

onClipEvent 影片事件處理程序

功能說明

OnClipEvent 是用來設定場景上影片片段物件的影片事件處理程序，這個指令只能用在已放置在場景上的實體影片片段物件。

對應版本 Flash 5 以上

快速鍵 ESC+o+c

性質 Handler

語法

```
onClipEvent(movieEvent){  
.....;  
}
```

參數

movieEvent：影片事件，有 load、unload、enterFrame、mouseMove、mouseDown、mouseUp、keyDown、keyUp、data 九種。詳細說明請參閱第四篇第三十四章「影片片段類別」的「事件回應」中對應字詞的說明。

statement：事件發生時執行的程式

敘述範例

```
onClipEvent (enterFrame) {  
    this._x = _xmouse;  
    this._y = _ymouse;  
}
```

當時間軸播放到含有該影片的圖格的時候，就開始執行事件以下的程式。

關聯參照

key (object)、on (movieEvent)

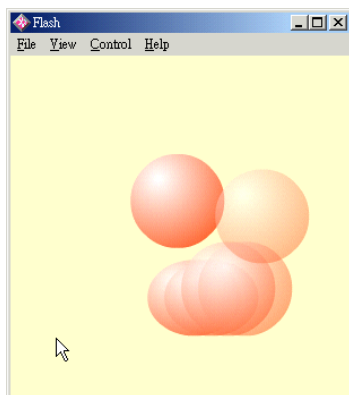
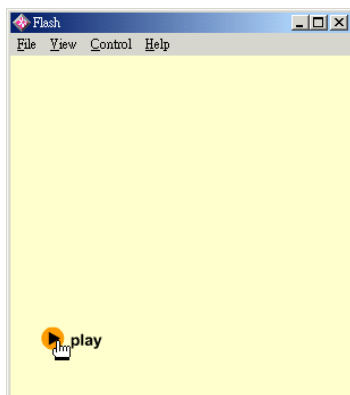
應用範例

```
▶ //movie clip 裡的 script  
▶ onClipEvent (enterFrame) {  
▶     if (_root.i>0) {  
▶         duplicateMovieClip (_root.ball, "ball_" add _root.i, _root.i);  
▶         setProperty ("_root.ball_" add _root.i, _alpha, 100-_root.i*10);  
▶         setProperty ("_root.ball_" add _root.i, _x, 145+_root.i*15);  
▶         _root.i--;  
▶     }  
▶ }
```



onClipEvent 影片事件處理程序

執行結果 執行後，按下按鈕跳至下一圖格，由於下一圖格包含 ball 這個影片物件，就會開始執行影片物件的 onClipEvent 事件處理程序。

**TIPS**

- 以下是 Clip Event 事件的含意：

Clip 事件	含意
Load	Movie Clip 被載入時執行程式(只執行一次)
EnterFrame	播放到 Movie Clip 所在圖格時執行程式(反覆執行)
Unload	Movie Clip 被釋放時執行程式
Mouse down	按下滑鼠左鍵時執行程式
Mouse up	放開滑鼠左鍵時執行程式
Mouse move	每次滑鼠移動時就執行程式
Key down	按下鍵盤上任何按鍵時執行程式
Key up	放開鍵盤上任何按鍵時執行程式
Data	當外部資料使用「Load Variable」或「Load Movie」載入時執行程式

removeMovieClip 移除電影短片

功能說明

主要功能為移除影片片段物件，把之前使用 duplicateMovieClip 或 AttachMovie 複製出的實體影片片段物件，從場景上移除。

對應版本 Flash 4 以上

快速鍵 ESC+r+m

性質 Action

語法

removeMovieClip(target);

參數

target：實體影片片段物件的目標路徑

敘述範例

```
removeMovieClip (star101);
```

移除實體名稱為 "star101" 的影片片段物件

用途

移除不需要的影片片段物件

關聯參照

attachMovie、duplicateMovieClip、MovieClip.duplicateMovieClip

應用範例

```
▶ // 場景上第一格 frame 的 script
▶ i = 5;
▶ for (j=1; j<i; j++) {
▶     duplicateMovieClip (_root.ball, "ball_" + j, j);
▶     setProperty ("ball_" + j, _x, 100+j*20);
▶ }
▶ stop ();
▶
▶ // 按鈕上的 script
▶ on (release) {
▶     removeMovieClip ("_root.ball_" + _root.i);
▶     _root.i--;
▶ }
```



