

فصل پانزدهم

هدف کلی فصل :

بهره‌گیری بیشتر از زبان ActionScript 3.0

اهداف رفتاری

پس از مطالعه این فصل، از فراگیر انتظار می‌رود که :

- بتواند با استفاده از Action Script به متعامل کردن انیمیشن بپردازد.
- بتواند دستورات شرطی را به کار برد.
- بتواند ترتیب اجرای دستورات برنامه را با دستور Switch Case تغییر دهد.
- بتواند دستورات را به تعداد دفعات معین و محدود با دستور For اجرا کند.
- مفهوم Class را شرح دهد.
- از ابزار آرماتور استفاده کند.

زمان (ساعت)	
عملی	تئوری
۸	۳



کلیات

لذت واقعی در Flash زمانی آغاز می‌شود که با دستورات کوتاه به Flash بگویید که چه کار کند. با ترکیب دستورات ساده می‌توانید مجموعه‌های پیچیده‌ای از دستورات را برای ایجاد گرافیک‌های متحرک جذاب تولید کنید. در این بخش به معرفی تعدادی از دستورات می‌پردازیم.

۱-۱۵- متغیر (Variable)

Variable یا متغیر، مقادیر مربوط به یک داده که ممکن است ثابت یا متغیر باشد را در خود نگه می‌دارد. نوع اطلاعاتی که یک متغیر می‌تواند در خود نگه دارد در زمان تعریف آن مشخص می‌شود. به عنوان مثال، یک متغیر رشته‌ای تنها قادر به نگهداری اطلاعات از نوع حروف و اعداد است در حالی که یک متغیر عددی تنها می‌تواند اعداد را در خود نگهداری کند.

نکاتی که در تعریف متغیر باید رعایت شود:

- ۱- نام متغیر باید با حروف یا زیر خط (Underscore) و یا علامت (\$) شروع شود.
 - ۲- نام متغیر می‌تواند شامل حروف، اعداد، زیر خط و یا علامت (\$) باشد.
 - ۳- از کاراکتر فاصله خالی Blankspace یا Whitespace در نام متغیر نمی‌توان استفاده کرد.
 - ۴- کلمات رزرو شده را به عنوان نام متغیر نمی‌توان استفاده کرد. مانند if, for, else, and, ...
 - ۵- بهتر است نام متغیر با حروف کوچک باشد.
 - ۶- تعیین نوع داده‌ای اختیاری است ولی بهتر است آن را تعیین کنید.
 - ۷- می‌توان با یک var چند متغیر تعریف کرد (علامت جداکننده بین متغیرها، می‌باشد).
- var i: int, m: int=12, s: string ;

۱-۱۵-۱- اعلام یک متغیر

قبل از این که از یک متغیر استفاده کنید، باید آن را تعریف کنید. برای تعریف یک متغیر باید از کلمه کلیدی var استفاده کرده و عبارتی را به عنوان نام متغیر تعریف کنید، سپس مقداری را به عنوان مقدار اولیه به آن اختصاص دهید.

فرم کلی تعریف متغیر

[مقدار]= نوع داده‌ای : نام متغیر var

علامت [] به مفهوم اختیاری بودن می‌باشد.

۲-۱-۱۵- انواع داده‌ها

Flash برای نگهداری و پردازش داده‌ها، انواع مختلفی از داده‌ها را استفاده می‌کند. نوع داده‌ها مشخص می‌کنند چه نوع اطلاعاتی می‌توانند در متغیرها ذخیره شوند. برای مثال متغیر نوع داده صحیح، می‌تواند فقط اعداد صحیح را در خود ذخیره کند. Flash از انواع داده‌های عددی و غیر عددی پشتیبانی می‌کند. داده‌ها به شرح جدول ۱-۱۵ است.

جدول ۱-۱۵- انواع داده‌ها

مثال	نوع داده	مفهوم
1	Unit	عدد صحیح بدون علامت
-5	Int	عدد صحیح
4.5	Number	عدد اعشاری
"Flash"	String	رشته‌ای
True	Boolean	منطقی

رشته ترکیبی از چند کاراکتر می‌باشد که داخل علامت نقل قول (زوج کوتیشن) قرار گرفته است. داده‌های رشته‌ای می‌توانند شامل هر کاراکتری باشند. برای مثال «Flash cs4» رشته‌ای به طول ۹ کاراکتر است.

متغیر Boolean می‌تواند فقط شامل دو مقدار True و False باشد.

۳-۱-۱۵- عملگرها

علایمی که برای انجام عملیات مختلف روی متغیرها به کار می‌روند، عملگر (Operator) نامیده می‌شوند. Flash دارای انواع عملگرهای مختلف است که روی متغیرها اعمالی را انجام می‌دهند. در ادامه با انواع عملگرهای Flash آشنا می‌شوید.

عملگرهای محاسباتی: از عملگرهای محاسباتی برای انجام عملیات ریاضی استفاده می‌شود. نتیجه عملگرهای محاسباتی، به صورت عدد می‌باشد. انواع عملگرهای محاسباتی در جدول ۲-۱۵ شرح داده شده است.



جدول ۱۵-۲ - انواع عملگرهای محاسباتی

مثال	مفهوم	عملگرهای محاسباتی
$2+3$	جمع	+
$8-5$	تفریق	-
$4*2$	ضرب	*
$24/4$	تقسیم	/
$A=A+1$ معادل $A++$	افزودن یک واحد به مقدار قبلی	++
$A=A-1$ معادل $A--$	کاستن یک واحد از مقدار قبلی	--

عملگرهای انتساب : در Flash برای قرار دادن یک مقدار ثابت یا نتیجه یک عبارت در یک متغیر، از دستور انتساب (=) استفاده می‌شود. انواع عملگرهای انتساب در جدول ۱۵-۳ شرح داده شده است.

جدول ۱۵-۳ - انواع عملگرهای انتساب

مثال	مفهوم	عملگرهای انتساب
$A=1$	انتساب	=
$A=A+3$ معادل $A+=3$	انتساب با اضافه کردن به مقدار قبلی	+=
$A=A-3$ معادل $A-=3$	انتساب با کاستن از مقدار قبلی	-=
$A=A*3$ معادل $A*=3$	انتساب با ضرب در مقدار قبلی	*=
$A=A/3$ معادل $A/=3$	انتساب با تقسیم در مقدار قبلی	/=

عملگرهای رابطه‌ای : عملگرهای رابطه‌ای برای مقایسه انواع داده‌ها به کار می‌روند. حاصل این عملگرها ارزش True یا False است. انواع عملگرهای رابطه‌ای در جدول ۱۵-۴ شرح داده شده است.

جدول ۴-۱۵- انواع عملگرهای رابطه‌ای

مثال	مفهوم	عملگرهای رابطه‌ای
$3 < 5$	کوچکتر	$<$
$2 \leq 3$	کوچکتر یا مساوی	\leq
$8 > 5$	بزرگتر	$>$
$9 \geq 7$	بزرگتر یا مساوی	\geq
$6 = 6$	مساوی	$==$
$6 != 5$	نامساوی	$!=$

عملگرهای منطقی: برای ایجاد ترکیبات مختلف شرطی و انجام عملیات منطقی روی عبارات، از عملگرهای منطقی استفاده می‌شود. در جدول ۵-۱۵ اسامی عملگرهای منطقی Flash آمده است.

