScroll	FScrollPane.setVScroll
setWidth	FListBox.setWidth
setYear	Date.setYear
shadow	FStyleFormat.shadow
shift (method)	Array.shift
SHIFT (constant)	Key.SHIFT
show	Mouse.show
sin	Math.sin
size	TextFormat.size

slice	Array.slice, String.slice
sort	Array.sort
sortItemsBy	FComboBox.sortItemsBy, FListBox.sortItemsBy
Sound	Sound (object)
_soundbuftime	_soundbuftime, TextField._soundbuftime, MovieClip._soundbuftime, Button._soundbuftime
SPACE	Key.SPACE
splice	Array.splice
split	String.split
sqrt	Math.sqrt
SQRT1_2	Math.SQRT1_2
SQRT2	Math.SQRT2
start	Sound.start
startDrag	startDrag, MovieClip.startDrag
status	XML.status
stop	stop, MovieClip.stop, Sound.stop
stopAllSounds	stopAllSounds
stopDrag	stopDrag, MovieClip.stopDrag
String	String (function), String (object)
substr	String.substring
substring	substring, String.substring
super	super
swapDepths	MovieClip.swapDepths
switch	switch
System	System (object)
TAB	Key.TAB
tabChildren	MovieClip.tabChildren
tabEnabled	Button.tabEnabled, TextField.tabEnabled, MovieClip.tabEnabled
tabIndex	Button.tabIndex, MovieClip.tabIndex, TextField.tabIndex

tabStops	<code>TextFormat.tabStops</code>
tan	<code>Math.tan</code>
target	<code>TextFormat.target</code>
_target	<code>Button._target, MovieClip._target, TextField._target</code>
targetPath	<code>targetPath</code>
tellTarget	<code>tellTarget</code>
text	<code>TextField.text</code>
textAlign	<code>FStyleFormat.textAlign</code>
textBold	<code>FStyleFormat.textBold</code>
textColor	<code>FStyleFormat.textColor, TextField.textColor</code>
textDisabled	<code>FStyleFormat.textDisabled</code>
TextField	<code>TextField</code> (object)
textFont	<code>FStyleFormat.textFont</code>
TextFormat	<code>TextFormat</code> (object)
textHeight	<code>TextField.textHeight</code>
textIndent	<code>FStyleFormat.textIndent</code>
textItalic	<code>FStyleFormat.textItalic</code>
textLeftMargin	<code>FStyleFormat.textLeftMargin</code>
textRightMargin	<code>FStyleFormat.textRightMargin</code>
textSelected	<code>FStyleFormat.textSelected</code>
textSize	<code>FStyleFormat.textSize</code>
textUnderline	<code>FStyleFormat.textUnderline</code>
textWidth	<code>TextField.textWidth</code>
this	<code>this</code>
toggleHighQuality	<code>toggleHighQuality</code>
toLowerCase	<code>String.toLowerCase</code>
toString	<code>Array.toString, Boolean.toString, Date.toString, Number.toString, Object.toString, XML.toString</code>
_totalframes	<code>MovieClip._totalframes</code>
toUpperCase	<code>String.toUpperCase</code>

trace	trace
trackAsMenu	Button.trackAsMenu, MovieClip.trackAsMenu
true	true
type	TextField.type
typeof	typeof
undefined	undefined
underline	TextFormat.underline
unescape	unescape
uninstall	CustomActions.uninstall
unloadMovie	unloadMovie, MovieClip.unloadMovie
unLoadMovieNum	unloadMovieNum
unshift	Array.unshift
unwatch	Object.unwatch
UP	Key.UP
updateAfterEvent	updateAfterEvent
url	TextFormat.url
_url	MovieClip._url, TextField._url, Button._url
useHandCursor	Button.useHandCursor, MovieClip.useHandCursor
UTC	Date.UTC
valueOf	Boolean.valueOf, Number.valueOf, Object.valueOf
var	var
variable	TextField.variable
version	System.capabilities.version
_visible	MovieClip._visible, Button._visible, TextField._visible
void	void
watch	Object.watch
while	while
width	Stage.width
_width	MovieClip._width, TextField._width, Button._width

with	with
wordwrap	TextField.wordWrap
_x	Button._x , MovieClip._x , TextField._x
XML	XML (object)
xmlDecl	XML.xmlDecl
XMLSocket	XMLSocket (object)
_xmouse	Button._xmouse , MovieClip._xmouse , TextField._xmouse
_xscale	Button._xscale , MovieClip._xscale , TextField._xscale
_y	Button._y , MovieClip._y , TextField._y
_ymouse	Button._ymouse , MovieClip._ymouse , TextField._ymouse
_yscale	Button._yscale , MovieClip._yscale , TextField._yscale

Accessibility (object :đối tượng) {Sự dễ bị ảnh hưởng}

Accessibility object là một tập hợp những phương pháp mà bạn có thể sử dụng để tạo ra các nội dung có thể áp dụng với ActionScript. Trong Flash MX, chỉ có một cách.

Object này chỉ có giá trị với Flash Player 6.

Tóm lược phương pháp cho đối tượng Arguments

Property (Thuộc tính)	Description (Mô tả)
Accessibility.isActive	Chỉ ra một chương trình đọc màn hình có hoạt động tích cực không

Accessibility.isActive

Availability (Có hiệu lực với)

Flash Player 6.

Usage (Sử dụng)

Accessibility.isActive()

Parameters (Tham số)

None. {không}

Returns (Trả về)

Một giá trị logic.

Description (Mô tả)

Phương pháp; chỉ ra xem một chương trình đọc màn hình có thực sự đang hoạt động hay không. Sử dụng cách thức này khi bạn muốn movie của bạn trình diễn khác nhau trong sự hiện diện của một người đọc màn ảnh (Use this method when you want your movie to behave differently in the presence of a screen reader).

See also (Xem thêm)

[System.capabilities.hasAccessibility](#)

Button (object) : Nút (đối tượng)

Tất cả các ký hiệu nút (button symbols) trong một Flash movie là các thể hiện (instance) của đối tượng Button (nút). Bạn có thể gán cho một button một tên minh họa (an instance name) trong Property inspector, và sử dụng các phương pháp (method) và các thuộc tính (properties) của đối tượng Button để thao tác các nút với ActionScript. Các tên minh họa (instance name) của nút được hiển thị trong hộp Movie Explorer và trong hộp thoại Insert Target Path trong panel Actions.

Đối tượng Button kế thừa từ đối tượng Object.

Đối tượng Button được hỗ trợ bởi Flash Player 6.

Cách thức tóm lược cho đối tượng Button

Method	Description
Button.getDepth	Trả lại độ dày của một thể hiện nút (button instance).

Thuộc tính tóm lược cho đối tượng Button

Property	Description
Button._alpha	Giá trị thuộc tính trong suốt của một thể hiện nút.
Button.enabled	Cho biết một nút có hoạt động hay không.
Button._focusrect	Chỉ báo liệu có phải một nút với tiêu điểm có một hình chữ nhật màu

	vàng bao quanh nó không.
<code>Button._height</code>	Chiều cao của một thể hiện nút, tính theo pixel.
<code>Button._highquality</code>	Cho biết chất lượng trình diễn của đoạn phim movie
<code>Button._name</code>	Tên minh họa (instance name) của một thể hiện nút.
<code>Button._parent</code>	Một sự liên hệ tới một thể hiện movie clip mà là cha mẹ của thể hiện này.
<code>Button._quality</code>	Cho biết chất lượng trình diễn của đoạn phim.
<code>Button._rotation</code>	Mức độ xoay vòng của một thể hiện nút.
<code>Button._soundbuftime</code>	Số giây cho một âm thanh để preload.
<code>Button.tabEnabled</code>	Cho biết một nút có bao gồm cả trong automatic tab ordering (Indicates whether a button is included in automatic tab ordering).
<code>Button.tabIndex</code>	Cho biết bảng thứ tự của một đối tượng (Indicates the tab order of an object).
<code>Button._target</code>	Đường dẫn tới đích của một thể hiện nút.
<code>Button.trackAsMenu</code>	Cho biết có các nút khác có thể nhận các sự kiện mouse release (Chẳng hiểu dịch thế nào).
<code>Button._url</code>	Địa chỉ của file SWF đã tạo ra thể hiện nút.
<code>Button.useHandCursor</code>	Cho biết có phải con chuột hình bàn tay được hiển thị khi chuột di chuyển qua một nút hay không.
<code>Button._visible</code>	Một giá trị Boolean xác định một thể hiện nút có tính ẩn (hidden) hay có thể nhìn thấy được (visible).
<code>Button._width</code>	Độ rộng của một thể hiện nút, tính theo pixel.
<code>Button._x</code>	Tọa độ x của một thể hiện nút.
<code>Button._xmouse</code>	Tọa độ x của con trỏ so với một thể hiện nút.
<code>Button._xscale</code>	Giá trị xác định tỷ lệ phần trăm theo chiều ngang của một thể hiện nút (The value specifying the percentage for horizontally scaling a button instance).
<code>Button._y</code>	Tọa độ y của một thể hiện nút.
<code>Button._ymouse</code>	Tọa độ y của con trỏ so với một thể hiện nút.
<code>Button._yscale</code>	Giá trị xác định tỷ lệ phần trăm theo chiều dài của một thể hiện nút (The value specifying the percentage for vertically scaling a button instance).

Button.onDragOver

Availability

Flash Player 6.

Usage

```
myButton.onDragOver
```

Parameters

None.

Returns

Nothing.

Description

Bộ quản lý sự kiện; được viện dẫn khi người dùng nhấn và di chuyển con trỏ chuột (trong khi vẫn giữ phím trên chuột) ra ngoài và sau đó lại lên trên nút.

Bạn phải định nghĩa một hàm để thực thi khi sự kiện được viện dẫn.

Example

Ví dụ dưới đây định nghĩa một hàm cho phương pháp `onKeyDown` để gửi một `trace` (dấu vết) ra cửa sổ hiển thị (Output window):

```
myButton.onDragOver = function () { trace ("onDragOver called");};
```

Color (object) : Màu sắc (đối tượng)

Đối tượng `Color` (Màu sắc) cho phép bạn thiết đặt giá trị màu RGB (ba màu cơ bản Red, Green và Blue) và thay đổi màu sắc của các movie clip và khôi phục lại các giá trị đó một khi chúng đã được thiết đặt.

Bạn phải sử dụng constructor (người xây dựng) `new Color()` để tạo một thể hiện (instance) của đối tượng `Color` trước khi gọi các method (phương pháp) của nó.

Đối tượng `Color` được hỗ trợ bởi Flash 5 và các phiên bản về sau của Flash Player.

Bảng tóm lược các method của đối tượng `Color`

Method	Description
--------	-------------

Color.getRGB	Trả lại tập hợp giá trị RGB thiết lập bởi lần cuối cùng gọi <code>setRGB</code> .
Color.getTransform	Trả lại tập hợp thông tin biến đổi thiết lập bởi lần gọi <code>setTransform</code> cuối cùng.
Color.setRGB	Thiết đặt sự thể hiện ở hệ 16 của giá trị RGB cho một đối tượng <code>Color</code> .
Color.setTransform	Thiết đặt sự biến đổi cho một đối tượng màu <code>Color</code> .

Người xây dựng cho đối tượng màu (Constructor for the Color object)

Availability

Flash Player 5.

Usage

```
new Color(target)
```

Parameters

`target` Tên minh họa của một movie clip.

Returns

Nothing.

Description

Người xây dựng (Constructor); tạo ra một thể hiện của đối tượng `Color` cho movie clip được chỉ rõ bởi tham số `target`. Bạn có thể dùng các method (cách thức, phương pháp) của đối tượng `Color` để thay đổi màu sắc của toàn bộ movie clip đích (entire target movie clip).

Example

Ví dụ dưới đây tạo một thể hiện của đối tượng màu sắc `Color` được gọi là `myColor` cho movie clip `myMovieClip` và thiết đặt giá trị RGB của nó:

```
myColor = new Color(myMovieClip);myColor.setRGB(0xff9933);
```

Color.getTransform

Availability

Flash Player 5.

Usage

`myColor.getTransform()`

Parameters

None.

Returns

Nothing.

Description

Phương pháp (Method); trả lại giá trị thay đổi thiết đặt bởi lần gọi `setTransform` cuối cùng.

See also

[Color.setTransform](#)

CustomActions (object) {Các hành vi tùy chỉnh (đối tượng)}

Các method của đối tượng CustomActions cho phép một Flash movie đang chạy trong công cụ thiết kế Flash được quản lý bất kỳ hành vi tùy chỉnh nào đã đăng ký bởi công cụ thiết kế. Một Flash movie có thể cài đặt và gỡ bỏ các hành vi tùy chỉnh, tiếp nhận định nghĩa XML của một hành vi tùy chỉnh, và truy lục danh sách các hành vi tùy chỉnh đã đăng ký.

Bạn có thể sử dụng các method này để xây dựng các Flash movies với các phần mở rộng của công cụ thiết kế Flash. Chẳng hạn như một phần mở rộng của movie có thể là, ví dụ, sử dụng giao thức trình ứng dụng Flash (Flash Application Protocol) để điều hướng một kho UDDI và tải xuống các dịch vụ Web vào trong hộp công cụ Actions toolbox.

Bảng tổng quan các method cho đối tượng CustomActions

Method	Description
CustomActions.get	Đọc các nội dung của một file XML định nghĩa hành vi tùy chỉnh.
CustomActions.install	Cài đặt một file định nghĩa hành vi tùy chỉnh XML mới (Installs a new custom action XML definition file).
CustomActions.list	Trả về một danh sách tất cả hành vi tùy chỉnh đã đăng ký.
CustomActions.uninstall	Gỡ bỏ một file định nghĩa hành vi tùy chỉnh XML .

Date (object) : Ngày tháng (đối tượng)

Đối tượng ngày tháng Date cho phép bạn gọi (retrieve) các giá trị ngày tháng và thời gian so với giờ quốc tế (Greenwich Mean Time là giờ chuẩn căn cứ theo kinh tuyến Greenwich, giờ GMT, hiện tại được gọi là giờ quốc tế hoặc UTC) hoặc so với hệ điều hành mà Flash Player đang chạy. Các phương pháp (method) của đối tượng ngày tháng Date không phải là tĩnh, nhưng chỉ áp dụng cho các thể hiện riêng lẻ của đối tượng Date được chỉ rõ khi method được gọi. Phương pháp `Date.UTC` là một ngoại lệ; nó là một method tĩnh.

Đối tượng Date quản lý thời gian kéo dài giờ làm việc ban ngày mùa hạ (để tiết kiệm điện ...) (tôi dịch từ nguyên gốc là **daylight saving time**) khác nhau tùy thuộc vào hệ điều hành và phiên bản Flash Player. Flash Player 6 điều khiển daylight saving time trên các hệ điều hành theo các cách sau:

- ♦ Windows—đối tượng Date tự động điều chỉnh đầu ra (thông tin do máy tính cung cấp) của nó cho daylight saving time. Đối tượng Date phát hiện ra **daylight saving time** được dùng trong địa phương hiện tại (current locale), và nếu như thế, nó nhận ra tiêu chuẩn chuyển tiếp tới thời gian kéo dài giờ làm việc ban ngày mùa hạ (it detects what the standard-to-daylight-saving-time transition date and times are). Tuy nhiên, những ngày tháng chuyển tiếp hiện thời trong thực tế được ứng dụng tới những ngày tháng trong quá khứ và tương lai, vì thế thời gian kéo dài giờ làm việc ban ngày mùa hạ thiên lệch có thể được tính toán không chính xác cho ngày tháng trong quá khứ khi địa phương có những ngày tháng chuyển tiếp khác nhau.
- ♦ Mac OS 8 và 9—đối tượng Date dùng thời gian kéo dài giờ làm việc ban ngày mùa hạ thực tế (object uses the current daylight-saving-time bias), không quan tâm đến ngày tháng hoặc thời gian đang được tính toán. Ví dụ, ở khu vực có chung giờ chuẩn U.S. Pacific trong thời gian tháng tám August, khi thời gian kéo dài giờ làm việc mùa hạ (DST) có hiệu lực, một đối tượng chứa đựng ngày tháng Jan 1 2001 vẫn tưởng trình thời gian DST mặc dù DST không có hiệu lực trong tháng một January. Vấn đề này không được sửa chữa trên Mac OS 8 hoặc 9 bởi vì một cơ sở dữ liệu thông tin khu vực có cùng giờ chuẩn không có hiệu lực (because a time zone information database is not available).
- ♦ Mac OS X—đối tượng Date tự động điều chỉnh đầu ra của nó cho thời gian kéo dài giờ làm việc mùa hè (daylight saving time). Cơ sở dữ liệu thông tin timezone trong Mac OS X được sử dụng để xác định liệu có phải bất kỳ ngày tháng hoặc thời gian nào trong hiện tại hoặc quá khứ cần phải có một thời gian kéo dài giờ làm việc mùa hè thiên lệch được ứng dụng hay không (used to determine whether any date or time in the present or past should have a daylight-saving-time bias applied).

Flash Player 5 xử lý thời gian kéo dài giờ làm việc mùa hè (daylight saving time) trên các hệ điều hành như sau:

- ♦ Mac OS 8 và 9—tương tự như đối với Flash Player 6.
- ♦ Windows—các quy tắc U.S. cho **daylight saving time** luôn luôn được áp dụng, dẫn tới sự chuyển tiếp không đúng ở Châu Âu (Europe) và các vùng khác dùng **daylight saving time** nhưng có thời gian chuyển tiếp khác nhau hơn là U.S. Flash phát hiện một cách chính xác DST có được sử dụng trong locale hiện tại không. (but have different transition times than the U.S. Flash correctly detects whether DST is employed in the current locale).

Để gọi các method của đối tượng Date, đầu tiên bạn phải tạo một thể hiện (instance) của đối tượng Date sử dụng người xây dựng (constructor) cho đối tượng Date.

Đối tượng Date đòi hỏi phải có Flash Player 5.

Bảng tóm lược các method của đối tượng Date object

Method	Description
Date.getDate	Trả lại ngày trong tháng theo thời gian cục bộ (local time : tôi nghĩ thời gian cục bộ ở đây chính là thời gian trên máy tính của bạn).
Date.getDay	Trả lại ngày trong tuần theo thời gian cục bộ.
Date.getFullYear	Trả lại năm bốn chữ số theo thời gian cục bộ.
Date.getHours	Trả lại giờ theo thời gian cục bộ.
Date.getMilliseconds	Trả lại các mili-giây (một phần nghìn giây) theo thời gian cục bộ.
Date.getMinutes	Trả lại phút theo thời gian cục bộ.
Date.getMonth	Trả lại tháng theo thời gian cục bộ.
Date.getSeconds	Trả lại giây theo thời gian cục bộ.
Date.getTime	Trả lại số mili-giây kể từ mười hai giờ đêm ngày 1 tháng Giêng, năm 1970, giờ thế giới.
Date.getTimezoneOffset	Trả lại sự chênh lệch, theo phút, giữa thời gian cục bộ trên các máy tính và thời gian quốc tế (universal time).
Date.getUTCDate	Trả lại ngày (date) của tháng theo thời gian quốc tế.
Date.getUTCDay	Trả lại ngày trong tuần theo thời gian quốc tế.
Date.getUTCFullYear	Trả lại năm bốn chữ số theo thời gian quốc tế.
Date.getUTCHours	Trả lại giờ theo thời gian quốc tế.
Date.getUTCMilliseconds	Trả lại mili-giây theo thời gian quốc tế.
Date.getUTCMinutes	Trả lại phút theo thời gian quốc tế.
Date.getUTCMonth	Trả lại tháng theo thời gian quốc tế.
Date.getUTCSeconds	Trả lại số giây đồng hồ theo thời gian quốc tế.
Date.getYear	Trả lại năm theo thời gian cục bộ.
Date.setDate	Thiết lập ngày trong tháng theo thời gian cục bộ. Trả lại thời gian mới theo mili-giây.

<u>Date.setFullYear</u>	Thiết lập năm đầy đủ theo thời gian cục bộ. Trả lại thời gian mới theo mili-giây.
<u>Date.setHours</u>	Thiết lập giờ theo thời gian cục bộ. Trả lại thời gian mới theo mili-giây.
<u>Date.setMilliseconds</u>	Thiết lập số mili-giây theo thời gian cục bộ. Trả lại thời gian mới theo mili-giây.
<u>Date.setMinutes</u>	Thiết lập số phút theo thời gian cục bộ. Trả lại thời gian mới theo mili-giây.
<u>Date.setMonth</u>	Thiết lập tháng theo thời gian cục bộ. Trả lại thời gian mới theo mili-giây.
<u>Date.setSeconds</u>	Thiết lập số giây theo thời gian cục bộ. Trả lại thời gian mới theo mili-giây.
<u>Date.setTime</u>	Thiết đặt ngày (date) theo mili-giây. Trả lại thời gian mới theo mili-giây.
<u>Date.setUTCDate</u>	Thiết lập ngày theo thời gian quốc tế. Trả lại thời gian mới theo mili-giây.
<u>Date.setUTCFullYear</u>	Thiết lập năm theo thời gian quốc tế. Trả lại thời gian mới theo mili-giây.
<u>Date.setUTCHours</u>	Thiết lập giờ theo thời gian quốc tế. Trả lại thời gian mới theo mili-giây.
<u>Date.setUTCMilliseconds</u>	Thiết lập số mili-giây theo thời gian quốc tế. Trả lại thời gian mới theo mili-giây.
<u>Date.setUTCMinutes</u>	Thiết lập số phút theo thời gian quốc tế. Trả lại thời gian mới theo mili-giây.
<u>Date.setUTCMonth</u>	Thiết lập tháng theo thời gian quốc tế. Trả lại thời gian mới theo mili-giây.
<u>Date.setUTCSeconds</u>	Thiết lập số giây theo thời gian quốc tế. Trả lại thời gian mới theo mili-giây.
<u>Date.setYear</u>	Thiết đặt năm theo thời gian cục bộ.
<u>Date.toString</u>	Trả lại một chuỗi giá trị đại diện cho ngày và giờ được lưu giữ trong đối tượng Date xác định.
<u>Date.UTC</u>	Trả lại thời số mili-giây giữa nửa đêm ngày 1 tháng Giêng, năm 1970, giờ quốc tế, và thời gian xác định.

Constructor (Người xây dựng) cho đối tượng Date object

Availability

Flash Player 5.

Usage

```
new Date()  
new Date(year, month [, date [, hour [, minute [, second [, millisecond ]]])
```

Parameters

year Một giá trị thuộc khoảng 0 đến 99 biểu thị khoảng thời gian từ 1900 đến năm 1999; nếu không thì cả bốn chữ số của năm phải được chỉ rõ.

month Một số nguyên từ 0 (January : tháng Giêng) đến 11 (December: tháng Mười Hai).

date Một số nguyên từ 1 đến 31. Thông số này là tùy chọn.

hour Một số nguyên từ 0 (midnight : nửa đêm) đến 23 (11 p.m.).

minute Một số nguyên từ 0 đến 59. Thông số này là tùy chọn.

second Một số nguyên từ 0 đến 59. Thông số này là tùy chọn.

millisecond Một số nguyên từ 0 đến 999. Thông số này là tùy chọn.

Returns

Một số nguyên.

Description

Đối tượng; xây dựng một đối tượng Date (ngày tháng) mới lưu giữ ngày và giờ hiện thời, hoặc ngày xác định.

Example

Ví dụ sau đây gọi ra ngày và giờ hiện thời.

```
now = new Date();
```

Ví dụ sau đây tạo một đối tượng ngày tháng (Date object) mới cho ngày sinh nhật của Gary (Gary's birthday), August 7, 1974.

```
gary_birthday = new Date (74, 7, 7);
```

Ví dụ sau tạo một đối tượng Date mới, kết hợp với các giá trị được trả lại của các method của đối tượng Date như `getMonth`, `getDate`, và `getFullYear`, và trình bày chúng trong trường văn bản (text field) được định nghĩa bởi biến `dateTextField`.

```
myDate = new Date();
dateTextField = ((myDate.getMonth() + 1) + "/" + myDate.getDate() + "/" +
myDate.getFullYear());
```

delete

Availability

Flash Player 5.

Usage

`delete reference`

Parameters

`reference` Tên của biến hoặc đối tượng để loại trừ.

Returns

Nothing.

Description

Toán tử (Operator); phá hủy đối tượng hoặc biến được chỉ rõ bởi tham số `reference`, và trả về `true` nếu đối tượng đã được xóa thành công; nếu không thì trả về một giá trị `false`. Toán tử này hữu ích cho việc giải phóng bộ nhớ được dùng bởi các script. Mặc dù `delete` là một toán tử, nó được dùng đặc thù như một phát biểu (statement), như sau:

```
delete x;
```

Toán tử `delete` có thể sai và trả về `false` nếu tham số `reference` không tồn tại, hoặc có thể không bị xoá. Các đối tượng và thuộc tính xác định trước, và các biến khai báo với `var`, có thể không bị xoá. Bạn không thể dùng toán tử `delete` để remove các movie clip.

