SwishMax

and

ActionScript 2

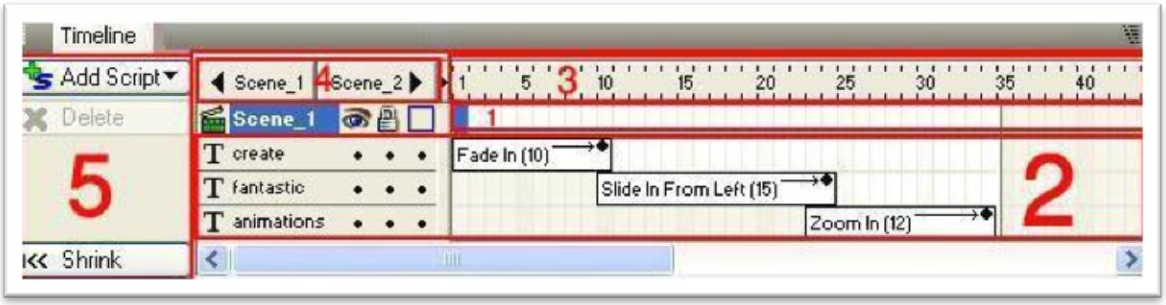
نویسنده : محمد رضا باغبانی
www.uc.blogfa.com
پرسش و پاسخ در سایت مجید آنلاین:
www.majidonline.com

به نام خدا

- معرفی Timeline و Frame
- معرفی آبجکت ها :
  - Shape(اشکال هندسی و غیر هندسی)
  - ایجاد دکمه
  - ایجاد یک Movie
  - ایجاد یک متن
  - ایجاد گروه اشیا
  - وارد کردن فایل صوتی
  - وارد کردن فایل ویدئویی
- معرفی ابزار
- افکت ها
- دیگر آبجکت ها
- Action Script 2

آشنایی با محیط نرم افزار

Frame , Timeline



تصویر 1

پنل Timeline جزئیات صحنه کنونی بر حسب زمان است. هر صحنه از یک سری فریم تشکیل می شود در حقیقت افکتها و تصاویر متحرکی که در اینترنت و رایانه می بینیم از فرم ها تشکیل شده اند. پنل Timeline ابتدای فریم را در سمت چپ و انتهای فریم را در سمت راست به صورت بصری نشان می دهد.

بالا ترین سطر (تصویر 1 بخش 1) رویداد های فریم و اکشن های صحنه جاری را نمایش می دهد. اکشن ها زمانی اجرا می شوند که نمایش ما به ستون معین شده ای برسد که اکشن در آنجا قرار دارد. در هر ستون فقط یک نماد اکشن دیده می شود ولی شما می توانید در هر ستون از فریم مقدار زیادی اکشن بنویسید.

سطر های بعدی (تصویر 1 بخش 2) مربوط به اشیای موجود در صحنه است. در اینجا هر شیئی بالاتر قرار بگیرد در صحنه جاری نیز این شی روی شی پایینتر قرار می گیرد.

این سطر ها نشان دهنده افکت هایی است که برای آبجکت ها استفاده کرده اید.

افکت ها ممکن است یک ستونِ یک خانه ای باشند یا شامل خانه های بیشتری شوند. اما در هر خانه فقط یک افکت می تواند قرار بگیرد. (همون طور که در تصویر1 بخش 2 می بینید آبجکت اولی دارای افکت 10 خانه ای و آبجکت زیری دارای افکت 15 خانه ای هستش)

هر لحظه از فریم با خطوط سیاه رنگ کوچک از یک دیگر جدا شده اند(تصویر 1 بخش 3). نرخ و تعداد این لحظه ها مطابق زمان نمایش ما تنظیم می شود.(به طور معمول هر ثانیه از 25 فریم تشکیل میشه)

نکته:

هر صحنه تا زمانی که افکت یا اکشنی را اضافه نکنید شامل صفر فریم است .

کنترل صحنه

(تصویر 1 بخش 4) این بخش زمانی فعال می شود که در نمایش بیش تر از یک صحنه داشته باشیم.

قسمت راست (تصویر 1 بخش 4) نمایش صحنه بعدی و قسمت چپ نمایش صحنه قبلی است.

هنگامی که اولین سطر را انتخاب می کنید در تصویر 1بخش 4 دکمه Add Script نمایان می شود با کلیک بر روی این دکمه منوی اسکریپت نمایش داده می شود.

هنگامی که سطر های بعدی را انتخاب می کنید دکمه Add Effect نمایش داده می شود با کلیک بر روی این دکمه منوی افکت ها نمایش داده می شود.

آبجکت ها

- آبجکت های ساده

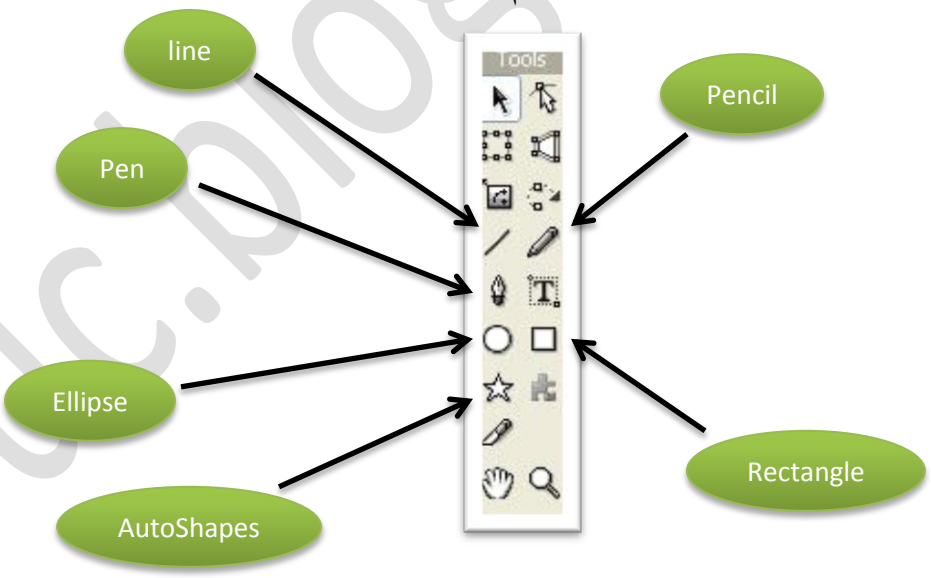
- Shape
- Button
- Movie Clip
- Input text
- Dynamic text
- Static text (وقتی تیک گزینه 'Target' فعال باشد)
- Group (وقتی نام گذاری شده و تیک گزینه 'Target' فعال باشد)
- Sound
- فایل ویدئویی داخل پروژه
- مدیا های خارجی (خارج از پروژه)

- آبجکت های مجتمع
  - static text(وقتی تیک گزینه 'Target' فعال نباشد)
  - Group (گزینه 'Target' فعال نباشد)

---

shape

Shape ها را می توانید با ابزار های زیر رسم کنید :



---

button

برای ساخت دکمه کافیست ابتدا یک shape ایجاد کنید سپس بر روی آن راست کلیک کنید و از Grouping گزینه Group as button را انتخاب کنید .

MovieClip

MovieClip ها به تنهایی جزو آبجکت های ساده هستند این آبجکت ها برای خود یک Timelineمخصوص دارند و می توانند شامل تعداد نامحدودی آبجکت زیر گروه از جمله Shape ها movieClip ها ، Buttonها و حتی Effect وScript باشند.

ایجاد یک MovieClip

از منوی بالایی بر روی گزینه Insert کلیک کنید از منوی باز شده بر روی MovieClip کلیک کنید.

Text

آبجکت text به سه نوع تقسیم می شود

- Static
- Dynamic
- Input

برای آبجکت text وقتی می توانیم script بنویسیم که از پنل Peroperties تیک گزینهTarget را فعال کنیم.

Static text : برای ایجاد فایل های متنی استاتیک در محتوایتان مورد استفاده قرار می گیرد statictextها در پنل peroperties قسمت target تیک ندارند.برای این آبجکت ها نمی توان اسکریپت نوشت.

Dynamic text : برای این آبجکت ها می توان اسکریپت نوشت (مثلا با کد نویسی محل این آبجکت رو تغییر بدیم یا رنگش رو عوض کنیم) این آبجکت ها حتما باید در پنل properties دارای نام یا در فیلد var یک مقداری داشته باشند

(اگر قسمت نام یا فیلد Var خالی باشه در هنگام اجرا با این خطا مواجه میشید:

WARNING : The dynamic/input text object containing the text '' should have a name

این متن یعنی آبجکت متنی دینامیک یا اینپوت ایجاد شده باید اسم داشته باشه)

Input text : این آبجکت امکان نوشتن متن توسط کاربر (کسی که با پروژه نهایی شما کار می کند) را فراهم می کند.

در این نرم افزار شما می توانید متن های خارجی (فقط از نوع txt) را نیز وارد کنید برای این کار از گزینه importText که با علامت (تصویر شماره 2) نمایش داده شده است استفاده کنید .
(این قسمت نوشته های فارسی رو پشتیبانی میکنه)

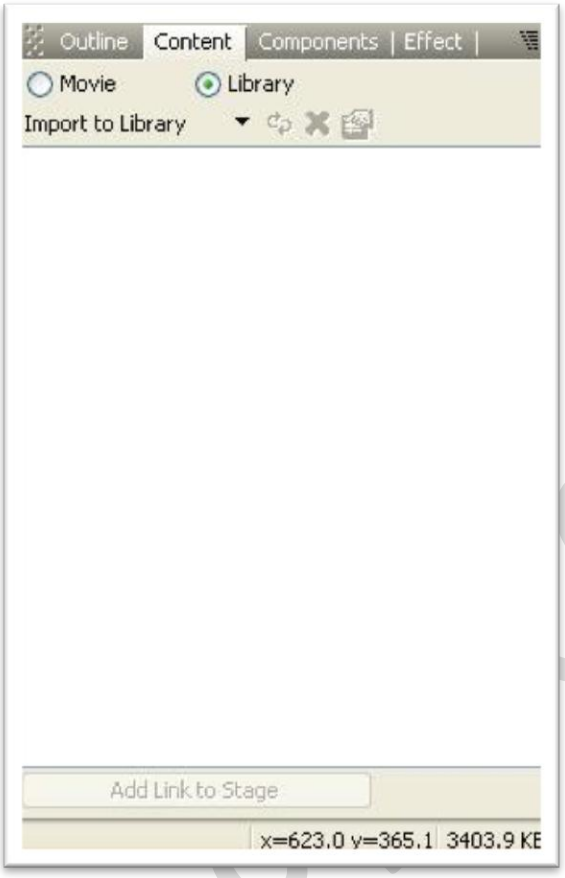
تصویر شماره2

Group

برای ایجاد Groupکافیست آبجکت هایی که در داخل صحنه خود ایجاد کردید را با selection toolانتخاب کرده سپس بر روی آن راست کلیک کنید از گزینه Groupingمقدار group as group را انتخاب کنید .
این آبجکت جزو آبجکت های مجتمع می باشد. ولی وقتی نامی برای آن انتخاب کنیم و تیک گزینه Target را برای آن فعال کنیم این آبجکت یک آبجکت ساده خواهد شد . وقتی به هر دو Group افکتی اعمال کنیم هریک نمایش متفاوتی خواهند .
Group همانند آبجکت movieClip است ولی تفاوت آنها این است که movieClip در داخل خود یک timeline مخصوص دارد ولیGroup این گونه نیست .

Sound

برای وارد کردن فایل صوتی ابتدا به تب content از پنل سمت راست مراجعه کنید (تصویر شماره3) اینجا دو گزینه را مشاهده می کنید.



تصویر 3

:Movie

روی دکمه رادیویی movie کلیک کنید (تصویر 3) تا فعال شود از import to stage گزینه soundرا انتخاب کنید سپس فایل صوتی مورد نظر را انتخاب کنید و وارد کنید مشاهده می کنید که دو پوشه ایجاد شد پوشه soundtracksکه فایل شما را نشان میدهد و پوشه audio Resources که مسیر فایل وارد شده را نشان می دهد در پنل timeline نیز مشاهده می کنید که فایل صوتی شما در داخل stage قرار گرفته (باعمل Drag هم میتونید فایل صوتی تون رو وارد کنید) با اجرای پروژه کلید ترکیبی (Ctr+T) فایل صوتی شما همراه پروژه اجرا می شود.

: Library

یک پروژه جدید بسازید کلید ترکیبی (Ctr+N) از پنل content، Library را انتخاب کنید.
از Import to libraryگزینه Sound را انتخاب کنید فایل صوتی مورد نظر را وارد کنید با اجرای پروژه فایل صوتی وارد شده اجرا نمی شود زیرا برای این کار نیاز به اسکریپت نویسی می باشد به شما توصیه می کنم در وارد کردن فایل های صوتی (چه در Swish یا Flash) به نکات زیر توجه نمایید:
فایل صوتی خود را از قسمت library وارد کنید یا آنها را از خارج پروژه فراخوانی کنید.

برای انتخاب موزیک Background از موزیک های (loop تکرار شونده) استفاده شود.

وارد کردن فایل های ویدئویی :

همانند وارد کردن صدا این فایل را نیز می توانید به قسمت movie یا Library وارد کنید .
توصیه هایی در رابطه با این بخش :
اگر می خواهید یک فایل ویدئویی را وارد کنید حتما از فرمت flv. استفاده کنید .اگر چه این نرم افزار قابلیت تبدیل برخی فرمت های ویدئویی را دارا ست ولی سرعت تبدیل در حجم های متوسط و زیاد کار آمد نیست بدین منظور از نرم افزار های تبدیل کننده استفاده شود.
حد المقدور فایل های ویدئویی را در داخل خود پروژه وارد نکنید بلکه آنها را در خارج پروژه قرار دهید و با استفاده از کد نویسی آنها را فراخوانی کنید .
Insert External media : از منو اصلی insert را انتخاب کنید بر روی External media کلیک کنید این آبجکت را از Stage در حالت انتخاب شده قرار دهید در پنل Peroperties مسیر فایل مورد نظر را نتخاب کنید (فقط از فرمت FLV پشتیبانی می کنه) این آبجکت را می توانید به غیر از Stage در داخل MovieClipوGroup نیز استفاده کنید.

اسکریپت نویسی

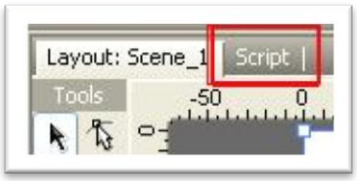
- اسکریپت به کاربر اجازه می دهد که اشیاء داخل پروژه را کنترل کند .
- با اسکریپت می توان تعریف کرد یک برنامه در یک چارچوب خاص عمل شود مثلا وقتی دو شی به یک دیگر بر خورد کنند یا مقداری تغییر کند و... .
- اسکریپت برای کنترل صدا بارگذاری فیلم یا متن در داخل پروژه کاربرد دارد .
- اسکریپت می تواند با اسکریپت های خارجی دیگر مانند PHP یا ASP ارتباط برقرار کند .
- با اسکریپت می توان خواص فیزیکی (physics) را برای اشیا تعریف کرد مانند اصطحکاک شتاب جاذبه .

اکشن های swishmax تفاوت های کمی با اکشن های برنامه flash دارد . و اکشنی که کار خواهیم کرد اکشن اسکریپت 2 خواهد بود .