جدول ۵-۱۵- انواع عملگرهای منطقی

شرح	مفهوم	عملگرهای منطقی
عملگر Not ارزش یک عبارت را معکوس می‌کند.	NOT	!
نتیجه عملگر منطقی And فقط وقتی درست است که هر دو عبارت منطقی آن درست باشند و اگر یکی یا هر دو عبارت نادرست باشند، نتیجه نادرست خواهد بود.	AND	&&
نتیجه عملگر منطقی Or فقط وقتی نادرست است که هر دو عبارت منطقی آن نادرست باشد و اگر یکی یا هر دو عبارت درست باشند، نتیجه درست خواهد بود.	OR	

در مثال‌های ارائه شده به نحوه معرفی و مقداردهی متغیرها دقت کنید.



مثال ۱-۱۵ : دو متغیر a و b از نوع متغیر عددی با یک مقدار اولیه تعریف شده‌اند. سپس مقدار متغیر a یک واحد افزایش و مقدار متغیر b یک واحد کاسته شده است. در آخر توسط فرمان trace حاصل جمع دو متغیر نمایش می‌یابد.

```
Var a: int = 1;
Var b: int = 2;

a++; // (a=a+1)
b--; // (b=b-1)
trace (a+b); //3
```

عبارت مقابل // یک توضیح اضافه (Comment) برای کاربر جهت کسب اطلاع است. برای این که کد Action Script خود را بعد از چندماه نیز درک کنید باید توضیحات را اضافه کنید که هدف دستورات را توضیح می‌دهند.

نکته : تابع trace تابعی است که نتیجه عبارت‌های نام برده شده درون پرانتز، به هنگام نمایش فیلم swf روی پانل output چاپ می‌شود.

مثال ۲-۱۵ : متغیر answer از نوع متغیر رشته‌ای تعریف شده است. از عملگر + برای اتصال رشته‌ها استفاده می‌شود. سپس رشته "two" توسط عملگرهای اتصال مقدار متغیر answer را تعیین می‌کند. در آخر توسط فرمان trace حاصل متغیر answer نمایش می‌یابد.

```
Var answer: string;
answer= "two" + "two" ;
trace(answer) ; // twotwo
```

مثال ۳-۱۵ : متغیر a با مقدار اولیه ۲ تعریف شده است. سپس فرمان trace حاصل عبارت منطقی را نمایش می‌دهد. مقدار متغیر a بزرگتر یا مساوی عدد ۳ نیست و نتیجه محاسبه مقدار منطقی false می‌باشد.

```
Var a: unit=2 ;
trace(a >= (1+2)) ; // false
```

تمرین ۱-۱۵: شرح دستورات زیر را بنویسید.

```
Var a: unit=2 ;
trace (a == (1+2)) ;
trace (a != (1+2)) ;
```

۱۵-۲ تابع (Function)

تابع، مجموعه‌ای از دستورات زبان ActionScript است که مجموعاً تحت یک نام قابل فراخوانی هستند. استفاده از توابع باعث می‌شود که در صورت نیاز به تکرار یک مجموعه از دستورات در برنامه، نیار به تکرار و تایپ مجدد کل دستورات نباشد.

به عنوان مثال فرض کنید که برای یک دکمه، یک قطعه مشتمل بر سه خط نوشته‌اید که کار آن تغییر اندازه باشد، اگر تنها یک دکمه داشته باشید، نوشتن سه خط برنامه مشکلی به نظر نمی‌رسد اما چنانچه مجبور باشید برای ۵ دکمه همین برنامه را تکرار کنید بهتر است که این سه خط برنامه را در قالب یک تابع تعریف کرده و برای هر دکمه تنها نام تابع را فراخوانی کنید.

نحوه معرفی توابع و فراخوانی آنها و همچنین به کارگیری آرگومان‌ها را در نمونه مثال‌ها ملاحظه

می‌کنید.

مثال ۴-۱۵: در این مثال همزمان با رها کردن کلید ماوس روی دکمه، توسط تابع A

ارتفاع و پهناى آن تغییر می‌کند.

۱- یک سند جدید از نوع ActionScript 3 باز کنید و آن را با نام size.fla ذخیره کنید.

۲- روی آیکن New Layer کلیک کنید و یک لایه جدید ایجاد کنید. نام لایه اول را Actions



و لایه دوم را Frames وارد کنید.

۳- در لایه Frames روی فریم ۱ یک تصویر رسم کنید. سپس توسط کلید F8، آن را به سمبل دکمه (Button) تبدیل کنید.

۴- در پانل Properties نام نمونه را mc وارد کنید.

۵- در لایه Actions روی فریم ۱ کلیک کنید و توسط کلید F9، پانل Action را باز کنید سپس کدهای زیر را به دقت وارد کنید.

```

1 mc.addEventListener(MouseEvent.CLICK, A);
2
3 function A (e:MouseEvent)
4 {
5     mc.width = 50;
6     mc.height = 100;
7 }
8

```

شکل ۱-۱۵

رویداد دکمه mc یعنی EventListener اعلام می‌کند که در صورت قرار گرفتن ماوس روی دکمه mc تابع A اجرا شود که با اجرای تابع A پهنا و ارتفاع نمونه mc تغییر می‌کند. این کد چه تغییری در مثال بالا ایجاد می‌کند؟

this.width=180;

مثال ۵-۱۵: در این مثال نحوه استفاده از یک تابع بدون آرگومان را مشاهده می‌کنید.

۱- یک سند جدید از نوع ActionScript 3 باز کنید و آن را با نام message.fla ذخیره کنید.

۲- نام لایه اول را Actions وارد کنید.

۳- در لایه Actions روی فریم ۱ کلیک کنید و توسط کلید F9، پانل Action را باز کنید سپس کدهای زیر را به دقت وارد کنید.

با اجرای فیلم تابع showMsg() اجرا شده و متن hello نمایش داده می‌شود.


```

1 function showMsg()
2 {
3   trace("hello");
4 }
5 showMsg();
6
7

```

Output
hello

شکل ۱۵-۲

مثال ۱۵-۶: در این مثال نحوه استفاده از یک تابع با آرگومان را مشاهده می‌کنید.

در این مثال تابع `showMsg()` دارای آرگومان رشته‌ای با نام `msg` است که هدف آن نمایش متن درون متغیر `msg` در پنجره Output است. با اجرای فیلم، تابع `showMsg()` اجرا شده و متن `goodbye` که به متغیر `msg` نسبت داده شده، نمایش داده می‌شود.

```

1 function showMsg(msg:String) {
2   trace(msg);
3 }
4 showMsg("goodbye");
5 //goodbye
6
7

```

Output
goodbye

شکل ۱۵-۳

مثال ۱۵-۷: این مثال 20° درجه سلیسیوس را به فارنهایت تبدیل می‌کند. خروجی عدد ۶۸ است.



```
1 function celToFahrenheit (cel:Number):Number {
2     return (9 / 5) * cel + 32;
3 }
4 trace(celToFahrenheit(20));
5 //00
```

OUTPUT
68

شکل ۱۵-۴

نام تابع `celToFahrenheit` است. آرگومان آن `cel` بوده و از نوع عددی است. خروجی نیز از نوع عدد می‌باشد. مقدار محاسبه شده توسط فرمان `return` بازگردانده می‌شود. در خط بعدی توسط فرمان `trace` تابع `celToFahrenheit` با مقدار ورودی 20° فراخوانی می‌شود.

مثال ۱۵-۸: این مثال 68° درجه فارنهایت را به سلسیوس تبدیل می‌کند. خروجی عدد 20° است.

```
1 function fahrenheitToCelsius (far:Number):Number {
2     return (5 / 9) * (far - 32);
3 }
4 var celDeg:Number = fahrenheitToCelsius(68);
5 trace(celDeg);
6 //10
```

OUTPUT
20

شکل ۱۵-۵

نام تابع `fahrenheitToCelsius` است. آرگومان آن `far` بوده و از نوع عددی است. خروجی نیز از نوع عدد می‌باشد. مقدار محاسبه شده توسط فرمان `return` بازگردانده می‌شود. در خط بعدی تابع `fahrenheitToCelsius` با مقدار 68° فراخوانی شده و در متغیر `celDeg` نگهداری

می‌شود.