Example

Ví dụ sau tạo một đối tượng, sử dụng nó, và sau đó xóa nó sau khi không cần dùng nữa.

```
account = new Object();account.name = 'Jon';account.balance = 10000;delete
account;
```

Example

Ví dụ sau xóa một thuộc tính của một đối tượng.

```
// tạo một đối tượng mới "account"account = new Object();// gán tên thuộc tính cho account account.name = 'Jon'; // xóa thuộc tínhdelete account.name;
```

Example

Sau đây là một ví dụ khác của việc xóa một thuộc tính đối tượng.

```
// tạo một đối tượng mảng Array với độ dài 0array = new Array(); // độ dài mảng Array.length bây giờ là 1array[0] = "abc";// thêm phần tử khác vào mảng,Array.length bây giờ là 2array[1] = "def";// thêm phần tử khác vào mảng,Array.length bây giờ là 3array[2] = "ghi";// array[2] bị xóa, nhưng Array.length không bị thay đổi,delete array[2];
```

Ví dụ sau đây minh họa hành vi của `delete` trên các đối tượng liên quan (on object references):

```
// tạo một đối tượng mới, và gán biến ref1// tham chiếu tới đối tượngref1 = new Object();ref1.name = "Jody";// copy biến tham khảo vào trong một biến mới// và xóa ref1ref2 = ref1;delete ref1;
```

Nếu `ref1` không được sao chép thành `ref2`, đối tượng sẽ bị xóa khi `ref1` bị xóa, bởi vì không có sự liên quan nào tới nó. Nếu bạn xóa `ref2`, sẽ không còn bất kỳ sự liên hệ nào tới đối tượng; nó sẽ bị phá hủy, và bộ nhớ (memory) mà nó sử dụng sẽ lại sẵn sàng dùng được.

See also

[var](#)

duplicateMovieClip

Availability

Flash Player 4.

Usage

```
duplicateMovieClip(target, newname, depth)
```

Parameters

target Đường dẫn của movie clip để nhân đôi (duplicate).

newname Một từ định danh (trong quản lý cơ sở dữ liệu, đây là một từ mô tả dùng để xác định tính duy nhất của thông tin chứa trong bản ghi dữ liệu) duy nhất cho movie clip sao lại (A unique identifier for the duplicated movie clip).

depth Một mức năng lực (hay mức chiều sâu) đơn nhất cho movie clip bản sao (A unique depth level for the duplicated movie clip). Mức chiều sâu (depth level) là một sắp xếp thứ bậc (stacking order) cho các movie clip bản sao. Sắp xếp thứ bậc này tương tự như sắp xếp thứ bậc của các layer trong Timeline; các movie clip với một mức năng lực thấp hơn ẩn ở dưới các clip có mức năng lực cao hơn. Bạn phải gán cho mỗi movie clip bản sao một mức năng lực (hay mức chiều sâu) duy nhất

để ngăn cản nó thay thế các movie trên các mức năng lực đang sử dụng (You must assign each duplicated movie clip a unique depth level to prevent it from replacing movies on occupied depths).

Returns

Nothing.

Description

Hành vi (Action); tạo một thể hiện của một movie clip trong khi movie (phim) đang được chơi. Thanh playhead (đây là thanh kéo ở trong Timeline) trong các movie clip bản sao luôn luôn bắt đầu tại Frame 1, không quan tâm đến playhead trong movie clip gốc (hoặc movie clip mẹ "parent"). Các biến trong movie clip gốc không được sao chép vào trong movie clip bản sao. Nếu movie clip gốc (parent movie clip) bị xóa thì movie clip bản sao cũng bị xóa. Sử dụng hành vi (action) hoặc phương pháp (method) `removeMovieClip` để xóa một movie clip thể hiện được tạo với `duplicateMovieClip`.

Example

Phát biểu (statement) này sao lại movie clip thể hiện (movie clip instance) `flower` mười lần. Biến `i` được sử dụng để tạo một tên minh họa (instance name) và một chiều sâu duy nhất (unique depth) mới cho mỗi movie clip bản sao.

```
on (release) { amount = 10;   while (amount>0) {           duplicateMovieClip
(_root.flower, "mc"+i, i);       setProperty ("mc"+i, _x, random(275));
random(50));                    i++;                       amount--;                } }
```

See also

[MovieClip.duplicateMovieClip](#), [removeMovieClip](#), [MovieClip.removeMovieClip](#)

FCheckBox (component: thành phần)

Thành phần CheckBox trong môi trường thiết kế Flash cung cấp chức năng kéo và thả (drag-and-drop) để thêm các check box (hộp kiểm tra) vào các tài liệu Flash; nó cũng cung cấp một giao diện người dùng để thiết lập các tham số cơ bản. Các method (phương pháp) của thành phần FCheckBox cho phép bạn điều khiển các check box khi chạy: bạn có thể tạo các check box, điều khiển các check box đã được tạo trong chương trình, thiết lập hoặc bỏ qua các tham số cơ bản, và thiết lập các tùy chọn bổ sung khi thực hiện. Bạn không cần sử dụng một constructor (người thiết kế) để truy cập các method của các thành phần.

Thành phần CheckBox được hỗ trợ bởi Flash Player 6.

Các method thành phần không thực hiện việc kiểm tra các lỗi cho kiểu (Component methods do not perform error checking for type), như các đối tượng và hành vi ActionScript tự nhiên khác; vì thế,

nó khuyên bạn nên phê chuẩn xác nhận tính hợp lệ của các tham số trước khi gửi chúng đến các method.

Để biết thêm thông tin về việc sử dụng thành phần CheckBox component, thiết lập các tham số trong suốt thời gian thiết kế, và thay đổi màu sắc và diện mạo của các thành phần (component), hãy xem [Customizing component colors and text](#) và [Customizing component skins](#) trong chương [Using Components](#) của Using Flash.

Phương pháp tóm lược cho thành phần FCheckBox

Method	Description
FCheckBox.setEnabled	Trả lại <code>true</code> nếu hộp kiểm tra check box được cho phép, <code>false</code> nếu nó bị vô hiệu hoá.
FCheckBox.getLabel	Trả lại nhãn ứng dụng tới hộp kiểm như một chuỗi.
FCheckBox.getValue	Trả lại <code>true</code> nếu hộp kiểm được lựa chọn, <code>false</code> nếu nó không được lựa chọn.
FCheckBox.registerSkinElement	Đăng ký một phần tử ngoài tới một thuộc tính (Registers a skin element to a property).
FCheckBox.setChangeHandler	Chỉ rõ một người điều khiển thay đổi (change handler) được gọi khi giá trị của hộp kiểm thay đổi.
FCheckBox.setEnabled	Xác định xem hộp kiểm được cho phép hay bị vô hiệu hóa.
FCheckBox.setLabel	Chỉ rõ text cho nhãn của hộp kiểm.
FCheckBox.setLabelPlacement	Chỉ rõ xem nhãn xuất hiện tới bên trái hay bên phải của hộp kiểm.
FCheckBox.setSize	Thiết lập chiều rộng của hộp kiểm, theo pixel, và vẽ lại nó.
FCheckBox.setStyleProperty	Thiết lập một thuộc tính style đơn cho một thành phần component.
FCheckBox.setValue	Lựa chọn hủy bỏ hộp kiểm và gây ra hàm điều khiển thay đổi (Selects or deselects the check box and triggers the change handler function).

FComboBox (component)

Thành phần ComboBox (tạm dịch là Hộp kết hợp) trong môi trường thiết kế Flash cung cấp chức năng kéo-và-thả cho việc thêm các danh sách lựa chọn riêng thả xuống có thể cuộn được (scrollable single-selection drop-down lists : xin lỗi các bạn vì cụm từ này dịch hơi lủng củng) vào các tài liệu Flash; nó cũng cung cấp một giao diện người dùng cho việc thiết lập các tham số cơ bản. Các

method (cách thức, phương pháp) của thành phần FComboBox cho phép bạn điều khiển các hộp kết hợp khi thực hiện (at runtime): bạn có thể tạo các hộp kết hợp combo box, điều khiển các combo box đã tạo trong trình thiết kế Flash, thiết đặt hoặc ghi đè các tham số cơ bản, và thiết đặt các tùy chọn bổ sung khi chạy. Bạn không cần dùng một constructor (người xây dựng) để truy xuất các method của các thành phần.

Thành phần ComboBox tạo ra cả hai loại hộp kết hợp (combo box) tĩnh và có thể hiệu chỉnh được. Một combo box tĩnh là một danh sách thả xuống có thể cuộn được cho phép người dùng lựa chọn các mục. Một combo box có thể hiệu chỉnh được là một danh sách thả xuống có thể cuộn được với một trường text ở trên cùng. Bạn có thể cho người dùng nhập text vào trường text để làm cho hộp combo box cuộn tới mục yêu cầu, hoặc bạn có thể sử dụng trường text để thiết đặt text được hiển thị khi chạy chương trình.

Cả hai phiên bản tĩnh và hiệu chỉnh của thành phần ComboBox đều liệt kê các mục từ trên xuống dưới sử dụng một hệ thống liệt kê hoá trên nền cơ sở zero (zero-based indexing system). Nếu số các mục trong danh sách combo box tạo một danh sách thả xuống mà vượt quá khoảng trống bên dưới mỗi component (thành phần), danh sách sẽ mở lên trên thay vì thả xuống dưới.

Những method thành phần không thực hiện sự kiểm tra lỗi cho kiểu , như đã làm với những hành vi (action) và đối tượng ActionScript bẩm sinh khác; bởi vậy, khuyên bạn nên thông qua (xác nhận tính hợp lệ) các tham số trước khi chuyển chúng tới các method.

Thành phần ComboBox được hỗ trợ bởi Flash Player 6 và các phiên bản về sau của Flash Player.

Để biết thêm thông tin về việc sử dụng thành phần ComboBox , hãy thiết đặt các tham số trong suốt quá trình thực hành, và thay đổi màu sắc và hình thức của các thành phần, xem [Customizing component colors and text](#) và [Customizing component skins](#) trong chương [Using Components](#) của Sử dụng Flash.

Phương pháp tóm lược cho thành phần FComboBox

Method	Description
FComboBox.addItem	Thêm một mục mới vào cuối của danh sách combo box.
FComboBox.addItemAt	Thêm một mục mới vào danh sách tại một chỉ mục xác định.
FComboBox.setEnabled	Trả lại true nếu combo box có hiệu lực, false nếu nó đã bị vô hiệu hóa.
FComboBox.getItemAt	Trả lại mục (item) tại chỉ mục xác định (specified index) như một đối tượng (object) với các thuộc tính label và data.
FComboBox.getLength	Trả lại số các mục đã liệt kê trong combo box.
FComboBox.getRowCount	Trả lại số các hàng thấy được trong combo box.

FComboBox.getScrollPosition	Trả lại eturns the index of the item at the top of the combo box.
FComboBox.getSelectedIndex	Returns the index of the currently selected item.
FComboBox.getSelectedItem	Returns the currently selected item as an object with the properties label and data.
FComboBox.getValue	Returns the text in the input field for editable combo boxes; returns the label or data of the selected item for static combo boxes.
FComboBox.registerSkinElement	Registers a skin element to a property.
FComboBox.removeAll	Removes all items from the combo box.
FComboBox.removeItemAt	Removes the item at the specified index.
FComboBox.replaceItemAt	Replaces the label and data of an item at the specified index.
FComboBox.setChangeHandler	Assigns a function to call each time an item is selected or the user enters text in the text field.
FComboBox.setDataProvider	Registers an outside object to the component as a data source.
FComboBox.setEditable	Determines whether the combo box is editable (<code>true</code>) or static (<code>false</code>).
FComboBox.setEnabled	Specifies whether the combo box is enabled (<code>true</code>) or disabled (<code>false</code>).
FComboBox.setItemSymbol	Registers a symbol linkage ID to use for displaying list items in the combo box.
FComboBox.setRowCount	Determines the number of items displayed in the combo box without a scroll bar.
FComboBox.setSelectedIndex	Selects the item at the specified index.
FComboBox.setSize	Sets the pixel width of the combo box.
FComboBox.setStyleProperty	Sets a single style property for an instance of a component.
FComboBox.setValue	Specifies the text displayed in the text field at the top of an editable combo box.
FComboBox.sortItemsBy	Sorts the items in the list box alphabetically or numerically by label or by data.

FPushButton (component)

Thành phần PushButton trong môi trường thiết kế Flash cung cấp chức năng kéo-và-thả để thêm các nút vào các tài liệu Flash; nó cũng cung cấp một giao diện người dùng cho việc thiết đặt các tham số cơ bản. Các method của thành phần FPushButton cho phép bạn điều khiển các nút khi thực thi chương trình: bạn có thể tạo các nút, điều khiển các nút đã tạo trong khi thiết kế, thiết đặt hoặc ghi đè lên các tham số cơ bản, và thiết đặt các tùy chọn bổ sung khi chạy chương trình. Bạn không cần sử dụng một constructor để truy lục các method của các thành phần.

Thành phần PushButton chấp nhận tất cả các tương tác chuột và bàn phím chuột. Bạn có thể dùng các method FPushButton để xác định một hàm điều khiển cho các push buttons, vô hiệu hoá hoặc hiệu lực hoá các nút, và điều chỉnh kích thước các nút mà không cần bóp méo khi thực hiện chương trình (disable or enable buttons, and resize buttons without distortion at runtime).

Các method thành phần không thực thi các kiểm tra lỗi cho kiểu, như với các đối tượng và hành vi ActionScript tự nhiên khác; vì thế, khuyên bạn nên thông qua các tham số trước khi chuyển chúng tới các methods.

Thành phần PushButton component được hỗ trợ bởi Flash Player 6 và các phiên bản về sau của Flash Player.

Để biết thêm thông tin về cách sử dụng thành phần (component) PushButton, thiết đặt các tham số trong suốt quá trình thiết kế, và thay đổi màu sắc và diện mạo của các components, xem [Customizing component colors and text](#) (Tùy biến màu sắc và text của thành phần) và [Customizing component skins](#) (Tùy biến các skin thành phần) trong chương [Using Components](#) (Sử dụng các thành phần) trong Using Flash.

Bảng tóm tắt các method cho thành phần FPushButton

Method	Description
FPushButton.setEnabled	Trả về true nếu nút được hiệu lực hoá, false nếu nó đã bị vô hiệu hoá.
FPushButton.getLabel	Trả về nhãn của nút như một chuỗi.
FPushButton.registerSkinElement	Đăng ký một phần tử giao diện (skin element) thành một thuộc tính.
FPushButton.setClickHandler	Xác định hàm được gọi khi người dùng nhấn nút.
FPushButton.setEnabled	Quyết định xem nút được hiệu lực hoá hay vô hiệu hoá.
FPushButton.setLabel	Thiết đặt nhãn của nút khi thực hiện chương trình.
FPushButton.setSize	Thiết đặt chiều cao và chiều rộng của nút, theo pixels.
FPushButton.setStyleProperty	Thiết đặt một thuộc tính style (kiểu dáng) đơn cho một thành phần (component).

FRadioButton (component: thành phần)

Các nút Radio là nhóm nút có thể chọn được từ chỉ một nút được chọn ở một thời điểm. Thành phần RadioButton component trong môi trường thiết kế Flash cung cấp chức năng kéo-và-thả cho việc thêm các nhóm nút radio vào các tài liệu Flash; nó cũng cung cấp một giao diện người dùng cho việc thiết đặt các tham số cơ bản. Các method của thành phần (component) FRadioButton cho phép bạn điều khiển các nút radio khi chạy chương trình: bạn có thể tạo các nút radio, điều khiển các nút radio đã tạo khi thiết kế, thiết đặt hoặc ghi đè lên các tham số cơ bản, và thiết đặt các tùy chọn bổ sung khi chạy chương trình. Bạn không cần sử dụng một constructor (trình xây dựng, người xây dựng) để truy cập các method của các components.

Thành phần RadioButton component được hỗ trợ bởi Flash Player 6 và các phiên bản về sau của Flash Player.

Các method thành phần (component methods) không thực hiện kiểm tra lỗi cho kiểu, như làm với các đối tượng và hành vi ActionScript tự nhiên khác; vì thế, khuyên bạn nên xem xét làm cho hợp lệ các tham số trước khi chuyển chúng đến cho các method.

Để biết thêm thông tin về việc sử dụng thành phần RadioButton component, thiết đặt các tham số trong suốt quá trình thiết kế, và thay đổi màu sắc và hình thức của các component, xem [Customizing component colors and text](#) (tùy chỉnh màu sắc và chữ của thành phần hợp thành) và [Customizing component skins](#) (tùy chỉnh các thành phần giao diện) trong chương [Using Components](#) (sử dụng các thành phần hợp thành) của Using Flash.

Bảng tóm tắt các method của FRadioButton component

Method	Description
FRadioButton.getData	Trả về một giá trị dữ liệu cho một instance (trường hợp, thể hiện) của nút radio.
FRadioButton.setEnabled	Trả về <code>true</code> nếu nút radio được hiệu lực hóa (cho phép hoạt động), <code>false</code> nếu nó bị vô hiệu hóa.
FRadioButton.getLabel	Trả về nhãn (label) của nút radio như một chuỗi.
FRadioButton.getState	Trả về trạng thái lựa chọn của một instance của nút radio.
FRadioButton.getValue	Trả về giá trị dữ liệu của nút radio được lựa chọn trong một nhóm, hoặc trả về nhãn nếu không có dữ liệu nào được xác định.
FRadioButton.registerSkinElement	Đăng ký một phần tử giao diện (skin element) thành một thuộc tính kiểu dáng (style).

FRadioButton.setChangeHandler	Chỉ định một hàm để gọi khi sự lựa chọn nút radio thay đổi.
FRadioButton.setData	Thiết đặt dữ liệu kết hợp với một instance của nút radio.
FRadioButton.setEnabled	Xác định xem nút radio được hiệu lực hóa hay bị vô hiệu hóa.
FRadioButton.setGroupName	Xác định một tên nhóm cho một instance (trường hợp, thể hiện) của nút radio, hoặc thiết đặt một tên mới cho một nhóm các nút radio.
FRadioButton.setLabel	Áp dụng một nhãn (label) cho nút radio khi chạy chương trình.
FRadioButton.setLabelPlacement	Xác định xem nhãn xuất hiện ở bên trái hay bên phải của nút radio.
FRadioButton.setSize	Thiết đặt chiều rộng của nút radio, theo pixels.
FRadioButton.setState	Thiết đặt trạng thái lựa chọn của một instance của nút radio.
FRadioButton.setStyleProperty	Thiết đặt một thuộc tính style (kiểu dáng) đơn cho một instance.
FRadioButton.setValue	Lựa chọn một nút radio trong một nhóm nút radio khi chạy chương trình.

FScrollBar (component)

Thành phần (component) ScrollBar trong môi trường thiết kế Flash cung cấp chức năng kéo-và-thả cho việc thêm các thanh cuộn vào các trường text động và text (chữ) nhập vào trong các tài liệu Flash; nó cũng cung cấp một giao diện người dùng cho việc thiết đặt các tham số cơ bản. Các method (phương pháp, cách thức) của thành phần FScrollBar cho phép bạn điều khiển các thanh cuộn khi chạy chương trình: bạn có thể tạo các thanh cuộn, điều khiển các thanh cuộn đã tạo trong khi thiết kế, thiết đặt hoặc ghi đè lên các tham số cơ bản, và thiết đặt các tùy chọn bổ sung khi chạy chương trình. Bạn không cần sử dụng một constructor (trình xây dựng) để truy cập các method của các component. Một số method của component FScrollBar không được khuyến khích để dùng với các thanh cuộn kèm theo các trường chữ (text field). Xem các mục từ method riêng lẻ để biết chi tiết.

Những người dùng và các lập trình viên cao cấp có thể dùng thành phần ScrollBar với các yếu tố Flash khác để tạo các giao diện người dùng tùy thích (custom user interfaces).

Các method thành phần (component methods) không thực hiện kiểm tra các lỗi do kiểu, như các đối tượng và hành vi ActionScript thông thường khác; vì thế, bạn nên kiểm tra tính hợp lệ của các tham số trước khi chuyển chúng tới các method.

Thành phần (component) ScrollBar được hỗ trợ bởi Flash Player 6 và các phiên bản sau này của Flash Player.

Để biết thêm thông tin trong việc sử dụng thành phần ScrollBar component, thiết đặt các tham số trong suốt quá trình thiết kế, và thay đổi màu sắc và hình thức của các component, xem [Customizing component colors and text](#) (tùy chỉnh các màu sắc và chữ của thành phần) và [Customizing component skins](#) trong chương [Using Components](#) (Sử dụng các thành phần) của Using Flash.

Bảng tóm tắt các method của FScrollBar component

Method	Description
FScrollBar.setEnabled	Trả về <code>true</code> nếu thanh cuộn được cho phép hoạt động, <code>false</code> nếu nó bị vô hiệu hóa.
FScrollBar.getScrollPosition	Trả về một số nguyên tương ứng với vị trí hiện tại của hộp cuộn (thumb).
FScrollBar.registerSkinElement	Đăng ký một yếu tố giao diện (skin element) thành một thuộc tính xác định cho một skin trong layer Read Me định vị trên khung Frame 1 của một skin movie clip trong thư viện.
FScrollBar.setChangeHandler	Chỉ định một hàm để gọi mỗi lần vị trí cuộn thay đổi. (Bạn không thể dùng method này với các trường chữ.)
FScrollBar.setEnabled	Xác định xem thanh cuộn được hiệu lực hóa (<code>true</code>) hay bị vô hiệu hóa (<code>false</code>).
FScrollBar.setHorizontal	Xác định xem thanh cuộn là ngang (<code>true</code>) hay dọc (<code>false</code>).
FScrollBar.setLargeScroll	Xác định số vị trí đã cuộn khi người dùng click vào rãnh cuộn.
FScrollBar.setScrollContent	Chỉ định instance (thể hiện, trường hợp) của trường chữ mà ở đó thanh cuộn được áp dụng.
FScrollBar.setScrollPosition	Thiết đặt vị trí của hộp cuộn như một số nguyên giữa <code>minPos</code> (vị trí cực tiểu) và <code>maxPos</code> (vị trí cực đại).
FScrollBar.setScrollProperties	Thiết đặt các thuộc tính <code>pageSize</code> (kích cỡ trang), <code>minPos</code> , và <code>maxPos</code> cho thanh cuộn. (Bạn không thể dùng method này với các trường chữ.)
FScrollBar.setScrollTarget	Chỉ định một trường chữ (text field) như đường dẫn (target: đích) cho thanh cuộn.
FScrollBar.setSize	Thiết đặt độ dài của thanh cuộn, theo pixel.

FScrollBar.setSmallScroll	Chỉ định số vị trí cuộn khi người dùng click vào một mũi tên cuộn.
FScrollBar.setStyleProperty	Thiết đặt một thuộc tính style (kiểu dáng) đơn cho một component.

FScrollPane (component)

Thành phần ScrollPane component trong môi trường thiết kế Flash cung cấp chức năng kéo-và-thả cho việc thêm các ô cuộn (scroll pane) để hiển thị (trình bày) các movie clip trong các tài liệu Flash; nó cũng cung cấp một giao diện người dùng cho quá trình thiết đặt các tham số cơ bản. Các method (phương pháp, cách thức) của component FScrollPane cho phép bạn điều khiển các ô cuộn khi chạy chương trình: bạn có thể tạo các ô cuộn, điều khiển các ô cuộn đã tạo khi thiết kế, thiết đặt hoặc ghi đè lên các tham số cơ bản, và thiết lập các tùy chọn bổ sung khi chạy chương trình. Bạn không cần sử dụng một constructor (trình xây dựng) để truy cập các method của những component.

Thành phần (component) ScrollPane cung cấp các thanh cuộn ngang và dọc cho phép bạn hiển thị các movie clip mà không cần tăng nhiều không gian trống của Stage (màn hình trình diễn). Các điều khiển chuột và bàn phím chuẩn đều được cài đặt sẵn.

Chú ý: Thành phần ScrollPane chỉ hiển thị các movie clip; để thêm các thanh cuộn vào các trường chữ động (dynamic text field) và trường chữ nhập vào (input text field), bạn sử dụng component ScrollBar. Thành phần ScrollPane không thể hiển thị bất kỳ nội dung nào mà có sử dụng các device fonts (không biết dịch cái từ **device** này thế nào cho hợp lý----> cứ để nguyên gốc) .