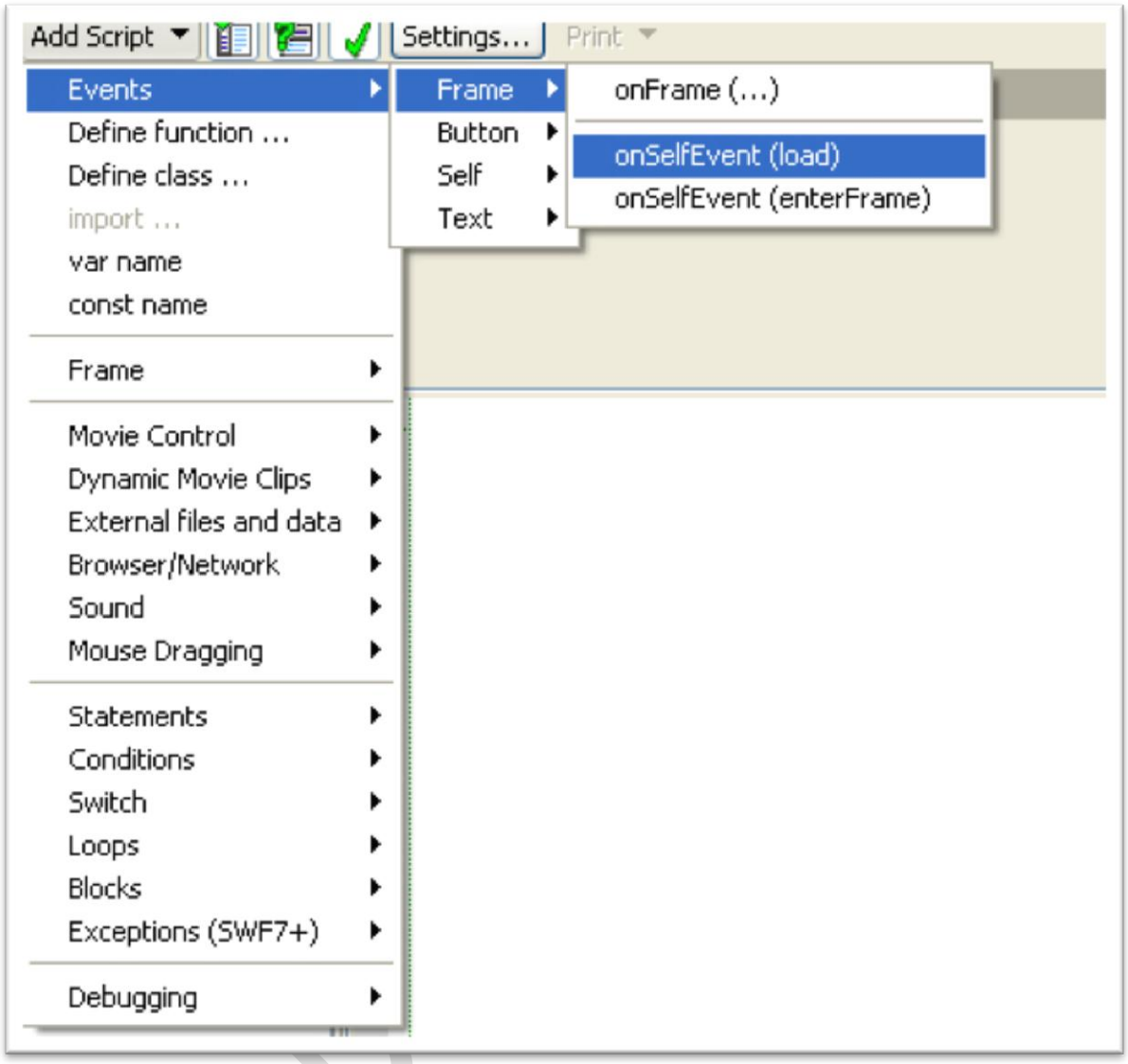
به پنل script بروید (تصویر1) از مسیر (تصویر 2) :

add script ‣ events ‣ frame ‣

onSelfEvent (load) را انتخاب کنید قطعه کدی به صفحه اسکریپت نویسی اضافه میشود این قطعه کد ، بیانگر این است که در هنگام اجرا (load)، دستور داخل آکولاد انجام شود . حالا به پنل layout برگردید و پروژه را با کلید ترکیبی (CTR+Enter) اجرا کنید .
مشاهده می کنید که هیچ اتفاقی نمی افتد حالا از منوی کنترل کلید Stop را بفشارید. قسمت مشخص شده در تصویر 3.



تصویر 1



تصویر 2

تصویر3

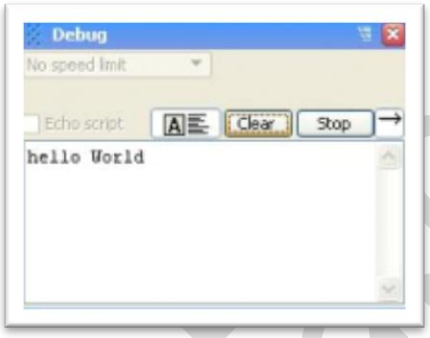
باز به پنل اسکریپت باز گردید بعد از اولین آکولاد یک بار کلید Enter را بفشارید تا قطعه کد زیر را در آن بنویسید :

```
trace("hello world");
```

در کل شکل اسکریپت شما به حالت زیر خواهد بود :

```
onSelfEvent (load) {
  trace("hello World");
}
```

حالا به layout بروید و پروژه را با (CTR+Enter) اجرا کنید .
پیغام hello world در پنل debug نمایش داده می شود (تصویر 4) که بیانگر صحیح بودن کد و اجرای درست پروژه است .



تصویر 4

تحلیل :

```
onSelfEvent (load) {
}
```

این کد زمانی اجرا میشود که پروژه شما اجرا شود(movie Clip/ object شما فراخوانی بشه یا همون load بشه) .
تابع load برای تعریف مرز (لحظه اولیه) شرایط و تعریف متغیرها بسیار مفید است .

```
trace(عبارت);
```

نشانوند:

trace : پیغامی را که شما تعریف کرده اید در پنجره debug برای شما نمایش میدهد .
عبارت : میتواند عدد ، **رشته** یا **متغیر** قرار بگیرد .
رشته : "reza" یا "hello" یا عبارات دیگری که میان "" قرار بگیرند .
متغیر : متغیر ها مقدار هایی هستند که از قبل برای پروژه تعریف کرده اید مانند :
tedad = 2 یا name = “reza”

onFrame

پروژه جدیدی بسازید و کد های زیر را در آن جایگذاری کنید .
ابتدا از مسیر زیر :

addScript ‣ Events ‣ Frame ‣

onSelfEvent(load) را انتخاب کنید .(تصویر2)

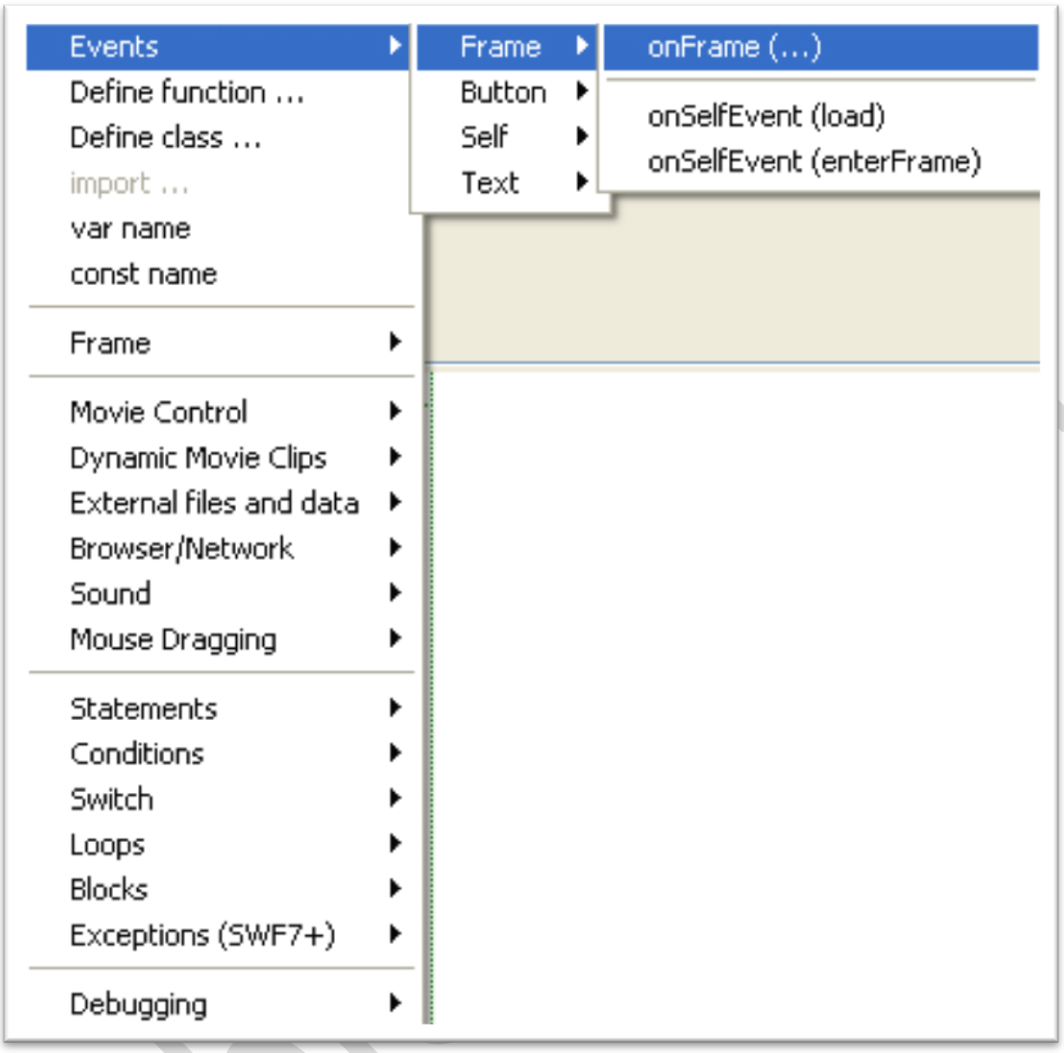
و در داخل آن بنویسید :

```
i = 0 ;
```

با این کار شما یک متغیر تعریف کرده اید به نام i و مقدار آن نیز در هنگام فراخوانی پروژه (Load)
برابر 0 قرار میگیرد .
اسکریپت شما به شکل زیر خواهد بود :

```
onSelfEvent (load) {
   i = 0 ;
}
```

سپس onFrame(…) ‣ Frame ‣ Events ‣ addScript را انتخاب کنید (تصویر 5)



تصویر 5

در داخل پرانتز onFrame عدد 25 را وارد کنید . اسکریپت شما به این شکل خواهد بود :

```
onSelfEvent (load) {
  i = 0 ;
}
onFrame (25) {
}
```

در داخل آکولاد ، کد زیر را جایگذاری کنید :

```
trace ( );
```

در داخل()trace کد زیر را بنویسد :

```
++i ;
```

یا :

```
i = i + 1 ;
```

یا :

```
i += 1 ;
```

اینها همه یعنی این که متغیر i که در هنگام فراخوانی برابر 0 قرار داده بودیم (منظورم در onSelfEvent(load) است) بعلاوه 1 شود .

در کل کد شما به شکل زیر خواهد بود :

```
onSelfEvent (load) {
   i=0;
}
onFrame (25) {
   trace(++i);
}
```

به پنل Layout باز گردید حالا پروژه را با کلید ترکیبی Ctr+Enter اجرا کنید . مشاهده می کنید که در هر 25 فریم یک عدد به عدد قبلی اضافه می شود .

تحلیل :

در اولین بار که نمایشگر فریم در پنل timeline به 25 میرسد پنجره debug عدد 1 را نشان میدهد چون :

```
++i ;
```

برابر است با :

0 +1 که میشود 1 و متغیر i برابر 1 قرار میگیرد .

سپس دوباره فریم آغاز میشود و به فریم 25 میرسد که متغیر i میشود 1+1 و پنجره debug عدد 2 را نشان میدهد و تا زمانی که پروژه در حال اجراست عدد 1 به عدد قبلی (متغیرi) اضافه میشود و این عدد در پنجره debug نمایش داده میشود .

(طبعا با اضافه یا کم کردن عدد 25 در onFrame سرعت عمل جمع هم تغییر می کنه این هم به خاطر اینه که اندازه فریم پیمایشی در timeline بیشتر یا کم خواهد شد)

Enter Frame

پروژه جدیدی بسازید و کد های زیر را در آن جایگذاری کنید .

ابتدا از مسیر زیر :

addScript ‣ Events ‣ Frame ‣

onSelfEvent(load) را انتخاب کنید .

در داخل آن بنویسید :

```
i = 0 ;
```

سپس

addScript ‣ Events ‣ Frame ‣ onSelfEvent(enterframe) را انتخاب کنید .

سپس

addScript ‣ Debugging ‣ trace(…) را انتخاب کنید. در داخل پرانتز trace بنویسید :

```
++i ;
```

کد نهایی شما به شکل زیر خواهد بود :

```
onSelfEvent (load){
  i = 0 ;
}
onSelfEvent (enterFrame){
  trace ( ++i )
}
```

سپس پروژه را با کلید ترکیبی Ctr+Enter اجرا کنید .

این بار هم مشاهده می کنید که در هر فریم (EnterFrame) یک عدد به عدد قبلی اضافه می شود .

_root

یعنی ریشه

ترکیب :

```
_root.property
_root.property = value
```

نشانوندها :

property : هر ویژگی که شی موجود در ریشه دارد مثلا برای یک shape این ویژگی ها وجود دارد _x , _y و ...

value : مقدار . به جای value میتونید مقدار جدید رو برای ویژگی تعریف کنید .

مثال :

```
on (release){
  _root.text1.text = "Hello World"
}
```

_parent

تعریف یک شاخه قبل .

ترکیب :

```
_parent.property
_parent.property = value
_parent._parent.property
```

نشانوندها :

property : هر ویژگی که شی موجود در شاخه قبلی دارد .

value : مقدار . میتونید مقدار جدید رو برای ویژگی تعریف کنید .

---

this

اشاره به شیئی که در حال کد نویسی در آن هستیم.

ترکیب :

```
this.property
this.property = value
```

نشانوندها :

property : هر ویژگی که شی که در حال کد نویسی روی آن هستید . مثلا برای یک shape این ویژگی ها وجود دارد _x , _y , ...

value : مقدار . میتونید مقدار جدید رو برای ویژگی تعریف کنید .

---

do...while

ترکیب :

```
do {
 statements;
} while (condition);
```

نشانوندها :

statements : تا وقتی که شرط درست با شد این دستور انجام میشود این دستور حداقل یکبار اجرا میشود .

condition : شرط . این یک عبارت بولی (boolean) است یعنی یا درست است یا غلط (true or false)

مثال :

```
onSelfEvent (load) {
   a = 1 ;
   do {
      trace(a);
   } while (a++ < 5) ;
}
// show 0,1,2,3,4,5 in debug window
```

---

for

ترکیب :

```
for (initial ; condition ; end of loop) {
   statements;
}
```

نشانوندها :

statements : تا وقتی که شرط درست با شد این دستور انجام میشود .

initial : این مورد برای تعریف شرط اولیه آغاز حلقه مورد استفاده قرار دارد . برای افزودن شرط های بیشتر میتوان از کاما برای جدا سازی شرط ها از یکدیگر استفاده کرد .
condition : این یک شرط بولی boolean صحیح یا غلط . (true or false) و حلقه تا زمانی که شرط درست باشد ادامه می یابد .
end of loop : پایان حلقه . دستوری که در پایان حلقه اجرامیشود . به طور معمول برای افزایش شمارنده حلقه مورد استفاده قرار میگیرد(گام حلقه). برای افزودن چند دستور با هم میتوان از کاما برای جدا سازی آنها از هم استفاده کنید.

مثال :

```
for (i = 0; i <= 5 ; i++) {
    trace(i) ;
 }
// show 0,1,2,3,4 in debug window
```

---

while

ترکیب :

```
while (condition) {
  statements;
 }
```

نشانوندها :

statements : تا وقتی که شرط درست باشد این دستور ادمه می یابد.
condition : شرط این یک مقدار بولی است ، درست یا غلط بودن (true or false) را بررسی میکند.

مثال :

```
onSelfEvent (load) {
    a = 1;
    while (a++ < 5) {
       trace(a);
    }
 }
```

for...in

ترکیب :

```
for ( var  in object ){
statement(s) ;
}
```

نشانوند ها :

var : برای نشان دادن هر یک از مقادیر یک شی یا عنصر آرایه یک متغیر (var) اختصاص داده است.
object : نام شی مورد نظر.
Statements : دستوری که در اجرای هر حلقه انجام میشود.

مثال :

```
onFrame( 1 )  {
  n = 1;
  products = new Array();
  products[0] = "SWiSHstudio";
  products[1] = "SWiSHsites";
  products[2] = "SWiSHpix";
  products[3] = "SWiSHmax";
  products[4] = "SWiSH 2.0";
  products[5] = "SWiSHlite";
for  (n in products) {
    trace ("Product" add n add " = " add products[n]);
    n += 1;
  }
}
/* The above script displays the following in the debug window:
Product1 = SWiSHlite
Product2 = SWiSH Max 2.0
Product3 = SWiSH Max
Product4 = SWiSHpix
Product5 = SWiSHsites
Product6 = SWiSHstudio
*/
```

Array

آرایه خانه های پشت سر هم از حافظه .

عنصر آرایه می تواند هر مقدار (به عنوان مثال رشته ای) باشد.ولی تمام عناصریک آرایه باید یک نوع باشند.

برای دسترسی به عناصر معمولا از یک شاخص استفاده میکنم ،شاخصِ ، اولین عنصر آرایه معمولا صفر است .

شاخص هر آرایه بعد از مقدار متغیر در داخل یک براکت [] قرار میگیرد .

براکت دسترسی عملگر ها را به آرایه ها ممکن میسازد .