توسط فرمان var متغیر celDeg از نوع عددی معرفی شده است.
توسط فرمان trace محتوای متغیر celDeg نشان داده می‌شود.

تمرین ۲-۱۵: دستورات ارائه شده را به دقت تجزیه و تحلیل کنید و هدف آن را به طور مختصر شرح دهید.

```
function m( a:int , b: int)
{
  var c: int;
  c=a+b;
}
trace ( m(1,2));
```

```
var k:string = "Hello";
function g ()
{
  trace (k);
}
g ();
trace (k);
```

۳-۱۵- استفاده از عبارات شرطی

عبارات شرطی ابزاری است که Action بر اساس آن در خصوص درستی یا نادرستی یک موضوع تصمیم گرفته و بر اساس این تصمیم عمل مناسبی را انجام می‌دهد.

۱-۳-۱۵- دستور if

چنانچه بخواهید در صورت برقرار بودن شرط، دستور یا دستورات اجرا شود، از دستور if به شکل زیر استفاده می‌کنیم:

```
if(عبارت شرطی)
{
  دستورات
}
```

هنگام اجرای دستور if، ابتدا عبارت شرطی بررسی می‌شود. در صورتی که نتیجه ارزیابی، درست (True) باشد، دستور یا دستورات اجرا می‌شود.



مثال ۱۵-۹: توسط فرمان var به متغیر a عدد یک و به متغیر b رشته "goodbye" و به متغیر c مقدار منطقی false اختصاص داده شده است. طبق دستور شرطی اگر مقدار متغیر a برابر عدد ۱ باشد رشته "option a" نمایش داده می‌شود.

```

1 var a:Number = 1;
2 var b:String = "goodbye ";
3 var c:Boolean = false;
4 if (a == 1) {
5   trace("option a");
6 }
7

```

شکل ۱۵-۶

مثال ۱۵-۱۰: طبق دستور شرطی اگر مقدار a برابر عدد ۱ و مقدار متغیر b رشته "goodbye" باشد رشته "options a and b" نمایش داده می‌شود.

```

1 if (a == 1 && b == "goodbye")
2   |
3   trace("options a and b");
4
5
6

```

شکل ۱۵-۷

مثال ۱۵-۱۱: در این مثال اگر مقدار متغیر a برابر عدد ۱ یا مقدار متغیر b رشته "goodbye" باشد رشته option a or b نمایش داده می‌شود.

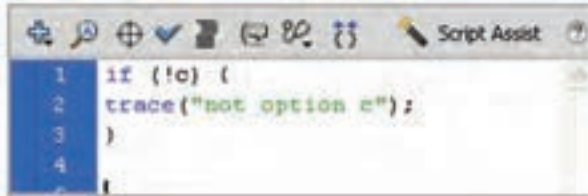
```

1 if (a == 1 || b == "goodbye")
2   trace("option a or b");
3
4

```

شکل ۱۵-۸

مثال ۱۲-۱۵: در این مثال اگر حاصل شرط برابر مقدار true باشد رشته "not options c" نمایش داده می‌شود (متغیر c مقدار منطقی دارد و داده‌های منطقی به طور پیش فرض مقدار اولیه false دارند و !c به معنای مقدار منطقی true است).



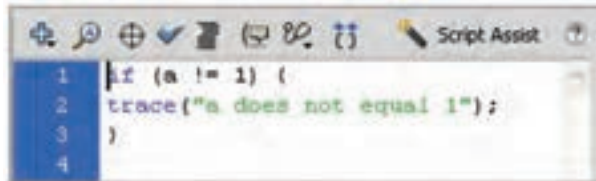
```

1  if (!c) {
2  trace("not option c");
3  }
4

```

شکل ۱۵-۹

مثال ۱۳-۱۵: در این دستور شرطی اگر مقدار متغیر a مخالف مقدار عددی ۱ باشد رشته "a does not equal 1" نمایش داده می‌شود.



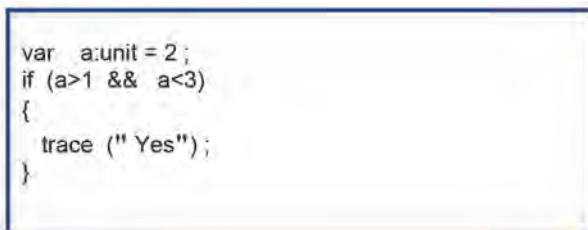
```

1  if (a != 1) {
2  trace("a does not equal 1");
3  }
4

```

شکل ۱۵-۱۰

تمرین ۳-۱۵: دستورات ارائه شده را به دقت تجزیه و تحلیل کنید و هدف آن را به طور مختصر شرح دهید.



```

var a:unit = 2;
if (a>1 && a<3)
{
    trace (" Yes");
}

```



۲-۳-۱۵- if-Else

هنگام اجرای دستور if، ابتدا عبارت شرطی بررسی می‌شود. در صورتی که نتیجه ارزیابی درست (True) باشد، دستورات ۱ اجرا می‌شوند و اگر نتیجه ارزیابی نادرست (False) باشد، دستورات ۲ اجرا خواهد شد.

```

(عبارت شرطی)if
{
دستورات ۱
}
else
{
دستورات ۲
}
    
```

مثال ۱۴-۱۵: در این دستور شرطی اگر مقدار متغیر a مخالف مقدار عددی ۱ باشد رشته "a does not equal 1" در غیر این صورت رشته "a does equal 1" نمایش داده شود.

```

1  if (a != 1) {
2  trace("a does not equal 1");
3  } else {
4  trace("a does equal 1");
5  }
    
```

شکل ۱۱-۱۵

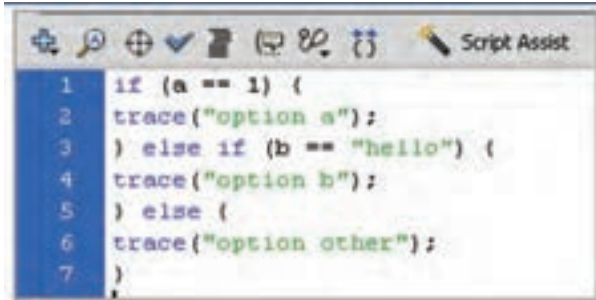
مثال ۱۵-۱۵: در این مثال از دستور if تودرتو استفاده شده است. اگر مقدار متغیر a مقدار عددی ۲ باشد رشته "a does not equal 1" نمایش داده می‌شود. در غیر این صورت اگر مقدار متغیر a مقدار عددی ۱ باشد رشته "a does equal 1" نمایش داده می‌شود.

```

1  if (a == 2) {
2  trace("a does not equal 1");
3  } else if (a == 1) {
4  trace("a does equal 1");
5  }
    
```

شکل ۱۲-۱۵

مثال ۱۶-۱۵: در این مثال از دستور if-else تودرتو استفاده شده است.