Các method của component không thực thi việc kiểm tra lỗi cho kiểu, như những đối tượng và hành vi ActionScript thông thường khác; bởi vậy, bạn nên kiểm tra tính hợp lệ của các tham số trước khi chuyển chúng tới các method.

Thành phần ScrollPane component được hỗ trợ bởi Flash Player 6 và các phiên bản sau này của Flash Player.

Để biết thêm thông tin về việc sử dụng thành phần (component) ScrollPane, việc thiết đặt các tham số trong suốt quá trình thiết kế, và thay đổi màu sắc và hình thức của các component, hãy xem [Customizing component colors and text](#) (tùy chỉnh màu sắc và text của các component) và [Customizing component skins](#) (tùy chỉnh bề mặt, giao diện của các component) trong chương [Using Components](#) của Using Flash.

Bảng các method tóm lược cho FScrollPane component

Method	Description
FScrollPane.getPaneHeight	Trả về chiều cao của ô cuộn.
FScrollPane.getPaneWidth	Trả về chiều rộng của ô cuộn.

FScrollPane.getScrollContent	Trả về một instance (trường hợp, thể hiện) của nội dung được trình bày trong ô cuộn.
FScrollPane.getScrollPosition	Trả về các tọa độ x và y của vị trí cuộn hiện tại.
FScrollPane.loadScrollContent	Nạp một file SWF hoặc JPEG vào trong một ô cuộn.
FScrollPane.refreshPane	Điều chỉnh lại kích thước của các thanh cuộn trong ô cuộn khi nội dung thay đổi kích cỡ.
FScrollPane.registerSkinElement	Đăng ký một yếu tố bề mặt (skin element) thành một thuộc tính xác định cho một skin trong layer Read Me định vị trên Frame 1 của một skin movie clip trong thư viện.
FScrollPane.setDragContent	Thiết đặt nội dung của ô cuộn thành có khả năng kéo (di chuyển con trỏ chuột trong khi vẫn ấn và giữ phím trên chuột).
FScrollPane.setHScroll	Thiết đặt kiểu dáng cuộn ngang (horizontal scrolling style) cho ô cuộn.
FScrollPane.setScrollContent	Thiết đặt một movie clip thành ô cuộn đích (scroll pane target).
FScrollPane.setScrollPosition	Làm cho ô cuộn tới các tọa độ x, y được chỉ định trước.
FScrollPane.setSize	Thiết đặt chiều rộng và chiều cao của ô cuộn, theo pixel.
FScrollPane.setStyleProperty	Thiết đặt một thuộc tính style (kiểu dáng) đơn cho một component.
FScrollPane.setVScroll	Thiết đặt style cuộn đơn của ô cuộn.

FStyleFormat (object)

Đối tượng FStyleFormat cho phép bạn thiết đặt hoặc thay đổi các thuộc tính trong định dạng kiểu dáng chung (global style format) đã gán cho tất cả các thành phần giao diện Flash UI components theo mặc định, hoặc tạo các custom style formats (các định dạng kiểu dáng do người dùng tự tạo) để sử dụng với các Flash UI components hoặc các custom components mà bạn tạo hoặc có được từ các nguồn khác. Định dạng kiểu dáng chung, hoặc đối tượng `globalStyleFormat`, là một instance (thể hiện) của đối tượng FStyleFormat mà xác định các thuộc tính màu sắc và định dạng chữ viết dùng để hiển thị tất cả các thành phần Flash UI components.

Để tạo một custom style format mới, bạn tạo một instance (thể hiện) của đối tượng FStyleFormat sử dụng constructor (người xây dựng) `new FStyleFormat()`, thiết đặt các thuộc tính FStyleFormat mà bạn muốn kèm theo trong định dạng kiểu dáng của bạn, và sau đó dùng method `FStyleFormat.addListener` để đăng ký các component instances thành một định dạng kiểu dáng (style format) mới. Một thể hiện của thành phần hợp thành (a component instance) có thể "listen"

(chỗ này chẳng biết dịch thế nào cả, dịch tạm là **nghe** vậy nha) nhiều style format, nhưng chỉ có thể lấy giá trị của một style format cho một thuộc tính cụ thể. Nếu bạn thêm một component (thành phần hợp thành) như một **listener** (???) thành một định dạng kiểu dáng (style format), nó sẽ sử dụng dạng style mới cho các thuộc tính đã được chỉ định trong định dạng, và nó dùng dạng style cũ cho tất cả các thuộc tính khác.

Bạn không cần dùng constructor (người xây dựng) FStyleFormat để thêm hoặc gỡ bỏ các listeners hoặc thiết đặt hoặc thay đổi các thuộc tính trong định dạng kiểu dáng chung bởi vì đối tượng `globalStyleFormat` tồn tại ngay khi bất kỳ thành phần giao diện Flash UI component được đặt trên Stage.

Bạn có thể thiết đặt bất kỳ thuộc tính nào trong các thuộc tính FStyleFormat cho một instance đơn của một component sử dụng method `setStyleProperty` có sẵn cho tất cả các Flash UI components. Việc sử dụng `setStyleProperty` cho phép bạn thiết đặt một thuộc tính cho một component mà không phải tạo một instance của đối tượng FStyleFormat. Việc sử dụng method `setStyleProperty` ghi đè lên thiết đặt cho một thuộc tính dạng style cụ thể gán cho component mà không phải thay đổi các thiết đặt cho những thuộc tính khác. Để biết thêm thông tin, xem mục từ `setStyleProperty` cho các component riêng lẻ.

Khi gán một giá trị màu vào một thuộc tính FStyleFormat, chỉ định một màu RGB theo định dạng `0xRRGGBB`.

Để biết thêm thông tin về định dạng kiểu dáng chung và tạo các định dạng kiểu dáng tùy chọn, xem [Customizing component colors and text](#) trong chương [Using Components](#) của Using Flash.

Method summary for the FStyleFormat object

Method	Description
<code>FStyleFormat.addListener</code>	Đăng ký một component thành một style format.
<code>FStyleFormat.applyChanges</code>	Áp dụng những thay đổi đã làm vào các giá trị thuộc tính trong một style format.
<code>FStyleFormat.removeListener</code>	Gỡ bỏ một component như một listener thành một style format.

Bảng tóm tắt các thuộc tính cho đối tượng FStyleFormat

Bảng sau liệt kê tóm tắt các thuộc tính cho đối tượng FStyleFormat.

Property	Description
FStyleFormat.arrow	Màu sắc của mũi tên dùng trong các thanh cuộn và các danh sách thả xuống.

<code>FStyleFormat.background</code>	Màu sắc phần nền phía sau của một thành phần (component).
<code>FStyleFormat.backgroundDisabled</code>	Màu sắc phần nền sau của một component bị vô hiệu hóa.
<code>FStyleFormat.check</code>	Màu sắc của dấu chọn đánh dấu trong một hộp kiểm được chọn.
<code>FStyleFormat.darkshadow</code>	Màu sắc đường viền trong hoặc phần bóng tối của một component.
<code>FStyleFormat.face</code>	Màu sắc chính của component.
<code>FStyleFormat.foregroundDisabled</code>	Màu tiền cảnh của một component bị vô hiệu hóa.
<code>FStyleFormat.highlight</code>	Màu sắc cho viền trong hoặc phần bóng tối của một component khi nó được lựa chọn.
<code>FStyleFormat.highlight3D</code>	Màu sắc cho viền ngoài hoặc phần bóng sáng của một component khi nó được lựa chọn.
<code>FStyleFormat.radioDot</code>	Màu sắc của chấm bên trong một nút radio đã được lựa chọn.
<code>FStyleFormat.scrollTrack</code>	Màu sắc của rãnh (track) trong một thanh cuộn.
<code>FStyleFormat.selection</code>	Màu sắc của thanh chọn làm nổi bật (highlighting: làm nổi bật, tô sáng) một danh sách mục trong một component.
<code>FStyleFormat.selectionDisabled</code>	Màu sắc của một thanh chọn (selection bar) mà tô sáng một danh sách mục trong một component bị vô hiệu hóa.
<code>FStyleFormat.selectionUnfocused</code>	Màu sắc của một thanh chọn khi component không có trọng tâm vào bàn phím (does not have key board focus).
<code>FStyleFormat.shadow</code>	Màu sắc của viền ngoài hoặc phần bóng sáng của một component.
<code>FStyleFormat.textAlign</code>	Sự canh lề (left: trái, right: phải, hoặc center: giữa) cho chữ (text) trình bày trong hoặc trên một component.
<code>FStyleFormat.textBold</code>	Xác định xem chữ là đậm (true) hay không (false).
<code>FStyleFormat.textColor</code>	Màu chữ mặc định trong tất cả các component gán cho định dạng kiểu dáng (style format).
<code>FStyleFormat.textDisabled</code>	Màu sắc của chữ trong một component bị vô hiệu hóa.

FStyleFormat.textFont	Tên của font để hiển thị chữ.
FStyleFormat.textIndent	Sự thụt dòng của chữ từ lề trái đến ký tự text đầu tiên, theo pixel.
FStyleFormat.textItalic	Xác định xem chữ có in nghiêng (<code>true</code>) hay không (<code>false</code>).
FStyleFormat.textLeftMargin	Lề trái của đoạn văn cho text, theo pixels.
FStyleFormat.textRightMargin	Lề phải của đoạn văn cho text, theo pixel.
FStyleFormat.textSelected	Màu sắc của một danh sách mục lựa chọn trong một component.
FStyleFormat.textSize	Kích cỡ của chữ, theo point (1 point = 1/72 inch).
FStyleFormat.textUnderline	Xác định xem chữ có được gạch chân (<code>true</code>) hay không (<code>false</code>).

Constructor (người xây dựng) cho đối tượng FStyleFormat

Availability

Flash Player 6.

Usage

```
new FStyleFormat()
```

Parameters

None.

Returns

Một instance của đối tượng FStyleFormat.

Description

Method (phương pháp, cách thức); tạo một đối tượng FStyleFormat mới. Bạn tạo các đối tượng FStyleFormat mới để định nghĩa các thuộc tính chữ viết (text) và màu sắc cho các custom style formats (các định dạng kiểu dáng do người dùng tạo ra) dùng với các custom components hoặc với các thành phần giao diện Flash UI components. Tất cả các thành phần giao diện Flash UI components được gán mặc định là `globalStyleFormat`, đây là một instance (thể hiện) của đối tượng FStyleFormat. Bạn không cần tạo một instance mới của đối tượng FStyleFormat để thay đổi

các thuộc tính trong định dạng kiểu dáng chung (global style format), bởi vì nó hiển nhiên đã tồn tại. Bạn cũng có thể dùng `setStyleProperty` để thay đổi các thuộc tính cho các instance của component cụ thể mà không sử dụng một constructor.

Để biết thêm thông tin, xem method `setStyleProperty` có sẵn đối với mỗi component—`FCheckBox.setStyleProperty`, `FComboBox.setStyleProperty`, và nhiều method khác. Ngoài ra bạn nên xem [Customizing component colors and text](#) của chương [Using Components](#) của Using Flash.

Example

Mẫu sau đây tạo định dạng kiểu dáng mới `formStyleFormat`.

```
formStyleFormat = new StyleFormat();
```

Function (object) : Hàm (đối tượng)

Đối tượng Function có sẵn trong Flash MX.

Bảng tóm tắt các thuộc tính cho đối tượng hàm Function object

Method	Description
Function.prototype	Đề cập tới một đối tượng mà đó là mẫu ban đầu (prototype) cho một lớp (class).

Bảng tóm lược các method cho đối tượng Function object

Method	Description
Function.apply	Hiệu lực hoá mã ActionScript để gọi một hàm.
Function.call	Viện dẫn (gọi) hàm được mô tả lại bởi một đối tượng Function object.

Function.prototype

Availability

Flash Player 6.

Usage

```
myFunction.prototype
```

Description

Property (thuộc tính); trong một hàm xây dựng (constructor function), thuộc tính `prototype` (mẫu ban đầu) đề cập tới một đối tượng là mẫu ban đầu của lớp đã được xây dựng. Mỗi thể hiện (instance) của lớp đều được tạo bởi hàm xây dựng thừa kế tất cả các thuộc tính và method từ đối tượng mẫu ban đầu.

fscommand

Availability

Flash Player 3.

Usage

```
fscommand("command", "parameters")
```

Parameters

`command` Một chuỗi chuyển tới trình ứng dụng máy chủ (host application) cho bất kỳ sử dụng hoặc một lệnh chuyển tới trình Flash Player độc lập.

`parameters` Một chuỗi chuyển đến trình ứng dụng máy chủ cho bất kỳ sử dụng hoặc một giá trị chuyển tới Flash Player.

Returns

Nothing.

Description

Action (hành vi); cho phép Flash movie truyền đạt (communicate) với cả trình Flash Player, hoặc chương trình có kèm theo Flash Player, chẳng hạn như một trình duyệt Web browser. Bạn cũng có thể sử dụng hành vi `fscommand` để chuyển các tin nhắn (messages) tới Macromedia Director, hoặc tới Visual Basic, Visual C++, và các chương trình khác có thể host các điều khiển ActiveX.

Cách dùng 1: Để gửi một message tới Flash Player, bạn phải dùng các lệnh và tham số được định nghĩa trước. Bảng sau cho thấy những giá trị mà bạn có thể chỉ định cho các tham số `command` và `parameters` của hành vi `fscommand` để điều khiển một phim movie trình diễn trong trình ứng dụng độc lập Flash player (kể cả các projectors : các file .exe tạo bởi Flash):

Command	Parameters	Purpose
quit	None	Đóng projector.
fullscreen	true hoặc false	Việc chỉ định là true thiết đặt Flash Player thành chế độ toàn màn hình full-screen. Chỉ định là false trả trình player về chế độ bình thường nhìn thấy thực đơn menu điều khiển.

allowscale	true hoặc false	Việc chỉ định là false thiết đặt player để movie luôn luôn được trình bày như kích thước gốc của nó và không bao giờ co dãn (bị thay đổi kích thước). Chỉ định là true bắt buộc movie phải thay đổi kích thước bằng 100% kích thước của player.
showmenu	true hoặc false	Việc chỉ định là true làm cho thiết lập đầy đủ của các mục thực đơn ngữ cảnh được hiển thị (thực đơn File, View, Control, Help... trên thanh menu trên cùng của Flash Player và khi bạn click chuột phải vào trình Flash Player). Chỉ định là false làm mờ đi (bạn sẽ không nhìn thấy được) tất cả các mục trên trình đơn ngữ cảnh ngoại trừ About Flash Player.
exec	Đường dẫn tới trình ứng dụng	Chạy một trình ứng dụng từ bên trong projector.
trapallkeys	true hoặc false	Việc chỉ định là true gửi tất cả các sự kiện phím, kể cả các phím gia tốc (accelerator keys), tới trình điều khiển <code>onClipEvent (keyDown/keyUp)</code> trong Flash Player.

Cách dùng 2: Sử dụng hành vi `fscCommand` để gửi một message tới một ngôn ngữ kịch bản (scripting language) như là JavaScript trong một Web browser, bạn có thể chuyển hai tham số bất kỳ trong các tham số `command` (lệnh) và `parameters` (thông số). Những tham số này có thể là các chuỗi hoặc các biểu thức được dùng trong một hàm JavaScript để "catches," (lấy) hoặc điều khiển, hành vi `fscCommand`.

Trong một Web browser, hành vi `fscCommand` gọi hàm JavaScript `moviename_DoFScCommand` trong trang HTML đang chứa Flash movie. Tham số `moviename` là tên của Flash Player được gán bởi thuộc tính `NAME` (tên) của thẻ `EMBED` hoặc đặc tính `ID` của thẻ `OBJECT` tag. Nếu bạn gán cho Flash Player tên `myMovie`, hàm JavaScript được gọi là `myMovie_DoFScCommand`.

Cách dùng 3: Hành vi `fscCommand` có thể gửi các message tới Macromedia Director mà được giải thích bởi Lingo như strings (các chuỗi), events (các sự kiện), hoặc mã Lingo có thể thực thi được. Nếu message là một chuỗi hoặc một sự kiện, bạn phải viết mã Lingo code để tiếp nhận message từ hành vi `fscCommand` và mang thực hiện một hành vi action trong Director. Để tìm hiểu thêm thông tin, xem [Director Support Center](#).

Cách dùng 4: Trong Visual Basic, Visual C++, và các chương trình khác mà có thể thực hiện (host) các điều khiển ActiveX, `fscCommand` gửi một sự kiện VB với hai chuỗi mà có thể được điều khiển trong môi trường ngôn ngữ lập trình. Để tìm hiểu thêm thông tin, dùng từ khoá Flash `method` để tìm kiếm [Flash Support Center](#).

Example

Cách dùng 1: Trong ví dụ sau, hành vi `fscCommand` thiết đặt trình Flash Player để co dãn movie thành kích cỡ toàn màn hình khi nút được nhả ra (release).

```
on(release){
    fscommand("fullscreen", true);
}
```

Cách dùng 2: Ví dụ sau dùng hành vi `fscommand` áp dụng cho một nút trong Flash để mở một hộp message JavaScript trong một trang HTML. Bản thân message được gửi tới JavaScript như tham số `fscommand`.

Bạn phải thêm một hàm vào trang HTML có chứa Flash movie. Hàm này, `myMovie_DoFSCommand` đặt trong trang HTML và chờ đợi một hành vi `fscommand` trong Flash. Khi nào một `fscommand` được khởi sự trong Flash (ví dụ, khi một người dùng nhấn nút), các chuỗi `command` và `parameter` được chuyển tới hàm `myMovie_DoFSCommand`. Bạn có thể dùng các chuỗi đã chuyển đến trong mã JavaScript hoặc VBScript của bạn theo bất kỳ cách nào mà bạn thích. Trong ví dụ này, hàm chứa một câu lệnh điều kiện `if` để kiểm tra việc quan sát nếu chuỗi lệnh là "messagebox". Nếu là nó, một JavaScript alert box (hay "messagebox") mở ra và hiển thị nội dung của chuỗi `parameters`.

```
function myMovie_DoFSCommand(command, args) {
    if (command == "messagebox") {
        alert(args);
    }
}
```

Trong tài liệu Flash, thêm một hành vi `fscommand` vào một nút:

```
fscommand("messagebox", "Đây là một hộp message được gọi từ bên trong Flash.")
```

Bạn cũng có thể dùng các biểu thức cho hành vi và các tham số `fscommand`, như trong ví dụ sau:

```
fscommand("messagebox", "Xin chào, " + name + ", Chào mừng bạn vào tham Web site của chúng tôi!")
```

Để kiểm tra movie, chọn File > Publish Preview > HTML.

Chú ý: Nếu bạn xuất bản movie của bạn bằng cách sử dụng Flash với khuôn mẫu (template) `FSCommand` trong HTML Publish Settings, hàm `myMovie_DoFSCommand` được chèn vào một cách tự động. Các thuộc tính `NAME` và `ID` của movie sẽ là tên file. Ví dụ, cho file `myMovie.fl`, các thuộc tính sẽ được thiết lập là `myMovie`.

function

Availability

Flash Player 5.

Usage

```
function functionname ([parameter0, parameter1,...parameterN]){
    statement(s)
}
function ([parameter0, parameter1,...parameterN]){
```



```
        statement(s)
    }
```

Parameters

`functionname` Tên của hàm mới.

`parameter` Một từ định danh đại diện cho một tham số để chuyển tới hàm. Các tham số này là tùy chọn.

`statement(s)` Bất kỳ lệnh ActionScript mà bạn đã định nghĩa cho phần thân của `function`.

Returns

Nothing.

Description

Action (hành vi); một thiết đặt của câu lệnh mà bạn định nghĩa để thực hiện một nhiệm vụ nào đó. Bạn có thể trình bày, hoặc định nghĩa, một hàm trong một vị trí xác định và gọi, hoặc viện dẫn nó từ các script khác nhau trong một movie. Khi bạn định nghĩa một hàm, bạn cũng có thể chỉ định các tham số cho hàm. Các tham số là trình giữ chỗ (placeholders) cho các giá trị mà hàm điều khiển. Bạn có thể chuyển các tham số khác nhau tới một hàm mỗi lần bạn gọi nó. Điều này cho phép bạn dùng lại một hàm trong nhiều hoàn cảnh khác nhau.

Sử dụng hành vi `return` trong `statement(s)` (câu lệnh) của một hàm để tạo ra một hàm để trả lại, hoặc tạo, một giá trị.

Cách dùng 1: Biểu thị một hàm `function` với `functionname` (tên hàm), `parameters` (các tham số), và `statement(s)` (các câu lệnh) xác định. Khi một hàm được gọi, việc khai báo hàm được viện dẫn. Sự giới thiệu tiếp tới được cho phép (Forward referencing is permitted); bên trong cùng danh sách Action, một hàm có thể được công khai sau khi nó được gọi. Một hàm công bố thay thế (function declaration) thay thế bất kỳ tuyên bố trước của cùng hàm (replaces any prior declaration of the same function). Bạn có thể dùng cú pháp này bất cứ khi nào một khai báo được thừa nhận (You can use this syntax wherever a statement is permitted).

Cách dùng 2: Tạo một hàm nặc danh và trả nó về. Cú pháp này được dùng trong các biểu thức, và đặc biệt hữu ích cho việc cài đặt các method trong các đối tượng.

Example

Cách dùng 1: Ví dụ sau định nghĩa hàm `sqr`, là hàm chấp nhận một tham số và trả về bình phương `square(x*x)` của tham số. Chú ý rằng nếu hàm được công khai và được dùng trong cùng script, sự công bố hàm có thể xuất hiện sau khi dùng hàm.

```
y=sqr(3);

function sqr(x) {
    return x*x;
}
```

```
}
```

Cách dùng 2: Hàm sau đây định nghĩa một đối tượng Circle (hình tròn):

```
function Circle(radius) {  
  this.radius = radius;  
}
```

Khai báo sau định nghĩa một hàm nặc danh để tính toán diện tích của một hình tròn và gán nó cho đối tượng `Circle` như một method:

```
Circle.prototype.area = function () {return Math.PI * this.radius * this.radius}
```

getProperty

Availability

Flash Player 4.

Usage

```
getProperty(instancename , property)
```

Parameters

`instancename` Tên minh họa của một movie clip cho cái mà thuộc tính đang được gọi (The instance name of a movie clip for which the property is being retrieved).

`property` Một thuộc tính của một movie clip.

Returns

Nothing.

Description

Hàm (Function); trả lại giá trị của thuộc tính xác định `property` cho movie clip `instancename`.

Example

Đoạn mã sau gọi trục tọa độ ngang (`_x`) cho movie clip `myMovie` và gán nó cho biến `myMovieX`:

```
myMovieX = getProperty(_root.myMovie, _x);
```

getTimer

Availability

Flash Player 4.

Usage

```
getTimer()
```

Parameters

None.

Returns

Nothing.

Description

Hàm; trả lại số mili-giây (một phần nghìn giây) đã trôi qua kể từ khi movie bắt đầu chơi.

getURL

Availability

Flash 2. Các tùy chọn GET (lấy) và POST (gửi, bổ sung dữ liệu) chỉ có có hiệu lực sử dụng trong Flash Player 4 và các phiên bản về sau của Player.

Usage

```
getURL(url [, window [, "variables"]])
```

Parameters

`url` Địa chỉ URL (địa chỉ tới một trạm Internet hay mạng nội bộ) mà ở đó chứa tài liệu.

`window` Một tham số tùy chọn chỉ định cửa sổ (window) hoặc HTML frame mà tài liệu sẽ nạp vào trong. Bạn có thể nhập tên của một cửa sổ xác định hoặc chọn từ các tên có sẵn sau:

- ✦ `_self` chỉ định frame hiện tại trong cửa sổ hiện tại.
- ✦ `_blank` chỉ định một cửa sổ mới.
- ✦ `_parent` chỉ định parent (mẹ) của frame hiện tại.
- ✦ `_top` chỉ định frame có mức cao nhất (top-level frame) trong cửa sổ hiện tại.

`variables` Một method GET hoặc POST cho việc gửi các biến. Nếu không có biến nào, bỏ qua tham số này. Method GET bổ sung các biến vào cuối của URL, và được dùng cho số biến nhỏ. Method POST gửi các biến trong một đầu đề HTTP được ngăn cách và được dùng để gửi các chuỗi dài của biến.

Returns

Nothing.

Description

Action (hành vi); nạp một tài liệu từ một URL xác định vào trong một cửa sổ hoặc chuyển các biến tới một trình ứng dụng khác tại một URL đã được xác định. Để kiểm tra hành vi này, hãy chắc chắn rằng file được nạp đặt ở một vị trí xác định. Để sử dụng một địa chỉ URL tuyệt đối (ví dụ, `http://www.myserver.com`), bạn cần một kết nối vào mạng máy tính.