مثال :

```
onFrame( 1 ) {
 i=0 ;
dayArray = new  Array ;
dayArray[ 0 ] = "Monday" ;
dayArray[ 1 ] = "Tuesday" ;
dayArray[ 2 ] = "Wednesday" ;
dayArray[ 3 ] = "Thursday" ;

while  (i <  3 ) {
    trace(dayArray[i++]);
}
}
/* The above script displays the following in the debug window:
Monday
Tuesday
Wednesday
*/
```

ایجاد آرایه :

آرایه ها را میتوان با هر یک از روش های زیر ساخت :

```
 name  =  new Array;
یا
 name[index] =  "value";
یا
 name  =  new Array(length);
یا
 name  =  new Array(value1, value2...);
```

if

ترکیب :

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

یا

if (condition) {
   statements(s)
} else {
   statements(s)
}

یا

if (condition) {
   statements(s)
} else if (condition) {
   statements(s)
} else {
   statements(s)
}
```

نشانوند ها :

condition : درستی یا نادرستی را بررسی میکند .

statement : وقتی که شرط درست باشد دستور ها انجام میشود .

مثال :

```
onSelfEvent (load) {
    a = 2;
    b = 1;
    if (a > b) {
       trace("a > b");
    }
```

اگر a>b باشد پیغام a>b در debug نمایش داده میشود . در غیر این صورت هیچ پیغامی نمایش داده نمیشود .

```
onSelfEvent (load) {
    a = 5;
    b = 3;
    if (a == b) {
        trace(a add " and " add b add " are equal);
    } else {
        trace(a add " and " add b add " are not equal);
    }
// show “5 and 3 are not equal” in debug window
```

```
onSelfEvent (load) {
    a = 10;
    b = 5;
    c = 5;
    if ((a+b) == c) {
        trace("Values are equal");
    } else if ((a - b) == c) {
        trace(a add " minus " add b add " is equal to " add c);
    } else {
        trace(a add " + " add b add " is not equal to " add c);
    }
 }
// show “10 minus 5 is equal to 5” in debug window
```

---

switch

ترکیب :

```
switch (expression1) {
case expression2 :
statements
break ;
case expression3:
statements
break ;
default :
statements
}
```

نشانوندها :

expression1 : هر عبارتی

expression2 : هر عبارتی

statement : دستوری که باید اجرا شود .

مثال :

```
onSelfEvent (load) {
   i=1
     switch (i) {
     case 1:
      trace("i is equal to 1");
      break;
     case 2:
      trace("i is equal to 2");
      break;
     case 3:
      trace("i is equal to 3");
      break;
      default:
      trace("i is not 1, 2 or 3");
      break;
   }
}
// show “i is equal to 1” in debug window
```

---

case

ترکیب :

```
case expression: statements
```

نشانوندها :

expression : هر عبارتی .

statements : هر دستوری که باید اجرا شود .

case : یک شرط برای انجام دستور switch است .

دستور زمانی اجرا میشود که عبارت برابر switch قرار گیرد .

---

break

هنگامی که این اکشن درون تابع یا یک عبارت شرطی قرار بگیرد و فراخوانی شود باعث پایان کار تابع یا شرط کورد نظر میگردد.

---

default

ترکیب :

```
default : statements
```

نشانوندها :

statements : هر دستوری که باید انجام شود .
وقتی که عبارت مورد نظر switch با هیچ یک از عبارات case برابر نبود در این صورت این دستور انجام میشود .

---

function()

ترکیب :

```
function name ([arg [ : type ] [, arg [ :  type ]]...]) [ : type ] {
   statements;
 }

   --یا--

function ([arg [ : type ] [, arg [ : type  ]]...]) [ : type ] {
   statements;
}
```

name : نام تابع تعریف شده .
arg : نام آرگومان تابع (اختیاری)
type : نوع آرگمان (اختیاری) این مورد نوع آرگمان را مشخص میکند مثل عدد ، رشته ای یا بولی (Number, Boolean or String)
statements : وقتی تابع فراخوانی میشود این دستور اجرا میشود .

اولین روش یک تابع با نام ایجاد میکند شما میتوانید با استفاده از نام ، آن تابع را فراخوانی کنید .
شما میتوانید به روش دوم برای تابع یک مقدار متغیر تعریف کنید و با استفاده ار آن تابع را فراخوانی کنید وحاصل را در متغیر قرار دهید.

مثال 1:

```
onSelfEvent (load) {
      a = 0;
      r = 3;
      a = _root.area(r);
      trace(a);
   }
   function area(p) {
     return  p * p;
   }
//show 9 in debug window
```

مثال 2:

```
onSelfEvent (load) {
      f = function (x) { return x*x; }
      y = f(10); // calls f with 10, returns 100
      trace(y);
   }
//show 100 in debug window
```

مثال 3:

تابعی به اصطلاح برای برگرداندن هَندل آبجکت ها :

```
   onFrame (10) {
      myMovieClip.onRollOver = function () {
        MyTextField.text = "Roll Me Over!";
      };
   }
   onFrame (20) {
      myMovieClip.onRollOver = function () {
        MyTextField.text = "Roll Me Over Again";
      };
   }
```

---

return

ترکیب :

```
   return;
   return expression;
```

expression : عبارتی برگشتی است : عبارت برگشتی میتواند رشته یا عدد باشد . این مقدار اختیاری است . واگر استفاده نشود مقدار null بازگردانده میشود .

این متد به کاربر اجازه میدهد که مقدار تابع تعریف شده اش را بازگرداند .

مثال:

```
function area(p) {
   return  p * p;
}
```

---

properties
methods
events
برخی properties ها شامل :

موقعیت در صفحه نمایش

Screen Position: _x, _y

اندازه

Size : _width, _height

مقیاس

Scale : _xscale, _yscale

زاویه چرخش

rotation angle : _rotation

میزان شفافیت

Opacity : _alpha.

نمایش (قابل دیدن بودن)

Visibility : _visible

نام

Name : _name

(text field) متن ها علاوه بر خصوصیات بالا خصوصیات دیگری نیز دارند:
حد اکثر اسکرول متن

max scroll

اسکرول کردن

scroll

متن

Text

---

methods

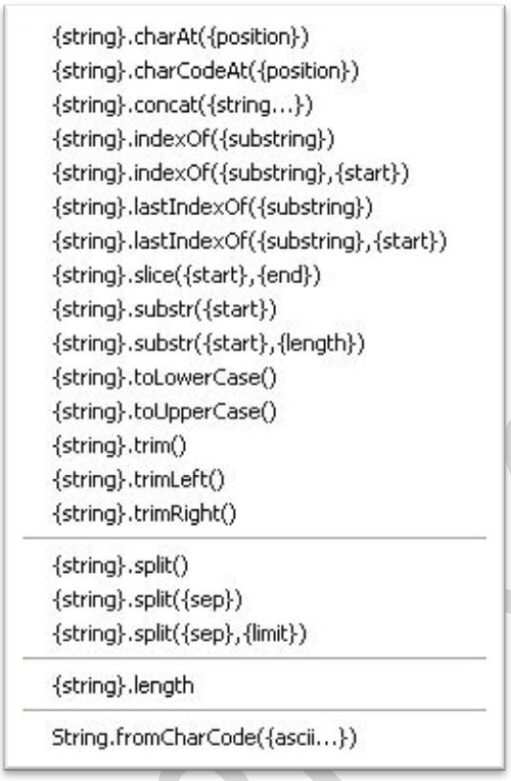
متد تعریف میکند که چه چیز آبجکت ها قابل تغییر است .
آبجکت های مختلف داری متد های مختلف هستند .
به طور مثال برخی متدهای movieclip شامل :
play()

stop()

duplicateMovieClip()

gotoAndPlay()

یا به طور مثال رشته ها دارای متدهای مختلفی هستند که اجازه ایجاد تغیرات در آنها را میسر میسازد:



Event ها یعنی رویداد هایی که برای یک آبجکت تعریف شده اند .
آبجکت های مختلف دارای رویداد های مختلفی هستند .
به طور مثال یک movieclip دارای رویدادهای مختلفی است .
این رویدادها شامل :

onFrame ()
onSelfEvent (load)
onSelfEvent (enterFrame)
onSelfEvent (mouseEvent)

در صفحات قبل این رویدادها مورد بررسی قرار گرفتند .
اما رویداد های دیگر:

---

onSelfEvent(changed)

ترکیب:

```
onSelfEvent (changed)
{
statements;
}
```

نشانوندها:

Statements :کدهایی که در صورت شامل شدن این رویداد اجرا میشوند .
دستورات در این رویداد زمانی اجرا خواهند شد که متن (text) را تغییر بدهیم .
این رویداد فقط برای "this" تعریف شده است .

مثال:

```
onSelfEvent (changed) {
   _yscale += 10;
}
```

---

on(changed)

ترکیب:

```
on (changed) {
statements;
}
```

نشانوند:

statements :کدهایی که در صورت شامل شدن این رویداد اجرا میشوند .

اشیا تعریف شده در اکشن اسکریپت 2 عبارت اند از :

Array

خانه های پشت سر هم حافظه که مقادیر میتوانند در آن قرار بگیرند .

Button

یک دکمه است .قابل افزودن رویداد ها و فانکشن هست .

Color

شیء رنگ .

Date

اجازه دسترسی به زمان محلی و جهانی را میسر میسازد .

Math

توابع ریاضی .

Movie Clip

یک movie clip .

Scripting Object

آبجکتی که تیک گزینه target فعال شده باشد .

Sound

مورد استفاده برای کنترل صدا .

String

رشته ها .

Array

در مطالب قبل روش ایجاد آرایه را آموختیم و دانستیم که چگونه اطلاعاتی در آن قرار دهیم و دریافت کنیم.

اما متد هایی که برای آرایه وجود دارد.

Array.concat()	Array.push()	**Array sorting**
Array.join()	Array.reverse()	Array.splice()
Array.length	Array.shift()	Array.toString()
Array.pop()	Array.slice()	Array.unshift()

Array.concat()

ترکیب:

```
arrayName.concat(value1, value2, ...)
```

نشانوندها:

value : عناصر، اعداد، و یا رشته های مرتبط با عناصر موجود در آرایه .

مثال:

```
onSelfEvent (load) {
    products = new Array("SWiSHlite", "SWiSH Max 2.0");
    more_products = new Array("SWiSH Max", "SWiSHpix");
    trace("Concatenated Array = " add products.concat(more_products));
    trace("Base Array = " add products);
    trace("Base Array with new elements = " add products.concat("Bob", 4));
 }
/* The above script displays the following in the debug window:
Concatenated  Array = SWiSHlite,SWiSH Max  2.0,SWiSH  Max,SWiSHpix
 Base Array = SWiSHlite,SWiSH Max  2.0
 Base Array with new elements =  SWiSHlite,SWiSH Max 2.0,Bob,4
/*
```

مختصر:

- این متد آرایه پایه را با مقادیر جدید پیوند میدهد و یک آرایه جدید میسازد .
- اگر دو آرایه به هم پیوند داده شوند تنها عناصر آن به آرایه پایه اضافه میشود .

که این مثال گویای هر دو مورد بود .

Array.join

ترکیب:

```
arrayName.join({sep});
```

نشانوندها:

sep : همان separator (جداکننده) کاراکتری دلخواه که عناصر آرایه برگشت داده شده را از هم جدا میکند .

بدون این آرگومان عناصر با کاما از یکدیگر جدا میشوند .

مثال 1 :

```
onSelfEvent (load) {
   days = new Array("Monday","Tuesday","Wednesday");
   trace(days.join());
}
/* The above script displays the following in the debug window:
Monday,Tuesday,Wednesday
*/
```

مثال2 :

```
onSelfEvent (load) {
    days = new Array("Monday","Tuesday","Wednesday");
    trace(days.join(":"));
 }
/* The above script displays the following in the debug window:
Monday:Tuesday:Wednesday
*/
```

مثال3 :

```
onSelfEvent (load) {
    pets = new Array("fluffy","spot","Mr.Kitty");
    trace(pets.join(" and "));
 }
/* The above script displays the following in the debug window:
fluffy and spot and Mr.Kitty
*/
```

مختصر:

مقدار بازگشتی ، یک رشته حاوی separator و عناصر به هم پیوسته است .

Array.length

ترکیب:

```
arrayName.length
```

مثال1:

```
onSelfEvent (load) {
    products = new Array();
    products[0] = "SWiSHlite";
    products[1] = "SWiSH Max 2.0";
    trace(products.length);
 }
//show 2 in debug window
```

مثال 2:

```
onSelfEvent (load) {
    products = new Array();
    products[0] = "SWiSHlite";
    products[1] = "SWiSH Max 2.0";
    products[2] = "SWiSH Max";
    products[3] = "SWiSHsites";
```

```
    products[4] = "SWiSHpix";
    trace(products.length);
 }
//show 5 in debug window
```

مثال3:

```
onSelfEvent (load) {
    pets = new Array("fluffy","spot","Mr.Kitty");
    trace(pets.length);
 }
//show 3 in debug window
```

مختصر:

مقدار بازگشتی تعداد عناصر موجود در آرایه است .

Array.pop()

ترکیب :

```
arrayName.pop()
```

مثال1:

```
onSelfEvent (load) {
    pets = new Array("fluffy","spot","Mr.Kitty");
    trace(pets.pop());
 }
//show 'Mr.Kitty'  in debug window
```

از آنجایی که این متد آخرین عنصر آرایه را حذف میکند با اجرای متوالی آن پیغام های متفاوتی میبینید:

مثال2:

```
onSelfEvent (load) {
   zoners = new Array();
   zoners[0] = "David M.";
   zoners[1] = "Hugh B.";
   zoners[2] = "Roger O.";
   zoners[3] = "David P.";
   trace("1st: " add zoners.pop());
   trace("2nd: " add zoners.pop());
   trace("3rd: " add zoners.pop());
```

```
}
/* The above script displays the following in the debug window:
1st: David P.
 2nd: Roger O.
 3rd: Hugh B.
*/
```

مختصر:

حذف آخرین عنصر آرایه و بازگشت مقدار آن .

Array.push()

ترکیب:

```
arrayName.push(value1, value2, ...)
```

نشانوندها:

value : یک یا چند مقدار برای افزودن به آخر آرایه .

مثال:

```
onSelfEvent (load) {
   products = new Array("SWiSHlite", "SWiSH Max 2.0");
   // array length is '2'
   trace("1st Length = " add products.push("SWiSH Max", "SWiSHsites"));
   // new array length of '4' is displayed in the debug window
   trace("2nd Length = " add products.push("SWiSHpix", "SWiSHstudio"));
   // new array length of '6' is displayed in the debug window
}
```

مختصر:

بعد از افزودن مقادیر جدید به پایان آرایه طول آرایه را باز میگرداند.

Array.reverse()

ترکیب:

```
arrayName.reverse();
```

مثال:

```
onSelfEvent (load) {
    days = new Array("Sunday","Monday","Tuesday","Wednesday");
    trace("Initial Array Order = " add days.join());
    days.reverse();
    trace("Reverse Array Order = " add days.join());
  }
/*The above script displays the following in the debug window:
  Initial Array Order = Sunday,Monday,Tuesday,Wednesday
  Reverse Array Order = Wednesday,Tuesday,Monday,Sunday
*/
```

مختصر:

عناصر آرایه را به صورت معکوس باز میگرداند .

---

array.shift()

ترکیب:

```
  arrayName.shift();
```

مثال:

```
  onSelfEvent (load) {
    days = new Array("Sunday","Monday","Tuesday","Wednesday");
    trace(days.shift());
    // displays 'Sunday' in the debug window
    trace(days.shift());
    // displays 'Monday' in the debug window
  }
```

مختصر:

اولین عنصر آرایه را حذف میکند ومقدار آن را باز میگرداند .

Array.slice()

ترکیب:

```
    arrayName.slice(start {, end});
```

start : عددی برای آغاز برش آرایه . اگر یک عدد منفی برای این مقدار قرار دهیم نقطه شروع پایان آرایه خواهد بود برای مثال 1- آخرین عنصر آرایه را نشان میدهد .یا مثلا 2- عنصر قبل از آخرین آرایه را نشان میدهد و همینطور... .

end : مقداری اختیاری است . مقدار پایانی برش را نشان میدهد . اگر این مورد را در برش دخیل کنیم عناصر بازگشتی مقادیر بین star و مقدار end خواهند بود.
اگر یک مقدار منفی قرار گیرد نقطه end از پایان آرایه است . مثلا 1- یک عنصر مانده به آخرین عنصر آرایه را نشان میدهد . 2- دو عنصر مانده به آخرین عنصر آرایه را نشان میدهد و همینطور... .

مثال1:

```
onSelfEvent (load) {
      days = new Array();
      days[0] = "Sunday";
      days[1] = "Monday";
      days[2] = "Tuesday";
      days[3] = "Wednesday";
      days[4] = "Thursday";
      days[5] = "Friday";
      days[6] = "Saturday";
      trace(days.slice(3));
   }
//show “Wednesday,Thursday,Friday,Saturday” in debug window
```

مثال2:

```
 onSelfEvent (load) {
      days = new Array();
      days[0] = "Sunday";
      days[1] = "Monday";
      days[2] = "Tuesday";
      days[3] = "Wednesday";
      days[4] = "Thursday";
      days[5] = "Friday";
      days[6] = "Saturday";
      trace(days.slice(3,-2));
   }
//show “    Wednesday,Thursday” in debug window
```

مختصر:

آرایه جدید ، آرایه ای حاوی تمام عناصر بین start و end است .
مقدار وارد شده برای end شامل برش نمیشود . و همچنین این متد بر آرایه پایه تاثیری ندارد .

Array sorting()

دوروش برای مرتب سازی آرایه ها وجود دارد :

Array.sort() , Array.sortOn()

Array.sort()

ترکیب:

```
arrayName.sort(compareFunction:Object[optional], options:Number [optional]);
```

نشانوندها:

compareFunction : یک پارامتر اختیاری است این دستور اجازه میدهد فانکشنی غیر استاندارد در این متد را دخیل کنیم.
مقادیر بازگشتی این فانکشن باید اینگونه باشد:
مقدار 1- را در مقایسهA<B
مقدار 0 در مقایسهA==B
مقدار 1 در مقایسهA<B

فرم این فانکشن باید اینگونه باشد:

```
fn(a,b)
optional:
```

مثال1:

```
onSelfEvent (load) {
    myNumbers = new Array("5","2","1","3","10");
    myNumbers.sort();
    trace(myNumbers.join());
 }// show 1,10,2,3,5 in debug window
```

مثال2:

```
onSelfEvent (load) {
    team = new Array("John","Brian","Alison","Stephan");
    team.sort();
```

```
    trace(team.join());
 }
// show Alison,Brian,John,Stephan in debug window
```

مثال3:

```
onSelfEvent (load) {
    myNumbers = new Array("one","two","three","five","six");
    myNumbers.sort(order);
    trace(myNumbers.join());
 }
 function order(a,b)
 {
    // sort strings according to length, if same length, sort alphabetically.
    if (a.length < b.length)
    {
       return -1;
    }
    if (a.length > b.length)
    {
       return 1;
    }
    return (a > b);    // alphabetic sort.
 }
// The above script displays the following in the debug window:
// one,six,two,five,three
```

مختصر:

چیدن آرایه به وسیله علامت > کوچکتری . از این متد میتوان برای چیدن آرایه های عددی و الفبایی استفاده کرد.

در این روش آرایه های عددی به صورت الفبایی چیده میشوند مثلا 10 قبل از 5 قرار میگیرد زیرا 1 مقدمتر از 5 است.

Array.sortOn()

ترکیب:

```
arrayName.sortOn(field, options);
```

نشانوندها:

field : حرفی برای شناسایی رشته ای از عنصر آرایه .

optional : مقدار اختیاری

مثال:

```
onSelfEvent (load) {
  person = new Array;
  person.push({first:"Tom", last:"Baker"});
  person.push({first:"Andrew", last:"Smith"});
  person.sorton("first");       // sorts the array by first names
  for (i=0; i<person.length; i++) {
        trace("First: " add person[i].first add " Last:" add person[i].last);
  }
  for (i=0; i<person.length; i++) {
        trace("First: " add person[i].first add " Last:" add person[i].last);
  }
  person.sorton("last");              // sorts the array by last names
 }
/* The above script displays the following in the debug window:
First: Tom Last:Baker
First: Andrew Last:Smith
First: Tom Last:Baker
First: Andrew Last:Smith
*/
```

مختصر:

روش مرتب سازی آرایه.

Array.splice()

ترکیب:

```
arrayName.splice(start,deletecount,{value...});
```

نشانوندها:

start : ارزش عنصری از آرایه که باید از آنجا اتصال انجام گیرد.

deletecount : ارزش عنصری از آرایه که باید حذف گردد . (از جمله عنصر مشخص شده توسط start)

اگر این مقدار برابر صفر قرار گیرد هیچ عنصری حذف نمیشود .

values : مقدار یا مقادیری که باید اضافه گردد .

مثال1:

```
onSelfEvent (load)  {
    nickNames = new Array("Snookie","Piggy","Shorty","Slim");
    nickNames.splice(3,0,"Tiny");
    trace(nickNames.join());
```

```
 }
/*The above script displays the following in the debug window:
Snookie,Piggy,Shorty,Tiny,Slim
*/
```

مثال 2:

```
onSelfEvent (load)  {
    nickNames = new Array("Snookie","Piggy","Shorty","Slim");
    nickNames.splice(3,1);
    trace(nickNames.join());
 }
/*The above script displays the following in the debug window:
Snookie,Piggy,Shorty
*/
```

مختصر:

افزودن یا حذف عنصری از آرایه.

Array.toString()

ترکیب:

```
arrayName.toString();
```

مثال:

```
onSelfEvent (load)  {
 days  = new Array("Sunday","Monday","Tuesday","Wednesday");
 trace(days.toString());
 }
// The above script displays the following in the debug window:
// Sunday,Monday,Tuesday,Wednesday
```

مختصر:

خروجی ، رشته ی ، تک تکه ، عناصر آرایه است .

Array.unshift()

ترکیب:

```
arrayName.unshift(value1, value2, ...);
```

نشانوند:

value : مقدار یامقادیری برای افزودن به اول آرایه.

مثال:

```
onSelfEvent (load) {
  days = new Array("Wednesday","Thursday","Friday","Saturday");
  trace(days.length);
  // displays '4' in the debug window
  week = days.unshift("Sunday","Monday","Tuesday");
  trace(week);
  // displays '7' in the debug window
}
//show 4 ,7 in debug window
```

مختصر:

افزودن مقادیر جدید به اول آرایه و بعد از آن بازگشت تعداد عناصر آرایه .

Button

ابتدا رویدادهای این آبجکت رابررسی میکنیم :

dragOut

dragOver

keyPress(key)

press

release

releaseOutside

rollOut

rollOver

دستورات press و release قبلا توضیح داده شده اند .

---

dragOut

مثال:

دکمه ای به نام Button_1 بسازید و کد زیر را در داخل آن جاگذاری کنید :

```
onSelfEvent (dragOut) {
   _xscale -= 10;
   _yscale -= 10;
}
```

سپس پروژه را اجرا کنید .
ماوس را بر روی Button_1 ببرید و حالت Press (کلید چپ ماوس را نگه دارید) حال در همان حالت سعی کنید نشانگر ماوس را به خارج از دکمه حرکت دهید مشاهده میکنید که اندازه دکمه با این کار کوچک میشود .

---

dragOver

مثال:

دکمه ای به نام Button_2 بسازید و کدهای زیر را در داخل آن جاگذاری کنید :

```
onSelfEvent (dragOut) {
   _xscale -= 10;
   _yscale -= 10;
}
onSelfEvent (dragOver) {
   _xscale += 10;
   _yscale += 10;
}
```

سپس پروژه را اجرا کنید .
ماوس را بر روی Button_2 ببرید و حالتPress (کلید چپ ماوس را نگه دارید) حال در همان حالت سعی کنید نشانگر ماوس را به خارج از دکمه حرکت دهید مشاهده میکنید که اندازه دکمه با این کار

کوچک میشود. و در همان حالت Press دوباره نشانگر ماوس را به داخل آبجکت حرکت دهید با این کار دکمه بزرگتر میشود .

---

keyPress(key)

این رویداد برای زمانی است که یک کلید از صفحه کلید را برای کار خود تعریف کنید.
برای مثال برای تعریف کلید a آن را میان " " قرار میدهیم :

```
on (keyPress("a")) {
دستور
    }
```

نکته:

نمیتوانیم برای همه آبجکت ها یک کلید تعریف کنیم (مثل تعریف کلید b برای دو آبجکت)
به صورت پیشفرض این دستور برای آبجکتی تعریف میشود که در تب (Tab) اول تری نسبت به سایر آبجکت ها قرار گرفته باشد.

مثال1:

```
onSelfEvent (keyPress("a")) {
    _xscale += 10;
    _yscale += 10;
}
```

تعریف دو کلید برای یک آبجکت :

مثال2:

```
onSelfEvent (keyPress("a")) {
   _xscale += 10;
   _yscale += 10;
}

onSelfEvent (keyPress("<Tab>")) {
   _xscale += 10;
   _yscale += 10;
}
```

---

releaseOutside

رویدادی که وقتی روی آبجکت مورد نظر حالت press را انجام میدهیم ولی در خارج از محیط آن آبجکت ، کلید ماوس را رها میکنیم. (releaseoutside)

مثال:

```
onSelfEvent (releaseOutside) {
    _xscale += 10;
    _yscale += 10;
}
```

---

rollover

حالتی که نشانگر ماوس را بر روی آبجکت مورد نظر میبریم.

مثال:

```
onSelfEvent (rollOut) {
    _xscale += 10;
    _yscale += 10;
}
```

---

rollOut

حالتی که نشانگر ماوس را از روی آبجکت به خارج ازآن حرکت میدهیم.

مثال:

```
onSelfEvent (rollOver) {
    _xscale += 10;
    _yscale += 10;
}
```

---

enabled

ترکیب:

```
    buttonName.enabled
```

مثال:

```
onFrame (10) {
   button1.enabled = false; // disables the specified
```

```
   button at Frame 10
   }
onFrame (20) {
   button1.enabled = true; // re-enables the specified
   button at Frame 20
   }
```

مختصر:

این متد یک مقدار بولی است که وضعیت فعال بودن یا غیر فعال بودن دکمه را تعیین میکند .

---

tabEnabled

ترکیب:

```
   buttonName.tabEnabled
```

مثال:

```
   onSelfEvent (load) {
      button1.tabEnabled = false;
   }
```

مختصر:

به صورت پیش فرض با کلید tab نیز میتوانیم دکمه ها را انتخاب کنیم با این متد این قابلیت غیر فعال میشود.

---

tabIndex

ترکیب:

```
      buttonName.tabIndex
```

مثال:

```
onSelfEvent (load) {
      button1.tabIndex = 1;
      button2.tabIndex = 2;
      button3.tabIndex = 3;
      myMovieClip1.tabIndex = 4;
      myTextField1.tabIndex = 5;
```

```
}
```

مختصر:

چیدمان tab آبجکت ها را تعریف میکند.

---

useHandCursor

ترکیب:

```
buttonName.useHandCursor
```

مثال:

```
onSelfEvent (load) {
      buttonName.useHandCursor = false ;
}
  // uses the standard mouse pointer when over the named  button
onFrame(10) {
      buttonName.useHandCursor = true ;
  }
 // displays the hand cursor when over the named button
```

مختصر:

وقتی مقدار برابر false قرار بگیرد هنگامی که نشانگر ماوس را بر روی دکمه مورد نظر میبرید دیگر کروسر ماوس به شکل دست ظاهر نمیشود.

---

trackAsMenu

ترکیب:

```
buttonName.trackAsMenu
```

مختصر:

این مقدار میتواند برابر true یا false قرار بگیرد در حالت true اگر در stage حالت press

را انجام دهید و سپس نشانگر ماوس را بر روی دکمه مورد نظر ببرید و عمل release را انجام دهید فرمان موجود در حالت release انجام میشود.

---

color

با color میتوانید برای آبجکتها رنگ (RGB) تعیین کنید.

new Color()	setRGB()	getRGB()

ترکیب:

```
new Color(target);
```

نشانوندها:

target : آبجکت (هدف) مورد نظر .

مثال:

```
myColor = new Color(_parent.myMovieClip);
```

مختصر:

ایجاد یک شی رنگ جدید است که با آن بتوان رنگ هدف را تغییر داد.

---

setRGB()

ترکیب:

```
colorObject.setRGB(0xRRGGBB);
```

نشانوند:

0xRRGGBB : رنگ بر مبنای هگزادسیمالred/green/blue

0x نشاندهنده این است که این یک مقدار هگزادسیمال

مثال:

```
onSelfEvent (load) {
    myColor = new Color(myMovieClip);
    myColor.setRGB(0xFF6600);
}
```

red=255

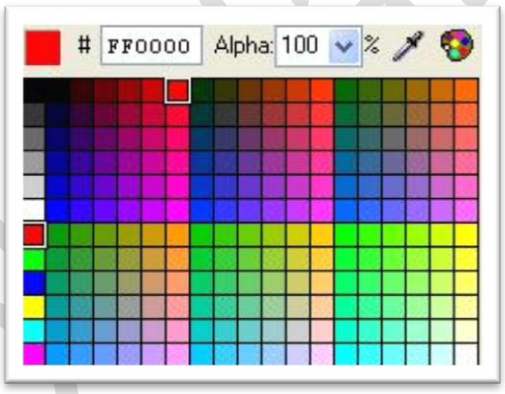
green=102

blue=0

رنگ حاصل شده = نارنجی

شما میتوانید این کد ها را در خود نرم افزار تولید کنید.

به طور مثال تصویر زیر کد رنگ قرمز را نشان میدهد که برابرff0000



مختصر:

تعریف رنگ RGB برای آبجکت هدف .

---

getRGB()

ترکیب:

```
colorObject.getRGB();
```

مثال:

```
onSelfEvent (load) {
    myColor = new Color(myMovieClip);
    myColor.setRGB(0xFF6600);
    trace(myColor.getRGB())
  }
```

مختصر:

مقدار عددی رنگ ساخته شده در آبجکت که با متد setRGB مشخص شده را بازمیگرداند .

Date

این شی به شما اجازه میدهد تا به خواص تاریخ و ساعت دسترسی پیدا کنید .
با استفاده از new Date شما میتوانید به تاریخ محلی و جهانی دسترسی پیدا کنید یا یک تاریخ منحصر بفرد برای خود ایجاد کنید .

---

localTime

زمان محلی بر اساس زمان سیستمی است که فلش پلیر در آن اجرا میشود.

متدهای مختلف date را در جدول زیر مشاهده میکنید:

new Date()	Date.getUTCFullYear()	Date.setSeconds()
Date.getDate()	Date.getUTCHours()	Date.setTime()
Date.getDay()	Date.getUTCMilliseconds()	Date.setUTCDate()
Date.getFullYear()	Date.getUTCMinutes()	Date.setUTCFullYear()
Date.getHours()	Date.getUTCMonth()	Date.setUTCHours()
Date.getMilliseconds()	Date.getUTCSeconds()	Date.setUTCMilliseconds()
Date.getMinutes()	Date.getYear()	Date.setUTCMinutes()
Date.getMonth()	Date.setDate()	Date.setUTCMonth()
Date.getSeconds()	Date.setFullYear()	Date.setUTCSeconds()
Date.getTime()	Date.setHours()	Date.setYear()
Date.getTimezoneOffset()	Date.setMilliseconds()	Date.toString()
Date.getUTCDate()	Date.setMinutes()	
Date.getUTCDay()	Date.setMonth()	

---

new Date()

ترکیب:

```
DateObj = new Date(year, month {, date {, hour {, min {, sec {, ms }}}}});
```

نشانوندها:

year : عددی چهار رقمی برای تعیین سال.
month : عددی از 0 تا 11 به نمایندگی از ماه ها (0 ژانویه ، 1 فوریه و...)
date: عددی از 1 تا 31 برای نشان دادن روزهای ماه . (این یک مقدار اختیاری است)
hour : عددی از 0 تا 23 برای نشان دادن ساعت (این یک مقدار اختیاری است)
min : عددی از 0 تا 59 برای نشان دادن دقیقه (این یک مقدار اختیاری است)
Sec : عددی از 0 تا 59 برای نشان دادن ثانیه (این یک مقدار اختیاری است)
ms : عددی از 0 تا 999 برای نشان دادن میلی ثانیه (این یک مقدار اختیاری است)

مختصر:

مقادیر بازگشتی به صورت integer است .
این متد برای ایجاد یک تاریخ جدید است .
میتوان از آن برای ایجاد زمان محلی استفاده کرد .
یا میتوان یک تاریخ مرجع برای دسترسی ایجاد کرد .

---

getDate()

ترکیب:

```
DateObj.getDate()
```

مثال:

```
onSelfEvent (load) {
  theDate = new Date();
  trace(theDate.getDate());
}
```

مختصر:

بازگشت روز از ماه .(بر اساس زمان محلی)

---

getDay()

ترکیب :

```
DateObj.getDay()
```

مثال:

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getDay());
}
```

مختصر:

دریافت روز هفته.(بر اساس زمان محلی)

---

getFullYear()

ترکیب:

```
DateObj.getFullYear()
```

مثال:

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getFullYear());
}
```

مختصر:

دریافت سال (بر اساس زمان محلی)

---

getHours()

ترکیب:

```
DateObj.getHours()
```

مثال:

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getHours());
}
```

مختصر:

دریافت ساعت (بر اساس زمان محلی)

---

getMilliseconds()

ترکیب:

```
DateObj.getMilliseconds()
```

مثال:

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getMilliseconds());
}
```

مختصر:

دریافت میلی ثانیه (بر اساس زمان محلی)

---

getMinutes()

ترکیب:

```
DateObj.getMinutes()
```

مثال:

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getMinutes());
}
```

مختصر:

دریافت دقیقه (بر اساس زمان محلی)

---

getMonth()

ترکیب:

```
DateObj.getMonth()
```

مثال:

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getMonth());
}
```

مختصر:

دریافت ماه (بر اساس زمان محلی)

---

getSeconds()

ترکیب:

```
DateObj.getSeconds()
```

مثال:

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getSeconds());
}
```

مختصر:

دریافت ثانیه (بر اساس زمان محلی)

---

getTime()

ترکیب:

```
DateObj.getTime()
```

مثال:

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getTime());
}
```

مختصر:

دریافت میلی ثانیه گذشته از زمان جهانی (نیمه شب ژانویه، 1، 1970)

---

getTimezoneOffset

ترکیب:

```
DateObj.getTimezoneOffset()
```

مثال:

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getTimezoneOffset());
}
```

مختصر:

نمایش تفاوت زمان محلی و زمان جهانی.