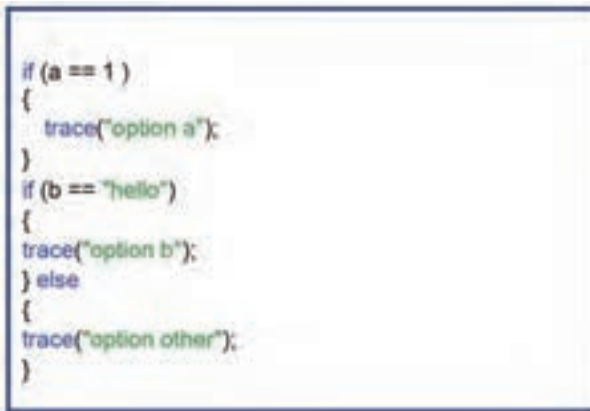
```

1  if (a == 1) {
2  trace("option a");
3  } else if (b == "hello") {
4  trace("option b");
5  } else {
6  trace("option other");
7  }

```

شکل ۱۳-۱۵

تمرین ۴-۱۵: نتیجه دستورات زیر چیست؟



```

if (a == 1)
{
  trace("option a");
}
if (b == "hello")
{
  trace("option b");
} else
{
  trace("option other");
}

```

۱۵-۴- دستور Switch Case

دستور Switch-case نیز مانند دستور if می تواند ترتیب اجرای دستورات برنامه را تغییر دهد و بر اساس شرایط خاص، باعث اجرای مجموعه ای از دستورات شود.

در این دستور، ابتدا مقدار عبارت مقابل Switch محاسبه شود. سپس این مقدار از بالا به پایین با مقادیر مقابل هر Case مقایسه شود. در صورتی که حاصل عبارت با مقدار مقابل Case یکسان باشد، دستور یا دستورات آن Case اجرا شده و با دستور break کنترل برنامه به اولین دستور بعد از Switch منتقل خواهد شد و نمونه های Case بعدی نادیده گرفته می شوند. اگر مقدار عبارت مقابل Switch با هیچ یک از مقادیر Case برابر نباشد، دستور یا دستورات مقابل default اجرا می شود.



عبارت یا متغیر) Switch

```
{  
مقدار ۱ :  
مجموعه دستورات ۱  
break ;  
مقدار ۲ :  
مجموعه دستورات ۲  
break ;  
.  
.  
.  
default  
مجموعه دستورات ۳  
}
```

نکته: اگر دو Case دارای مقادیر یکسان باشند، دستورات اولین Case اجرا می‌شود و فرمان Switch خاتمه می‌یابد.

مثال ۱۷-۱۵: در این مثال نحوه استفاده از دستور Switch-case را مشاهده می‌کنید.

اگر مقدار متغیر a مقدار عددی ۱ باشد رشته "one" نمایش داده می‌شود.
اگر مقدار متغیر a مقدار عددی ۲ باشد رشته "two" و اگر مقدار عددی ۳ باشد رشته "three"
نمایش داده می‌شود. در غیر این صورت رشته other نمایش می‌یابد.

```
1 switch (a) {  
2   case 1 :  
3     trace("one");  
4     break;  
5   case 2 :  
6     trace("two");  
7     break;  
8   case 3 :  
9     trace("three");  
10    break;  
11   default :  
12     trace("other");  
13   break;  
14 }
```

شکل ۱۴-۱۵

۱۵-۵- دستور FOR

از دستور For زمانی استفاده می‌شود که لازم است دستورات را به تعداد دفعات معین و محدود اجرا کنیم. ساختار حلقه For بصورت زیر است:

```

(گام حرکت یا پرش ; شرط اتمام حلقه ; مقدار اولیه = نام شمارنده)
For
{
مجموعه دستورات
}

```

دقت کنید که نام شمارنده باید از قبل تعریف شده باشد. همچنین مقدار شمارنده باید کوچکتر از مقدار نهایی باشد تا حلقه پایان یابد. اگر در گام حرکت از عملگر ++ استفاده شود هدف گام افزایشی است و اگر از عملگر -- استفاده شود هدف گام کاهشی است.

مثال ۱۵-۱۸: در این مثال مقدار اولیه متغیر i عدد صفر بوده و در هر بار اجرای حلقه یک واحد به مقدار آن اضافه می‌شود. دستورات حلقه تا زمانی که مقدار متغیر i کوچکتر از عدد ۳ باشد اجرا می‌شود. همانطور که ملاحظه می‌کنید ۳ بار رشته hello نمایش داده می‌شود.

```

Script Assist
1 for (var i:int = 0; i < 3; i++)
2   trace("hello");
3
4
5
6
7
8
9
10
11
12

```

OUTPUT

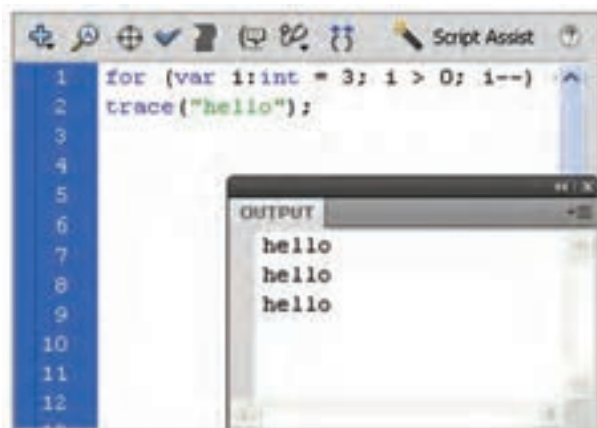
```

hello
hello
hello

```

شکل ۱۵-۱۵

مثال ۱۵-۱۹: این دستور حلقه از نوع کاهشی بوده و طبق دستور ۳ بار رشته hello نمایش داده می‌شود.



شکل ۱۶-۱۵

مثال ۲۰-۱۵: این دستور حلقه از نوع افزایشی بوده و طبق دستور اعداد کوچکتر از ۵ نمایش داده می‌شود.

```
for (var i:int= 0 ; i<5 ; i++)
{
  trace (i); //0 1 2 3 4
}
```

تمرین ۵-۱۵: خروجی دستورات زیر چیست؟

```
for (var i:int= 1 ; i<=10 ; i++)
{
  trace ("5 × ",i, " = ", 5* i);
}
```

```
for (var i:int= 3 ; i>0 ; i--)
{
  trace (i);
}
```

۱۵-۶- دستور While

این حلقه، دستورات داخل خود را تا زمانی که شرط تعیین شده درست باشد، اجرا می‌کند. شکل کلی این ساختار تکرار به صورت زیر است:

```
While (عبارت شرطی)
{
    دستورات
}
```

مثال ۱۵-۲۱: طبق این دستور با هر بار اجرای برنامه تعدادی عدد تصادفی بین ۰ تا ۱۰۰ نمایش داده می‌شود. متغیر num از نوع عددی و با مقدار اولیه صفر تعریف شده و تا زمانی که مقدار متغیر num کوچکتر از مقدار ۰.۵ باشد، اعداد تصادفی نشان داده می‌شود.

The screenshot shows a script editor window titled 'Script Assist' with the following code:

```
1 var num:Number = 0;
2 while (num < 0.5) {
3   num = Math.random();
4   trace (num*100);
5 }
6
7
8
9
10
11
12
```

An 'OUTPUT' window is open in the foreground, displaying the results of the trace statements:

```
OUTPUT
1.560756703838706
16.991470335051417
35.72255680337548
82.77999651618302
```

شکل ۱۷-۱۵

روند اجرای حلقه Do...While برخلاف قبل است. یعنی ابتدا دستورات داخل حلقه یک بار اجرا شده و سپس شرط حلقه ارزیابی می‌شود. شکل کلی این ساختار تکرار به صورت زیر است:

```
do
{
    مجموعه دستورات
} While (عبارت شرطی)
```



تمرین ۶-۱۵: دستورات زیر را شرح دهید.

```

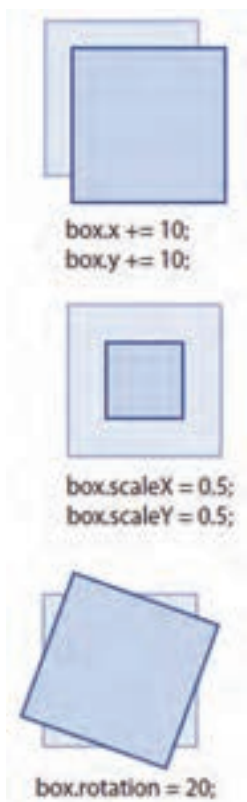
var i:int = 0;
While (i<5)
{
    trace (i);
    i++;
}

```

مثال ۲۲-۱۵: یک movie clip به شکل مربع ایجاد کنید و نام نمونه را box قرار

دهید.