Example

Ví dụ này nạp một địa chỉ URL mới vào trong một cửa sổ trình duyệt trống (blank browser window). Hành vi `getUrl` nhắm tới biến `incomingAd` như tham số `url` với kết quả là bạn có thể thay đổi địa chỉ URL đã nạp mà không phải chỉnh sửa Flash movie. Giá trị của biến `incomingAd` được chuyển vào trong Flash nhanh hơn movie sử dụng một hành vi `loadVariables`.

```
on(release) {
    getUrl(incomingAd, "_blank");
}
```

See also

[loadVariables](#), [XML.send](#), [XML.sendAndLoad](#), [XMLSocket.send](#)

_global

Availability

Flash Player 6.

Usage

```
_global.identifier
```

Parameters

None.

Returns

Một tham chiếu tới đối tượng chung (global object) nắm giữ phần lõi của các lớp ActionScript, như là String (chuỗi), Object (đối tượng), Math (toán), và Array (mảng).

Description

Identifier (từ định danh); tạo các biến, đối tượng, hoặc lớp chung. Ví dụ, bạn có thể tạo một thư viện được đặt vào như một đối tượng ActionScript chung, giống như đối tượng Math hoặc Date. Khác

với Timeline-công khai hoặc mang tính cục bộ-công khai các biến và hàm, các biến và hàm chung đều nhìn thấy được trong mọi Timeline và phạm vi trong Flash movie, miễn là chúng không bị làm mờ đi bởi các từ định danh cùng tên với các phạm vi bên trong.

Example

Ví dụ sau tạo một hàm mức cao nhất `factorial` có hiệu lực với mọi Timeline và phạm vi (scope) trong một Flash movie:

```
_global.factorial = function (n) {  
  if (n <= 1) {  
    return 1;  
  } else {  
    return n * factorial(n-1);  
  }  
}
```

See also

[var](#), [set variable](#)

Key (object)

Đối tượng Key (phím) là một đối tượng cao cấp mà bạn có thể truy cập (access) mà không cần sử dụng một constructor (trình xây dựng). Sử dụng các method (phương pháp) của đối tượng Key để tạo một giao diện mà có thể được điều khiển bởi một người dùng với một bàn phím chuẩn. Các thuộc tính của đối tượng Key object là những hằng số đại diện cho các phím vốn hầu như thường xuyên được sử dụng để điều khiển các trò chơi. Để biết một danh sách đầy đủ của các giá trị mã phím, hãy xem phụ lục [Keyboard Keys and Key Code Values](#) (Các phím trên bàn phím và các giá trị mã phím) trong phần Using Flash.

Example

Đoạn script sau đây sử dụng đối tượng Key để nhận biết các phím trên bàn phím bất kỳ để người sử dụng có thể điều khiển một movie clip.

```
onClipEvent (enterFrame) {  
  if(Key.isDown(Key.RIGHT)) {  
    this._x=_x+10;  
  } else if (Key.isDown(Key.DOWN)) {  
    this._y=_y+10;  
  }  
}
```

Tóm tắt các method của đối tượng Key object

Method	Description
Key.addListener	Đăng ký một đối tượng để tiếp nhận thông báo khi các method

	onKeyDown và onKeyUp được viện dẫn.
Key.getAscii	Trả về giá trị ASCII của phím cuối cùng đã nhấn.
Key.getCode	Trả về mã phím tượng trưng (virtual key code) của phím cuối cùng đã nhấn.
Key.isDown	Trả về true nếu phím đã xác định trong tham số được nhấn.
Key.isToggled	Trả về true nếu phím Num Lock hoặc Caps Lock được kích hoạt.
Key.removeListener	Gỡ bỏ một đối tượng mà trước đó đã đăng ký với addListener.

Tóm tắt các thuộc tính của đối tượng Key object

Tất cả các thuộc tính cho đối tượng Key đều là các hằng số.

Property	Description
Key.BACKSPACE	Hằng số kết hợp với giá trị mã phím (key code value) cho phím Backspace (8).
Key.CAPSLock	Hằng số kết hợp với giá trị mã phím đại diện cho phím Caps Lock (20).
Key.CONTROL	Hằng số kết hợp với giá trị mã phím đại diện cho phím Control (17).
Key.DELETEKEY	Hằng số kết hợp với giá trị mã phím đại diện cho phím Delete (46).
Key.DOWN	Hằng số kết hợp với giá trị mã phím đại diện cho phím mũi tên xuống Down Arrow (40).
Key.END	Hằng số kết hợp với giá trị mã phím đại diện cho phím End (35).
Key.ENTER	Hằng số kết hợp với giá trị mã phím đại diện cho phím Enter (13).
Key.ESCAPE	Hằng số kết hợp với giá trị mã phím đại diện cho phím Escape (27).
Key.HOME	Hằng số kết hợp với giá trị mã phím đại diện cho phím Home (36).
Key.INSERT	Hằng số kết hợp với giá trị mã phím đại diện cho phím Insert (45).
Key.LEFT	Hằng số kết hợp với giá trị mã phím đại diện cho phím mũi tên trái Left Arrow (37).
Key.PGDN	Hằng số kết hợp với giá trị mã phím đại diện cho phím Page Down (34).
Key.PGUP	Hằng số kết hợp với giá trị mã phím đại diện cho phím Page Up (33).
Key.RIGHT	Hằng số kết hợp với giá trị mã phím đại diện cho phím mũi tên sang phải Right

	Arrow (39).
Key.SHIFT	Hằng số kết hợp với giá trị mã phím đại diện cho phím Shift (16).
Key.SPACE	Hằng số kết hợp với giá trị mã phím đại diện cho phím Spacebar (32).
Key.TAB	Hằng số kết hợp với giá trị mã phím đại diện cho phím Tab (9).
Key.UP	Hằng số kết hợp với giá trị mã phím đại diện cho phím mũi tên lên Up Arrow (38).

Listener summary for the Key object

Method	Description
Key.onKeyDown	Được thông báo khi một phím được nhấn
Key.onKeyUp	Được thông báo khi một phím được nhả (release).

LoadVars (object) {Nạp các biến (đối tượng)}

Đối tượng LoadVars là một sự lựa chọn đến hành vi `loadVariables` để chuyển giao các biến giữa một Flash movie và một máy chủ server.

Bạn có thể sử dụng đối tượng LoadVars để thu được các thông tin lỗi, các số chỉ tiến trình, và dữ liệu stream trong khi nó đang tải xuống. Đối tượng LoadVars làm việc tương tự như đối tượng XML; nó sử dụng các method `load`, `send`, và `sendAndLoad` để giao tiếp với một server. Điểm khác nhau cơ bản giữa đối tượng LoadVars và đối tượng XML là ở chỗ LoadVars tên và các cặp giá trị ActionScript, hơn là một cây XML DOM tree cất giữ trong đối tượng XML.

Đối tượng LoadVars cũng theo cùng những hạn chế bảo mật như đối tượng XML.

Bạn phải sử dụng constructor `new LoadVars()` để tạo một instance (thể hiện) của đối tượng LoadVars trước khi gọi các method của nó.

Đối tượng LoadVars được hỗ trợ bởi Flash Player 6 và các phiên bản sau đó.

Bảng tổng quan các method cho đối tượng LoadVars

Method	Description
LoadVars.load	Tải xuống các biến từ một địa chỉ URL xác định.
LoadVars.getBytesTotal	Trả về số byte đã nạp từ một method <code>load</code> hoặc <code>sendAndLoad</code> .

LoadVars.getBytesTotal	Trả về tổng số byte mà sẽ được tải xuống bởi một method <code>load</code> hoặc <code>sendAndLoad</code> .
LoadVars.send	Gửi các biến từ một đối tượng <code>LoadVars</code> tới một địa chỉ URL.
LoadVars.sendAndLoad	Gửi các biến từ một đối tượng <code>LoadVars</code> tới một địa chỉ URL và tải xuống sự trả lời của server tới một đối tượng đích (target object).
LoadVars.toString	Trả về một địa chỉ URL mã hóa chuỗi mà chứa tất cả các biến có thể liệt kê trong đối tượng <code>LoadVars</code> .

Bảng tổng quan các thuộc tính cho đối tượng `LoadVars`

Tất cả các thuộc tính cho đối tượng `Key` đều là những hằng số.

Property	Description
LoadVars.contentType	Cho biết kiểu MIME của dữ liệu
LoadVars.load	Một giá trị logic Boolean cho biết xem một thao tác <code>load</code> hoặc <code>sendAndLoad</code> đã được hoàn thành hay chưa.

Bảng tổng quan các sự kiện cho đối tượng `LoadVars`

Method	Description
LoadVars.onLoad	Viện dẫn khi một thao tác <code>load</code> (nạp) hoặc <code>sendAndLoad</code> (gửi và nạp) được hoàn thành.

Constructor (người xây dựng) cho đối tượng `LoadVars`

Availability

Flash Player 6.

Usage

```
new LoadVars()
```


Parameters

None.

Returns

Nothing.

Description

Constructor (trình xây dựng); tạo một instance (thể hiện, minh họa) của đối tượng LoadVars. Sau đó bạn có thể sử dụng các method của đối tượng LoadVars đó để gửi và nạp dữ liệu.

Example

Ví dụ sau tạo một instance của đối tượng LoadVars đã gọi myLoadVars:

```
myLoadVars = new LoadVars();
```

_level

Availability

Flash Player 4.

Usage

```
_levelN
```

Description

Property (thuộc tính); một tham chiếu tới Timeline của movie gốc của _levelN. Bạn phải dùng hành vi loadMovieNum để nạp các movie vào trong trình Flash Player trước khi bạn sử dụng thuộc tính _level để hướng tới chúng. Bạn có thể dùng _levelN để hướng (target) tới một movie đã được nạp ở tại level (mức) đã gán bởi N.

Movie ban đầu đã nạp vào trong một instance (thể hiện) của Flash Player được tự động nạp vào trong _level0. Movie trong _level0 thiết đặt tốc độ frame, màu nền, và kích thước các frame đối với tất cả các movie đã được nạp sau đó. Các movie sau đó được xếp ngăn trong các mức (level) có số lớn hơn ở trên movie trong _level0.

Bạn phải gán một level vào mỗi movie mà bạn nạp vào trong Flash Player sử dụng hành vi loadMovieNum. Bạn có thể gán các mức trong bất kỳ thứ bậc nào. Nếu bạn gán một level mà đã chứa một file SWF (bao gồm cả _level0) movie ở tại level mà đã bị unload và thay thế bởi một movie mới.

Example

Ví dụ sau dừng đầu playhead trong Timeline chính của movie bên trong `_level19`.

```
_level19.stop();
```

Ví dụ sau gửi playhead trong Timeline chính của movie trong `_level14` tới frame 5. Movie trong `_level14` phải được nạp trước đó với một hành vi `loadMovieNum`.

```
_level14.gotoAndStop(5);
```

See also

[loadMovie](#), [MovieClip.swapDepths](#)

Math (object)

Đối tượng Math là một đối tượng cấp cao nhất mà bạn có thể truy xuất mà không dùng một constructor (người xây dựng).

Sử dụng các method và các thuộc tính của đối tượng này để truy xuất (access) và vận dụng các hàm và hằng số toán học. Tất cả các thuộc tính và method của đối tượng Math đều tĩnh, và phải được gọi sử dụng cú pháp `Math.method(parameter)` hoặc `Math.constant`. Trong ActionScript, các hằng số được xác định với độ chính xác lớn nhất của gấp đôi-độ chính xác IEEE-754 chữ số thập phân.

Một số method của đối tượng Math nhận radian của một góc như một tham số. Bạn có thể dùng phương trình dưới đây để tính toán các giá trị radian, hoặc đơn giản chuyển qua phương trình (nhập vào một giá trị theo độ) cho thông số radian.

Để tính toán một giá trị radian, dùng công thức sau:

```
radian = Math.PI/180 * degree
```

Sau đây là một ví dụ của việc chuyển đổi một phương trình thành như một tham số để tính sin của một góc 45 độ:

```
Math.SIN(Math.PI/180 * 45) cũng giống hệt như Math.SIN(.7854)
```

Đối tượng Math được hỗ trợ đầy đủ trong Flash Player 5. Trong Flash Player 4, bạn có thể sử dụng các method cho đối tượng Math, nhưng chúng cạnh tranh với việc dùng các phép xấp xỉ và có thể không chính xác như các hàm toán học không xung đột được hỗ trợ bởi Flash Player 5.

Bảng tóm lược các method của đối tượng Math object

Method	Description
Math.abs	Tính toán một giá trị tuyệt đối.
Math.acos	Tính toán một cung cosin.

Math.asin	Tính toán một cung sin.
Math.atan	Tính toán một cung tang.
Math.atan2	Tính toán một góc từ trục x đến điểm.
Math.ceil	Làm tròn một số lên số nguyên gần nhất.
Math.cos	Tính cosin.
Math.exp	Tính toán một giá trị mũ.
Math.floor	Làm tròn một số xuống số nguyên gần nhất.
Math.log	Tính toán một loga tự nhiên.
Math.max	Trả lại số lớn của hai số nguyên.
Math.min	Trả lại số nhỏ hơn trong hai số nguyên
Math.pow	Computes x raised to the power of the y .
Math.random	Trả lại một số giả ngẫu nhiên ở giữa 0.0 và 1.0.
Math.round	Làm tròn thành số nguyên gần nhất
Math.sin	Tính một sin.
Math.sqrt	Tính toán một số căn bình phương.
Math.tan	Tính tang.

Bảng tóm lược các thuộc tính của đối tượng Math object

Tất cả các thuộc tính (properties) cho đối tượng Math đều là các hằng số.

Property	Description
Math.E	Hằng số Euler và cơ số của loga tự nhiên (khoảng 2.718).
Math.LN2	Loga tự nhiên của 2 (khoảng 0.693).
Math.LOG2E	Loga cơ số 2 của e (khoảng 1.442).
Math.LN10	Loga tự nhiên của 10 (khoảng 2.302).
Math.LOG10E	Loga cơ số 10 của e (khoảng 0.434).
Math.PI	Tỉ lệ giữa chu vi của một đường tròn với đường kính của nó (khoảng 3.14159).

Math.SQRT1_2	Số nghịch đảo của căn bậc hai của 1/2 (khoảng 0.707).
Math.SQRT2	Số căn bậc hai của 2 (khoảng 1.414).

Mouse (object) {Con trỏ chuột (đối tượng)}

Đối tượng Mouse là một đối tượng cấp cao cho phép bạn truy cập mà không cần sử dụng một constructor (trình xây dựng). Bạn có thể sử dụng các method của đối tượng Mouse để ẩn và hiện con trỏ (cursor) trong movie. Con trỏ chuột được nhìn thấy theo mặc định, nhưng bạn có thể ẩn nó đi và bổ sung một con trỏ do bạn tạo bằng cách sử dụng một movie clip.

Bảng tóm lược các method của đối tượng Mouse

Method	Description
Mouse.addListener	Đăng ký một đối tượng để tiếp nhận thông báo <code>onMouseDown</code> , <code>onMouseMove</code> , và <code>onMouseUp</code> .
Mouse.hide	Ẩn con trỏ chuột trong movie.
Mouse.removeListener	Gỡ bỏ một đối tượng mà đã được đăng ký với method <code>addListener</code> .
Mouse.show	Hiện thị con trỏ chuột trong movie.

Bảng tóm lược các Mouse listener

Method	Description
MovieClip.onMouseDown	Thông báo khi nút bấm chuột được nhấn xuống.
MovieClip.onMouseMove	Thông báo khi nút bấm chuột được di chuyển.
MovieClip.onMouseUp	Thông báo khi nút bấm chuột được nhả ra (release).

MovieClip (object)

Các method cho đối tượng MovieClip cung cấp tính năng tương tự như các action hướng tới các movie clip. Cũng có cả các method bổ sung mà không có các action tương tự trong hộp công cụ Actions toolbox trong Actions panel.

Bạn không cần sử dụng một method xây dựng (constructor method) để gọi các method của đối tượng MovieClip; thay vì thế, bạn tham chiếu các instance movie clip bởi tên, sử dụng cú pháp sau:

```
myMovieClip.play();myMovieClip.gotoAndPlay(3);
```

Bảng tổng quan các method cho đối tượng MovieClip object

Method	Description
MovieClip.attachMovie	Đính kèm một movie trong thư viện (library).
MovieClip.createEmptyMovieClip	Tạo một movie clip rỗng.
MovieClip.createTextField	Tạo một trường text rỗng.
MovieClip.duplicateMovieClip	Sao chép movie clip đã chỉ định.
MovieClip.getBounds	Trả về các tọa độ x và y nhỏ nhất và lớn nhất của một movie trong một khoảng tọa độ xác định.
MovieClip.getBytesLoaded	Trả về số byte đã nạp cho movie clip đã chỉ định.
MovieClip.getBytesTotal	Trả về kích cỡ của movie clip theo byte.
MovieClip.getDepth	Trả về độ dày của một movie clip.
MovieClip.getURL	Truy lục một tài liệu từ một địa chỉ URL.
MovieClip.globalToLocal	Chuyển đổi đối tượng trỏ (point object) từ các tọa độ Stage thành các tọa độ cục bộ của movie clip đã chỉ định.
MovieClip.gotoAndPlay	Gửi đầu chạy playhead tới một frame xác định trong movie clip và trình diễn movie.
MovieClip.gotoAndStop	Gửi đầu chạy playhead tới một frame xác định trong movie clip và dừng movie lại.
MovieClip.hitTest	Trả về true nếu khung giới hạn của movie clip đã chỉ định giao với khung giới hạn của movie clip đích (target movie clip).
MovieClip.loadMovie	Nạp movie đã chỉ định vào trong movie clip.
MovieClip.loadVariables	Nạp các biến từ một địa chỉ URL hoặc vị trí khác vào trong movie clip.
MovieClip.localToGlobal	Chuyển đổi một đối tượng Point object từ các tọa độ cục bộ của movie clip thành các tọa độ Stage chung.
MovieClip.nextFrame	Gửi đầu chạy playhead tới frame kế tiếp của movie clip.
MovieClip.play	Trình diễn movie clip đã chỉ định.
MovieClip.prevFrame	Gửi đầu chạy playhead đến frame liền trước của movie

	clip.
MovieClip.removeMovieClip	Gỡ bỏ movie clip từ Timeline nếu nó được tạo bởi action <code>duplicateMovieClip</code> hoặc method <code>attachMovie</code> .
MovieClip.setMask	Chỉ định một movie clip như một lớp mặt nạ (mask) cho movie clip khác.
MovieClip.startDrag	Chỉ định một movie clip có khả năng kéo (draggable) và bắt đầu kéo movie clip.
MovieClip.stop	Dừng movie đang trình diễn.
MovieClip.stopDrag	Dừng việc kéo bất kỳ movie clip nào đang được kéo.
MovieClip.swapDepths	Đổi mức dày (depth level) của hai movie.
MovieClip.unloadMovie	Gỡ bỏ một movie mà đã được nạp bởi hành vi <code>loadMovie</code> .

Drawing method summary cho MovieClip

Method	Description
MovieClip.beginFill	Bắt đầu vẽ (drawing) một fill (vùng đồ họa hiển thị tất cả cùng một màu hoặc cùng một mẫu) trên Stage.
MovieClip.beginGradientFill	Bắt đầu vẽ một khoảng đầy chênh lệch (gradient fill: trong đồ họa vi tính, khoảng đầy bao gồm một sự kết hợp hài hòa từ từ màu đầu tiên tới màu cuối cùng) trên Stage.
MovieClip.clear	Gỡ bỏ tất cả các lệnh vẽ kết hợp với một movie clip instance.
MovieClip.curveTo	Vẽ một đường cong bằng cách sử dụng kiểu đường thẳng sau cùng (Draws a curve using the latest line style).
MovieClip.endFill	Kết thúc fill đã xác định bởi <code>beginFill</code> hoặc <code>beginGradientFill</code> .
MovieClip.lineStyle	Xác định nét của các đường (stroke of lines) đã tạo bởi các method <code>lineTo</code> và <code>curveTo</code> .
MovieClip.lineTo	Vẽ một đường kẻ bằng cách sử dụng kiểu đường (line style) hiện tại.
MovieClip.moveTo	Di chuyển vị trí đang vẽ hiện tại tới các tọa độ xác định.

Bảng tổng quan các thuộc tính cho đối tượng MovieClip object

Property	Description
<u>MovieClip._alpha</u>	Giá trị trong suốt của một movie clip instance.
<u>MovieClip._currentframe</u>	Số của frame mà tại đó đầu chạy playhead đang được định vị.
<u>MovieClip._droptarget</u>	Đường dẫn tuyệt đối trong ký hiệu cú pháp vạch xiên (slash syntax notation) của instance movie clip trên đó một movie clip có khả năng kéo (draggable) được thả (was dropped).
<u>MovieClip.enabled</u>	Cho biết xem một nút movie clip (button movie clip) có được hiệu lực hóa, cho phép hoạt động hay không.
<u>MovieClip.focusEnabled</u>	Hiệu lực hóa một movie clip để tiếp nhận tiêu điểm.
<u>MovieClip._focusrect</u>	Cho biết xem một movie clip được đặt tiêu điểm có một hình chữ nhật màu vàng bao quanh nó hay không.
<u>MovieClip._framesloaded</u>	Số các frame đã được nạp từ một streaming movie (dữ liệu movie lưu trữ trong một dạng thức cho phép phần đầu của file được thi hành ngay cả trước khi các phần sau của file được đọc).
<u>MovieClip._height</u>	Chiều cao của một movie clip instance, theo pixel.
<u>MovieClip.hitArea</u>	Chỉ rõ một movie clip để dùng như vùng bấm (hit area) cho một button movie clip.
<u>MovieClip._highquality</u>	Thiết đặt chất lượng biến đổi hình ảnh của một movie.
<u>MovieClip._name</u>	Tên minh họa (instance name) của một instance movie clip.
<u>MovieClip._parent</u>	Một tham chiếu tới movie clip mà nó chứa movie clip.
<u>MovieClip._rotation</u>	Độ xoay vòng của một movie clip instance.
<u>MovieClip._soundbuftime</u>	Số giây trước khi một âm thanh bắt đầu stream.
<u>MovieClip.tabChildren</u>	Cho biết xem các "con" (children) của một movie clip được bao gồm trong trình tự tab tự động (Indicates whether the children of a movie clip are included in automatic tab ordering).
<u>MovieClip.tabEnabled</u>	Cho biết xem một movie clip được bao gồm trong trình tự tab hay không (Indicates whether a movie clip is included in tab ordering).
<u>MovieClip.tabIndex</u>	Cho biết trình tự tab của một đối tượng (Indicates the tab order of an object).

MovieClip._target	Đường dẫn đích của một movie clip instance.
MovieClip._totalframes	Tổng số frame trong một instance movie clip.
MovieClip.trackAsMenu	Cho biết xem các nút khác có thể tiếp nhận các sự kiện chuột được nhả không.
MovieClip._url	Địa chỉ URL của file SWF từ đó một movie clip được tải xuống.
MovieClip.useHandCursor	Quyết định xem bàn tay có được hiển thị khi một người dùng cuộn lên trên một nút button movie clip.
MovieClip._visible	Một giá trị logic Boolean quyết định xem một instance movie clip là ẩn hay nhìn thấy được.
MovieClip._width	Chiều rộng của một movie clip instance, theo pixels.
MovieClip._x	Tọa độ x của một instance movie clip
MovieClip._xmouse	Tọa độ x của con trỏ bên trong một movie clip instance.
MovieClip._xscale	Giá trị xác định tỷ lệ phần trăm cho việc thu phóng một movie clip theo chiều ngang.
MovieClip._y	Tọa độ y của một instance movie clip
MovieClip._ymouse	Tọa độ y của con trỏ bên trong một movie clip instance.
MovieClip._yscale	Giá trị xác định tỷ lệ phần trăm cho việc thu phóng một movie clip theo chiều dọc.

Event handler summary cho đối tượng MovieClip object

Property	Description
MovieClip.onData	Viện dẫn khi tất cả dữ liệu được nạp vào trong một movie clip.
MovieClip.onDragOut	Viện dẫn khi con trỏ ở ngoài nút; nút chuột đã được nhấn ở bên trong, và sau đó cuộn ra ngoài vùng nút.
MovieClip.onDragOver	Viện dẫn trong khi con trỏ ở trên nút; nút chuột đã được nhấn sau đó đã cuộn ra ngoài nút, và sau đó lại cuộn trở lại lên trên nút.
MovieClip.onEnterFrame	Viện dẫn liên tục theo vận tốc frame của movie. Các action kết hợp với sự kiện clip enterFrame đều được giải quyết trước bất kỳ frame action nào được gắn kèm vào các frame bị gắn.