removeMovieClip 移除電影短片

執行結果 執行後，則場景上會先行複製出 5 個跳動的球，每按一次按鈕，則依序刪除每一個球。

第

1

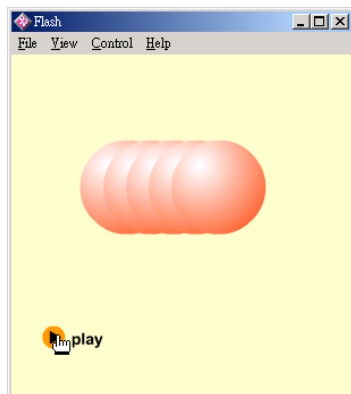
章

全

域

函

數



TIPS

- 移除的 movie clip 必須是 duplicateMovieClip 或 AttachMovie 複製出的 instance，一般的 movie clip 並無法使用 removeMovieClip 從場景上移除。

setProperty 設定影片片段物件屬性

功能說明

改變影片片段物件目前的屬性。影片片段物件在場景上稱之為實體，而實體的屬性是可以隨時修改的，如 alpha 透明度，X、Y 座標值，實體的寬、高、角度等等。

對應版本 Flash 4 以上

快速鍵 ESC+s+p

性質 Action

語法 `setProperty(target, property, expression);`

參數 target：影片片段物件的目標路徑

property：物件的屬性

expression：物件屬性的值

敘述範例

```
setPproperty (dog, _x, 250);
```

把名稱為 dog 的影片片段物件的 X 座標值設定成 250

用途

修改實體目前的屬性

關聯參照

`getProperty`

應用範例

```
▶ on (rollOver) {  
▶     setProperty (_root.boy, _xscale, 150);  
▶     setProperty (_root.boy, _yscale, 150);  
▶ }  
▶ on (release) {  
▶     n++;  
▶     setProperty (_root.boy, _rotation, 90*n);  
▶ }  
▶ on (rollOut) {  
▶     setProperty (_root.boy, _xscale, 100);  
▶     setProperty (_root.boy, _yscale, 100);  
▶ }
```



setProperty 設定影片片段物件屬性

執行結果 滑鼠指向按鈕時，場景上的 boy 會放大，每按一下按鈕，則 boy 就會順時針旋轉 90 度。滑鼠移開按鈕，場景上的 boy 會變回原來大小。



startDrag 拖曳影片片段物件

功能說明

用來拖曳場景上的實體，執行時，被拖曳的實體會跟著滑鼠游標的位置移動。

對應版本 Flash 4 以上

快速鍵 ESC+d+r

性質 Action

語法

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

參數

target：影片物件的目標路徑

lock：以布林值（true、false）判斷物件是否鎖定滑鼠游標中心點，當布林值為 true 的時候，影片物件的中心點鎖定滑鼠游標的中心點。

left、top、right、down：物件在場景上可拖曳的上下左右邊界，當 lock 為 true 的時候，才能設定邊界的引數。

敘述範例

```
startDrag ("squareA");
```

開始拖曳 squareA 物件

```
startDrag (_root.squareB, true);
```

開始拖曳場景上的 squareB 物件，拖曳時物件的中心點自動鎖定滑鼠游標的中心點。

```
startDrag ("", true, 0,0,200,200);
```

開始拖曳物件本身，拖曳時物件的中心點自動鎖定滑鼠游標的中心點，並且設定上下左右邊界

用途

製作拖曳互動介面

關聯參照

stopDrag、_droptarget

應用範例

- ▶ `// "drag 輪廓" 按鈕裡的 script`
- ▶ `on (press) {`
- ▶ `startDrag (_root.boy,true, 65,80,235,220);`
- ▶ `}`
- ▶ `// "drag 陰影" 按鈕裡的 script`
- ▶ `on (press) {`



startDrag 拖曳影片片段物件

```
▶ startDrag (_root.shadow);  
▶ }
```

第

1

章

全
域

函

數

執行結果 按下"drag 輪廓"按鈕，可拖曳 boy 輪廓，拖曳時 boy 輪廓的中心點會所定滑鼠游標的中心點，並且不會超出場景的邊界。按下"drag 陰影"按鈕，可拖曳陰影，而陰影會超出場景的邊界。



stopDrag 停止拖曳影片片段物件

功能說明

用來解除滑鼠拖曳物件的動作。

對應版本 Flash 4 以上

快速鍵 ESC+s+d

性質 Action

語法 stopDrag();