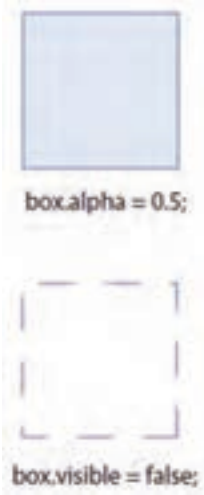
برای تغییر خصوصیات آن به مثال‌های زیر توجه کنید. شکل‌های تیره تغییرات را پس از اعمال دستور نشان می‌دهد.



الف) موقعیت نمونه box در راستای محور افقی و عمودی به اندازه ۱۰ پیکسل افزایش می‌یابد.

ب) مقیاس نمونه box در راستای محور افقی و عمودی نصف می‌شود.

ج) نمونه box به اندازه ۲۰ درجه چرخش می‌یابد.



د) میزان شفافیت نمونه box به اندازه نصف کاهش می‌یابد.

ه) نمونه box غیر فعال می‌شود.

مثال ۲۳-۱۵: در این مثال با هر با فشردن دکمه ماوس، در لحظه رها کردن K1 ۲۰ درجه می‌چرخد.

- ۱- یک فایل Flash از نوع ActionScript 3 باز کنید و آن را با نام rotate fla ذخیره کنید.
- ۲- روی آیکن New Layer کلیک کنید و یک لایه جدید ایجاد کنید. نام لایه اول را Actions و لایه دوم را Frames وارد کنید.
- ۳- در لایه Frames روی فریم ۱ یک تصویر رسم کنید. سپس توسط کلید F8، آن را به سمبل (movie clip) تبدیل کنید.
- ۴- در پانل Properties نام نمونه را K1 قرار دهید.
- ۵- در لایه Actions روی فریم ۱ کلیک کنید و توسط کلید F9، پانل Action را باز کنید سپس کدهای زیر را به دقت وارد کنید.

```

Script Assist
1 K1.addEventListener(MouseEvent.CLICK, A);
2 function A(e:MouseEvent):void {
3     K1.rotation += 20;
4 }
5

```

شکل ۱۸-۱۵



۷-۱۵- کلاس (Class)

در زبان ActionScript 3.0 هر شیء توسط یک Class تعریف می‌شود که این Class شامل نمایش خلاصه‌ای از خواص آن شیء است. یک Class شامل یک نوع از اشیا است که دارای خواص و متدهای مشترکی هستند. شما تا به حال با مفاهیمی مانند کلاس‌های MovieClip و Buttons آشنا شده‌اید. به منظور استفاده از متدها و خواص‌های مربوط به یک Class کافی است نمونه‌ای از آن کلاس را از طریق اعلام متغیر ایجاد و نوع داده (data type) آن را تعیین کنید.

Var نام کلاس = New نام کلاس : نام متغیر

به عنوان مثال می‌خواهیم متغیری از نوع MovieClip و با نام clip1 تعریف کنیم :

Var Clip1: movie Clip=New movie Clip() ;

مثال ۲۴-۱۵ : در این مثال اطلاعات مربوط به کلاس MovieClip را به پروژه

Flash وارد می‌کنید.

۱- پانل ActionScript را باز کنید.

۲- روی دکمه Script Assist در سمت راست پنجره Action کلیک کنید.

۳- در پانل سمت چپ Methods → MovieClip → flash.display را انتخاب کرده

و سپس روی Stop در فهرست متدها دابل کلیک کنید.

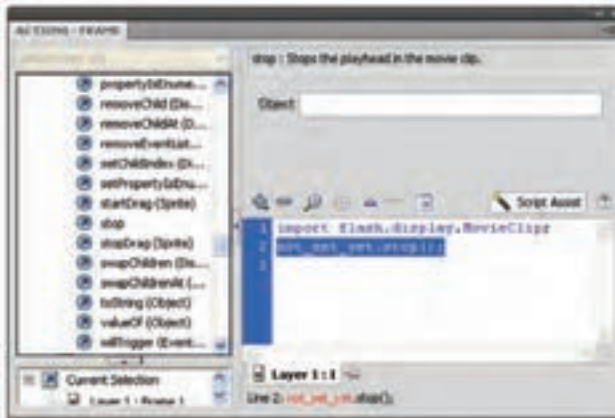
۴- این کد در پانل Script در سمت راست پانل Action ظاهر می‌شود.

۵- اولین عبارت ActionScript، اطلاعات مربوط به کلاس MovieClip را به پروژه Flash

وارد می‌کند به گونه‌ای که فیلم شما به متدها و خصوصیات تعبیه شده مربوط به کلاس MovieClip دسترسی خواهد داشت.

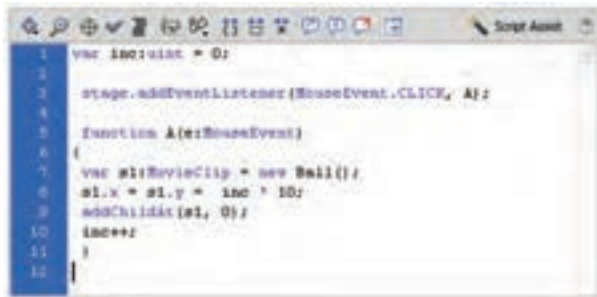
۶- عبارت دوم از متد Stop برای متوقف کردن نمونه not-set-yet استفاده می‌کند. می‌توانید

not-set-yet را به دلخواه خود تغییر دهید و مثلاً نام S1 وارد کنید.



شکل ۱۹-۱۵

مثال ۲۵-۱۵: در این مثال با هر بار کلیک ماوس یک MovieClip که نام کلاس آن Ball است به لیست نمایش اضافه می‌شود.



شکل ۲۰-۱۵

در خط ۱ متغیری به نام `inc` با مقدار اولیه صفر تعریف شده است.
 در خط ۲ `EventListener` گوش می‌دهد تا در صورت کلیک روی صفحه (stage) تابع `A` فراخوانی شود. عملکرد تابع `A` به شرح زیر است:
 متغیر `S1` یک `MovieClip` از نوع `MovieClip` با نام `Ball` می‌باشد.
 در خط بعد موقعیت مکانی `MovieClip` جدید بر اساس مضرری از متغیر `inc` تعیین می‌شود.
 توسط فرمان `addChildAt()` یک نمونه از کلیپ `S1` به صفحه نمایش اضافه می‌شود.
 در پایان توسط دستور `inc++` یک واحد به متغیر `inc` اضافه می‌شود تا در کلیک بعدی مکان `MovieClip` جدید تغییر یابد.