MovieClip.onKeyDown	Viện dẫn khi một phím được nhấn. Sử dụng các method <code>Key.getCode</code> và Key.getAscii để tiếp nhận thông tin về phím cuối cùng đã nhấn.
MovieClip.onKeyUp	Viện dẫn khi một phím được thả ra (release).
MovieClip.onKillFocus	Viện dẫn khi tiêu điểm bị gỡ bỏ khỏi một nút.
MovieClip.onLoad	Viện dẫn khi movie clip được thuyết minh (instantiated) và xuất hiện trong Timeline.
MovieClip.onMouseDown	Viện dẫn khi nút chuột trái được nhấn.
MovieClip.onMouseMove	Viện dẫn mỗi khi chuột được di chuyển.
MovieClip.onMouseUp	Viện dẫn khi nút chuột trái được thả.
MovieClip.onPress	Viện dẫn khi chuột được nhấn trong khi con trỏ đang ở trên một nút.
MovieClip.onRelease	Viện dẫn khi chuột được thả ra trong khi con trỏ đang ở trên một nút.
MovieClip.onReleaseOutside	Viện dẫn khi chuột được thả trong khi con trỏ ra ngoài nút sau khi nút được nhấn khi con trỏ ở trong nút.
MovieClip.onRollOut	Viện dẫn khi con trỏ cuộn ra ngoài một vùng nút.
MovieClip.onRollOver	Viện dẫn khi con trỏ chuột cuộn lên trên một nút.
MovieClip.onSetFocus	Viện dẫn khi một nút có một tiêu điểm nhập vào (input focus) và một phím được thả.
MovieClip.onUnload	Viện dẫn trong frame đầu tiên sau khi movie clip được gỡ bỏ từ Timeline. Các hành vi kết hợp với sự kiện movie clip Unload đều được xử lý trước khi các hành vi bất kỳ được gắn vào các affected frame(các frame bị ảnh hưởng).

MovieClip._url

Availability

Flash Player 4.

Usage

myMovieClip._url

Description

Property (read only) {Thuộc tính (chỉ đọc)}; truy lục (gọi ra) địa chỉ URL của file SWF từ đó movie clip đã được tải xuống.

MovieClip.useHandCursor

Availability

Flash Player 6.

Usage

```
myMovieClip.useHandCursor
```

Description

Property (thuộc tính); một giá trị logic Boolean cho biết xem một con trỏ hình bàn tay có hiển thị khi một người dùng cuộn ngang qua một nút button movie clip hay không. Giá trị mặc định của `useHandCursor` là `true`. Nếu `useHandCursor` được thiết đặt là `true`, con trỏ chuẩn hình bàn tay sẽ dùng cho các nút sẽ được hiển thị khi một người dùng cuộn trên một button movie clip. Nếu `useHandCursor` là `false`, con trỏ hình mũi tên sẽ được dùng thay thế.

Bạn có thể thay đổi thuộc tính `useHandCursor` bất kỳ lúc nào; button movie clip đã sửa đổi ngay lập tức nhận một xử lý con trỏ mới (new cursor behavior). Thuộc tính `useHandCursor` có thể được đọc bên ngoài một đối tượng mẫu ban đầu.

MovieClip._visible

Availability

Flash Player 4.

Usage

```
myMovieClip._visible
```

Description

Property (thuộc tính); một giá trị logic Boolean xác định xem movie đã chỉ định bởi tham số `MovieClip` là nhìn thấy được hay không. Các movie clip không nhìn thấy được (thuộc tính `_visible` thiết đặt là `false`) đều bị vô hiệu hóa. Ví dụ, một nút trong một movie clip với thuộc tính `_visible` thiết đặt thành `false` không thể click được.

MovieClip._width

Availability

Flash Player 4 as a read-only property.

Usage

```
myMovieClip._width
```

Description

Property (thuộc tính); thiết đặt và truy gọi chiều rộng của movie clip, theo pixel.

Example

Ví dụ sau thiết đặt các thuộc tính chiều cao và chiều rộng của một movie clip khi người dùng click chuột.

```
onClipEvent(mouseDown) {      _width=200;      _height=200; }
```

See also

[MovieClip._height](#)

MovieClip._x

Availability

Flash Player 3.

Usage

```
myMovieClip._x
```

Description

Property (thuộc tính); một số nguyên thiết đặt tọa độ x của movie ứng với các tọa độ địa phương (local coordinates) của movie clip mẹ (parent movie clip). Nếu một movie clip ở trong Timeline chính, khi đó hệ tọa độ của nó quy cho góc trên bên trái của Stage là (0, 0). Nếu movie clip ở bên trong movie clip khác mà có sự biến đổi, movie clip ở trong hệ tọa độ địa phương (cục bộ) của movie clip bao quanh. Do đó, với một movie clip quay 90° theo chiều kim đồng hồ, các "con" của movie clip (movie clip's children) thừa hưởng một hệ tọa độ cục bộ cũng quay 90° theo chiều kim đồng hồ. Các tọa độ của movie clip dựa vào vị trí của điểm đăng ký.

MovieClip._xmouse

Availability

Flash Player 5.

Usage

```
myMovieClip._xmouse
```

Description

Property (read-only) {Thuộc tính (chỉ đọc)}; trả về tọa độ x của vị trí chuột.

MovieClip._xscale

Availability

Flash Player 4.

Usage

```
myMovieClip._xscale
```

Description

Property (thuộc tính); xác định độ co giãn theo chiều ngang (percentage: tỷ lệ phần trăm) của movie clip áp dụng từ điểm đăng ký của movie clip. Điểm đăng ký mặc định là (0,0).

Việc chia tỷ lệ hệ tọa độ cục bộ ảnh hưởng đến các thiết đặt thuộc tính `_x` và `_y`, được định nghĩa theo pixel. Ví dụ, nếu movie clip mẹ được co lại thành 50%, việc thiết đặt thuộc tính `_x` di chuyển một đối tượng trong movie clip bằng một nửa số pixel mà nó có thể nếu movie được thiết đặt ở 100%.

Number (function)

Availability

Flash Player 4.

Usage

```
Number(expression)
```

Parameters

`expression` Một biểu thức để chuyển thành một số.

Returns

Nothing.

Description

Function (hàm); chuyển đổi tham số `expression` thành một số và trả về một giá trị như sau:

Nếu `expression` là một số, giá trị trả về là `expression`.

Nếu `expression` là một giá trị logic Boolean, giá trị trả về là 1 nếu `expression` là `true`, 0 nếu `expression` là `false`.

Nếu `expression` là một chuỗi, hàm cố gắng để phân tích `expression` như một số thập phân với số mũ kéo theo tùy chọn, đó là, `1.57505e-3`.

Nếu `expression` là `undefined`, giá trị trả về là 0.

Hàm này được dùng để chuyển đổi các file Flash 4 đang chứa những toán tử không được tán thành mà được nhập (imported) vào trong môi trường thiết kế Flash 5. Hãy xem toán tử & để biết thêm thông tin.

Object (object) {Đối tượng (đối tượng)}

Đặc điểm chung của đối tượng Object là ở gốc (root) của phân lớp ActionScript. Điểm chung của đối tượng ActionScript Object đều chứa một nhóm nhỏ các tính năng cung cấp bởi đối tượng JavaScript Object. Trong Flash MX, đối tượng Object trở thành một đối tượng tự nhiên (có sẵn). Vì vậy, bạn sẽ trải qua những cái tiến đáng ngạc nhiên trong khi thực thi.

Điểm chung của đối tượng Object là đều được hỗ trợ trong Flash Player 5.

Bảng tóm lược các method của đối tượng Object

Method	Description
Object.addProperty	Tạo một thuộc tính lấy/thiết lập (getter/setter property) trên một đối tượng.
Object.registerClass	Gán một lớp ActionScript vào một instance của movie clip.
Object.toString	Chuyển đổi đối tượng đã chỉ định thành một chuỗi, và trả nó về.
Object.unwatch	Gỡ bỏ đăng ký mà một method <code>Object.watch</code> đã tạo ra.
Object.valueOf	Trả về giá trị nguyên gốc của một đối tượng Object.
Object.watch	Đăng ký một hàm gọi trở về (callback function) để viện dẫn khi một thuộc tính đã chỉ định của một đối tượng ActionScript thay đổi.

Bảng tóm lược các thuộc tính của đối tượng Object

Property	Description
Object.__proto__	Một tham chiếu tới thuộc tính <code>prototype</code> (mẫu ban đầu) của hàm xây dựng của đối tượng.

Constructor cho Object object

Availability

Flash Player 5.

Usage

```
new Object([value])
```

Parameters

`value` Một số, giá trị logic Boolean, hoặc chuỗi được chuyển đổi thành một đối tượng. Tham số này là tùy chọn. Nếu bạn không chỉ định giá trị `value`, trình xây dựng constructor sẽ tạo một đối tượng mới với các thuộc tính không được định nghĩa.

Description

Constructor; tạo một đối tượng Object mới.

Object.registerClass

Availability

Flash Player 6

Usage

```
Object.registerClass(symbolID, theClass)
```

Parameters

`symbolID` Từ định dạng liên kết của movie clip symbol, hoặc chuỗi từ định danh của lớp `ActionScript`.

`theClass` Một tham chiếu tới trình xây dựng hàm (constructor function) của lớp `ActionScript`, hoặc `null` với symbol chưa đăng ký.

Returns

Nếu sự đăng ký lớp thành công, một giá trị `true` sẽ được trả về; nếu không thì, `false` được trả về.

Decription

Method (phương pháp, cách thức); kết hợp một movie clip symbol với một lớp đối tượng `ActionScript`. Nếu một symbol không tồn tại, Flash tạo một sự kết hợp giữa một chuỗi định danh (a string identifier) và một lớp đối tượng (an object class).

Khi một instance của movie clip symbol xác định được đặt vào Timeline, nó được đăng ký thành lớp xác định bởi tham số `theClass` hơn là thành lớp `MovieClip`.

Khi một instance của movie clip symbol xác định được tạo bằng cách sử dụng các method `attachMovie` hoặc `duplicateMovieClip`, nó được đăng ký thành lớp xác định bởi tham số `theClass` parameter hơn là thành lớp `MovieClip`.

Nếu `theClass` là `null`, `Object.registerClass` sẽ gỡ bỏ bất kỳ sự định nghĩa lớp `ActionScript` kết hợp với movie clip symbol đã xác định hoặc lớp định danh (class identifier). Đối với các movie clip symbol, bất kỳ instance nào đang tồn tại của movie clip vẫn giữ nguyên không thay đổi, nhưng các instance mới của symbol đều kết hợp với lớp `MovieClip` mặc định.

Nếu một biểu tượng (symbol) đã được đăng ký thành một lớp, method `Object.registerClass` sẽ thay thế nó với sự đăng ký mới.

Khi một instance movie clip được đặt bởi Timeline hoặc đã tạo bằng cách sử dụng `attachMovie` hoặc `duplicateMovieClip`, `ActionScript` viện dẫn constructor (trình xây dựng) cho lớp thích hợp với từ khóa `this` nhằm vào đối tượng. Hàm xây dựng (constructor function) được viện dẫn không có tham số nào cả.

Nếu bạn sử dụng method `Object.registerClass` để đăng ký một movie clip với một lớp `ActionScript` khác so với `MovieClip`, movie clip symbol không được thừa hưởng các method, các thuộc tính (properties), và sự kiện gắn liền với lớp `MovieClip` trừ phi bạn kèm cả (include) lớp `MovieClip` trong chuỗi ban đầu của lớp mới. Đoạn mã sau tạo một lớp `ActionScript` mới gọi là `theClass` thừa hưởng các thuộc tính của lớp `MovieClip`:

```
theClass.prototype = new MovieClip();
```

Example

Ví dụ này tạo một component cho một standard check box UI widget (hộp kiểm giao diện người dùng...).

Đầu tiên bạn tạo một movie clip symbol gọi là Check Box (hộp kiểm) trong thư viện library. Sau đó bạn tạo một lớp `CheckBox` trong `ActionScript` cái mà sẽ đại diện cho check box.

```
// Định nghĩa constructor cho (và do đó định nghĩa) CheckBox classfunction
CheckBox() {...} // Thiết đặt chuỗi ban đầu CheckBox// để thừa kế từ
MovieClipCheckBox.prototype = new MovieClip();// Định nghĩa các method cho lớp
CheckBox CheckBox.prototype.setLabel = function (newLabel) { this.label =
newLabel; ...};CheckBox.prototype.setValue = function (newValue) {
this.value = value; ...};CheckBox.prototype.getValue = function () {
return this.value;};CheckBox.prototype.getLabel = function () { return
this.label;};
```

Bây giờ bạn phải kết hợp lớp `CheckBox` với movie clip symbol Check Box. Đầu tiên, bạn cần có khả năng nhận dạng movie clip symbol Check Box với `ActionScript`. Để làm điều đó, nhập một từ định danh trong hộp thoại `Linkage` (sự liên kết) trong thư viện (library) và chọn `Export for ActionScript` (Xuất ra cho `ActionScript`).

Tiếp đó, viết ActionScript để kết hợp lớp CheckBox với symbol CheckBox:

```
Object.registerClass("CheckBox" /*symbolID*/, CheckBox /*theClass*/ );
```

Cách dùng 1(Timeline sắp đặt): Bây giờ bạn đặt các instance (thể hiện, trường hợp) của CheckBox trên stage trong công cụ thiết kế (chính là cái chương trình Macromedia Flash đó), và tại thời gian chạy (at runtime) các instance sẽ tự động tiếp nhận lớp ActionScript là CheckBox. Nếu bạn đặt hai instance, myCheckBox1 (hộp kiểm thứ nhất của tôi) và myCheckBox2, bạn có thể điều khiển chúng bằng cách viện dẫn các method, như sau:

```
myCheckBox1.setValue(true);myCheckBox2.setValue(false);myCheckBox2.setLabel("new label for #2");
```

Cách dùng 2 (Dynamic instances: các instance động): Bạn có thể sử dụng method attachMovie (đính kèm Movie) để tạo một instance mới của hộp kiểm check box trên Stage chẳng hạn như movie plays (trình diễn movie). Bởi vì symbol CheckBox được đăng ký thành lớp ActionScript là CheckBox, instance động mới sẽ tự động tiếp nhận lớp đó.

```
// createCheckBox là một hàm trợ giúp // tích cực trong việc tạo các CheckBoxesfunction createCheckBox(name, depth) { attachMovie("CheckBox", name, depth);}createCheckBox("myCheckBox3", 100);myCheckBox3.setValue(false);myCheckBox3.setLabel("new label for #3");
```

See also

onClipEvent

Availability

Flash Player 5.

Usage

```
onClipEvent(movieEvent){ statement(s);}
```

Parameters

Một movieEvent (sự kiện movie) là một hành động nhanh được gọi là một sự kiện. Khi sự kiện được đặt vào, các câu lệnh, phát biểu theo sau nó bên trong các dấu ngoặc xoắn sẽ được thực thi. Bất kỳ giá trị nào sau đây đều có thể được chỉ định làm tham số cho movieEvent:

- ♦ load (nạp) Hành vi được khởi xướng ngay khi movie clip được thể hiện và xuất hiện trong Timeline.
- ♦ unload Hành vi được khởi xướng trong frame (khung) đầu tiên sau khi movie clip bị gỡ bỏ khỏi Timeline. Các action (hành vi) kết hợp với sự kiện movie clip Unload được xử lý trước khi bất kỳ hành vi nào được gắn vào frame bị ảnh hưởng (processed before any actions are attached to the affected frame).
- ♦ enterFrame (nhập frame) Hành vi được khởi sự liên tục theo tốc độ frame của movie. Các hành

viết kết hợp với sự kiện clip `enterFrame` được xử lý trước khi bất kỳ action cho frame nào được đính kèm vào các affected frame.

- ♦ `mouseMove` (di chuyển chuột) Hành vi được bắt đầu mỗi lần con trỏ chuột được di chuyển. Sử dụng các thuộc tính `_xmouse` (hoành độ vị trí chuột) và `_ymouse` (tung độ vị trí chuột) để xác định vị trí con chuột hiện tại.
- ♦ `mouseDown` (nhấn chuột) Hành vi được khởi xướng khi phím chuột trái được nhấn.
- ♦ `mouseUp` (nhả chuột) Hành vi được khởi xướng khi phím trái chuột được nhả ra.
- ♦ `keyDown` (nhấn phím) Hành vi được khởi xướng khi một phím được nhấn. Sử dụng method `Key.getCode` để gọi (truy lục) thông tin về phím đã được nhấn cuối cùng.
- ♦ `keyUp` (nhả phím) Hành vi được khởi xướng khi một phím được nhả ra. Sử dụng method `Key.getCode` để truy lục thông tin về phím cuối cùng đã được nhấn.
- ♦ `data` (dữ liệu) Hành vi được khởi xướng khi dữ liệu được tiếp nhận trong một hành vi `loadVariables` (nạp các biến) và `loadMovie` (nạp movie). Khi đã xác định với một hành vi `loadVariables`, sự kiện `data` xảy ra chỉ một lần, khi biến cuối cùng được nạp. Khi chỉ định với một hành vi `loadMovie`, sự kiện `data` xảy ra lặp đi lặp lại nhiều lần, như mỗi phần của dữ liệu được truy lục.

`statement(s)` Các lệnh để thực thi khi sự kiện `mouseEvent` xảy ra.

Description

Event handler (bộ quản lý sự kiện); các hành vi xảy ra rất nhanh định nghĩa cho một instance xác định của một movie clip.

Example

Lệnh sau bao gồm script từ một file bên ngoài khi movie được xuất ra; các hành vi bên trong script kèm theo được chạy khi movie clip mà chúng được đính kèm được nạp:

```
onClipEvent(load) { #include "myScript.as" }
```

Ví dụ sau đây sử dụng `onClipEvent` với sự kiện movie `keyDown`. Sự kiện movie `keyDown` thường được sử dụng trong sự kết hợp với một hoặc nhiều method và thuộc tính của đối tượng `Key` (phím). Đoạn script bên dưới sử dụng method `Key.getCode` để tìm ra phím nào mà người dùng vừa nhấn; nếu phím vừa nhấn khớp với thuộc tính `Key.RIGHT`, movie được gửi tới frame tiếp theo; nếu phím đã nhấn khớp với thuộc tính `Key.LEFT`, movie được chuyển tới frame trước đó (previous frame).

```
onClipEvent(keyDown) { if (Key.getCode() == Key.RIGHT) {  
    _parent.nextFrame(); } else if (Key.getCode() == Key.LEFT){  
    _parent.prevFrame(); } }
```

Ví dụ sau sử dụng `onClipEvent` với sự kiện movie `mouseMove`. Các thuộc tính `_xmouse` và `_ymouse` theo dõi vị trí của con chuột mỗi lần chuột di chuyển.

```
onClipEvent(mouseMove) { stageX=_root.xmouse; stageY=_root.ymouse; }
```

See also

[Key \(object\)](#), [MovieClip._xmouse](#), [MovieClip._ymouse](#), [on](#)

_parent

Availability

Flash Player 4.

Usage

```
_parent.property_parent._parent.property
```

Description

Property (thuộc tính); xác định hoặc trả về một tham chiếu tới movie clip hoặc đối tượng mà chứa movie clip hoặc đối tượng hiện tại (current). Đối tượng hiện tại là đối tượng đang chứa mã ActionScript tham chiếu `_parent`. Sử dụng `_parent` để chỉ định một đường dẫn tương đối tới các movie clip hoặc các đối tượng cấp cao hơn (ở trên, `parent` có nghĩa là cha mẹ) movie clip hoặc đối tượng hiện tại.

Example

Trong ví dụ sau, movie clip `desk` (cái bàn học sinh) là một "đứa con" (child) của movie clip `classroom` (phòng lớp học). Khi đoạn script ở dưới thực thi bên trong movie clip `desk`, đầu chạy playhead sẽ nhảy tới Frame 10 trong Timeline của movie clip `classroom`.

```
_parent.gotoAndStop(10);
```

See also

[_root](#), [targetPath](#)

on

Availability

Flash 2. Không phải tất cả các sự kiện đều được hỗ trợ trong Flash 2.

Usage

```
on(mouseEvent) { statement(s); }
```

Parameters

`statement(s)` Các lệnh để thực thi khi `mouseEvent` xảy ra.

Một `MouseEvent` là một hành động diễn ra nhanh được gọi là một "sự kiện" (event). Khi sự kiện xảy ra, các lệnh theo sau nó bên trong dấu ngoặc xoắn được thực thi. Các giá trị bất kỳ trong những giá trị sau có thể được chỉ định cho tham số `MouseEvent`:

- ◆ `press` Phím chuột được nhấn trong khi con trỏ ngang qua nút.
- ◆ `release` Phím chuột được nhả ra trong khi con trỏ ngang qua nút.
- ◆ `releaseOutside` Phím chuột được nhả ra trong khi con trỏ ra bên ngoài nút sau khi nút được nhấn trong khi con trỏ ở giữa nút.
- ◆ `rollOut` Con trỏ cuộn ngoài vùng nút.
- ◆ `rollOver` Con trỏ chuột cuộn qua nút.
- ◆ `dragOut` Trong khi con trỏ qua nút, phím chuột được nhấn và sau đó cuộn ra ngoài vùng nút.
- ◆ `dragOver` Trong khi con trỏ qua nút, phím chuột vừa được nhấn và sau đó đã cuộn ra ngoài của nút và sau đó đã cuộn trở lại ngang qua nút.
- ◆ `keyPress` ("key") Phím `key` xác định đã được nhấn. Phần `key` của tham số được xác định bằng cách sử dụng bất kỳ mã phím nào được liệt kê trong Phụ lục B, [Keyboard Keys and Key Code Values](#) (Các phím trên bàn phím và các giá trị mã phím) của Using Flash hoặc bất kỳ hằng phím nào được liệt kê trong [Property summary for the Key object](#) (Thuộc tính tổng quan cho đối tượng Phím).

Description

Event handler (bộ quản lý sự kiện); chỉ định sự kiện chuột, hoặc phím nhấn (keypress) mà khởi sự một hành vi.

Example

Trong đoạn script sau, hành vi `startDrag` thực thi khi con chuột được nhấn và script điều kiện được thực thi khi con chuột được nhả ra và đối tượng thả (dropped).

```
on(press) { startDrag("rabbit"); } on(release) { trace(_root.rabbit._y); } trace(
```

print

Availability

Flash Player 4.20.

Usage

```
print (level)print (level, "Bounding box")print ("target")print ("target",  
"Bounding box")printAsBitmap (level)printAsBitmap (level, "Bounding  
box")printAsBitmap ("target")printAsBitmap ("target", "Bounding box")
```

Parameters

`print` (in) Trong chế độ bình thường (normal mode) trong panel Actions, chọn `As vectors` (như là vectơ) để in các frame không chứa các hình ảnh bitmap hoặc sử dụng tính trong suốt transparency (alpha) hoặc các hiệu ứng màu sắc (color effects); chọn `As bitmap` (dạng bitmap) để in các frame chứa các hình ảnh nhị phân bitmap, tính trong suốt transparency hoặc các hiệu ứng màu sắc. Nếu bạn lựa chọn tham số in là `As bitmap`, cú pháp `ASBitmap` được bổ sung vào hành vi `print` trong panel Actions.

`level` (mức, cấp) Cấp trong Flash Player được in. Trong panel Actions trong chế độ normal mode, nếu bạn chọn một cấp level, hành vi `print` chuyển đổi thành `printNum` hoặc `printASBitmapNum`; trong chế độ chuyên môn (expert mode), bạn phải chỉ định hoặc `printNum` hoặc `printASBitmapNum`. Theo mặc định, tất cả các frame trong level được in. Nếu bạn muốn in các frame xác định trong level, hãy gán một nhãn frame `#p` vào các frame đó.

`target` (đích dẫn) Tên thể hiện (instance name) của một movie clip được in. Theo mặc định, tất cả các frame trong instance đích sẽ được in. Nếu bạn muốn in các frame xác định trong movie clip, gán một nhãn frame `#p` vào những frame đó.