---

getUTCDate()

ترکیب:

```
DateObj.getUTCDate()
```

مثال:

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getUTCDate());
}
```

مختصر:

عددی از 1 تا 31 نمایش روز جاری ماه با توجه به زمان جهانی.

---

getUTCDay()

ترکیب:

```
DateObj.getUTCDay()
```

مثال:

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getUTCDay());
}
```

مختصر:

عددی از 0 تا 6 نمایش روز جاری هفته با توجه به زمان جهانی.

---

getUTCFullYear()

ترکیب:

```
DateObj.getUTCFullYear()
```

مثال:

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getUTCFullYear());
}
```

مختصر:

نمایش عددی 4 رقمی به نمایندگی سال جاری با توجه به زمان جهانی .

---

getUTCHours()

ترکیب :

```
DateObj.getUTCHours()
```

مثال :

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getUTCHours());
}
```

مختصر :

عددی از 0 تا 23 نمایش ساعت جاری با توجه به زمان جهانی .

---

getUTCMilliseconds()

ترکیب :

```
DateObj.getUTCMilliseconds()
```

مثال :

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getUTCMilliseconds());
}
```

مختصر :

عددی از 0 تا 999 نمایش میلی ثانیه جاری با توجه به زمان جهانی .

---

getUTCMinutes()

ترکیب :

```
DateObj.getUTCMinutes()
```

مثال :

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getUTCMinutes());
}
```

مختصر:

عددی از 0 تا 59 نمایش دقیه در ساعت جاری با توجه به زمان جهانی .

---

getUTCMonth()

ترکیب:

```
DateObj.getUTCMonth()
```

مثال:

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getUTCMonth());
}
```

مختصر:

عددی از 0 تا 12 نمایش ماه جاری با توجه به زمان جهانی .

---

getUTCSeconds()

ترکیب :

```
DateObj.getUTCSeconds()
```

مثال :

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getUTCSeconds());
}
```

مختصر :

عددی از 0 تا 59 نمایش ثانیه در دقیقه جاری با توجه به زمان جهانی .

---

getYear()

ترکیب :

```
DateObj.getYear()
```

مثال :

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.getYear());
}
```

مختصر:

نمایش سال به صورت عدد چهار رقمی بر اساس زمان محلی .

---

setDate()

ترکیب:

```
DateObj.setDate(date)
```

نشانوند:

date : مقدار integer از 1 تا 31

مثال:

```
onFrame (1) {
  theDate = new Date();
  theDate.setDate()
}
```

مختصر:

روز از ماه را بر اساس زمان محلی قرار میدهد .
(توجه کنید زمان رایانه شما تغییر نمیکند بلکه این کار داخل فلش انجام میشود)

---

setFullYear()

ترکیب:

```
DateObj.setFullYear(year {,month{,date}})
```

نشانوند:

year: عدد چهار رقمی برای مشخص کردن سال
month: عددی از 0 تا 11 برای نمایش ماه
date: عددی از 1 تا 31 برای نمایش روز از ماه

مثال:

```
onFrame (1) {
  theDate = new Date();
 theDate. setFullYear(1999,7,11);
 trace(theDate.getYear()) ;
 trace(theDate.getMonth()) ;
 trace(theDate.getDate());
}
```

مختصر:

قرار دادن سال برای شی Data . تمام مقدار های year,month,date بر اساس زمان محلی است.

---

setHours()

ترکیب:

```
DateObj.setHours(hour)
```

نشانوند:

hour : عددی از 0 تا 23

مثال:

```
onFrame (1){
  theDate = new Date();
  theDate.setHours(3);
  trace(theDate.getHours());
}
```

مختصر:

قرار دادن ساعت برای شی Data بر اساس زمان محلی است.

---

setMilliseconds()

ترکیب:

```
DateObj.setMilliseconds(ms)
```

نشانوند:

ms : عددی از 0 تا 999

مثال:

```
onFrame (1){
   theDate = new Date();
   theDate.setMilliseconds(3);
   trace(theDate.getMilliseconds());
}
```

مختصر:

قرار دادن میلی ثانیه برای شی Data بر اساس زمان محلی است.

---

setMinutes()

ترکیب :

```
DateObj.setMinutes(min)
```

نشانوند :

min : عددی از 0 تا 59

مثال :

```
onFrame (1){
  theDate = new Date();
 theDate.setMinutes(3)
  trace(theDate.getMinutes())
}
```

مختصر :

قرار دادن دقیقه برای آبجکت Data بر اساس زمان محلی است .

---

setMonth()

ترکیب :

```
DateObj.setMonth(month {, date})
```

نشانوند :

mont: عددی از 0 تا 11

date: عددی از 1تا 31

مثال :

```
onFrame (1) {
```

```
  theDate = new Date();
  theDate.setMonth(3)
  trace(theDate.getMonth())
}
```

مختصر :

قرار دادن ماه برای آبجکت Data بر اساس زمان محلی است .

---

setSeconds()

ترکیب :

```
DateObj.setSeconds(sec)
```

نشانوند :

sec: عددی از 0 تا 59

مثال :

```
onFrame (1) {
  theDate = new Date();
  theDate.setSeconds(3);
  trace(theDate.getSeconds());
}
```

مختصر :

قرار دادن ثانیه برای آبجکت Data بر اساس زمان محلی است .

---

setTime()

ترکیب :

```
DateObj.setTime(ms)
```

نشانوند :

ms : از نوع عددی . استفاده از0 برای نمایش 0:00 GMT 1970 Jan 1

مثال :

```
onFrame (1) {
  theDate = new Date();
  theDate.setTime(0);
  trace(theDate.getFullYear());
  trace(theDate.getMonth());
  trace(theDate.getDate());
}
```

مختصر :

قرار دادن زمان در 12:00am January 1, 1970 برای آبجکت Data بر اساس زمان محلی است .
مقدار بازگشتی به میلی ثانیه است .

SetUTCDate()

ترکیب :

```
DateObj.setUTCDate(date)
```

نشانوند :

date : مقدار integer از 1 تا 31

مثال :

```
onFrame (1) {
  theDate = new Date();
  theDate.setUTCDate()
}
```

مختصر :

روز از ماه را بر اساس زمان جهانی قرار میدهد .
(توجه کنید زمان رایانه شما تغییر نمیکند بلکه این کار داخل فلش انجام میشود)

---

setUTCFullYear()

ترکیب :

```
DateObj.setUTCFullYear(year {,month{,date}})
```

نشانوند :

year : عدد چهار رقمی برای مشخص کردن سال
month : عددی از 0 تا 11 برای نمایش ماه
date : عددی از 1 تا 31 برای نمایش روز از ماه

مثال :

```
onFrame (1) {
  theDate = new Date();
 theDate. setUTCFullYear(1999,7,11);
 trace(theDate.getUTCYear()) ;
 trace(theDate.getUTCMonth()) ;
 trace(theDate.getUTCDate());
}
```

مختصر :

قرار دادن سال برای آبجکت Data . تمام مقدار های year,month,date بر اساس زمان جهانی است.

---

setUTCHours()

ترکیب :

```
DateObj.setUTCHours(hour)
```

نشانوند :

hour : عددی از 0 تا 23

مثال:

```
onFrame (1){
   theDate = new Date();
   theDate.setUTCHours(3);
   trace(theDate.getUTCHours());
```

```
}
```

مختصر:

قرار دادن ساعت برای آبجکت Data بر اساس زمان جهانی است.

---

setUTCMilliseconds()

ترکیب:

```
DateObj.setUTCMilliseconds(ms)
```

نشانوند :

ms : عددی از 0 تا 999

مثال :

```
onFrame (1){
  theDate = new Date();
  theDate.setUTCMilliseconds(3);
  trace(theDate.getUTCMilliseconds());
}
```

مختصر:

قرار دادن میلی ثانیه برای آبجکت Data بر اساس زمان جهانی است .

---

setUTCMinutes()

ترکیب:

```
DateObj.setUTCMinutes(min)
```

نشانوند:

min : عددی ار 0 تا 59

مثال :

```
onFrame (1){
  theDate = new Date();
 theDate.setUTCMinutes(3)
  trace(theDate.getUTCMinutes())
}
```

مختصر:

قرار دادن دقیقه برای آبجکت Data بر اساس زمان جهانی است.

---

setUTCMonth()

ترکیب :

```
DateObj.setUTCMonth(month {, date})
```

نشانوند :

mont : عددی از 0 تا 11

date : عددی از 1تا 31

مثال :

```
onFrame (1) {
  theDate = new Date();
  theDate.setUTCMonth(3)
  trace(theDate.getUTCMonth())
}
```

مختصر :

قرار دادن ماه برای آبجکت Data بر اساس زمان جهانی است.

---

setUTCSeconds()

ترکیب :

```
DateObj.setUTCSeconds(sec)
```

نشانوند :

sec : عددی از 0 تا 59

مثال :

```
onFrame (1) {
  theDate = new Date();
  theDate.setUTCSeconds(3);
  trace(theDate.getUTCSeconds());
}
```

مختصر:

قرار دادن ثانیه برای آبجکت Data بر اساس زمان جهانی است.

---

setYear()

ترکیب :

```
DateObj.setYear(year)
```

نشانوند :

year : عددی ازبین 1 تا 99

مثال :

```
onFrame (1) {
  theDate = new Date();
  theDate.setYear(0);
  trace(theDate.getFullYear());
  theDate.setYear(52);
  trace(theDate.getFullYear());
  theDate.setYear(99);
  trace(theDate.getFullYear());
}
```

مختصر :

قرار دادن سال برای آبجکت Data بر اساس زمان محلی است . (قرار دادن سال از 1900 تا 1999)

---

toString()

ترکیب :

```
DateObj.toString()
```

مثال :

```
onFrame (1) {
  theDate = new Date();
  trace(theDate.toString());
}
```

مختصر :

بازگشت مقدار رشته ای تاریخ آبجکت Data

Math

از طریق این آبجکت شما میتوانید به توابع گوناگون ریاضی دست یابید .
متدهای این آبجکت را در جدول زیر مشاهده میکنید .

Math.abs(number)	Math.clamp(number,lo,hi)	Math.random()
Math.acos(number)	Math.cos(radians)	Math.randomInt(max)
Math.acosdeg(number)	Math.cosdeg(degrees)	Math.randomRange(lo,hi)
Math.approach(number,dest,factor)	Math.degrees(radians)	Math.round(number)
Math.asin(number)	Math.exp(number)	Math.sign(number)
Math.asindeg(number)	Math.floor(number)	Math.sin(radians)
Math.atan(number)	Math.log(number)	Math.sindeg(degrees)
Math.atandeg(number)	Math.max(number,number...)	Math.sqrt(number)
Math.atan2(y,x)	Math.min(number,number...)	Math.SQRT1_2
Math.atan2deg(y,x)	Math.Pi	Math.SQRT2
Math.ceil(number)	Math.pow(base,power)	Math.tan(radians)
Math.chance(percent)	Math.radians(degrees)	Math.tandeg(degrees)

Math.abs(number)

ترکیب :

```
Math.abs(x)
```

نشانوند :

x : یک عدد

مثال :

```
onFrame (1) {
  trace(Math.abs(3.4));
  trace(Math.abs(-3.4));
}
```

مختصر:

محاسبه قدر مطلق x

Math.acos(number)

ترکیب :

```
Math.acos(x)
```

نشانوند :

x: یک عدد (-1<= x <=1)

مثال :

```
onFrame (1) {
  trace(Math.acos(0.5));
}
```

مختصر :

مقدار arc cos (x) به واحد رادیان

Math.acosdeg(number) +

ترکیب :

```
Math.acosdeg(x)
```

نشانوند :

x: یک عدد (1 <= x <= -1)

مثال :

```
onFrame (1) {
  trace(Math.acosdeg(0.5));
}
```

مختصر:

مقدار arc cos (x) به واحد درجه

Math.approach(number,dest,factor) +

ترکیب :

```
Math.approach(number, dest, factor)
```

نشانوند :

همه به صورت عددی هستند

Number : موقیت کنونی (x یا y)

dest : میل به (x یاy)

factor : میزان نزدیک شدن به dest

مثال :

```
onSelfEvent (enterFrame) {
   this._X = Math.approach(this._X, 50, 0.95);
    trace(this._X)
```

```
}
```

مختصر:

حرکت آبجکت به موقعیت جدید (dest)

Math.asin(number)

ترکیب :

```
Math.asin(x)
```

نشانوند :

x: یک عدد (1 => x => -1)

مثال :

```
onSelfEvent (load) {
   trace(Math.asin(0.5))
}
```

مختصر:

مقدار arc sin (x) به واحد رادیان

Math.asindeg(number) +

ترکیب :

```
Math.asindeg(x)
```

نشانوند :

x : یک عدد (1 => x => -1)

مثال :

```
onSelfEvent (load) {
   trace(Math.asindeg(0.5))
}
```

مختصر:

مقدار arc sin (x) به واحد درجه

Math.atan(number)

ترکیب :

```
Math.atan(x)
```

نشانوند :

x : یک عدد

مثال :

```
onSelfEvent (load) {
   trace(Math.atan(1))
}
```

مختصر :

مقدار arc tan (x) به واحد رادیان

Math.atandeg(number) +

ترکیب :

```
Math.atandeg(x)
```

نشانوند :

x : یک عدد

مثال :

```
onSelfEvent (load) {
   trace(Math.atandeg(1))
}
```

مختصر :

مقدار arc tan (x) به واحد درجه

Math.atan2(y,x)

ترکیب :

```
Math.atan2(y,x)
```

نشانوند :

y,x : یک عدد

مثال :

```
onSelfEvent (load) {
   trace(Math.atan2(1,1))
   trace(Math.atan2(1,0))
}
```

مختصر :

مقدار arc tan (y/x) به واحد رادیان

Math.atan2deg(y,x) +

ترکیب :

```
Math.atan2deg(y,x)
```

نشانوند :

x,y : یک عدد

مثال :

```
onSelfEvent (load) {
  trace(Math.atan2deg(1,1));
  trace(Math.atan2deg(1,0));
}
```

مختصر:

مقدار arc tan (y/x) به واحد درجه

Math.ceil(number)

ترکیب :

```
Math.ceil(x)
```

نشانوند :

x : یک عدد

مثال :

```
onSelfEvent (load) {
  trace(Math.ceil(4.2));
  trace(Math.ceil(-3.8));
}
```

مختصر:

گرد شده عدد x (مقدار بازگشتی همیشه به عدد بزرگتر گرایش دارد)

Math.chance(percent) +

ترکیب :

```
Math.chance(percent)
```

نشانوند :

percent : میزان شانش نمایش true or false نمایش صحیح و غلط. (0 تا 100)

مثال :

```
onSelfEvent (enterFrame) {
  trace(Math.chance(50));
}
```

مختصر :

مقدار بازگشتی بر حسب اتفاق ، true یا false خواهد بود هر چه مقدار percent نزدیکتر به 0 باشد امکان نمایش true کمتر میشود و هرچه به 100 نزدیک باشد امکان نمایش false کاهش میابد .

Math.clamp(number,lo,hi) +

ترکیب :

```
Math.clamp(number, lo, hi)
```

نشانوند :

number : مقداری برای مقایسه میانه

lo : کمترین عدد میانه

hi : بیشترین عدد میانه

مثال:

```
onSelfEvent (load) {
   trace(Math.clamp(-5,2,15));
   trace(Math.clamp(4.6,2,15));
   trace(Math.clamp(103,2,15));
}
```

مختصر:

برگرداندن مقدار میانی

Math.cos(radians)

ترکیب :

```
Math.cos(x)
```

نشانوند :

x : یک عدد

مثال :

```
onSelfEvent (load) {
  trace(Math.cos(0));
}
```

مختصر :

محاسبه درجه بر حسب cos رادیان x

Math.cosdeg(degrees) +

ترکیب :