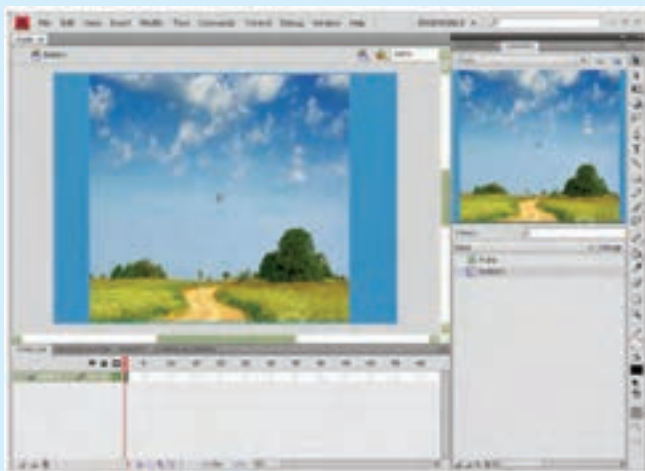
۱۵-۸- ایجاد آرماتور

اگر بخواهید مجموعه‌ای از عناصر را به هم متصل کنید از ابزار Bone استفاده کنید. این ابزار ساختار شاخه‌دار ایجاد می‌کند مانند شاخه‌های درخت، پاها و بازوهای انسان.

مثال ۱۵-۲۶: محو کردن یک سمبل نمایشی توسط Action و به کارگیری دستور

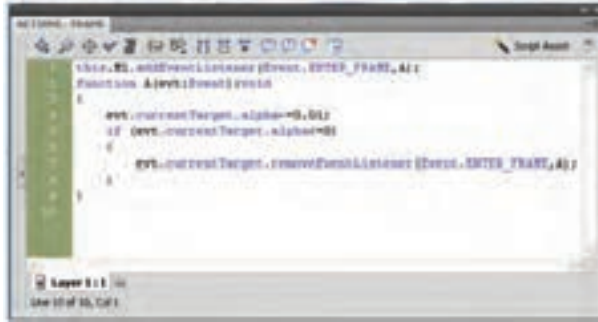
شرطی if

- ۱- یک فایل جدید ایجاد کنید و با نام Fade.fla ذخیره کنید.
- ۲- یک سمبل نمایشی همانند شکل ۱۵-۲۱ ایجاد کنید.
- ۳- نمونه سمبل نمایشی را در صفحه اصلی قرار دهید.



شکل ۱۵-۲۱

- ۴- در پانل Properties در قسمت Instant Name نام نمونه سمبل نمایشی را M1 وارد کنید.
- ۵- کلید F9 را بزنید تا پانل Action ظاهر شود و کد را مطابق شکل ۱۵-۲۲ در آن وارد کنید.

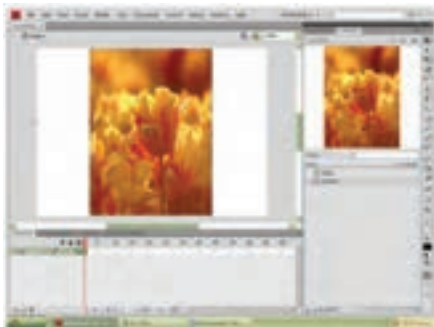


شکل ۲۲-۱۵

- ۶- اگر بخواهید به نمونه‌ای از سمبل نمایشی دسترسی داشته باشید که نام آن M1 است و در خط زمان اصلی قرار دارد، باید آدرس آن را به صورت this.M1 ذکر کنید.
- ۷- Event.ENTER_FRAME نام رویدادی است که باید سمبل نمایشی جاری به آن گوش فرا دهد و سپس تابع A را اجرا کند.
- ۸- تابع A دارای آرگومان است. توابع در هنگام بازگشت می‌توانند اطلاعاتی را بازگردانند و یا خالی باشند که در این مثال کلمه Void به خالی بودن آن اشاره می‌کند.
- ۹- با اجرای تابع A، میزان Alpha نمونه سمبل نمایشی جاری به میزان 0.01 کاهش می‌یابد.
- ۱۰- توسط دستور if میزان Alpha کنترل می‌شود و در صورتی که میزان آن کوچکتر یا مساوی صفر باشد اجرای رویداد توسط removeEventListener خاتمه می‌یابد.
- ۱۱- با اجرای فیلم مشاهده می‌کنید که نمونه سمبل نمایشی به تدریج محو می‌شود.

مثال ۲۷-۱۵: اعمال جلوه حرکتی (Transition) به یک سمبل نمایشی توسط

Action و استفاده از کلاس

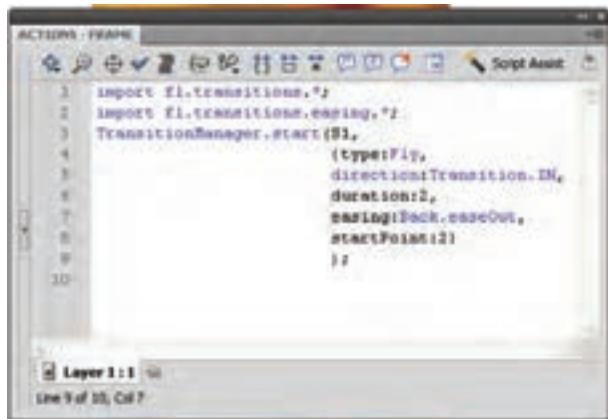


شکل ۲۳-۱۵

- ۱- یک فایل جدید ایجاد کنید و با نام Transition.fla ذخیره کنید.
- ۲- یک سمبل نمایشی ایجاد کرده و نمونه سمبل نمایشی را مطابق شکل ۲۳-۱۵ در صفحه اصلی قرار دهید.



- ۳- در پانل Properties در قسمت Instant Name نام نمونه سمبل نمایشی را S1 وارد کنید.
- ۴- کلید F9 را بزنید تا پانل Action ظاهر می‌شود و کد را مطابق شکل ۲۴-۱۵ در آن وارد کنید.



شکل ۲۴-۱۵

- ۵- پکیج transitions شامل کلاس‌های متفاوت برای اعمال جلوه حرکتی می‌باشد. اگر بخواهید به پکیج transitions دسترسی داشته باشید، باید دستور آن را به صورت *import fl.transitions.* ذکر کنید.
- ۶- کلاس easing شتاب‌های متفاوت به حرکت می‌دهد. برای دسترسی به کلاس easing که در پکیج transitions است، باید دستور *import fl.transitions.easing.* را ذکر کنید.
- ۷- TransitionManager.start برای اعمال جلوه حرکتی به نمونه S1 است که دارای پارامترهای مختلف است.
- ۸- با اجرای فیلم مشاهده می‌کنید که نمونه سمبل نمایشی از سمت بالا وارد صحنه می‌شود و دارای حرکت نوسانی در راستای عمود است.

نکته: برای تغییر نقطه شروع جلوه حرکتی مقدار پارامتر Start Point را مطابق جدول زیر تغییر دهید.

مقدارهای پارامتر StartPoint	نقطه شروع حرکت
1	بالا و سمت چپ
2	بالا و مرکز
3	بالا و راست
4	چپ و مرکز
5	مرکز
6	راست و مرکز
7	پایین و چپ
8	پایین و مرکز
9	پایین و راست

تمرین ۷-۱۵: مدل‌های مختلف حرکت را توسط پارامتر easing طبق جدول زیر مشاهده کنید.