`Bounding box` (hộp giới hạn) Một sửa đổi thiết đặt vùng in của movie. Tham số này là tùy chọn. Bạn có thể chọn lựa một trong các loại sau:

- ♦ `bmovie` Chỉ rõ khung giới hạn của một frame xác định trong một movie như vùng in cho tất cả những frame có thể in được trong movie. Gán một nhãn frame `#b` vào frame mà bạn muốn dùng khung giới hạn của nó như vùng in.
- ♦ `bmax` Chọn lựa một bản tổng hợp của tất cả các khung giới hạn của tất cả các frame có thể in được như vùng in. Chỉ định tham số `bmax` khi các frame có thể in được trong movie của bạn có kích thước khác nhau.
- ♦ `bframe` Chọn lựa khi khung giới hạn của mỗi frame có thể in được (printable frame) được dùng như vùng in cho frame đó. Điều này thay đổi vùng in với mỗi frame và điều chỉnh kích thước (thu, phóng tỷ lệ) các đối tượng cho vừa với vùng in. Sử dụng `bframe` nếu bạn có các đối tượng có kích thước khác nhau trong mỗi frame và muốn mỗi đối tượng lấp đầy trang in.

Returns

None.

Description

Action (hành vi); in movie clip đích `target` theo các đường biên chỉ định trong tham số (`bmovie`, `bmax`, hoặc `bframe`). Nếu bạn muốn in các frame xác định trong movie đích, hãy đính kèm một nhãn frame `#p` vào những frame đó. Mặc dù hành vi `print` cho kết quả in có chất lượng cao hơn hành vi `printASBitmap`, nhưng nó không thể được dùng để in các movie sử dụng thuộc tính trong suốt alpha hoặc các hiệu ứng màu sắc xác định.

Nếu bạn không chỉ định một tham số in đường biên ranh giới (a print boundary parameter), thì theo mặc định vùng in được quyết định bởi kích cỡ Stage của movie đã nạp. Movie không thừa hưởng kích cỡ Stage của movie chính. Bạn có thể điều khiển vùng in bằng cách chỉ định các tham số `bmovie`, `bmax`, hoặc `bframe` parameters.

Tất cả các phần tử có khả năng in trong một movie phải được nạp đầy đủ trước khi quá trình in có thể bắt đầu.

Tính năng in Flash Player hỗ trợ các máy in PostScript (một máy in, chủ yếu là loại laser, trong đó có lắp mạch xử lý giải mã và biên dịch các lệnh in đã được viết thành câu theo PostScript-một loại ngôn ngữ mô tả trang phức tạp được dùng cho việc in chất lượng cao bằng máy laser và các thiết bị in độ phân giải cao khác) và non-PostScript. Các máy in Non-PostScript chuyển đổi các hình ảnh vector thành bitmap.

Example

Ví dụ sau đây sẽ in tất cả các frame có khả năng in trong movie clip `myMovie` với vùng in định nghĩa bởi khung giới hạn của frame với nhãn frame #b gắn kèm:

```
print("myMovie", "bmovie");
```

Ví dụ sau đây sẽ in tất cả các frame có thể in được trong `myMovie` với vùng in định nghĩa bởi khung giới hạn của mỗi frame:

```
print("myMovie", "bframe");
```

See also

[printNum](#), [printAsBitmap](#), [printAsBitmapNum](#)

random {ngẫu nhiên}

Availability

Flash Player 4. Hàm này bị phân đổi trong Flash 5; được thay thế bởi việc sử dụng method `Math.random`.

Usage

```
random(value)
```

Parameters

`value` Một số nguyên.

Returns

Một số nguyên.

Description

Function (hàm); trả về một số nguyên ngẫu nhiên giữa 0 một số nhỏ hơn số nguyên được chỉ định trong tham số.

Example

Cách sử dụng sau đây của hàm `random` trả về một trong các giá trị 0, 1, 2, 3, hoặc 4:

```
random(5);
```

See also

[Math.random](#)

removeMovieClip

Availability

Flash Player 4.

Usage

```
removeMovieClip(target)
```

Parameters

`target` Đường dẫn đích của một instance movie clip đã tạo với `duplicateMovieClip`, hoặc tên thể hiện (instance name) của movie clip đã tạo với các method `attachMovie` hoặc `duplicateMovieClip` của đối tượng `MovieClip`.

Returns

None.

Description

Action (hành vi); xoá một movie clip instance mà đã được tạo với các method `attachMovie` (đính kèm movie) hoặc `duplicateMovieClip` (nhân đôi movie clip) của đối tượng `MovieClip`, hoặc với hành vi `duplicateMovieClip`.

See also

[duplicateMovieClip](#), [MovieClip.duplicateMovieClip](#), [MovieClip.attachMovie](#), [MovieClip.removeMovieClip](#)

_root

Availability

Flash Player 4.

Usage

```
_root.movieClip_root.action_root.property
```

Parameters

`movieClip` Tên minh họa (instance name) của một movie clip.

`action` Một hành vi hoặc method.

`property` Một thuộc tính của đối tượng MovieClip.

Description

Property (thuộc tính); xác định hoặc trả về một tham chiếu tới Timeline của movie gốc (root movie Timeline). Nếu một movie có nhiều level (cấp), root movie Timeline ở trên level đang chứa đoạn script hiện tại đang thực thi. Ví dụ, nếu một script trong level 1 ước lượng `_root`, `_level1` sẽ được trả về.

Việc chỉ định `_root` giống như khi sử dụng ký hiệu vạch xiên (/) để xác định một đường dẫn tuyệt đối bên trong level hiện tại.

Example

Ví dụ sau dừng Timeline của level chứa đoạn script hiện đang thực thi.

```
_root.stop();
```

Ví dụ sau gửi Timeline trong level hiện tại tới frame 3:

```
_root.gotoAndStop(3);
```

See also

[_parent](#), [targetPath](#)

scroll

Availability

Flash Player 4.

Usage

```
textFieldVariableName.scroll = x
```

Description

Property (thuộc tính); một thuộc tính bị phản đối để điều khiển sự hiển thị thông tin trong một trường chữ (text field) kết hợp với một biến. Thuộc tính `scroll` định nghĩa nơi trường chữ bắt đầu hiển thị (trình bày) nội dung; sau khi bạn thiết đặt nó, trình Flash Player cập nhật nó khi người dùng

cuộn lên, cuộn xuống trong suốt trường chữ. Thuộc tính cuộn `scroll` rất hữu ích cho người dùng điều khiển một đoạn văn cụ thể trong một bài văn dài, hoặc việc tạo khả năng cuộn trong những trường chữ. Thuộc tính này có thể được khôi phục và thay đổi.

Example

Đoạn mã sau được gắn vào một nút Up (nút lên) để cuộn trường chữ `myText`:

```
on (release) {  
    myText.scroll = myText.scroll + 1;  
}
```

See also

[TextField.maxscroll](#), [TextField.scroll](#)

Selection (object) {Sự lựa chọn (đối tượng)}

Đối tượng vùng chọn Selection cho phép bạn thiết đặt và điều khiển trong trường chữ mà con trỏ được định vị trong một Flash movie. Trường chữ (text field) được gọi là có "tiêu điểm" (focus) là trường mà con trỏ hiện đang định vị. Các chỉ mục nối của vùng chọn (selection-span indexes) lấy cơ sở từ 0 (cho ví dụ, vị trí đầu tiên là 0, vị trí thứ hai là 1, và cứ như thế).

Không có constructor method nào cho đối tượng Selection, vì thế mỗi lần chỉ có một trường hiện tại được tập trung.

Bảng tóm lược các method cho đối tượng vùng chọn Selection

Method	Description
Selection.addListener	Đăng ký một đối tượng để nhận thông báo khi method <code>onSetFocus</code> được viện dẫn.
Selection.getBeginIndex	Trả về chỉ mục (index) tại mở đầu của chiều dài vùng chọn (selection span). Trả về -1 nếu không chỉ mục hoặc trường nào hiện tại đang được lựa chọn.
Selection.getCaretIndex	Trả về vị trí dấu sót (để đánh dấu chỗ phải thêm chữ còn sót) hiện tại trong chiều dài vùng chọn hiện đang đặt tiêu điểm (đang xét). Trả về -1 nếu không có vị trí dấu sót hoặc chiều dài vùng chọn nào hiện đang được đặt trọng tâm (if there is no caret position or currently focused selection span).
Selection.getEndIndex	Trả về chỉ mục tại cuối của selection span. Trả về -1 nếu không có chỉ mục nào hoặc trường nào hiện đang được lựa chọn.
Selection.getFocus	Trả về tên biến cho trường text hiện tại đang được chú ý. Trả về <code>null</code> nếu hiện tại không có trường text nào được đặt trọng tâm

	(được tập trung xét).
Selection.removeListener	Gỡ bỏ một đối tượng mà đã được đăng ký với <code>addListener</code> .
Selection.setFocus	Tập trung vào trường text đã kết hợp với biến đã chỉ định trong tham số.
Selection.setSelection	Thiết đặt các chỉ mục mở đầu và kết thúc của chiều dài vùng chọn (selection span).

Bảng tóm lược các listener ("người nghe", tức là chịu sự tác động, chịu ảnh hưởng) cho đối tượng Selection

Method	Description
Selection.onSetFocus	Thông báo khi tiêu điểm nhập vào thay đổi (Notified when the input focus changes).

Sound (object) {Âm thanh (đối tượng)}

Đối tượng Sound cho phép bạn điều khiển âm thanh trong một movie. Bạn có thể thêm các âm thanh vào một movie clip từ thư viện Library trong khi phim movie đang trình diễn và điều khiển những âm thanh đó. Nếu bạn không chỉ định một đường dẫn `target` khi bạn tạo một đối tượng âm thanh mới (new Sound object), bạn có thể sử dụng các method (phương pháp, cách thức) để điều khiển âm thanh cho toàn bộ phim movie. Bạn phải sử dụng constructor (trình xây dựng) `new Sound` để tạo một thể hiện (instance) của đối tượng Sound trước khi gọi các method của đối tượng Sound.

Đối tượng Sound được hỗ trợ trong Flash Player 5 và Flash Player 6.

Bảng tóm lược các method dành cho đối tượng Sound object

Method	Description
Sound.attachSound	Đính kèm âm thanh đã chỉ định trong tham số.
Sound.getBytesLoaded	Trả về số byte đã nạp của âm thanh đã chỉ định.
Sound.getBytesTotal	Trả về kích cỡ của âm thanh theo bytes.
Sound.getPan	Trả về giá trị của lệnh gọi <code>setPan</code> (điều khiển loa) liền trước.

Sound.getTransform	Trả về giá trị của lệnh gọi <code>setTransform</code> (điều khiển sự biến đổi) liền trước.
Sound.getVolume	Trả về giá trị của lệnh gọi <code>setVolume</code> (thiết đặt âm lượng) trước đó.
Sound.loadSound	Nạp một file MP3 vào trong Flash Player.
Sound.setPan	Thiết đặt sự cân bằng giữa hai loa trái/phải của âm thanh.
Sound.setTransform	Thiết đặt số lượng của mỗi kênh, trái và phải, để trình diễn trong mỗi loa.
Sound.setVolume	Thiết đặt mức âm lượng cho một âm thanh.
Sound.start	Bắt đầu trình diễn một âm thanh từ đầu hoặc, do tùy chọn, từ một điểm trống (offset point) thiết đặt trong tham số.
Sound.stop	Dừng âm thanh đã chỉ định hoặc tất cả các âm thanh hiện đang được trình diễn.

Bảng tóm lược các thuộc tính của đối tượng Sound object

Method	Description
Sound.duration	Độ dài của một âm thanh theo mili-giây (một phần nghìn giây).
Sound.position	Số mili-giây âm thanh đã được trình diễn.

Event handler summary for the Sound object

Method	Description
Sound.onLoad	Được viện dẫn khi một âm thanh được nạp.
Sound.onSoundComplete	Được viện dẫn khi một âm thanh ngừng trình diễn.

Constructor dành cho đối tượng Sound object

Availability

Flash Player 5.

Usage

```
new Sound([target])
```

Parameters

target Thể hiện của movie clip trên đó đối tượng âm thanh Sound hoạt động. Tham số này là tùy chọn.

Returns

Nothing.

Description

Constructor (trình xây dựng, biên tập); tạo một đối tượng âm thanh mới (new Sound object) dành cho một movie clip xác định. Nếu bạn không chỉ định một đường dẫn xác định, đối tượng Sound sẽ điều khiển tất cả các âm thanh trong movie.

Example

Ví dụ sau tạo một thể hiện (instance) mới của đối tượng Sound object gọi là `GlobalSound`. Dòng thứ hai gọi method `setVolume` và điều chỉnh âm lượng trên tất cả các âm thanh trong movie thành 50%.

```
globalsound = new Sound();  
globalsound.setVolume(50);
```

Ví dụ sau tạo một thể hiện mới của đối tượng Sound, chuyển nó tới movie clip đích `myMovie`, và gọi method `start`, nó sẽ bắt đầu chạy bất kỳ âm thanh nào có trong `myMovie`.

```
moviesound = new Sound(myMovie);  
moviesound.start();
```

Stage (object) {Sân khấu (đối tượng)}

Đối tượng Stage là một đối tượng cấp cao nhất mà bạn có thể truy cập mà không cần sử dụng một constructor.

Sử dụng các method và thuộc tính của đối tượng này để truy cập (access) và thao tác thông tin về các đường biên ranh giới của một movie Flash.

Đối tượng Stage có sẵn trong Flash Player 6 và các phiên bản về sau.

Bảng tóm lược các method dành cho đối tượng Stage

Method	Description
Stage.addListener	Thêm một listener object (ta tạm hiểu là đối tượng chịu sự tác động, "người nghe") vào đối tượng Stage.
Stage.removeListener	Gỡ bỏ một listener object từ đối tượng Stage.

Bảng các thuộc tính của đối tượng Stage object

Method	Description
Stage.align	Sự canh chỉnh vị trí của Flash movie trong trình duyệt.
Stage.height	Chiều cao của Stage, theo pixel.
Stage.width	Chiều rộng của Stage, theo pixel.
Stage.scaleMode	Tỷ lệ thu phóng hiện thời của Flash movie.

Bộ quản lý sự kiện của đối tượng Stage

Method	Description
Stage.onResize	Cho biết movie đã được điều chỉnh lại kích thước.

startDrag

Availability

Flash Player 4.

Usage

```
startDrag(target,[lock ,left , top , right, bottom])
```

Parameters

target Đường dẫn tới movie clip để kéo (drag: di chuyển con trỏ chuột trong khi vẫn nhấn và giữ phím trên chuột).

`lock` Một giá trị logic Boolean xác định xem movie clip có khả năng kéo (draggable movie clip) bị khoá đối với điểm giữa của vị trí con trỏ (`true`), hay bị khoá với điểm mà ở đó người dùng click lên trên movie clip lần đầu tiên (`false`). Tham số này là tùy chọn.

`left`, `top`, `right`, `bottom` Các giá trị liên quan đến các tọa độ của movie clip's parent ("mẹ" của movie clip, ta hiểu là instance cấp cao hơn, nơi movie clip được đặt) xác định một hình chữ nhật chứa movie clip (a constraint rectangle for the movie clip). Những tham số này là tùy chọn.

Returns

Nothing.

Description

Action (hành vi); tạo một `target` movie clip có thể kéo được khi movie đang trình diễn. Ở một thời điểm chỉ một movie clip có thể kéo được. Ngay khi một thao tác `startDrag` (bắt đầu kéo) được thực thi, movie clip giữ nguyên khả năng kéo được (remains draggable) cho đến khi dứt khoát được dừng lại bởi một hành vi `stopDrag`, hoặc cho đến khi hành vi `startDrag` đối với một movie clip khác được gọi.

Example

Để tạo một movie clip mà người dùng có thể định vị trong bất kỳ vị trí nào, kèm theo các hành vi `startDrag` và `stopDrag` vào một nút bên trong movie clip.

```
on(press) {
    startDrag(this,true);
}
on(release) {
    stopDrag();
}
```

String (function)

Availability

Flash Player 4.

Usage

`String(expression)`

Parameters

`expression` Một biểu thức để chuyển đổi thành một chuỗi.

Returns

Nothing.

Description

Function (hàm); trả về một chuỗi đại diện của tham số đã được chỉ định như sau đây:

Nếu `expression` là một giá trị logic Boolean, chuỗi trả về là `true` hoặc `false`.

Nếu `expression` là một số, chuỗi trả về là một text (chỉ bao gồm các ký tự ASCII) đại diện của số.

Nếu `expression` là một chuỗi, chuỗi trả về là biểu thức `expression`.

Nếu `expression` là một đối tượng, giá trị trả về là một chuỗi đại diện của đối tượng đã tạo bởi việc gọi thuộc tính chuỗi (string property) cho đối tượng, hoặc bởi việc gọi `Object.toString` nếu không có thuộc tính nào tồn tại.

Nếu `expression` là một movie clip, giá trị trả về là đường dẫn của movie clip trong ký hiệu vạch chéo (/).

Nếu `expression` là không xác định (undefined), giá trị trả về là một chuỗi rỗng ("").

See also

[Number.toString](#), [Object.toString](#), [String\(object\)](#), [" " \(string delimiter\)](#)

System.capabilities (object)

Bạn có thể dùng đối tượng `System.capabilities` (năng lực hệ thống) để xác định những năng lực của hệ thống và player đang chạy một Flash movie. Điều này cho phép bạn biến đổi nội dung cho nhiều định dạng. Cho ví dụ, màn hình của một điện thoại dạng ô (đen và trắng, 100 điểm ảnh hình vuông) khác khác với màn hình màu máy vi tính 1000 điểm ảnh hình vuông. Để chuẩn bị nội dung thích hợp với nhiều người dùng là điều có thể thực hiện được, bạn có thể dùng đối tượng `Capabilities` để xác định loại thiết bị (device) mà một người dùng có. Sau đó bạn có thể hoặc chỉ định server (máy chủ) để gửi các file SWFs khác nhau dựa trên cơ sở năng lực của thiết bị, hoặc ảnh hưởng đến Flash movie để biến đổi bản trình bày của nó dựa trên các năng lực của thiết bị.

Bạn có thể gửi thông tin năng lực (capabilities information) sử dụng một method GET hoặc POST HTTP. Sau đây là một ví dụ của một chuỗi máy chủ (server string) cho một thiết bị mà không hỗ trợ MP3 và có một màn hình 400 x 200 pixel, 8 x 4 centimet:

```
"A=t&MP3=f&AE=gsm&VE=h11&ACC=f&V=WIN%206%2C0%2C0%2C129&M=Macromedia%WINDOWS&R=400x200&DP=72&COL=color&AR=1.0&OS=WINDOWS%2000&L=en-US"
```

Đối tượng `Capabilities` có sẵn trong Flash Player 6.

Bạn phải truy cập tất cả các thuộc tính của đối tượng `Capabilities` cho đến đối tượng `System.capabilities`.

Bảng tóm lược các thuộc tính của đối tượng Capabilities

Property	Description
System.capabilities.hasAudioEncoder	Cho biết trình mã hoá hỗ trợ âm thanh.
System.capabilities.hasAccessibility	Cho biết xem thiết bị có phù hợp với các tiêu chuẩn có khả năng truy cập không.
System.capabilities.hasAudio	Cho biết thiết bị (device) có khả năng âm thanh không.
System.capabilities.hasMP3	Cho biết xem thiết bị có bộ giải mã MP3 không.
System.capabilities.language	Cho biết ngôn ngữ mà Flash Player hỗ trợ.
System.capabilities.manufacturer	Cho biết người sản xuất của Flash Player.
System.capabilities.os	Cho biết hệ thống hoạt động chứa Flash Player.
System.capabilities.pixelAspectRatio	Cho biết số pixel về chiều rộng của màn hình hiển thị so với chiều cao của nó.
System.capabilities.screenColor	Cho biết xem màn hình là màu, đen trắng, hay thang độ xám (grayscale).
System.capabilities.screenDPI	Cho biết các điểm màn hình theo inch.
System.capabilities.screenResolution.x	Cho biết kích thước chiều ngang của màn hình.
System.capabilities.screenResolution.y	Cho biết kích thước chiều dọc của màn hình.
System.capabilities.version	Cho biết phiên bản thấp nhất hỗ trợ Flash Player.
System.capabilities.hasVideoEncoder	Cho biết có hỗ trợ trình mã hoá video.

targetPath

Availability

Flash Player 5.

Usage

```
targetpath(movieClipObject)
```

Parameters

`movieClipObject` Tham chiếu (ví dụ, `_root` hoặc `_parent` : cả hai từ này đều chỉ gốc rễ, nguồn) tới movie clip cho đường dẫn đang được truy gọi.

Returns

Nothing.

Description

Hàm số (Function); trả lại một chuỗi chứa đựng đường dẫn đích (target path) của `movieClipObject`. Đường dẫn đích được trả về tại ký hiệu dấu chấm (dot). Để gọi đường dẫn đích tại ký hiệu gạch xiên (/), sử dụng thuộc tính `_target`.

Example

Ví dụ sau đây hiển thị đường dẫn đích của một movie clip ngay khi nó được tải.

```
onClipEvent(load){    trace(targetPath(this));}
```

tellTarget

Availability

Flash Player 3. (Không dùng trong Flash 5; việc sử dụng ký hiệu chấm (dot) và hành vi `with` được khuyến cáo.)

Usage

```
tellTarget("target") { statement(s);}
```

Parameters

`target` Một chuỗi mà chỉ rõ đường dẫn đích của Timeline sẽ được điều khiển.

`statement(s)` Các lệnh để thực thi nếu điều kiện xác định là `true`.

Returns

Nothing.

Description

Hành vi (Action); áp dụng các chỉ dẫn chỉ định trong tham số `statements` tới Timeline được chỉ định trong tham số `target`. Hành vi `tellTarget` hữu dụng cho các điều khiển điều hướng (navigation controls). Gán `tellTarget` cho các nút button để dừng hoặc khởi động các đoạn phim (movie clip) ở một nơi nào khác ở trên Stage. Bạn cũng có thể tạo các đoạn phim nhảy tới một

frame riêng biệt trong đoạn clip đó. Ví dụ, bạn có thể gán `tellTarget` tới các nút để dừng hoặc khởi động các movie clip trên Stage hoặc nhắc các movie clip nhảy tới một frame riêng biệt.

Trong Flash 5, bạn có thể dùng ký hiệu dấu chấm (dot) để thay thế cho hành vi `tellTarget`. Bạn có thể dùng hành vi (action) `with` để phát hành nhiều action tới cùng một Timeline đó. Bạn có thể dùng hành vi `with` để nhắm tới bất kỳ đối tượng object nào, trong khi hành vi `tellTarget` chỉ có thể nhắm đích tới các đoạn phim movie clip.

Example

Câu lệnh `tellTarget` này điều khiển movie clip minh họa (movie clip instance) `ball` trên Timeline chính. Frame 1 của `ball` instance (movie clip thể hiện quả bóng) để trống và có một action là `stop` vì thế nó không thể thấy được trên Stage. Khi nút với action sau đây được click, `tellTarget` ra lệnh cho playhead trong `ball` để nhảy tới frame 2 nơi đoạn hoạt cảnh bắt đầu.

```
on(release) { tellTarget("ball") { gotoAndPlay(2); } }
```

Ví dụ sau đây dùng ký hiệu dấu chấm (dot) để cho kết quả tương tự.

```
on(release) { ball.gotoAndPlay(2); }
```

Nếu bạn cần phát hành nhiều mệnh lệnh cho instance `ball`, bạn có thể dùng hành vi `with`, như trong phát biểu sau.

```
on(release) { with(ball) { gotoAndPlay(2); _alpha = 15; } } // _scale
```

See also

[with](#)

TextField (object)

Tất cả các trường text động (dynamic text) và các trường text nhập dữ liệu vào trong một Flash movie là các trường hợp riêng của đối tượng `TextField`. Bạn có thể đặt cho một trường text một tên minh họa trong Property inspector (thanh tra thuộc tính) và dùng các method (phương pháp) và thuộc tính của đối tượng `TextField` để chế tác nó với ActionScript. Những tên thể hiện `TextField` được trình bày trong hộp thoại Movie Explorer và trong hộp thoại Insert Target Path trong bảng Actions.

Đối tượng `TextField` thừa kế từ đối tượng `Object`.

Để tạo ra một trường text động, bạn có thể dùng method `MovieClip.createTextField`.

Đối tượng `TextField` được hỗ trợ bởi Flash Player 6 và các phiên bản sau này của Flash Player.

Phương pháp tóm lược cho đối tượng `TextField`

Method	Description
TextField.addListener	Đăng ký một đối tượng để tiếp nhận thông báo khi các sự kiện <code>onChanged</code> và <code>onScroller</code> được viện dẫn.
TextField.getDepth	Trả lại độ dày (depth) của một trường text.
TextField.getNewTextFormat	Mang định dạng text mặc định được gán cho text mới được nhập vào.
TextField.removeListener	Gỡ bỏ một đối tượng người nghe (listener objec).
TextField.removeTextField	Gỡ bỏ một trường text mà được tạo với <code>MovieClip.createTextField</code> .
TextField.setNewTextFormat	Thiết đặt một đối tượng định dạng text (chữ) cho text mà được nhập vào bởi một người dùng hoặc bởi một method.
TextField.replaceSel	Thay thế vùng lựa chọn hiện tại.
TextField.setTextFormat	Thiết đặt định dạng text mặc định gán cho đoạn text mới được chèn vào.