```
Math.cosdeg(x)
```

نشانوند :

x : یک عدد

مثال :

```
onSelfEvent (load) {
  trace(Math.cosdeg(90));
}
```

مختصر :

محاسبه رادیان بر حسب cos درجه x

Math.degrees(radians) +

ترکیب :

```
Math.degrees(x)
```

نشانوند :

x : یک عدد (طبق رادیان)

مثال :

```
onSelfEvent (load) {
  trace(Math.degrees(3.14159265358979));
}
```

مختصر:

محاسبه درجه بر طبق رادیان از فرمول زیر:

degrees = radians * 360 / (2 * pi).

Math.distance(x1,y1,x2,y2) +

ترکیب :

```
Math.distance(x1,y1,x2,y2)
```

نشانوند :

x1,y1,x2,y2 : اعدادی از دو نقطه

مثال :

```
onSelfEvent (load) {
  trace(Math.distance(1,1,10,10));
  trace(Math.distance(3,5,30,40));
}
```

مختصر:

محاسبه فاصله دو نقطه (x1,y1) و (x2,y2)

Math.distanceSq(x1,y1,x2,y2) +

ترکیب :

```
Math.distanceSq(x1,y1,x2,y2)
```

نشانوند :

x1,y1,x2,y2 : اعدادی از دونقطه

مثال :

```
onSelfEvent (load) {
  trace(Math.distanceSq(1,1,10,10));
  trace(Math.distanceSq(3,5,30,40));
}
```

مختصر :

محاسبه مربع فاصله دو نقطه (x1,y1) و (x2,y2)

Math.exp(number)

ترکیب :

```
Math.exp(x)
```

نشانوند :

x : یک عدد

مثال :

```
onSelfEvent (load) {
  trace(Math.exp(2));
}
```

مختصر:

محاسبه e (2.71828) با توان x

Math.floor(number)

ترکیب :

```
Math.floor(x)
```

نشانوند :

x : یک عدد

مثال :

```
onSelfEvent (load) {
   trace(Math.floor(4.8));
   trace(Math.floor(-3.2));
}
```

مختصر :

گرد شده عدد x (مقدار بازگشتی همیشه به عدد کوچکتر گرایش داره)

Math.log(number)

ترکیب :

```
Math.log(x)
```

نشانوند :

x : یک عدد بزرگتر از 0

مثال :

```
onSelfEvent (load) {
   trace(Math.log(2));
   trace(Math.log(100)/Math.log(10));
}
```

مختصر:

محاسبه لوگاریتم x

Math.log10(x) +

ترکیب :

```
Math.log(x)
```

نشانوند :

x : یک عدد بزرگتر از 0

مثال :

```
onSelfEvent (load) {
  trace(Math.log10(2));
  trace(Math.log10(100));
}
```

مختصر:

محاسبه لوگاریتم x بر مبنای 10

Math.max(number,number...) +

ترکیب :

```
Math.max(n1, n2[, n3 ....])
```

نشانوند :

n1 : یک عدد

n2 : یک عدد

n3 : لیستی از اعداد

مثال :

```
onSelfEvent (load) {
  trace(Math.max(1,4,-3));
}
```

مختصر :

نمایش بزرگترین مقدار یک مجموعه

Math.min(number,number...) +

ترکیب :

```
Math.min(n1, n2[, n3 ....])
```

نشانوند :

n1 : یک عدد

n2 : یک عدد

n3 : لیستی از اعداد

مثال :

```
onSelfEvent (load) {
  trace(Math.min(1,4,-3));
}
```

مختصر :

نمایش کوچکترین مقدار یک مجموعه

Math.PI

ترکیب :

```
Math.PI
```

مختصر :

نمایش عدد پی

Math.pow(base,power)

ترکیب :

```
Math.pow(base,power)
```

نشانوند :

base : پایه

power : توان

مثال :

```
onSelfEvent (load) {
  trace(Math.pow(3,2)); // returns 9 (3 * 3)
  trace(Math.pow(2,3));  // returns 8 (2 * 2 * 2)
  trace(Math.pow(-2,3));// returns -8
  trace(Math.pow(2,-1));// returns 0.5 (1/2)
  trace(Math.pow(2,0.5));// returns 1.414213... (square root of 2)
  trace(Math.pow(2,-0.5)); // reutrns 0.707... (1/(square root of 2)
  trace(Math.pow(-2,0.5));  // returns 0. (Error condition, base <0, non integer power)
}
```

مختصر :

محاسبه عدد base به توان (power)

Math.radians(degrees) +

ترکیب :

```
Math.radians(x)
```

نشانوند :

x : یک عدد (طبق درجه)

مثال :

```
onSelfEvent (load) {
  trace(Math.radians(360));
}
```

مختصر:

محاسبه رادیان بر حسب درجه طبق فرمول زیر:

radians = degrees * 2 * pi / 360

Math.random()

ترکیب :

```
Math.random()
```

مثال :

```
onSelfEvent (load) {
  trace(Math.random());
}
```

مختصر :

انتخاب یک عدد به صورت تصادفی (0 <= n < 1)

Math.random()

ترکیب :

```
Math.random()
```

مثال :

```
onSelfEvent (load) {
  trace(Math.random());
}
```

مختصر :

انتخاب یک عدد به صورت تصادفی (0 <= n < 1)

Math.randomInt(max) +

ترکیب :

```
Math.randomInt(max)
```

نشانوند :

max : عدد صحیح (بزرگتر از 0)

مثال :

```
onSelfEvent (load) {
  Math.randomInt(5);
}
```

مختصر :

انتخاب یک عددصحیح به صورت تصادفی از 0 تا max

Math.randomRange(lo,hi) +

ترکیب :

```
Math.randomRange(lo, hi)
```

نشانوند :

lo: کوچکترین عدد در محدوده مورد نظر

hi: بزرگترین عدد در محدوده مورد نظر

مثال :

```
onSelfEvent (load) {
  trace(Math.randomRange(-10,10));
}
```

مختصر:

انتخاب یک عدد به صورت تصادفی از مجموعه (lo,hi)

Math.round(number)

ترکیب :

```
Math.round(x)
```

نشانوند :

x : یک عدد

مثال :

```
onSelfEvent (load) {
  trace(Math.round(4.8);// returns the value 5
  trace(Math.round(4.2);// returns the value 4
  trace(Math.round(-4.8);// returns the value -5
}
```

مختصر:

محاسبه گرد شده یک عدد

Math.sign(number) +

ترکیب :

```
Math.sign(number)
```

نشانوند :

number : یک عدد (اگر 0 باشه مقدار بازگشتی 0 است)

مثال :

```
onSelfEvent (load) {
 trace(Math.sign(3.8));      // returns +1
 trace(Math.sign(0));       // returns 0
 trace(Math.sign(-9));       // returns -1
}
```

مختصر:

مشخص کردن منفی یا مثبت بودن یک عدد

Math.sin(radians)

ترکیب :

```
Math.sin(x)
```

نشانوند :

x : یک عدد (طبق رادیان)

مثال :

```
onSelfEvent (load) {
  Math.sin(Math.PI / 2);       // returns 1 (sin 90deg = 1)
}
```

مختصر:

محاسبه درجه بر حسب sin رادیان x

Math.sindeg(degrees) +

ترکیب :

```
Math.sindeg(x)
```

نشانوند :

x : یک عدد (طبق درجه)

مثال :

```
onSelfEvent (load) {
  trace(Math.sindeg(90));       // returns 1 (sin 90deg = 1)
}
```

مختصر :

محاسبه رادیان بر حسب sin درجه x

Math.sqrt(number)

ترکیب :

```
Math.sqrt(number)
```

نشانوند :

number : یک عدد (بزرگتر یا مساوی 0)

مثال :

```
onSelfEvent (load) {
  trace(Math.sqrt(3));       // returns 1.732050807...
}
```

مختصر:

محاسبه مجذور number

Math.SQRT1_2

ترکیب :

```
Math.SQRT1_2
```

مثال :

```
onSelfEvent (load) {
    trace(Math.SQRT1_2);       // returns 0.707106
    trace(Math.sindeg(45) - Math.SQRT1_2);       // returns -1.11022302462515e-16 which is pretty close to 0
}
```

مختصر :

Math.SQRT1_2یک عدد ثابت -> مجذور عدد ½ = 0.70710678

Math.SQRT2

ترکیب :

```
Math.SQRT2
```

مثال :

```
onSelfEvent (load) {
  trace(Math.SQRT2);       // returns 1.414213
  trace(Math.cosdeg(45) - 1/Math.SQRT2);       // returns 1.11022302462515e-16 which is pretty close to 0
}
```

مختصر:

Math.SQRT2یک عدد ثابت -> مجذور عدد 2 = 1.41421356

Math.tan(radians)

ترکیب:

```
Math.tan(x)
```

نشانوند :

x: یک عدد (طبق رادیان)

مثال:

```
onSelfEvent (load) {
  trace(Math.tan(Math.PI / 4)); // returns 1 (tan 45deg = 1)
}
```

مختصر:

محاسبه درجه بر حسب tan رادیان x

Math.tandeg(degrees) +

ترکیب :

```
Math.tandeg(x)
```

نشانوند :

x : یک عدد (طبق درجه)

مثال :

```
onSelfEvent (load) {
  trace(Math.tandeg(45));       // returns 1 (tan 45deg = 1)
}
```

مختصر :

محاسبه رادیان بر حسب tan درجه x

Movie Clip

MovieClip._alpha	MovieClip.gotoAndPlay	MovieClip.startDrag
MovieClip.attachMovie	MovieClip.gotoAndStop	MovieClip.stop
MovieClip.beginFill	MovieClip._height	MovieClip.stopDrag
MovieClip.beginGradientFill	MovieClip._highquality	MovieClip.swapDepths
MovieClip.clear	MovieClip.hitArea	MovieClip.tabChildren
MovieClip.createEmptyMovieClip	MovieClip.hitTest	MovieClip.tabEnabled
MovieClip.createTextField	MovieClip.lineStyle	MovieClip.tabIndex
MovieClip._currentframe	MovieClip.lineTo	MovieClip._target
MovieClip.curveTo	MovieClip.loadMovie	MovieClip._totalframes
MovieClip._droptarget	MovieClip.loadVariables	MovieClip.trackAsMenu
MovieClip.duplicateMovieClip	MovieClip.localToGlobal	MovieClip.unloadMovie
MovieClip.enabled	MovieClip.moveTo	MovieClip._url
MovieClip.endFill	MovieClip._name	MovieClip.useHandCursor
MovieClip.focusEnabled	MovieClip.nextFrame	MovieClip._visible
MovieClip._focusrect	MovieClip._parent	MovieClip._width
MovieClip._framesloaded	MovieClip.play	MovieClip._x
MovieClip.getBounds	MovieClip.prevFrame	MovieClip._xmouse

MovieClip.getBytesLoaded	MovieClip._quality	MovieClip._xscale
MovieClip.getBytesTotal	MovieClip.removeMovieClip	MovieClip._y
MovieClip.getDepth	MovieClip._rotation	MovieClip._ymouse
MovieClip.getURL	MovieClip.setMask	MovieClip._yscale
MovieClip.globalToLocal	MovieClip._soundbuftime	

_alpha

ترکیب :

```
instanceName._alpha = value
```

نشانوند :

instanceName : نام آبجکت مورد نظر

value : عددی بین 0 تا 100

مثال :

```
onSelfEvent (load) {
  _root.rectangle1._alpha = 20;  // make rectangle partly transparent
  _root.rectangle2._alpha = 40;  // make rectangle partly transparent
  _root.rectangle3._alpha = 60;  // make rectangle partly transparent
  _root.rectangle4._alpha = 80;  // make rectangle partly transparent
}
```

مختصر :

تغییر شفافیت

MovieClip.attachMovie

ترکیب :

```
instanceName._alpha = value
```

نشانوند :

instanceName : نام آبجکت مورد نظر

value : عددی بین 0 تا 100

مثال :

```
onSelfEvent (load) {
  _root.rectangle1._alpha = 20;  // make rectangle partly transparent
  _root.rectangle2._alpha = 40;  // make rectangle partly transparent
  _root.rectangle3._alpha = 60;  // make rectangle partly transparent
  _root.rectangle4._alpha = 80;  // make rectangle partly transparent
}
```

مختصر :

تغییر شفافیت

---

beginFill()

ترکیب :

```
myMovieClip.beginFill({rgb {,alpha})
```

نشانوند :

alpha : میزان شفافیت

rgb : رنگ بر اساس کدهای هگزا (نمونه: 0xFF8800 رنگ نارنجی)

مثال :

```
onSelfEvent (load){
   this.beginFill(0xFF8800,100);
   this.moveTo(-50,-50);
   this.lineTo(-50,50);
   this.lineTo(50,50);
   this.lineTo(50,-50);
   this.endFill();
}
```

مختصر :

متدی برای آغاز رسم

---

beginGradientFill()

ترکیب :

```
myMovieClip.beginFill(fillType, colors, alphas, ratios, matrix)
```

نشانوند :

fillType : "linear"یا"radial"

colors : آرایه ای از رنگ ها

alpha : آرایه ای از شفافیت (تعداد عناصر آرایه برابر تعدا عناصر رنگ ها است)

ratios : آرایه ای از نسبت توزیع رنگ ها

matrix : اطلاعات بیشتر در مورد این قسمت را در help نرم افزار بیابید

مثال :

```
onSelfEvent (load){
   var colors = [0x00FF00,0xFF0000];   // colors green, red
   var alphas = [100,100];
   var ratios = [0x00,0xFF];
   matrix = {a:71,b:71,c:0,d:-56.8,e:56.8,f:0,g:0,h:0,i:1};
   this.lineStyle(1,0);
   this.beginGradientFill("linear",colors,alphas,ratios,matrix);// draw rectangle
   this.moveTo(-50,-40);   // TLHC
   this.lineTo(50,-40);    // TRHC
   this.lineTo(50,40);      // BRHC
   this.lineTo(-50,40);      // BLHC
   this.lineTo(-50,-40);   // back to TLHC
   this.endFill();
}
```

مختصر:

متدی برای آغاز رسم طیفی

---

clear()

ترکیب :

```
myMovieClip.clear()
```

مثال :

```
onSelfEvent (load){
   this.beginFill(0xFF8800,100);
   this.moveTo(-50,-50);
   this.lineTo(-50,50);
   this.lineTo(50,50);
   this.lineTo(50,-50);
   this.endFill();
}
onFrame (15) {
   this.clear()
}
```

مختصر :

متدی برای پاک کردن تمامی رسم ها

---

createEmptyMovieClip()

ترکیب :

```
myMovieClip.createEmptyMovieClip(instanceName, depth)
```

نشانوند :

instanceName: نام آبجکت ایجاد شده

depth : عمق ایجاد آبجکت هدف

مثال :

```
onSelfEvent (load) {
   this.createEmptyMovieClip("movie1", 1);
}
```

مختصر :

متدی برای ساخت یک MovieClip خالی

---

createTextField()

ترکیب :

```
myMovieClip.createTextField(instanceName, depth, x, y, width, height)
```

نشانوند :

instanceName : نام آبجکت ایجاد شده

depth : عمق ایجاد آبجکت هدف

x : موقیت در محور x

y : موقعیت در محور y

width : عرض textfield ایجاد شده

height : ارتفاع textfield ایجاد شده

مثال :

```
onSelfEvent (load) {
   this.createTextField("movie2",2,50,50,20,120);
}
```

مختصر:

متدی برای ساخت یک TextField خالی

_currentframe

ترکیب :

```
instanceName._currentframe
```

مثال :

```
onSelfEvent (enterFrame) {
   if (_root._currentframe > 10)
```

```
 {
   trace("passed frame 10");
 }
}
```

مختصر :

عدد فریم کنونی را باز میگرداند

---

curveTo()

ترکیب :

```
myMovieClip.curveTo(controlX, controlY, anchorX, anchorY)
```

نشانوند :

controlX: میزان قوس در محور x

controlY: میزان قوس در محور y

anchorX: محل آغازین قوس در محور x

anchorY: محل آغازین قوس در محور y

مثال :

```
onSelfEvent (enterFrame) {
   this.moveTo(100,0);
   this.lineStyle(1,0xFF0000);
   this.lineTo(0,0);    // draw horizontal red line
   this.lineTo(0,100);  // draw vertical line
   this.lineStyle(1,0); // black line
   this.curveTo(0,0,100,0); // note pen was at 0,100
}
```

مختصر :

متدی برای ترسیم خطوط قوس دار

_droptarget

ترکیب :

```
draggableInstance._droptarget
```

مثال :

```
onEnterFrame() {
  if (this._droptarget ==_parent.s2._target) {
     trace("zap");
  }
}
onSelfEvent (release) {
  stopDrag();
}
onSelfEvent (press) {
  this.startDragLocked();
}
```

مختصر:

Drop شدن یک آبجکت بر روی آبجکت دیگر (آبجکت هدف) را باز میگرداند .

---

duplicateMovieClip()

ترکیب :