مقادیر مختلف پارامتر easing
Back
Bounce
Elastic
Regular
Strong
None

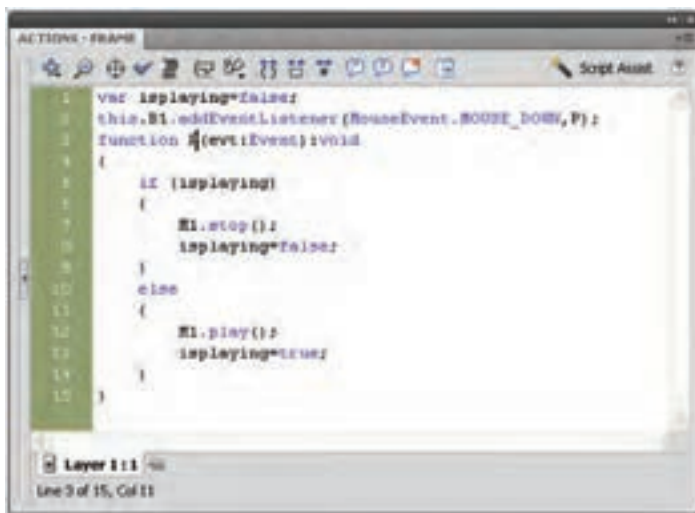
مثال ۲۸-۱۵: کنترل حرکت یک انیمیشن توسط Action و به کارگیری دستور شرطی if



شکل ۱۵-۲۵

- ۱- یک فایل جدید ایجاد کنید و با نام Animation.fla ذخیره کنید.
- ۲- یک سمبل نمایشی ایجاد کرده و نمونه سمبل نمایشی را در صفحه اصلی قرار دهید.
- ۳- در پانل Properties در قسمت Instant Name نام نمونه سمبل نمایشی را M1 وارد کنید.
- ۴- یک سمبل کلیدی رسم کرده و نمونه آن را در صفحه اصلی قرار دهید و نام آن را B1 قرار دهید (شکل ۱۵-۲۵).

۵- کلید F9 را بزنید تا پانل Action ظاهر شود و کد را مطابق شکل ۱۵-۲۶ در آن وارد کنید.

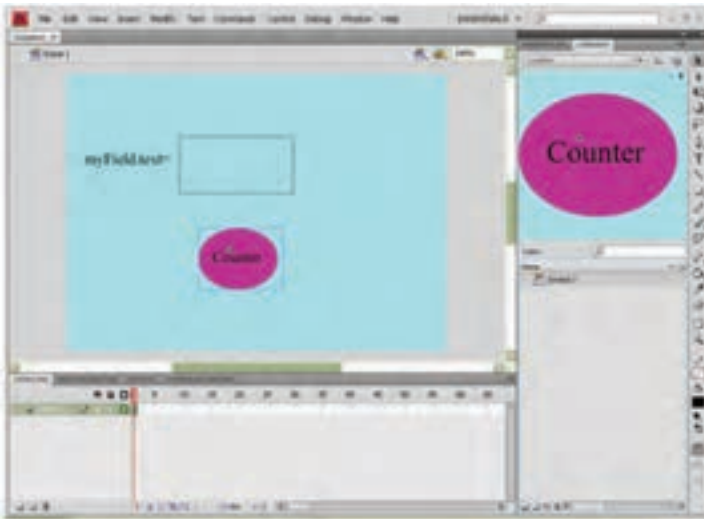


شکل ۱۵-۲۶

- ۶- متغیر isplaying با مقدار اولیه false مقدار دهی می‌شود.
- ۷- رویداد دکمه B1 یعنی EventListener اعلام می‌کند که در صورت فشردن دکمه B1 توسط ماوس تابع P اجرا شود.
- ۸- تابع P دارای آرگومان است و کلمه Void به خالی بودن آن اشاره می‌کند.
- ۹- تابع P در هنگام اجرا مقدار متغیر isplaying را بررسی می‌کند در صورتی که مقدار آن true بود حرکت نمونه سمبل نمایشی متوقف می‌شود و اگر مقدار آن برابر false بود به حرکت ادامه می‌دهد.

مثال ۲۹-۱۵: ایجاد یک کادر متنی دینامیک برای شمارش توسط Action

- ۱- یک فایل جدید ایجاد کنید و با نام counter.fla ذخیره کنید.
- ۲- در صفحه اصلی یک کادر متنی قرار دهید.
- ۳- در پانل Properties در قسمت Text type نوع کادر متنی را Dynamic Text انتخاب کنید و نام نمونه را T1 قرار دهید.
- ۴- یک سمبل کلیدی رسم کرده و نمونه آن را در صفحه اصلی قرار دهید و نام آن را B1 قرار دهید (شکل ۲۷-۱۵).



شکل ۲۷-۱۵



۵- کلید F9 را بزنید تا پانل Action ظاهر شود و کد را مطابق شکل ۱۵-۲۸ در آن وارد کنید.



شکل ۱۵-۲۸

- ۶- رویداد دکمه B1 یعنی EventListener اعلام می‌کند که در صورت فشردن دکمه B1 توسط ماوس تابع A اجرا شود.
- ۷- تابع A در هنگام اجرا مقدار متغیر clicks را یک واحد افزایش می‌دهد و مقدار آن را در کادر متنی T1 نمایش می‌دهد.

مثال ۱۵-۳۰: کنترل چرخش یک سمبل نمایشی توسط Action به کارگیری دستور if

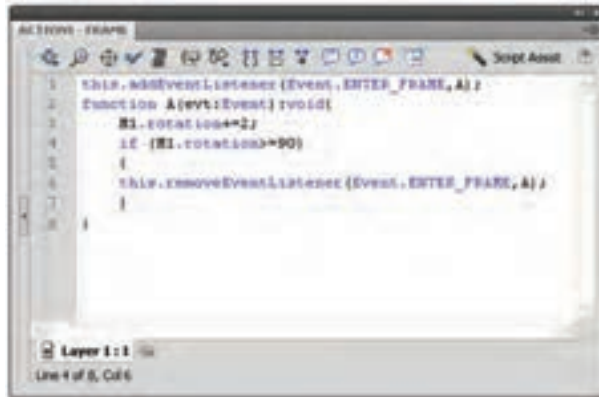
- ۱- یک فایل جدید ایجاد کنید و با نام Rotation.fla ذخیره کنید.
- ۲- یک سمبل نمایشی رسم کرده و نمونه سمبل نمایشی را مطابق شکل ۱۵-۲۹ در صفحه اصلی قرار دهید. (در این مثال عقربه ساعت سمبل نمایشی مورد نظر است).



شکل ۱۵-۲۹

۳- در پانل Properties در قسمت Instant Name نام نمونه سمبل نمایشی را M1 وارد کنید.

۴- کلید F9 را بزنید تا پانل Action ظاهر شود و کد را مطابق شکل ۱۵-۳۰ در آن وارد کنید.



شکل ۱۵-۳۰

۵- اگر بخواهید به نمونه‌ای از سمبل نمایشی اشاره کنید که در خط زمان اصلی قرار دارد، باید از this استفاده کنید.

۶- EVENT.ENTER-FRAME نام رویدادی است که باید سمبل نمایشی جاری به آن گوش فرا دهد و سپس تابع A را اجرا کند.

۷- تابع A دارای آرگومان است. توابع در هنگام بازگشت می‌توانند اطلاعاتی را بازگردانند و یا خالی باشند که در این مثال کلمه Void به خالی بودن آن اشاره می‌کند.

۸- با اجرای تابع A، نمونه سمبل نمایشی جاری به میزان ۲ درجه می‌چرخد.

۹- توسط دستور if میزان چرخش نمونه سمبل نمایشی کنترل می‌شود و در صورتی که میزان آن بزرگتر یا مساوی ۹۰ درجه باشد اجرای رویداد خاتمه می‌یابد.

۱۰- با اجرای فیلم مشاهده می‌کنید که نمونه سمبل نمایشی می‌چرخد و پس از گردش ۹۰ درجه متوقف می‌شود.