Các thuộc tính tóm lược cho đối tượng TextField object

Property	Description
TextField._alpha	Giá trị trong suốt của một thể hiện (instance) của trường chữ.
TextField.autoSize	Điều khiển sự canh thẳng hàng một cách mặc định và điều chỉnh kích thước của một trường chữ.
TextField.background	Chỉ ra nếu trường text có một nền đủ (background fill).
TextField.backgroundColor	Cho biết màu sắc của nền đầy đủ.
TextField.border	Biểu thị nếu trường text có một đường viền.
TextField.borderColor	Cho biết màu sắc của đường viền.
TextField.bottomScroll	Dòng ở dưới cùng nhìn thấy được trong trường text.
TextField.embedFonts	Cho biết xem trường text sử dụng các font phác thảo được nhúng vào (embedded font outlines) hay các font ngoại vi (device font).
TextField._highquality	Cho biết chất lượng biến đổi hình vẽ phác đồ hoạ ra hình ảnh ba chiều của movie.

<code>TextField._height</code>	Chiều cao của một instance của trường text theo pixel. Điều này chỉ có tác dụng với khung giới hạn của trường text, nó không ảnh hưởng đến bề dày đường viền hoặc kích thước font chữ.
<code>TextField.hscroll</code>	Cho biết giá trị cuộn sang ngang của một trường text.
<code>TextField.html</code>	Cho biết vị trí cuộn cực đại hiện tại (current maximum scrolling position) của một trường text.
<code>TextField.htmlText</code>	Chứa biểu diễn HTML của các nội dung của một trường text.
<code>TextField.length</code>	Số ký tự trong một trường text.
<code>TextField.maxChars</code>	Số ký tự cực đại mà một trường text có thể chứa.
<code>TextField.maxhscroll</code>	Giá trị cực đại của <code>TextField.hscroll</code> .
<code>TextField.maxscroll</code>	Giá trị cực đại của <code>TextField.scroll</code> .
<code>TextField.multiline</code>	Chỉ ra nếu trường text bao gồm nhiều dòng.
<code>TextField._name</code>	Tên minh họa (instance name) của một thể hiện của trường text.
<code>TextField._parent</code>	Một tham chiếu tới instance mà là "mẹ" (parent: tức là cấp cao hơn) của instance này; hoặc là nút Button hoặc MovieClip.
<code>TextField.password</code>	Biểu thị nếu một trường text giấu các ký tự nhập vào.
<code>TextField._quality</code>	Biểu thị chất lượng biến đổi hình phác thảo thành hình ba chiều (rendering) của một movie.
<code>TextField.restrict</code>	Sự thiết đặt các ký tự mà một người dùng có thể nhập vào trong một trường text.
<code>TextField._rotation</code>	Mức độ xoay vòng của một instance của trường text.
<code>TextField.scroll</code>	Biểu thị vị trí đang cuộn của một trường text.
<code>TextField.selectable</code>	Cho biết xem một trường text có khả năng lựa chọn hay không.
<code>TextField._soundbuftime</code>	Tổng số thời gian một âm thanh phải nhớ đệm trước (prebuffer) trước khi nó stream (hoạt động, chảy).
<code>TextField.tabEnabled</code>	Cho biết xem một movie clip được bao gồm trong chỉ thị tab tự động (Indicates whether a movie clip is included in automatic tab ordering).
<code>TextField.tabIndex</code>	Cho biết lệnh tab (tab order) của một đối tượng.
<code>TextField.text</code>	Text hiện tại trong trường text.
<code>TextField.textColor</code>	Màu sắc của text hiện thời trong trường text.

<code>TextField.textHeight</code>	Chiều cao khung giới hạn của trường text.
<code>TextField.textWidth</code>	Chiều rộng khung giới hạn của trường text.
<code>TextField.type</code>	Chỉ ra xem một trường text là một trường text nhập vào (input text field) hay trường text động (dynamic text field).
<code>TextField._url</code>	Địa chỉ URL của file SWF đã tạo text field instance.
<code>TextField.variable</code>	Tên biến đã kết hợp với trường text.
<code>TextField._visible</code>	Một giá trị logic Boolean quyết định xem một instance của trường text là ẩn hay nhìn thấy được.
<code>TextField._width</code>	Chiều rộng của text field instance theo pixel. Cái này chỉ có tác dụng với khung giới hạn (bounding box) của trường text, nó không ảnh hưởng đến độ dày của đường viền hay kích cỡ font chữ.
<code>TextField.wordWrap</code>	Cho biết xem trường text có khả năng ngắt từ không (word-wraps: chức năng ngắt các từ xuống đầu dòng kế tiếp nếu chúng vượt quá lề phải).
<code>TextField._x</code>	Tọa độ x của một text field instance
<code>TextField._xmouse</code>	Tọa độ x của con trỏ liên quan đến một instance của trường text.
<code>TextField._xscale</code>	Giá trị chỉ định tỷ lệ phần trăm cho việc chia tỷ lệ (scaling) theo chiều ngang của một text field instance.
<code>TextField._y</code>	Tọa độ y của một instance của trường text.
<code>TextField._ymouse</code>	Tọa độ y của con trỏ liên quan đến một text field instance.
<code>TextField._yscale</code>	Giá trị chỉ định tỷ lệ phần trăm cho việc chia tỷ lệ (scaling) theo chiều dọc của một text field instance.

Tổng quan về các bộ quản lý sự kiện (Event handler) cho đối tượng TextField

Method	Description
<code>TextField.onChanged</code>	Được viện dẫn khi trường text được thay đổi.
<code>TextField.onKillFocus</code>	Được viện dẫn khi trường text mất tiêu điểm (loses focus).
<code>TextField.onScroller</code>	Được viện dẫn khi thuộc tính <code>scroll</code> , <code>maxscroll</code> , <code>hscroll</code> , <code>maxhscroll</code> , hoặc <code>bottomscroll</code> của một trường text thay đổi

TextField.onSetFocus	Được viện dẫn khi trường text tiếp nhận tiêu điểm.
--------------------------------------	--

Bảng tóm lược listener cho đối tượng TextField

Method	Description
TextField.onChangeed	Thông báo khi trường text bị thay đổi.
TextField.onScroller	Thông báo khi thuộc tính <code>scroll</code> hoặc <code>maxscroll</code> của một trường text thay đổi.

TextFormat (object) {Định dạng Text (đối tượng)}

Đối tượng TextFormat đại diện cho thông tin về định dạng ký tự.

Bạn phải sử dụng constructor `new TextFormat` để tạo một instance (thể hiện) của đối tượng TextFormat trước khi gọi các method của nó.

Bạn có thể thiết đặt các tham số TextFormat thành `null` để chỉ ra rằng chúng không được định nghĩa. Khi bạn áp dụng một đối tượng TextFormat vào một trường text bằng cách sử dụng method `setTextFormat`, chỉ có những thuộc tính đã định nghĩa là được áp dụng, như trong ví dụ sau đây:

```
myTextFormat = new TextFormat();myTextFormat.bold = true;myTextField.setTextFormat(myTextFormat);
```

Đoạn mã này đầu tiên tạo một đối tượng TextFormat rỗng với tất cả các thuộc tính chưa định nghĩa, sau đó thiết đặt thuộc tính `bold` cho một giá trị đã được định nghĩa.

Mã `myTextField.setTextFormat(myTextFormat)` chỉ thay đổi thuộc tính đậm `bold` property của dạng text mặc định của trường text, bởi vì thuộc tính `bold` chỉ là một thuộc tính đã định nghĩa trong `myTextFormat`. Tất cả các mặt khác của dạng text mặc định của trường text vẫn giữ nguyên không thay đổi.

Khi `getTextFormat` được viện dẫn, đối tượng TextFormat được trả về với tất cả các thuộc tính đã định nghĩa của nó; không có thuộc tính nào là `null`.

Bảng tổng quan các method cho đối tượng TextFormat

Method	Description
TextFormat.getTextExtent	Trả về một đối tượng với hai thuộc tính, chiều rộng <code>width</code> và chiều cao <code>height</code> , cho biết kích thước của text trong một trường text.

bảng tổng quan về các thuộc tính property đối với đối tượng TextFormat

Property	Description
TextFormat.align	Cho biết sự canh thẳng của một đoạn văn.
TextFormat.blockIndent	Cho biết mẫu thụt dòng theo point (1 point = 1/72 inch).
TextFormat.bold	Cho biết text có kiểu chữ đậm hay không.
TextFormat.bullet	Cho biết có hay không text bên trong một danh sách bulleted list (danh sách được đánh dấu ở đầu dòng).
TextFormat.color	Cho biết màu sắc của text
TextFormat.font	Cho biết tên font của text với một định dạng text.
TextFormat.indent	Cho biết sự thụt dòng từ lề trái tới ký tự đầu tiên trong đoạn.
TextFormat.italic	Cho biết text có in nghiêng hay không.
TextFormat.leading	Cho biết khoảng cách giữa các dòng.
TextFormat.leftMargin	Cho biết lề trái của đoạn văn, theo point.
TextFormat.rightMargin	Cho biết lề phải của đoạn văn, theo point.
TextFormat.tabStops	Chỉ định trình tự tab lựa chọn (custom tab) dừng
TextFormat.target	Biểu thị cửa sổ trong một trình duyệt khi một siêu liên kết được hiển thị.
TextFormat.size	Cho biết cỡ point của text
TextFormat.underline	Cho biết một text có được gạch dưới không.
TextFormat.url	Cho biết địa chỉ URL mà text liên kết tới.

Constructor cho đối tượng TextFormat object

Usage

```
new TextFormat([font, [size, [color, [bold, [italic, [underline, [url, [target, [align, [leftMargin, [rightMargin, [indent, [leading]]]]]]]]]]]]])
```

Parameters

`font` Tên của một font cho text dưới dạng một chuỗi.

`size` Một số nguyên cho biết kích cỡ point (đơn vị đo cỡ chữ 1 point = 1/72 inch).

`color` Màu sắc của text sử dụng định dạng text này. Một số mang 3 component 8-bit RGB; ví dụ, 0xFF0000 là đỏ, 0x00FF00 là xanh lá cây.

`bold` Một giá trị logic Boolean cho biết xem text có được in đậm không.

`italic` Một giá trị logic Boolean cho biết text có được in nghiêng không.

`underline` Một giá trị logic Boolean cho biết xem text được gạch dưới.

`url` Địa chỉ URL mà text trong dạng text này siêu liên kết. Nếu `url` là một chuỗi rỗng, text không có một siêu liên kết.

`target` Cửa sổ đích (target window) nơi siêu liên kết được hiển thị. Nếu target window là một chuỗi rỗng, text được hiển thị trong cửa sổ target mặc định `_self`. Nếu thuộc tính `TextFormat.url` được thiết đặt thành chuỗi rỗng hoặc thành giá trị `null`, thuộc tính này có thể lấy (get) nhưng không có ảnh hưởng (has no effect).

`align` Sự canh chỉnh của đoạn văn, biểu thị như một chuỗi. Nếu là "left", đoạn văn được canh lề trái. Nếu là "center", đoạn văn được canh giữa. Nếu là "right", đoạn văn được canh lề phải.

`leftMargin` Cho biết lề trái của đoạn văn, theo point.

`rightMargin` Cho biết lề phải của đoạn văn, theo point.

`indent` Một số nguyên cho biết sự thụt dòng từ lề trái tới ký tự đầu tiên trong đoạn văn.

`leading` Một số cho biết số lượng khoảng trống dọc giữa các dòng.

Description

Constructor; tạo một instance (thể hiện) của đối tượng `TextFormat` với các thuộc tính đã được chỉ định. Sau đó bạn có thể thay đổi các thuộc tính của đối tượng `TextFormat` để thay đổi định dạng của các trường text.

Mỗi tham số có thể được thiết đặt thành giá trị `null` để chỉ ra rằng nó không được xác định. Tất cả các tham số để tùy chọn, tham số nào bị bỏ qua đều được xem là `null`.

Availability

Flash Player 6.

this

Availability

Flash Player 5.

Usage

`this`

Description

Keyword (từ khóa); tham chiếu một instance (thể hiện) của đối tượng hoặc movie clip. Khi script thực thi, `this` tham chiếu movie clip instance chứa script. Khi một method được gọi, `this` chứa một tham chiếu tới đối tượng mà có chứa method đã gọi.

Bên trong một bộ quản lý sự kiện hành vi `on` gắn vào một nút, `this` chuyển đến Timeline mà chứa nút. Bên trong một hành vi quản lý sự kiện `onClipEvent` gắn vào một movie clip, `this` chuyển tới Timeline của chính movie clip

Example

Trong ví dụ sau, từ khoá `this` tham chiếu đối tượng Circle (hình tròn).

```
function Circle(radius) {      this.radius = radius;  this.area = Math.PI *
radius * radius;}
```

Trong câu lệnh sau được gán cho một frame, từ khoá `this` tham chiếu movie clip hiện tại.

```
// thiết đặt thuộc tính trong suốt alpha của movie clip thành 20this._alpha =
20;
```

Trong câu lệnh sau gán cho bộ quản lý sự kiện `onClipEvent`, từ khoá `this` tham chiếu movie clip hiện tại.

```
// khi movie clip nạp, một thao tác startDrag// được bắt đầu với movie clip hiện
thời.onClipEvent (load) {      startDrag (this, true);}
```

See also

[new](#)

toggleHighQuality

Availability

Flash 2.

Usage


```
toggleHighQuality()
```

Parameters

None.

Returns

Nothing.

Description

Action (hành vi); bật và tắt chế độ khử răng cưa trong Flash Player. Chế độ khử răng cưa làm trơn các rìa của đối tượng và làm chậm sự phát lại của movie. Hành vi `toggleHighQuality` ảnh hưởng đến tất cả các movie trong Flash Player.

Example

Đoạn mã sau có thể áp dụng cho một nút mà, khi được click, sẽ bật hoặc tắt chế độ khử răng cưa:

```
on(release) {  
    toggleHighQuality();  
}
```

See also

[_quality](#), [_highquality](#)

trace

Availability

Flash Player 4.

Usage

```
trace(expression)
```

Parameters

`expression` Một biểu thức để ước lượng. Khi một file SWF được mở trong công cụ thiết kế Flash (qua lệnh Test Movie), giá trị của tham số `expression` được trình bày trong cửa sổ Output.

Returns

Nothing.

Description

Action (hành vi); đánh giá biểu thức `expression` và hiển thị kết quả trong cửa sổ Output ở chế độ kiểm tra.

Sử dụng `trace` (ghi lại chuỗi trong đó các phát biểu của chương trình được thực thi và một cách tùy chọn, các giá trị của biến chương trình được thực hiện trong các phát biểu ấy) để ghi lại những lưu ý, chú giải khi lập trình hoặc để hiển thị các thông báo trong cửa sổ Output trong khi đang kiểm tra một movie. Dùng tham số `expression` để kiểm tra nếu một điều kiện tồn tại, hoặc để hiển thị các giá trị trong cửa sổ Output. Hành vi `trace` tương tự như hàm `alert` (cảnh báo) trong JavaScript.

Bạn có thể sử dụng các lệnh hành động Omit Trace (bỏ qua Trace) trong Publish Settings để loại bỏ các hành vi `trace` từ file SWF xuất ra.

Example

Ví dụ này là từ một trò chơi tại đó một thể hiện (instance) của movie clip có thể kéo được (draggable) đặt tên là `rabbi` phải được nhả ra trên một target (đích) xác định. Một câu lệnh có điều kiện đánh giá thuộc tính `_droptarget` và thực thi các hành vi khác nhau tùy thuộc vào nơi `rabbi` được released (giải thoát). Hành vi `trace` được dùng ở cuối của đoạn script để đánh giá vị trí của movie clip `rabbi`, và hiển thị kết quả trong cửa sổ Output. Nếu `rabbi` không chạy như mong đợi (ví dụ, nếu nó canh tới đường dẫn sai), các giá trị gửi tới cửa sổ Output bởi hành vi `trace` sẽ giúp bạn xác định trục trặc trong đoạn script.

```
on(press) {
    rabbi.startDrag();
}

on(release) {
    if(eval(_droptarget) != target) {
        rabbi._x = rabbi._x;
        rabbi._y = rabbi._y;
    } else {
        rabbi._x = rabbi._x;
        rabbi._y = rabbi._y;
        target = "_root.pasture";
    }
    trace("rabbi_y = " + rabbi._y);
    trace("rabbi_x = " + rabbi._x);
    stopDrag();
}
```

typeof

Availability

Flash Player 5.

Usage

```
typeof expression
```

Parameters

`expression` Một chuỗi (string), movie clip, nút (button), đối tượng (object), hoặc hàm (function).

Description

Operator (toán tử); một toán tử đơn nguyên (chỉ có một số hạng) đặt trước một tham số đơn (single parameter). Toán tử `typeof` ra lệnh cho Flash interpreter (người biên dịch, trình diễn Flash) ước lượng biểu thức `expression`; kết quả là một chuỗi xác định xem biểu thức là một chuỗi, movie clip, đối tượng, hay hàm. Bảng sau đây cho biết kết quả của toán tử `typeof` trên mỗi kiểu của biểu thức `expression`:

Parameter	Output
String (chuỗi)	string
MovieClip	movieclip
Button (nút)	object
Text field (trường chữ)	object
Number (số)	number
Boolean (giá trị logic)	boolean
Object (đối tượng)	object
Function (hàm)	function

undefined

Availability

Flash Player 5.

Usage

undefined

Parameters

None.

Returns

Nothing.

Description

Một giá trị đặc biệt, thường được sử dụng để cho biết một biến đã được gán một giá trị. Một sự chuyển đến một giá trị không xác định trả lại giá trị đặc biệt `undefined`. Đoạn mã ActionScript `typeof(undefined)` trả lại chuỗi `"undefined"`. Giá trị duy nhất của kiểu `undefined` là `undefined`.

Khi `undefined` được chuyển thành một chuỗi, nó chuyển đổi thành chuỗi rỗng (it converts to the empty string).

Giá trị `undefined` tương tự như giá trị đặc biệt `null`. Trong thực tế, khi `null` và `undefined` được so sánh với dấu bằng thức (equality operator), chúng ngang nhau.

Example

Trong ví dụ sau, biến `x` không được khai báo và vì thế có giá trị không xác định `undefined`. Trong phần đầu tiên của đoạn mã, toán tử bằng thức (`==`) so sánh giá trị của `x` với giá trị `undefined` và kết quả thích hợp được gửi tới cửa sổ xuất ra Output window. Trong phần thứ hai của đoạn mã, toán tử bằng thức so sánh các giá trị `null` và `undefined`.

```
// x không được khai báo
trace ("The value of x is " + x);
if (x == undefined) {
  trace ("x is undefined");
} else {
  trace ("x is not undefined");
}
trace ("typeof (x) is " + typeof (x));
if (null == undefined) {
  trace ("null and undefined are equal");
} else {
  trace ("null and undefined are not equal");
}
```

Kết quả như sau được hiển thị trong cửa sổ Output window:

```
The value of x is x is undefined
typeof (x) is undefined
null and undefined are equal
```

Chú ý: Trong ECMA-262 specification (đặc điểm kỹ thuật), `undefined` chuyển thành chuỗi `"undefined"`, không phải là chuỗi rỗng. Đây là điểm khác biệt giữa ActionScript và ECMA-262 specification.

unescape

Availability

Flash Player 5.

Usage

```
unescape(x)
```

Parameters

`x` Một chuỗi với các dãy thuộc hệ 16 để thoát ra (A string with hexadecimal sequences to escape).

Returns

Nothing.

Description

Hàm cấp cao (Top-level function); ước lượng tham số x như một chuỗi, đọc chuỗi từ URL-dạng được mã hoá (chuyển đổi tất cả các dãy hệ 16 thành các ký tự ASCII), và trả lại chuỗi.

Example

Ví dụ sau minh họa quá trình chuyển đổi escape-tới-unescape .

```
escape("Hello{[World]}");
```

Kết quả thoát ra (escaped result) như sau:

```
("Hello%7B%5BWorld%5D%7D");
```

Dùng unescape để trả lại dạng nguyên gốc:

```
unescape("Hello%7B%5BWorld%5D%7D")
```

Kết quả như sau:

```
Hello{[World]}
```

unloadMovie

Availability

Flash Player 3.

Usage

```
unloadMovie[Num](level/"target")
```

Parameters

level Mức (`_levelN`) của một movie được load. Khi bạn ngưng nạp (unload) một movie từ một mức (level), hành vi `unloadMovie` trong bảng Action ở chế độ normal (bình thường) chuyển đổi thành `unloadMovieNum`; trong chế độ expert mode, bạn phải chỉ định `unloadMovieNum` hoặc chọn nó từ hộp công cụ Actions.

target Đích đường dẫn của một movie clip.

Returns

None.

Description

Hành vi (Action); gỡ bỏ một movie hoặc một movie clip (đoạn phim) đã được nạp từ Flash Player. Để ngưng nạp (unload) một movie mà đã được nạp vào trong một mức trong Flash Player, sử dụng thông số `level` . Để unload một movie clip đã được nạp, dùng tham số `target` .

Example

Ví dụ sau đây unload movie clip `draggable` trên Timeline chính, và nạp movie `movie.swf` vào level 4.

```
on (press) {  
    unloadMovie ("_root.draggable");  
    loadMovieNum ("movie.swf", 4);  
}
```

Ví dụ sau unload movie đã được nạp vào level 4:

```
on (press) {  
    unloadMovieNum (4);  
}
```

See also

[loadMovie](#), [loadMovieNum](#), [unloadMovieNum](#)

unloadMovieNum

Availability

Flash Player 3.

Usage

```
unloadMovieNum(level)
```

Parameters

`level` Mức (`_levelN`) của một movie đã được nạp.

Returns

Nothing.

Description

Hành vi (Action); gỡ bỏ một movie đã được nạp từ trình Flash Player.

See also

[loadMovie](#), [loadMovieNum](#)

updateAfterEvent

Availability

Flash Player 5.

Usage

```
updateAfterEvent ( )
```

Parameters

None.

Returns

Nothing.

Description

Hành vi (Action); cập nhật sự hiển thị (sự độc lập của các khung frame mỗi giây thiết đặt cho movie) khi bạn gọi nó bên trong một trình xử lý (handler) `onClipEvent` hoặc như một phần của một hàm hoặc method mà bạn chuyển thành (pass to) `setInterval`. Flash bỏ qua các lệnh gọi tới `updateAfterEvent` mà không ở trong một trình xử lý `onClipEvent` handler hoặc một phần của một hàm hoặc method đã được chuyển thành `setInterval`.

See also

[onClipEvent](#), [setInterval](#)

var

Availability

Flash Player 5.

Usage

```
var variableName1 [= value1] [...,variableNameN [=valueN]]
```

Parameters

`variableName` (có nghĩa là tên biến) Một từ định danh.

`value` Giá trị được gán cho biến.

Returns

Nothing.

Description

Hành vi (Action); được dùng để biểu thị các biến cục bộ. Nếu bạn trình bày các biến cục bộ bên trong một hàm, các biến được định nghĩa cho hàm và hết hiệu lực khi kết thúc lệnh gọi hàm. Nếu các biến không được trình bày trong một khối ({}), nhưng danh sách hành vi (action list) đã được thực thi với một hành vi gọi `call`, các biến là cục bộ và hết hiệu lực ở kết thúc của danh sách hiện tại. Nếu các biến không được biểu thị bên trong một khối và danh sách hành vi hiện tại không được thực thi với một hành vi gọi `call` action, các biến không phải là cục bộ.

Example

Các ví dụ sau dùng hành vi `var` để biểu thị và gán các biến:

```
var x;  
var y = 1;  
var z = 3, w = 4;  
var s, t, u = z;
```

void

Availability

Flash Player 5.

Usage

```
void (expression)
```

Description

Toán tử (Operator); một toán tử đơn nguyên (a unary operator: toán tử số học chỉ có một số hạng) để loại bỏ giá trị `expression` (biểu thức) và trả lại một giá trị không xác định. Toán tử `void` (bỏ trống) thường được dùng trong sự so sánh sử dụng toán tử `==` để kiểm tra cho các giá trị không xác định.

while

Availability

Flash Player 4.

Usage

```
while(condition) {  
    statement(s);  
}
```


Parameters

`condition` (điều kiện) Biểu thức mà được ước lượng lại mỗi lần hành vi `while` được thực thi. Nếu khai báo ước lượng là đúng `true`, thì `statement(s)` sẽ chạy.

`statement(s)` Đoạn mã để thực thi nếu điều kiện ước lượng là `true`.