```
MovieClipName.duplicateMovieClip(newname, depth)
```

نشانوند :

newname : نام آبجکت ایجاد شده

depth : عمق ایجاد آبجکت هدف

مثال :

```
onSelfEvent (load) {
   mc1.duplicateMovieClip("mc2",1);
   mc1._X += 50;
   mc1._Y += 50;
   mc2._X += 150;
   mc2._Y += 50;
}
```

مختصر:

ایجاد یک کپی از آبجکت مورد نظر

---

enabled

ترکیب :

```
buttonName.enabled
```

نشانوند :

buttonName : نام دکمه مورد نظر

مثال :

```
onFrame (50) {
   button1.enabled = false; // disables the specified button at Frame 50
}
onFrame (100) {
   button1.enabled = true; // re-enables the specified button at Frame 100
}
onSelfEvent (load) {
   button1.onRelease=function(){
    trace("hellllllllllo");
   }
}
```

مختصر :

فعال یا غیر فعال کرن یک دکمه

---

endFill()

ترکیب :

```
myMovieClip.endFill()
```

مثال :

```
onSelfEvent (load)
{
   this.beginFill(0xFF8800,100);
   this.moveTo(-50,-50);
   this.lineTo(-50,50);
   this.lineTo(50,50);
   this.lineTo(50,-50);
   this.endFill();
```

```
}
```

مختصر :

پایان رسم

---

focusEnabled

ترکیب :

```
myMovieClip.focusEnabled
```

مختصر :

وضعیت فعلی focusEnabled را بازمیگرداند .

_focusrect

ترکیب :

```
_focusrect = Boolean;
```

مثال :

```
onSelfEvent (load) {
  _focusrect = false;      // turn off yellow rectangle around selected button.
}
```

مختصر:

مربع زرد رنگی که در انتخاب دکمه (با استفاده از کلید Tab) نمایش داده میشود را غیر فعال میکند.

_framesloaded

ترکیب :

```
a = MovieClipName._framesloaded
```

نشانوند :

a : یک متغیر

مثال:

```
onSelfEvent (load) {
  a = _framesloaded / _totalframes * 100;      // a contains % complete of download.
}
```

مختصر:

تعداد فریم های بارگذاری شده .

_framesloaded

ترکیب :

```
a = MovieClipName._framesloaded
```

نشانوند :

a : یک متغیر

مثال :

```
onSelfEvent (load) {
  a = _framesloaded / _totalframes * 100;      // a contains % complete of download.
}
```

مختصر :

تعداد فریم های بارگذاری شده .

---

getBounds()

ترکیب :

```
myMovieClip.getBounds(targetCoordinateSpace)
```

مختصر:

تعیین مرز های یک آبجکت xMin yMin xMax yMax

---

getBytesLoaded()

ترکیب :

```
MovieClipName.getBytesLoaded()
```

مثال :

```
onFrame (1) {
   bl = _root.getBytesLoaded();
}
// the variable 'bl' will have a value indicating the total number of bytes loaded at Frame 1.
```

مختصر :

حجم بارگذاری شده .

---

getBytesTotal()

ترکیب :

```
MovieClipName.getBytesTotal()
```

مثال :

```
onFrame (1) {
   bt = _root.getBytesTotal();
}
// the variable 'bt' will have a value indicating the total number of bytes contained within the root path.
```

مختصر:

حجم کل .

---

getDepth()

ترکیب :

```
myMovieClip.getDepth()
```

مثال :

```
// A movie contains two movie clips, mc1 and mc2
onFrame (10)
{
   var mc1_depth:Number;
   var mc2_depth:Number;

   mc1_depth = mc1.getDepth();
   mc2_depth = mc2.getDepth();
   trace("mc1:" add mc1_depth add " mc2:" add mc2_depth);   // show the depth of mc1, mc2

   mc1.swapDepths(mc2_depth);   // give mc1 the mc2 depth. Note mc2 gets the mc1 depth.

   mc1_depth = mc1.getDepth();
   mc2_depth = mc2.getDepth();
   trace("mc1:" add mc1_depth add " mc2:" add mc2_depth);   // show the depth of mc1, mc2
}

*/ The following is displayed in the debug window:
mc1:-16380 mc2:-16381
mc1:-16381 mc2:-16380/*
```

مختصر:

دریافت سطح قرار گیری آبجکت مورد نظر .

---

getURL(url [, window])

ترکیب :

```
getURL(url {,window, method})
```

نشانوند :

url : مسیر مورد نظر

window : مقداری اختیاری مشخص کردن پنجره برای باز کردن صفحه مورد نظر. یکی از : "_self" "_blank" "_top" "_parent"

method : مقداری اختیاری برای دریافت یا ارسال مقادیر. (GET or POST)

مثال:

```
getURL("http://www.swishzone.com","_self");      // replace the current movie with the quoted URL.
```

مختصر:

بارگیری مسیر مورد نظر در پنجره مورد نظر .

---

globalToLocal()

ترکیب :

```
myMovieClip.globalToLocal(point);
```

نشانوند :

point : نام یا شناسه آبجکت .

مختصر:

تبدیل مقادیر کلی به مقادیر محلی .

---

gotoAndPlay()

ترکیب :

```
[object.]gotoAndPlay(frame / label)
```

نشانوند :

object : نام ابجکت مورد نظر.

frame : عدد فریم مورد نظر یا نام برچسب مشخص شده.

مثال :

```
onSelfEvent (load) {
```

```
    gotoAndPlay(16); // Movie Clip / Sprite starts at Frame 16 when loaded.
}
```

مختصر:

اجرای فریم تایم لاین از فریم یا برچسب مشخص شده

---

gotoAndStop()

ترکیب :

```
[object.]gotoAndStop(frame / label)
```

نشانوند :

object : نام ابجکت مورد نظر

frame : عدد فریم مورد نظر یا نام برچسب مشخص شده

مثال :

```
onSelfEvent (load) {
    gotoAndStop(16); // Movie Clip / Sprite goes to Frame 16 and stops.
}
```

مختصر :

توقف فریم تایم لاین از فریم یا برچسب مشخص شده

_height

ترکیب:

```
MovieClipName._height
```

مثال:

```
h = rect._height;       // h contains the current height of the Object rect.
```

مختصر:

ارتفاع آبجکت مورد نظر

_highquality

ترکیب :

```
_highquality
_highquality = value
```

نشانوند :

value : شامل سه عدد 0 یا 1 یا 2 برای تعیین کیفیت نمایش

مثال :

```
onSelfEvent (load) {
   _highquality = 0;
}
```

مختصر :

تعیین کیفیت نمایش در فلش پلیر

---

hitArea

ترکیب :

```
myMovieClip.hitArea
```

مختصر:

ناحیه بر خورد .

---

hitTest()

ترکیب :

```
myMovieClip.hitTest(x, y {,shapeFlag})
myMovieClip.hitTest(target)
```

نشانوند :

x : مشخص کردن ناحیه افقی برخورد

y : مشخص کردن ناحیه عمودی برخورد

shapeflag : مقداری بولی (true or false) اگر false باشد فقط خود آبجکت مد نظر قرار میگیرد اگر true باشد کادری که آبجکت در آن قرار میگیرد مد نظر قرار میگیرد.

target : آبجکت مورد نظر برای برخورد.

مثال:

```
onSelfEvent (enterFrame) {
  if (myMovieClip_1.hitTest(_xmouse,_ymouse,true)) {
    trace("Hey, quit touching me!");
  }
}
/* will display "Hey, quit touching me!" in the debug window
whenever the mouse cursor is moved over the star shape inside 'myMovieClip' */
onSelfEvent (enterFrame) {
  if (myMovieClip_2.hitTest(_xmouse,_ymouse,false)) {
    trace("Hey, quit touching me!");
  }
}
/* will display "Hey, quit touching me!" in the debug window
whenever the mouse cursor is moved over the outer boundaries
of the Movie Clip itself (the rectangle container holding the star shape) */
```

مختصر:

برای تعیین برخورد مورد استفاده قرار میگیرد.

---

lineStyle()

ترکیب :

```
myMovieClip.lineStyle({thickness {,rgb {,alpha}}})
```

نشانوند :

thickness : ضخامت خط از 0 تا 255

rgb : رنگ از نوع هگزا (Black is 0x000000, white is 0xFFFFFF, و دیگر...)

alpha : شفافیت خط از 0 تا 100

مختصر :

تعیین نوع خط .

---

lineTo()

ترکیب :

```
myMovieClip.lineTo(x,y)
```

نشانوند :

x : مشخص کردن نقطه در محور x

y : مشخص کردن نقطه در محور y

مثال :

```
onSelfEvent (load)
{
   this.beginFill(0xFF8800,100);
   this.moveTo(-50,-50);
   this.lineTo(-50,50);
   this.lineTo(50,50);
   this.lineTo(50,-50);
   this.endFill();
}
```

مختصر:

متدی برای ترسیم خط به مختصات مشخص شده .

---

loadMovie(name[,variables])

ترکیب :

```
MovieClipName.loadMovie("url"[, variables])
```

نشانوند :

url : مشخص کردن مسیر Movie (فایلهایی مانند swf ، jpg و برخی دیگر)
variables : مقداری اختیاری از نوع GET یا POST برای دریافت یا ارسال مقادیر.

مثال :

```
onSelfEvent (load) {
   this.loadMovie("http://www.swishzone.com/script_samples/testswf.swf");
}
```

مختصر:

بارگیری movie مورد نظر از مسیر مشخص شده .

---

loadVariables(name[,variables])

ترکیب :

```
MovieClipName.loadVariables ("url" [, variables])
```

نشانوند :

url : مشخص کردن مسیر .
variables : مقداری اختیاری از نوع GET یا POST برای دریافت یا ارسال مقادیر.

مثال :

```
onSelfEvent (load) {
   this.date = "date";
   this.time = "time";
   this.loadVariables("http://www.swishzone.com/script_samples/date.php",'GET');
}
```

کد موجود در php :

```
<?php
 // example script that returns date and time.
 // demonstrates use of load variable function.
 echo "date=";
 print (date ("d-M-Y"));
 echo "&time=";
 print (date ("H:i"));
?>
```

مختصر :

بارگیری مقادیر مورد نظر از مسیر مشخص شده .

---

localToGlobal()

ترکیب :

```
myMovieClip.localToGlobal(point);
```

نشانوند :

point : نام یا شناسه آبجکت .

مثال :

```
onSelfEvent (enterFrame) {
   location = new Object();
   location.x = thisMovieClip.shape1._X;
   location.y = thisMovieClip.shape1._Y;
   thisMovieClip.localToGlobal(location);
   trace("Shape1's Local X location is: " add thisMovieClip.shape1._X);
   trace("Shape1's Local Y location is: " add thisMovieClip.shape1._Y);
   trace("Shape1's Global X location is: " add location.x);
   trace("Shape1's Global Y location is: " add location.y);
}
```

مختصر:

تبدیل مقادیر محلی به مقادیر کلی .

---

moveTo()

ترکیب :

```
myMovieClip.moveTo(x,y)
```

نشانوند :

x : نقطه ای در محور x

y : نقطه ای در محور y

مثال :

```
onSelfEvent (load)
{
   this.beginFill(0xFF8800,100);
   this.moveTo(-50,-50);
   this.lineTo(-50,50);
   this.lineTo(50,50);
   this.lineTo(50,-50);
   this.endFill();
}
```

مختصر:

حرکت نقطه رسم به مختصات مورد نظر .

_name

ترکیب :

```
mc._name
mc._name = value
```

نشانوند :

mc : آبجکت مورد نظر .

value : قرار دادن نام جدید برای ابجکت مورد نظر .

مثال :

```
onSelfEvent (load) {
  trace(this._name);  // show name of this Object / Sprite
}
```

مختصر:

بازگرداندن نام آبجکت مورد نظر یا تغییر نام آن .

---

nextFrameAndPlay()

ترکیب :

```
[object.]nextFrameAndPlay()
```

نشانوند :

object : آبجکت MoveClip .

مثال :

```
onSelfEvent (load) {
   nextFrameAndPlay();  // Movie Clip / Sprite starts playing from second Frame
}
```

مختصر:

رفتن به فریم بعد و اجرا .

_parent

ترکیب :

```
_parent.property
_parent.property = value
_parent._parent.property is also valid.
```

نشانوند :

property : ویژگی های آبجکت یک شاخه قبل

value : قرار دادن مقدار جدید در ویژگی های آبجکت شاخه قبل

مثال :

```
_parent.play();      // restart parent Movie Clip
```

مختصر:

اشاره به آبجکت شاخه قبل

---

prevFrameAndPlay()

ترکیب :

```
[object.]prevFrameAndPlay()
```

نشانوند :

object : آبجکت MoveClip .

مثال :

```
onFrame (5) {
   prevFrameAndPlay();  // continues playing from frame 4
}
```

مختصر:

رفتن به فریم قبل و اجرا .

_quality

ترکیب :

```
_quality
```

نشانوند :

property : ویژگیب های تعریف شده :

LOW
MEDIUM
HIGH
BEST

value : مقدار جدید برای ویژگی (LOW، MEDIUM، HIGH، BEST)

مختصر:

تعیین کیفیت نمایش .

---

removeMovieClip()

ترکیب :

```
MovieClipName.removeMovieClip()
```

نشانوند :

MovieClipName : نام movieclip مورد نظر

مثال :

```
onSelfEvent (load) {
   mc1.duplicateMovieClip("mc2",1);
   mc1._X += 50;
   mc1._Y += 50;
   mc2._X += 150;
   mc2._Y += 50;
}

onFrame (12) {
   mc2.removeMovieClip();  // remove the duplicated Movie Clip
}
```

مختصر:

حذف movieclip ایجاد شده

_rotation

ترکیب :

```
MovieClipName._rotation
MovieClipName._rotation = value;
```

نشانوند :

MovieClipName : نام movieclip مورد نظر

value : مقدار جدید از نوع درجه

مثال :

```
_root.rect._rotation += 10;      // rotate rect Object 10 degrees clockwise.
```

مختصر:

مقدار چرخش یک آبجکت را مشخص میکند

---

setMask()

ترکیب :

```
myMovieClip.setMask(maskMovieClip)
```

نشانوند :

myMovieClip : نام movieclip مورد نظر

maskMovieClip : نام آبجکتی که باید به صورت ماسک قرار گیرد

مختصر:

ایجاد ماسک برای movieclip مورد نظر

_soundbuftime

ترکیب :

```
_soundbuftime
_soundbuftime = value
```

نشانوند :

value : مقدار از نوع عددی ، تعیین مقدار ثانیه برای قرار گیری صدا در حافظه بافر

مثال :

```
_soundbuftime = 10;       // buffer 10 seconds of audio.
```

مختصر:

تعیین مقدار ثانیه برای قرار گیری صدا در حافظه بافر

---

startDrag()

ترکیب :

```
target.startDrag(lockCenter:Boolean[,left, top, right, bottom])
```

نشانوند :

target : آبجکت مورد نظر

lockCenter : اگر true باشد نشانگر ماوس در وسط آبجکت قرار میگیرد .

مثال :

```
onSelfEvent (press)
{
   this.startDrag(false);
}
onSelfEvent(release)
{
   stopDrag();
}
```

مختصر:

آغاز عمل جابجای آبجکت مورد نظر ، به وسیله نشانگر ماوس .

---

stop()

ترکیب :

```
[object.]stop()
```

نشانوند :

object : آبجکت مورد نظر

مثال :

```
intro.stop();      // stops the Movie Clip / Sprite intro from playing
_root.stop();      // stop the main Timeline
```

مختصر:

متوقف کردن تایم لاین .

---

stopDrag()

ترکیب :

```
stopDrag()
```

مثال :

```
onSelfEvent (press) {
   this.startDrag(true);
}
onSelfEvent (release) {
   stopDrag();
}
```

مختصر:

متوقف کردن جابجایی .

---

swapDepths()

ترکیب :

```
myMovieClip.swapDepths(depth)
myMovieClip.swapDepths(target)
```

نشانوند :

depth : عمق مورد نظر

target : آبجکت moviClip مورد نظر

مثال :

```
onFrame(10) {
   myMovieClip1.swapDepths(20); // moves myMovieClip1 to a depth level of 20
}
onFrame(10) {
   myMovieClip1.swapDepths(myMovieClip2); // swaps the depth level of myMovieClip1 with myMovieClip2
}
```

مختصر:

تعیین عمق قرار گیری آبجکت مورد نظر .

---

tabChildren

ترکیب :

```
myMovieClip.tabChildren
```

مثال :