參數 無

敘述範例

```
stopDrag ();
```

停止滑鼠拖曳物件的動作

用途

配合 startDrag 製作拖曳互動介面

關聯參照

startDrag 、 _droptarget

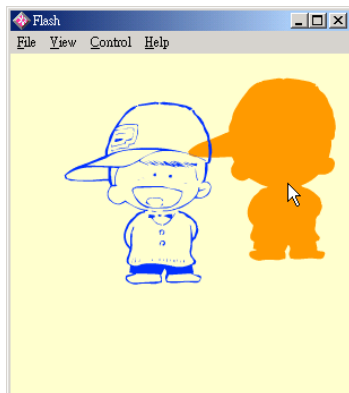
應用範例

```
▶ // "boy" 物件裡的 onClipEvent
▶ onClipEvent (mouseDown) {
▶     n++;
▶     if (n%2 == 1) {
▶         startDrag (_root.boy, true, 65, 80, 235, 220);
▶     } else {
▶         stopDrag ();
▶     }
▶ }
▶ // "shadow" 物件裡的 onClipEvent
▶ onClipEvent (keyDown) {
▶     startDrag (_root.shadow);
▶ }
▶ onClipEvent (keyUp) {
▶     stopDrag ();
▶ }
```



stopDrag 停止拖曳影片片段物件

執行結果 按一下 boy 物件，可拖曳 boy 輪廓，再按一下 boy 物件，則可停止拖曳。按下鍵盤上任一個按鍵，可拖曳陰影，放開鍵盤按鍵，則可停止拖曳。



targetPath 取得指定影片片段物件的完整路徑

功能說明

對應版本 Flash 5 以上

性質 Function

取得指定影片片段物件的完整路徑值並以 " . " 點標記法傳回字串。

語法 targetPath(movieClipObject);

參數 movieClipObject：欲傳回完整路徑的參考點，如：_root、_parent。

敘述範例

```
myVar=targetPath(_root.ball);
```

將 _root.ball 的完整路徑傳回並字串值指定給 myVar。

用途

取得完整路徑。

關聯參照

eval、_target

應用範例

```
▶ on (release) {  
▶   p = targetPath(ball);  
▶ }
```

執行結果 按一下按鈕，畫面上就會顯示 movie clip 的物件路徑。





updateAfterEvent 更新事件處理函式結果顯示

功能說明

在執行完 onClipEvent 事件處理函式的程式後或利用 setInterval() 呼叫此函式時，更新處理執行結果的畫面顯示。

對應版本 Flash 5 以上

性質 Action

語法 updateAfterEvent();

參數 無。

敘述範例

```
onClipEvent (load) {  
    updateAfterEvent();  
}
```

當 onClipEvent(load) 的動作執行完成後，執行 updateAfterEvent 以更新執行結果的畫面顯示。

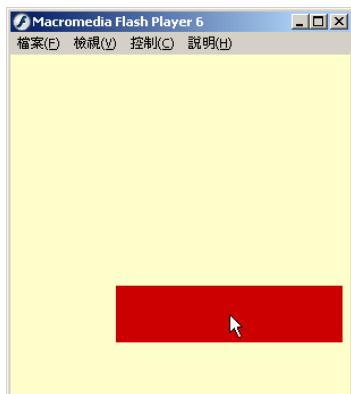
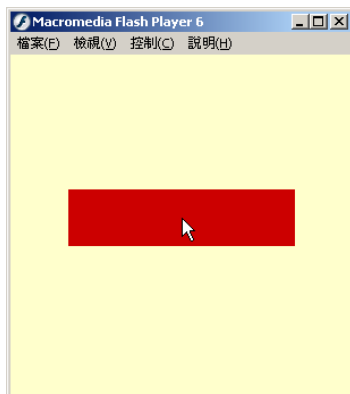
關聯參照 onClipEvent()、setInterval()

應用範例

```
► //movie clip 裡的 script  
► onClipEvent (mouseDown) {  
►     click += 1;  
► }  
► onClipEvent (mouseMove) {  
►     _x = _root._xmouse;  
►     _y = _root._ymouse;  
►     if (click%2 == 0) {  
►         updateAfterEvent(mouseMove);  
►     }  
► }
```

updateAfterEvent 更新事件處理函式結果顯示

執行結果 執行後，紅色矩形方塊會隨著滑鼠游標移動，按一下滑鼠左鍵，並觀察前後的變化。



TIPS

- 以下是 Clip Event 事件的含意：

Clip 事件	含意
Load	Movie Clip 被載入時執行程式(只執行一次)
EnterFrame	播放到 Movie Clip 所在圖格時執行程式(反覆執行)
Unload	Movie Clip 被釋放時執行程式
Mouse down	按下滑鼠左鍵時執行程式
Mouse up	放開滑鼠左鍵時執行程式
Mouse move	每次滑鼠移動時就執行程式
Key down	按下鍵盤上任何按鍵時執行程式
Key up	放開鍵盤上任何按鍵時執行程式
Data	當外部資料使用「Load Variable」或「Load Movie」載入時執行程式