Returns

Nothing.

Description

Hành vi (Action); kiểm tra một biểu thức và thực hiện một câu lệnh hoặc một chuỗi các lệnh lặp lại trong một vòng lặp cho đến khi biểu thức nhận giá trị `true`.

Trước khi khối lệnh được chạy, điều kiện `condition` được kiểm tra; nếu quá trình kiểm tra trả về là đúng `true`, khối lệnh sẽ được chạy. Nếu điều kiện là sai `false`, khối lệnh sẽ được bỏ qua và câu lệnh đầu tiên sau khối lệnh của hành vi `while` sẽ được thực thi.

Quá trình lặp thông thường được dùng để thực hiện một hành vi trong khi một biến đếm (counter variable) nhỏ hơn một giá trị xác định. Tại điểm kết của mỗi vòng lặp, máy đếm được gia tăng một trị số cho đến khi đạt tới giá trị đã xác định ở trên. Ở điểm đó, điều kiện `condition` không trả lại giá trị `true` nữa, và vòng lặp kết thúc.

Câu lệnh `while` thực hiện chuỗi các bước như sau. Mỗi bản sao của các bước từ 1-4 được gọi là một phép lặp của vòng lặp. Điều kiện `condition` được kiểm tra lại ở điểm bắt đầu của mỗi phép lặp, như trong các bước sau:

- 1 Biểu thức điều kiện `condition` được ước lượng.
- 2 Nếu `condition` ước lượng là đúng `true` hoặc một giá trị mà khi chuyển đổi sang giá trị logic Boolean là `true`, chẳng hạn như một số khác không, chuyển sang bước 3.
Nếu không thì, câu lệnh `while` được hoàn thành và thực thi lại ở câu lệnh tiếp theo sau vòng lặp `while` .
- 3 Thực thi khối lệnh `statement(s)`.
- 4 Chuyển tới bước 1.

Example

Ví dụ sau sao lại 5 movie clip trên Stage, mỗi cái với một vị trí `x` và `y` được tạo ngẫu nhiên, thuộc tính `xscale` và `yscale`, và `_alpha` để thực hiện một hiệu ứng rải rác (a scattered effect). Biến `foo` được khởi chạy với giá trị 0. Tham số điều kiện `condition` được thiết đặt với mục đích vòng lặp `while` sẽ chạy năm lần, cho đến khi nào giá trị của biến `foo` còn nhỏ hơn 5. Bên trong vòng lặp `while` , một movie clip được sao lại (duplicate) và `setProperty` được dùng để điều chỉnh các thuộc tính khác nhau của movie clip bản sao. Câu lệnh cuối cùng của vòng lặp gia tăng một trị số `foo` với

mục đích đến khi giá trị đạt tới 5, tham số điều kiện `condition` ước lượng là sai `false`, và vòng lặp sẽ không được thực thi.

```
on(release) {
    foo = 0;
    while(foo < 5) {
        duplicateMovieClip("_root.flower", "mc" + foo, foo);
        setProperty("mc" + foo, _x, random(275));
        setProperty("mc" + foo, _y, random(275));
        setProperty("mc" + foo, _alpha, random(275));
        setProperty("mc" + foo, _xscale, random(200));
        setProperty("mc" + foo, _yscale, random(200));
        foo++;
    }
}
```

See also

[do while](#), [continue](#), [for](#), [for..in](#)

with

Availability

Flash Player 5.

Usage

```
with (object) {
    statement(s);
}
```

Parameters

`object` Một instance (trường hợp) của một đối tượng ActionScript hoặc một movie clip.

`statement(s)` Một hành vi hoặc một nhóm hành vi bao bên trong các dấu ngoặc xoắn.

Returns

Nothing.

Description

Hành vi (Action); cho phép bạn chỉ định một đối tượng (như là một movie clip) với tham số `object` và ước lượng các biểu thức và các hành vi bên trong đối tượng đó với tham số `statement(s)`. Điều này ngăn cản bạn từ việc phải viết đi viết lại tên đối tượng hoặc đường dẫn tới đối tượng.

Tham số `object` trở thành ngữ cảnh trong đó các thuộc tính, các biến, và các hàm trong tham số `statement(s)` đã được đọc. Ví dụ, nếu `object` là `myArray`, và hai thuộc tính đã được chỉ định là `length` và `concat`, các thuộc tính đó tự động được đọc như `myArray.length` và `myArray.concat`.

Trong ví dụ khác, nếu `object` là `state.california`, bất kỳ hành vi hoặc câu lệnh nào bên trong hành vi `with` được gọi từ bên trong `california` instance.

Để tìm giá trị của một từ định danh trong tham số `statement(s)`, `ActionScript` bắt đầu tại mở đầu của `scope chain` (tôi tạm dịch là phạm vi của chuỗi) được chỉ định bởi `object` và tìm kiếm cho từ định danh tại mỗi level (mức) của `scope chain`, theo một thứ tự xác định.

The `scope chain` dùng bởi hành vi `with` để phân tích các từ định danh bắt đầu với mục đầu tiên trong danh sách sau và tiếp tục tới mục cuối cùng:

- ♦ Đối tượng đã chỉ định trong tham số `object` ở tận trong cùng của hành vi (action) `with`.
- ♦ Đối tượng đã chỉ định trong tham số `object` ở phía ngoài cùng của hành vi `with`.
- ♦ Đối tượng kích hoạt. (Một đối tượng tạm thời được tự động tạo khi một hàm được gọi chứa các biến cục bộ đã gọi trong hàm.)
- ♦ Đoạn movie clip chứa đoạn script hiện đang được thực thi.
- ♦ Đối tượng chung :The Global object (gắn liền với các đối tượng chẳng hạn như `Math` và `String`).

Để thiết đặt một biến bên trong một hành vi (action) `with`, biến phải được khai báo (declare) bên ngoài hành vi `with` hoặc bạn phải nhập đường dẫn đầy đủ đến `Timeline` mà trên đó bạn muốn đặt biến. Nếu bạn thiết đặt một biến trong một hành vi `with` mà không khai báo nó, hành vi `with` sẽ tìm kiếm giá trị theo `scope chain`. Nếu biến chưa tồn tại, giá trị mới sẽ được thiết lập trên `Timeline` từ chỗ hành vi `with` được gọi.

Trong `Flash 5`, hành vi `with` thay thế hành vi không được chấp nhận `tellTarget`. Bạn nên hưởng ứng việc sử dụng `with` thay vì `tellTarget` bởi vì nó là một phần `ActionScript` tiêu chuẩn mở rộng của chuẩn `ECMA-262`. Sự khác nhau cơ bản giữa các hành vi `with` và `tellTarget` là ở chỗ `with` đưa một tham chiếu tới một movie clip hoặc đối tượng khác như tham số của nó, trong khi `tellTarget` đưa một chuỗi đường dẫn để nhận biết một movie clip như tham số của nó, và không thể sử dụng các đối tượng đường dẫn.

Example

Ví dụ sau thiết đặt các thuộc tính `x` và `y` của instance (thể hiện, trường hợp) `someOtherMovieClip`, và sau đó chỉ thị cho `someOtherMovieClip` đi đến `frame 3` và dừng lại.

```
with (someOtherMovieClip) {
    _x = 50;
    _y = 100;
    gotoAndStop(3);
}
```

Đoạn mã sau cho ta thấy cách viết code có trước mà không cần dùng hành vi `with`.

```
someOtherMovieClip._x = 50;
someOtherMovieClip._y = 100;
someOtherMovieClip.gotoAndStop(3);
```

Bạn cũng có thể viết đoạn mã này bằng cách sử dụng hành vi `tellTarget`. Tuy nhiên, nếu `someOtherMovieClip` không phải là một movie clip, nhưng là một đối tượng, bạn không thể sử dụng hành vi `with`.

```
tellTarget ("someOtherMovieClip") {
    _x = 50;
    _y = 100;
    gotoAndStop(3);
}
```

Hành vi `with` rất hữu dụng trong việc truy cập vào nhiều mục phức tạp trong một danh sách scope chain (chuỗi mục đích, chuỗi kiến thức...--> chẳng biết dịch thế nào) xảy ra cùng một lúc (is useful for accessing multiple items in a scope chain list simultaneously). Trong ví dụ sau, đối tượng được cài đặt sẵn `Math` được đặt ở phía trước của scop chain. Việc thiết lập `Math` như một đối tượng mặc định phân tích các từ định danh `cos`, `sin`, và `PI` to `Math.cos`, `Math.sin`, và `Math.PI`, một cách lần lượt. Các từ định danh `a`, `x`, `y`, và `r` không phải là các method (phương pháp, cách thức) hoặc các thuộc tính (properties) của đối tượng `Math`, nhưng từ khi chúng tồn tại trong phạm vi kích hoạt của hàm `polar`, chúng phân tích tới các biến cục bộ tương ứng (they resolve to the corresponding local variables).

```
function polar(r) {
    var a, x, y;
    with (Math) {
        a = PI * r * r;
        x = r * cos(PI);
        y = r * sin(PI/2);
    }
    trace("area = " + a);
    trace("x = " + x);
    trace("y = " + y);
}
```

Bạn có thể dùng các hành vi `with` lồng vào nhau để truy cập thông tin trong nhiều scope (phạm vi). Trong ví dụ sau, instance `fresno` và instance `salinas` là có mức là con của instance `california`. Câu lệnh thiết lập các giá trị `_alpha` của `fresno` và `salinas` mà không làm thay đổi giá trị `_alpha` của `california`.

```
with (california){
    with (fresno){
        _alpha = 20;
    }
    with (salinas){
        _alpha = 40;
    }
}
```

See also

[tellTarget](#)

XML (object) : XML (đối tượng)

Sử dụng các method và thuộc tính của đối tượng XML để nạp, phân tích, gửi, xây dựng, và thao tác các cây tài liệu XML. Trong Flash MX, đối tượng XML đã trở thành một đối tượng tự nhiên (tức là đã có sẵn từ các phiên bản trước). Theo nghĩa thông thường, bạn sẽ thấy sự cải tiến ẩn tượng trong khi tiến hành công việc.

Bạn phải dùng constructor (người xây dựng) `new XML()` để tạo một thể hiện (instance) của đối tượng XML trước khi gọi bất kỳ đối tượng XML nào.

XML được hỗ trợ bởi Flash Player 5 và Flash Player 6.

Bản tóm tắt các method cho đối tượng XML.

Method	Description
XML.appendChild	Bổ sung thêm một nút dữ liệu vào cuối của danh sách đối tượng con xác định (Appends a node to the end of the specified object's child list).
XML.cloneNode	Bắt chước nút dữ liệu xác định và, mặc nhiên, sao chép có tính đệ quy tất cả các thành phần con (Clones the specified node and, optionally, recursively clones all children).
XML.createElement	Tạo một phần tử XML mới.
XML.createTextNode	Tạo một XML text node mới.
XML.getBytesLoaded	Trả về số byte đã nạp cho tài liệu XML xác định.
XML.getBytesTotal	Trả về kích cỡ của tài liệu XML theo byte.
XML.hasChildNodes	Trả về <code>true</code> nếu nút dữ liệu đã chỉ định có các nút con; nếu không thì, trả về là <code>false</code> .
XML.insertBefore	Chèn một nút vào trước một nút đã có trong danh sách các nút con xác định (Inserts a node in front of an existing node in the specified node's child list).
XML.load	Nạp một tài liệu (định rõ bởi đối tượng XML) từ một URL.
XML.parseXML	Phân tích một tài liệu XML vào trong cây đối tượng XML xác định.
XML.removeNode	Loại bỏ nút đã chỉ định từ mẹ của nó (Removes the specified node from its parent).
XML.send	Gửi đối tượng XML xác định tới một URL.
XML.sendAndLoad	Gửi đối tượng XML xác định tới một URL và nạp câu trả lời của máy chủ (server response) vào trong một đối tượng XML khác.

XML.toString	Chuyển đổi nút dữ liệu xác định và bất kỳ các nút con của chúng thành văn bản XML.
------------------------------	--

Bản tóm tắt các thuộc tính của đối tượng XML

Property	Description
XML.contentType	Cho biết kiểu MIME đã truyền tới server.
XML.docTypeDecl	Thiết lập và trả về thông tin về một tuyên bố kiểu văn bản của một tài liệu XML (Sets and returns information about an XML document's DOCTYPE declaration).
XML.firstChild	Chuyển đến (liên hệ) nút con đầu tiên trong danh sách cho các nút xác định.
XML.ignoreWhite	Khi thiết lập là true, các nút dữ liệu văn bản (text nodes) mà chỉ chứa dấu cách trắng bị loại bỏ trong suốt quá trình phân tích.
XML.lastChild	Chuyển đến nút con cuối cùng trong danh sách các nút xác định.
XML.load	Kiểm tra nếu đối tượng XML được chỉ định đã được nạp.
XML.nextSibling	Chuyển đến sibling (anh chị em ruột) tiếp theo trong danh sách những đứa con của nút dữ liệu mẹ (References the next sibling in the parent node's child list).
XML.nodeName	Trả về tên tag của một phần tử XML
XML.nodeType	Trả về kiểu của nút dữ liệu đã được chỉ rõ (phần tử XML hoặc nút văn bản).
XML.nodeValue	Trả về text (văn bản) của nút dữ liệu xác định nếu nút là một nút văn bản (text node).
XML.parentNode	Trả về nút mẹ (parent node) của nút đã được chỉ rõ (specified node).
XML.previousSibling	Trả về sibling ngay trước trong danh sách các nút con của nút mẹ (References the previous sibling in the parent node's child list).
XML.status	Trả về một mã trạng thái cho biết việc phân tích một tài liệu XML thành công hay thất bại (Returns a numeric status code indicating the success or failure of an XML document parsing operation).
XML.xmlDecl	Thiết lập và trả về thông tin về một "tuyên bố" của tài liệu XML (Sets and returns information about an XML document's document declaration).

Collections summary for the XML object

Method	Description
XML.attributes	Trả về một mảng kết hợp chứa tất cả các thuộc tính (attribute) của nút dữ liệu đã được chỉ rõ.
XML.childNodes	Trả về một mảng chứa các liên hệ (references) tới các nút con của nút dữ liệu đã được chỉ rõ (specified node).

Bản tóm tắt bộ quản lý sự kiện (Event handler) cho đối tượng XML

Method	Description
XML.onData	Một hàm gọi trở về (callback function) để viện dẫn khi văn bản XML được download trọn vẹn từ máy chủ (server), hoặc một lỗi xảy ra trong khi đang download văn bản XML từ một server.
XML.onLoad	Một callback function dành cho <code>load</code> (nạp) và <code>sendAndLoad</code> (gửi và nạp).

Constructor (người xây dựng) cho đối tượng XML object

Availability

Flash Player 5.

Usage

```
new XML([source])
```

Parameters

`source` Văn bản XML đã phân tích để tạo đối tượng XML mới.

Returns

Nothing.

Description

Constructor; tạo một đối tượng XML mới. Bạn phải dùng method constructor để tạo một thể hiện (instance) của đối tượng XML trước khi gọi bất kỳ method nào của đối tượng XML.

Chú ý: Các method (phương pháp, cách thức) `createElement` (tạo phần tử) và `createTextNode` (tạo nút text) là các method xây dựng 'constructor' cho việc tạo các phần tử và nút dữ liệu text (text nodes) trong một cây tài liệu XML.

Example

Cách dùng 1: Ví dụ sau tạo một đối tượng XML mới và rỗng (không có gì ở trong).

```
myXML = new XML();
```

Cách dùng 2: Ví dụ sau tạo một đối tượng XML bằng cách phân tích văn bản XML được chỉ rõ trong tham số `source`, và đưa (populates) đối tượng XML mới được tạo với việc đưa đến cây tài liệu XML (populates the newly created XML object with the resulting XML document tree).

```
anyOtherXML = new XML("<state>California<city>san francisco</city></state>");
```

See also

[XML.createElement](#), [XML.createTextNode](#)

XML.send

Availability

Flash Player 5.

Usage

```
myXML.send(url, [window])
```

Parameters

`url` Địa chỉ URL đích đối với đối tượng XML đã chỉ định.

`window` Cửa sổ trình duyệt để hiển thị dữ liệu trả về bởi server: `_self` chỉ định khung hiện tại trong cửa sổ hiện tại, `_blank` chỉ định một cửa sổ mới, `_parent` chỉ định "mẹ" (cấp cao hơn) của frame hiện tại, và `_top` chỉ định frame cấp cao nhất trong cửa sổ hiện tại. Tham số này là tùy chọn; nếu không tham số `window` nào được chỉ định, sẽ tương tự như chỉ định `_self`.

Returns

Nothing.

Description

Method (phương pháp, cách thức); mã hóa đối tượng XML đã chỉ định vào trong một tài liệu XML và gửi nó tới địa chỉ URL xác định bằng cách dùng method POST.

XML.status

Availability

Flash Player 5.

Usage

```
myXML.status
```

Description

Property (thuộc tính); tự động thiết đặt và trả về một giá trị số cho biết xem một tài liệu XML đã được phân tách thành công vào trong một đối tượng XML. Mã trạng thái số và một phần mô tả của mỗi mã đó được liệt kê như dưới đây:

- ◆ 0 Không có lỗi; việc phân tích đã hoàn toàn thành công (No error; parse was completed successfully).
- ◆ -2 Một phần CDATA đã không được hoàn thành đúng cách (A CDATA section was not properly terminated).
- ◆ -3 Tuyên bố XML đã không được hoàn thành đúng cách (The XML declaration was not properly terminated).
- ◆ -4 Tuyên bố kiểu tài liệu đã không được hoàn thành một cách thích hợp (The DOCTYPE declaration was not properly terminated).
- ◆ -5 Một nội dung đã không được hoàn thành đúng cách (A comment was not properly terminated).
- ◆ -6 Một phần tử XML bị dị dạng (An XML element was malformed).
- ◆ -7 Xuất bộ nhớ (Out of memory).
- ◆ -8 Một giá trị thuộc tính không được hoàn thành một cách thích hợp (An attribute value was not properly terminated).
- ◆ -9 Một thẻ bắt đầu không khớp với một thẻ kết thúc (A start-tag was not matched with an end-tag).
- ◆ -10 Một thẻ kết thúc được bắt gặp mà không khớp với một thẻ bắt đầu (An end-tag was encountered without a matching start-tag).

XMLSocket (object) {Ổ cắm XML (đối tượng)}

Bổ sung kiến thức: *Socket*: ổ cắm : trong việc ghi địa chỉ trên Internet là một sự kết hợp một địa chỉ IP (địa chỉ này xác định một máy chủ) và một địa chỉ cổng (địa chỉ này xác định một trình ứng dụng chạy trên máy chủ này). Hai địa chỉ này cùng được sử dụng để xác định một dịch vụ đặc biệt trên khắp Internet. Điều này không cần thiết cho các dịch vụ nổi tiếng chẳng hạn như FTP và Telnet, bởi lẽ các dịch vụ này sử dụng các cổng tiêu chuẩn. Như bạn muốn sử dụng một chương trình duy nhất

như một MUD (trò chơi săn tìm trên Internet) đang chạy trên một máy chủ từ xa nào đó, thì bạn sẽ phải có một ổ cắm. Ví dụ, để truy cập một MUD bạn sử dụng Telnet và đánh địa chỉ số của ID và sau đó là số cổng.

Đối tượng XMLSocket thực thi các ổ cắm khách (client socket) mà cho phép máy tính chạy Flash Player để giao tiếp với một máy tính chủ nhận dạng bởi một địa chỉ IP hoặc tên miền (domain name).

Sử dụng đối tượng XMLSocket

Để sử dụng đối tượng XMLSocket, máy chủ (server computer) phải chạy một daemon (trong các hệ thống UNIX, đây là một trình tiện ích hoạt động kín đáo trong nền sau, thực hiện các nhiệm vụ cần thiết như tiếp nhận thư điện tử gửi đến chẳng hạn, người dùng không nhận biết sự có mặt của daemon) hiểu được giao thức sử dụng bởi đối tượng XMLSocket. Giao thức như là dưới đây:

- ✦ Các thông báo XML được gửi lên trên một song công hoàn toàn (một định ước truyền thông, trong đó kênh truyền thông có thể gửi và nhận các tín hiệu đồng thời cùng một lúc) TCP/IP làm thông kết nối socket (XML messages are sent over a full-duplex TCP/IP stream socket connection).
- ✦ Mỗi thông báo XML là một tài liệu XML hoàn thành, kết thúc bởi một byte không (Each XML message is a complete XML document, terminated by a zero byte).
- ✦ Một số không giới hạn các thông báo XML có thể được gửi và tiếp nhận trên một kết nối XMLSocket đơn (An unlimited number of XML messages can be sent and received over a single XMLSocket connection).

Đối tượng XMLSocket rất hữu ích đối với các trình ứng dụng khách-chủ (client-server) đòi hỏi thời gian chờ thấp, chẳng hạn như các hệ thống chat thời gian thực (real-time chat system). Một giải pháp chat dựa trên HTTP truyền thông thường xuyên thăm dò máy chủ và tải xuống những tin nhắn mới bằng cách sử dụng một yêu cầu HTTP. Trái lại, một giải pháp chat XMLSocket duy trì một kết nối mở tới máy chủ, cho phép máy chủ ngay lập tức gửi các tin nhắn mới đến mà không cần một yêu cầu từ máy khách.

Thiết lập trên một máy chủ để giao tiếp với đối tượng XMLSocket có thể bị không thừa nhận. Nếu trình ứng dụng của bạn không phụ thuộc vào tương tác thời gian thực, hãy sử dụng hành vi `loadVariables`, hoặc Flash HTTP-dựa vào XML server kết nối (`XML.load`, `XML.sendAndLoad`, `XML.send`), thay vì sử dụng đối tượng XMLSocket.

Để sử dụng các method của đối tượng XMLSocket, đầu tiên bạn phải dùng trình xây dựng constructor, `new XMLSocket`, để tạo một đối tượng XMLSocket mới.

XMLSocket và bảo mật

Bởi vì đối tượng XMLSocket thiết lập và duy trì một kết nối mở tới máy chủ, các hạn chế sau đây được đặt trên đối tượng XMLSocket cho các lý do bảo mật:

- ✦ Method `XMLSocket.connect` chỉ có thể kết nối tới TCP các cổng số lớn hơn hoặc bằng 1024. Một kết quả của hạn chế này là các daemon máy chủ mà giao tiếp với đối tượng `XMLSocket` cũng phải được gán vào các số cổng lớn hơn hoặc bằng 1024. Các số cổng nhỏ hơn 1024 thường được sử dụng bởi các hệ thống dịch vụ như FTP, Telnet, và HTTP, do đó đối tượng `XMLSocket` bị chặn từ những cổng này bởi những lý do bảo mật. Sự hạn chế số cổng giới hạn tiềm năng mà những tài nguyên này sẽ bị truy cập và lạm dụng một cách không thích hợp.
- ✦ Method `XMLSocket.connect` chỉ có thể kết nối tới các máy tính trong cùng miền con nơi đặt file SWF (movie). Hạn chế này không áp dụng cho các movie chạy khỏi một đĩa cục bộ (running off a local disk). (Hạn chế này giống hệt các quy tắc bảo mật đối với `loadVariables`, `XML.sendAndLoad`, và `XML.load`.)

Bảng tóm lược các method đối với đối tượng `XMLSocket`

Method	Description
XMLSocket.close	Đóng một kết nối socket mở
XMLSocket.connect	Thiết lập một kết nối tới máy chủ xác định.
XMLSocket.send	Gửi một đối tượng XML tới máy chủ.

Event handler cho đối tượng `XMLSocket`

Method	Description
XMLSocket.onClose	Một hàm (chức năng) gọi ngược lại được viện dẫn khi một kết nối <code>XMLSocket</code> bị đóng.
XMLSocket.onConnect	Một hàm gọi ngược lại (callback function) được viện dẫn khi một kết nối <code>XMLSocket</code> được thiết lập.
XMLSocket.onData	Một hàm gọi ngược lại được viện dẫn khi một thông báo XML được tải xuống từ máy chủ.
XMLSocket.onXML	Một hàm gọi ngược lại được viện dẫn khi một đối tượng XML đến từ máy chủ.

Constructor cho đối tượng `XMLSocket` object

Availability

Flash Player 5.

Usage

```
new XMLSocket()
```

Parameters

None.

Returns

Nothing.

Description

Constructor (trình xây dựng); tạo một đối tượng XMLSocket mới. Đối tượng XMLSocket ban đầu không kết nối với bất kỳ máy chủ nào. Bạn phải gọi method `XMLSocket.connect` để kết nối đối tượng tới một server.

Example

```
myXMLSocket = new XMLSocket();
```

See also

[XMLSocket.connect](#)