```
onSelfEvent (load) {
  b1.tabChildren=false
}
```

مختصر:

اگر مقدار false قرار گیرد در هنگام استفاده از کلید tab آبجکت مورد نظر انتخاب نخواهد شد .

---

tabEnabled

ترکیب :

```
buttonName.tabEnabled
```

نشانوند :

buttonName : نام دکمه مورد نظر

مثال :

```
onSelfEvent (load) {
   button1.tabEnabled = false;
}
// disables tab ordering for this button
```

مختصر:

اگر مقدار false قرار گیرد در هنگام استفاده از کلید tab دکمه مورد نظر انتخاب نخواهد شد .

---

tabIndex

ترکیب :

```
buttonName.tabIndex
```

مثال :

```
onSelfEvent (load) {
   button1.tabIndex = 1;
   button2.tabIndex = 2;
   button3.tabIndex = 3;
   myMovieClip1.tabIndex = 4;
   myTextField1.tabIndex = 5;
}
```

مختصر:

ترتیب انتخاب در هنگام استفاده از کلید tab

_target

ترکیب :

```
MovieClip._target
```

مثال :

```
onEnterFrame() {
   if (this._droptarget ==_parent.s2._target) {
      trace("zap");
   }
}
onSelfEvent (release) {
   stopDrag();
}
onSelfEvent (press) {
   this.startDragLocked();
}
// if s2 is a MovieClip owned by the main Movie Clip then its _target name is "/s2".
```

مختصر:

مسیر ، هدف مورد نظر

_totalframes

ترکیب :

```
a = MovieClipName._totalframes
```

نشانوند :

a : یک متغیر

مثال :

```
a = _framesloaded / _totalframes * 100;      // a contains % complete of download.
```

مختصر:

نشاندهنده تعداد کل فریم ها .

---

trackAsMenu

ترکیب :

```
buttonName.trackAsMenu
```

مختصر:

بازگشت حالت کنونی traceAsMenu

---

unloadMovie()

ترکیب :

```
MovieClipName.unloadMovie()
```

نشانوند :

MovieClipName : نام movieclip مورد نظر

مثال :

```
onSelfEvent (load) {
   mc1.unloadMovie();      // unload mc1
}
```

مختصر:

عدم بارگذاری .

_url

ترکیب :

```
a = MovieClipName._url
```

نشانوند :

a : یک متغیر

مثال :

```
onSelfEvent (load) {
   a=_root._url
   trace(a)
}
```

مختصر:

نمایش مسیر بارگذاری فایل swf .

---

useHandCursor

ترکیب :

```
buttonName.useHandCursor
```

مثال :

```
onSelfEvent (load) {
   buttonName.useHandCursor = false;
}
// uses the standard mouse pointer when over the named button
onFrame(10) {
   buttonName.useHandCursor = true;
}
// displays the hand cursor when over the named button
```

مختصر:

نمایش یا عدم نمایش HandCursor بر روی دکمه مورد نظر .

_visible

ترکیب :

```
_MovieClipName._visible
_MovieClipName._visible = value
```

نشانوند :

MovieClipName : نام movieclip مورد نظر

value : مقدار true یا false

مثال :

```
onSelfEvent (load) {
_root.rect._visible = false; // hide rectangle.
}
```

مختصر:

نمایش یا عدم نمایش آبجکت مورد نظر .

_width

ترکیب :

```
MovieClipName._width
```

نشانوند :

MovieClipName : نام movieclip مورد نظر

مثال :

```
onSelfEvent (load) {
w = rect._width;      // w contains the current width of the Object rect.
Trace(w);
}
```

مختصر:

عرض آبجکت مورد نظر .

_x

ترکیب :

```
MovieClipName._x
MovieClipName._x = value;
```

نشانوند :

MovieClipName : نام movieclip مورد نظر

Value : مقدار عددی

مثال :

```
onSelfEvent (load) {
_root.rect._x += 10;      // moves the Object rect 10 pixels to the right.
}
```

مختصر:

موقعیت آبجکت در محور x

_xmouse

ترکیب :

```
myMovieClip._xmouse
```

مختصر:

موقعیت نشانگر ماوس در محور x

_xscale

ترکیب :

```
MovieClipName._xscale
MovieClipName._xscale = percentage;
```

نشانوند :

MovieClipName : نام movieclip مورد نظر

percentage : مقدار عددی

مثال :

```
onSelfEvent (load) {
_root.rect._xscale = 200;       // double the width of the Movie Clip / Sprite / Object.
 }
```

مختصر:

تغییر سایز در محور افقی

_y

ترکیب :

```
MovieClipName._y
MovieClipName._y = value;
```

نشانوند :

MovieClipName : نام movieclip مورد نظر

Value : مقدار عددی

مثال :

```
onSelfEvent (load) {
_root.rect._y += 10;       // moves the Object rect 10 pixels to the bottom.
}
```

مختصر:

موقعیت آبجکت در محور y

_ymouse

ترکیب :

```
myMovieClip._ymouse
```

مختصر:

موقعیت نشانگر ماوس در محور y

_yscale

ترکیب :

```
MovieClipName._yscale
MovieClipName._yscale = percentage;
```

نشانوند :

MovieClipName : نام movieclip مورد نظر

percentage : مقدار عددی

مثال :

```
onSelfEvent (load) {
_root.rect._yscale = 200;       // double the height of the Movie Clip / Sprite / Object.
 }
```

مختصر:

تغییر سایز در محور عمودی .

Sound

آبجکت sound برای ایجاد و کنترل صدا در پروژه به کار میرود .

new Sound()	getPan()	getVolume()
loadSound()	setPan()	setVolume()
start()		

---

new Sound()

ترکیب :

```
new Sound({target});
```

نشانوند :

target : نام آبجکت مورد نظر

مثال :

```
onSelfEvent (load) {
mySound = new Sound(myMovieClip);
mySound.setVolume(75);
 }
// creates a sound object to control the sound playing within the Movie Clip / Sprite "myMovieClip" and sets its volume to 75%
```

مختصر:

ایجاد آبجکت sound برای تعیین ویژگی های صدا .

---

getPan()

ترکیب :

```
soundObject.getPan();
```

مختصر:

بازگشت بالانس صدا بین 100-(تمام چپ) تا 100 (تمام راست)

---

getVolume()

ترکیب :

```
soundObject.getVolume();
```

مختصر:

دریافت شدت صدا بین (بیصدا) 0 تا 100(آخرین شدت صدا)

---

loadSound()

ترکیب :

```
soundObject.loadSound("url", isStreaming);
```

نشانوند :

url : مسیر بارگذاری صدا با فرمت mp3

isStreaming : مقدار بولی (true or false) که مشخص میکند آیا در هنگام بارگذاری صدا اجرا شود یا نه

مثال :

```
onFrame (1) {
   rockSong = new Sound(myMovieClip);
   rockSong.loadSound("http://www.yoursite.com/yoursong.mp3", true);
}
```

مختصر:

بارگذاری فایل صدا از مسیر مورد نظر .

---

setPan()

ترکیب :

```
soundObject.loadSound("url", isStreaming);
```

نشانوند :

amount : تعیین بالانس صدا 100- تمام چپ 100 تمام راست

مثال :

```
onSelfEvent (load) {
   introSound = new Sound(myMovieClip);
   introSound.setPan(-100);
}
// creates a new Sound object "introSound" for the target "myMovieClip" and sets the pan level to full left
```

مختصر:

تعیین بالانس صدا .

---

setVolume()

ترکیب :

```
soundObject.setVolume(volume);
```

نشانوند :

volume : تعیین شدت صدا بین 0 تا 100

مثال :

```
onSelfEvent (load) {
   introSound = new Sound(myMovieClip);
   introSound.setVolume(75);
}

// sets the volume level of the Sound object "introSound" to 75 percent.
```

مختصر:

تعیین شدت صدا .

---

start()

ترکیب :

```
soundObject.start({secondOffset, loop});
```

نشانوند :

secondOffset : نقطه شروع

loop : میزان تکرار

مثال :

```
onSelfEvent (load) {
   introSound = new Sound(myMovieClip);
   introSound.start(10, 100);
}
// begins playing the Sound object "introSound" at the 10-second mark and loops through it 10 times
```

مختصر:

اجرای صدا .

String

String.charAt(position)	String.charCodeAt(position)	String.concat(string)
String.fromCharCode(ascii)	String.indexOf(substring [,start])	String.lastIndexOf(substring [,start])
String.length	String.slice(start,end)	String.split(sep,{limit})
String.substr(start [,length])	String.substring(start,{end})	String.toLowerCase()
String.toUpperCase()	String.trim()	String.trimLeft()
String.trimRight()		

از طریق این متد ها میتوانید رشته ها را ویرایش کنید .

String.charAt(position)

ترکیب :

```
myString.charAt(index)
```

نشانوند :

index : مقدار عددی برای نمایش یک کاراکتر از رشته مورد نظر

مثال :

```
onSelfEvent (load) {
   i = 0;
   t1 = $version;
   trace(t1);
   do {
      trace(t1.charAt(i));
   } while (i++ < t1.length());
}
```

مختصر:

نمایش یک کاراکتر از رشته مورد نظر .

String.charCodeAt(position)

ترکیب :

```
myString.charCodeAt(index)
```

نشانوند :

index : مقدار عددی برای نمایش یک کد کاراکتر از رشته مورد نظر

مثال :

```
onSelfEvent (load) {
   i = 0;
   t1 = $version;
   trace(t1);
   do {
      trace(t1.charCodeAt(i));
   } while (i++ < t1.length);
}
```

مختصر:

نمایش یک کد کاراکتر از رشته مورد نظر . کدهای کاراکتر شامل 0 تا 65535 است .

String.concat(string)

ترکیب :

```
myString.concat(value1,...valueN)
```

نشانوند :

value1,...valueN : مقادیری برای پیوست دادن به رشته مورد نظر

مثال :

```
onSelfEvent (load) {
   str = "test2";
   trace(str.concat("hi","this", 4 > 3, "more"));  // returns "test2hithis1more"
   trace (str);  // returns "test2"
}
```

مختصر:

پیوست دادن مقادیر یا رشته ها به رشته مورد نظر

String.fromCharCode(ascii)

ترکیب :

```
String.fromCharCode(c1,c2,...cN)
```

نشانوند :

c1,c2,...cN : کد کاراکترها

مثال :

```
onSelfEvent (load) {
str = String.fromCharCode(60,61,62,63,64);
trace(str);       // returns the string "<=>?@"
 }
```

مختصر:

بازگرداندن کاراکتر ها طبق کد کارکترهای مشخص شده .

String.indexOf(substring [,start])

ترکیب :

```
myString.indexOf(substring, [startIndex])
```

نشانوند :

Substring : رشته یا عدد مورد نظر برای جستجو

startIndex : مقدار اختیاری برای تعیین آغاز جستجو از این نقطه (مقدار عددی)

مثال :

```
onSelfEvent (load) {
   str = "abcdefgbc";
   trace (str.indexOf("bc"));      // returns 1, position of "bc"
   trace (str.indexOf("bd"));      // returns -1, not found
   trace (str.indexOf("bc",3));      // returns 7, second occurrence
   trace (str.indexOf("cd",4));      // returns -1 as "cd" exists before index 4
   trace (str);
}
```

مختصر:

جستجو مقدار مورد نظر در رشته مورد نظر و بازگرداندن مکان آن .

String.lastIndexOf(substring [,start])

ترکیب :

```
myString.lastIndexOf(substring, [startIndex])
```

نشانوند :

Substring : رشته یا عدد مورد نظر برای جستجو

startIndex : مقدار اختیاری برای تعیین آغاز جستجو از این نقطه (مقدار عددی)

مثال :

```
onSelfEvent (load) {
   str = "abcdefgbc";
   trace (str.lastIndexOf("bc"));            // returns 7, (last position of "bc")
   trace (str.lastIndexOf("bd"));            // returns -1, not found
   trace (str.lastIndexOf("bc",3));      // returns 7, (last position of "bc")
   trace (str.lastIndexOf("cd",3));      // returns 2, found because 'd' of "cd" is at 3. See note below.
   trace (str.lastIndexOf("cd",4));      // returns -1 as "cd" exists before index 4
   trace (str);
}
```

مختصر:

جستجو مقدار مورد نظر در رشته مورد نظر از انتها به ابتدا و بازگرداندن مکان آن .

String.length

ترکیب :

```
myString.length
```

مثال :

```
onSelfEvent (load) {
   str = "abcdefgbc";
   trace (str.length);      // returns 9
   trace (str.charAt(str.length - 1));      // returns the last character in the string, 'c'
}
```

مختصر:

تعداد کاراکترهای یک رشته را باز میگرداند .

String.slice(start,end)

ترکیب :

```
myString.slice(start [, end])
```

نشانوند :

start : آغاز برش رشته از این قسمت

end : پایان برش نقطه تا این قسمت

مثال :

```
onSelfEvent (load) {
   str = "abcdefgbc";
   trace ("[" + str.slice(3,6) + "]");       // returns [def]
}
```

مختصر :

برش رشته و بازگرداندن مقدار آن از مکان کاراکتر مشخص شده در نقطه آغاز تا مکان کاراکتر مشخص شده در نقطه پایان

String.split(sep,{limit})

ترکیب :

```
myString.split({sep},{limit})
```

نشانوند :

sep : رشته ای که باید جدا شود

limit : مشخص میکند که باید تا چه تعداد جدا شود

مثال 1 :

```
onSelfEvent (load) {
   myString = "SWiSHmax";
   myString = myString.split();
   trace(myString); // displays "S,W,i,S,H,m,a,x" in the debug window
}
```

مثال 2 :

```
onSelfEvent (load) {
   myString = "Dog and Cat";
   myString = myString.split(" and ");
   trace(myString); // displays "Dog,Cat" in the debug window
}
```

مثال 3 :

```
onSelfEvent (load) {
   myString = "Dog and Cat and Fish and Bird";
   myString = myString.split(" and ", 3);
   trace(myString); // displays "Dog,Cat,Fish" in the debug window
}
```

مختصر:

جدا کردن رشته مورد نظر از رشته اصلی و بازگرداندن مقدار آن

String.substr(start [,length])

ترکیب :

```
myString.substr(start [, length])
```

نشانوند :

start : رشته ای که باید جدا شود

length : مشخص میکند که باید تا چه تعداد کاراکتر بعد از نقطه آغاز جدا شود

مثال :

```
onSelfEvent (load) {
   str = "abcdefgbc";
   trace (str.substr(3,3));      // returns "def"
   trace (str);             // returns "abcdefgbc"
}
```

مختصر:

جدا کردن رشته مورد نظر از نقطه آغاز به طول در نظر گرفته شده .

String.substring(start,{end})

ترکیب :

```
myString.substring(start {, end})
```

نشانوند :

start : رشته ای که باید جدا شود

end : نقطه پایان جدا سازی را مشخص میکند

مثال1 :

```
onSelfEvent (load) {
   str = "SWiSH Max is Great!";
   trace(str.substring(0,5));
}
// displays "SWiSH" In the Debug window
```

مثال2 :

```
onSelfEvent (load) {
   str = "SWiSH Max is Great";
   trace(str.substring(0,9));
}
// displays "SWiSH Max" In the Debug window
```

مختصر:

جدا کردن رشته مورد نظر از نقطه آغاز به طول در نظر گرفته شده .

String.toLowerCase()

ترکیب :

```
myString.toLowerCase()
```

مثال :

```
onSelfEvent (load) {
   str = "Hello World, how are YOU doing?";
   str = str.toLowerCase();
   trace(str);  // returns "hello world, how are you doing?"
}
```

مختصر:

تبدیل کاراکترهای یک رشته به حروف کوچک .

String.toUpperCase()

ترکیب :

```
myString.toUpperCase()
```

مثال :

```
onSelfEvent (load) {
   str = "Hello World, how are YOU doing?";
   str = str.toUpperCase();
   trace(str);  // returns "HELLO WORLD, HOW ARE YOU DOING?"
}
```

مختصر:

تبدیل کاراکترهای یک رشته به حروف بزرگ .

String.trim()

ترکیب :

```
myString.trim()
```

مثال :

```
onSelfEvent (load) {
   str = "   Hello World   ";
   trace("[" add str.trim() add "]"); // returns "[Hello World]"
}
```

مختصر:

جدا کردن فاصله ها (space) از نقطه آغاز و پایان یک رشته .

String.trimLeft()

ترکیب :

```
myString.trimLeft()
```

مثال :

```
onSelfEvent (load) {
   str = "   Hello World   ";
   trace(str.trimLeft()); // returns "Hello World   "
}
```

مختصر:

جدا کردن فاصله ها (space) از طرف چپ یک رشته .

String.trimRight()

ترکیب :

```
myString.trim()
```

مثال :

```
onSelfEvent (load) {
   str = "   Hello World   ";
   trace(str.trimRight()); // returns "   Hello World"
}
```

مختصر:

جدا کردن فاصله ها (space) از طرف راست یک رشته .

پایان