مثال ۱۵-۳۱: در این مثال با کلیک روی سمبل کلیدی Fast سرعت انیمیشن افزایش یافته و با کلیک روی سمبل کلیدی Slow سرعت انیمیشن کاهش می‌یابد.



- ۱- یک فایل جدید ایجاد کنید و با نام Speed.fla ذخیره کنید.
- ۲- یک انیمیشن رسم کرده و آن را در صفحه اصلی قرار دهید.
- ۳- دو سمبل کلیدی رسم کرده و نمونه آن را در صفحه اصلی قرار دهید و نام آنها را B1 و B2 قرار دهید.
- ۴- کلید F9 را بزنید تا پانل Action ظاهر شود و کد را مطابق شکل ۳۱-۱۵ در آن وارد کنید.

```

1
2 B1.addEventListener(MouseEvent.CLICK,S);
3 B2.addEventListener(MouseEvent.CLICK,K);
4
5 function S(evt:MouseEvent):void
6 {
7     stage.frameRate+=5;
8 }
9
10
11 function K(evt:MouseEvent):void
12 {
13     if (stage.frameRate>5)
14     {
15         stage.frameRate-=5;
16     }
17 }
18

```

شکل ۳۱-۱۵

- ۵- رویداد دکمه B1 یعنی EventListener اعلام می‌کند که در صورت کلیک دکمه B1 توسط ماوس تابع S اجرا شود.
- ۶- با اجرای تابع S، مقدار Frame rate پنج واحد افزایش می‌یابد. همان‌طور که می‌دانید برای تنظیم سرعت نمایش باید مقدار Frame rate را تغییر دهید. هر چه عدد بزرگتر انتخاب شود سرعت نمایش بالا می‌رود.
- ۷- رویداد دکمه B2 یعنی EventListener اعلام می‌کند که در صورت کلیک دکمه B2 توسط ماوس تابع K اجرا شود.
- ۸- با اجرای تابع K، مقدار Frame rate اگر بزرگتر از پنج بود، پنج واحد کاهش می‌یابد. هر چه عدد کوچک‌تر انتخاب شود سرعت نمایش کاهش می‌یابد.



● Variable یا متغیر، مقادیر مربوط به یک داده را که ممکن است ثابت یا متغیر باشد در خود نگه می‌دارد. نوع اطلاعاتی که یک متغیر می‌تواند در خود نگه دارد در زمان تعریف آن مشخص می‌شود.

● برای تعریف یک متغیر باید از کلمه کلیدی Var استفاده کرده و عبارتی را که به عنوان نام متغیر تعریف کنید، سپس مقداری را به عنوان مقدار اولیه به آن اختصاص دهید.

● Flash برای نگهداری و پردازش داده‌ها، انواع مختلفی از داده‌ها را استفاده می‌کند. نوع داده‌ها مشخص می‌کنند چه نوع اطلاعاتی می‌توانند در متغیرها ذخیره شوند.

● عملگر (Operator) عملگر (Operator) برای انجام عملیات مختلف روی متغیرها به کار می‌رود، عملگر (Operator) نامیده می‌شوند.

● تابع، مجموعه‌ای از دستورات زبان Action Script است که مجموعاً تحت یک نام قابل فراخوانی هستند. استفاده از توابع باعث می‌شود که در صورت نیاز به تکرار یک مجموعه از دستورات در برنامه، نیاز به تکرار و تایپ مجدد کل دستورات نباشد.

● عبارات شرطی ابزاری است که Action بر اساس آن در خصوص درستی یا نادرستی یک موضوع تصمیم گرفته و بر اساس این تصمیم عمل مناسبی را انجام می‌دهد.

● چنانچه بخواهیم در صورت برقرار بودن شرط، دستور یا دستورات اجرا شود، از دستور if استفاده می‌کنیم.

● هنگام اجرای دستور if، ابتدا عبارت شرطی بررسی می‌شود. در صورتی که نتیجه ارزیابی درست (True) باشد، دستورات بعد if اجرا می‌شوند و اگر نتیجه ارزیابی نادرست (False) باشد، دستورات بعد Else اجرا خواهند شد.

● دستور Switch Case نیز مانند دستور if می‌تواند ترتیب اجرای دستورات برنامه را تغییر دهد و بر اساس شرایط خاص، باعث اجرای مجموعه‌ای از دستورات شود.

● از دستور For زمانی استفاده می‌شود که لازم است دستورات را به تعداد دفعات معین و محدود اجرا کنیم.



● حلقه While، دستورات داخل خود را تا زمانی که شرط تعیین شده درست باشد، اجرا می‌کند.

- یک Class شامل یک نوع از اشیا است که دارای خواص و متدهای مشترکی هستند.
- اگر بخواهید مجموعه‌ای از عناصر را به هم متصل کنید از ابزار Bone استفاده کنید.

واژه‌نامه

Boolean	منطقی
Bone	استخوان
Class	کلاس
Comment	یادداشت
Default	پیش فرض
Function	تابع
Operator	عملگر
Output	خروجی
Switch	تعویض
String	رشته
Trace	ردیابی، دنبال کردن
Variable	متغیر
While	تا زمانی که

آزمون تئوری

درستی یا نادرستی گزینه‌های زیر را تعیین کنید.

۱- در Flash برای قرار دادن یک مقدار ثابت یا نتیجه‌ی یک عبارت در یک متغیر، از دستور انتساب (=) استفاده می‌شود.

۲- عملگرهای انتساب برای مقایسه‌ی انواع داده‌ها به کار می‌روند.

۳- برای ایجاد ترکیبات مختلف شرطی و انجام عملیات منطقی روی عبارات، از عملگرهای محاسباتی استفاده می‌شود.

معادل عبارتهای سمت راست را از ستون سمت چپ انتخاب کرده و مقابل آن بنویسید.

Operator ۴- متغیر

Boolean ۵- عملگر

Variable ۶- منطقی

چهارگزینه‌ای

۷- خروجی دستور زیر چیست؟

```
Trace("a" & "b");
```

الف) a & b (ب) ab

ج) "a" & "b" (د) aob

۸- خروجی دستور زیر چیست؟

```
trace (2+2*3);
trace ((2+2)*3);
trace (((2+2)/6)*3);
```

الف) 12,8,4 (ب) 8,12,2

ج) 12,12,4 (د) 12,12,0

۹- خروجی دستور زیر چیست؟

```
function passParameter(theParameter)
{
    theParameter++;
    trace(theParameter)
}

passParameter(3);
```

الف) 4 (ب) 5

ج) 6 (د) 3

۱۰- خروجی دستور زیر چیست؟

```
var a:uint =10;
a++;
trace (a);
a--;
trace(a);
```

الف) 0,20 (ب) 9,11

ج) 10,11 (د) 10,12

۱۱- خروجی دستور زیر چیست؟

```
function m (a:int, b:int):int
{
    var c:int
    c = a + b;
    return c;
}

trace(m(3,2))
```

الف) 3+2 (ب) 5

ج) 3 (د) 6

در جای خالی عبارت مناسب بنویسید.

۱۲- عملگر Not ارزش یک عبارت را می‌کند.

۱۳- نتیجه عملگر منطقی And فقط وقتی درست است که هر دو عبارت منطقی آن باشند.



۱۴- نتیجه‌ی عملگر منطقی Or فقط وقتی نادرست است که هر دو عبارت منطقی آن باشد.

باشد.

به سؤالات زیر پاسخ تشریحی دهید.

۱۵- متغیر چیست؟

۱۶- نحوه عملکرد دستور While چیست؟

دستور کار در کارگاه

۱- سندی ایجاد کنید که مطابق شکل با فشردن دکمه‌ها نمونه موجود تغییر یابد.

