زبان برنامه نویسی فلش Action script نام دارد و مثل هر زبان برنامه‌نویسی دیگری امکان نوشتن دستورالعمل‌هایی را جهت انجام کار فراهم می‌سازد. در این فصل به معرفی اصول اولیه و مهم برنامه نویسی در فلش می‌پردازیم.

پس از پایان این فصل انتظار می‌رود که فراگیر بتواند:

- نحوه کار با لوحه Actions را شرح دهد.
- دستورات مختلف برنامه نویسی در فلش را بیان کند.
- به یک قاب، دکمه و یا فیلم ویدئویی، یک فعل[1] اضافه کند.

۱-۸-آشنایی با لوحه Actions

همان طور که اشاره شد Action script یک زبان برنامه نویسی است که توسط آن می‌توان جلوه‌های جذاب ایجاد کرده و به برقراری ارتباط با کاربران در کلیپ ویدئویی‌های فلش پرداخت. همچنین می‌توان با استفاده از scriptهای مناسب روی کلیپ ویدئویی کنترل بیشتری داشت. به عنوان مثال روی کلیپ ویدئویی‌هایی که تاکنون ساخته‌اید هیچ کنترلی از نظر متوقف شدن و یا شروع دوباره آن‌ها ندارید ولی با کمک scriptها می‌توانید به راحتی کنترل بیشتری روی کلیپ ویدئویی‌های فلش اعمال کنید.

شکل (۱-۸) نمایی از لوحه Actions

نکته: در صورتی که لوحه Actions دیده نمی‌شود برای به نمایش درآوردن آن فرمان Window→Action را صادر کنید.

لوحه Actions در نرم‌افزار فلش از ۲ حالت مختلف تشکیل شده است. حالت معمولی و حالت حرفه‌ای. در حالت معمولی می‌توانید از دستورات آمادهای که در جعبه ابزار واقع در سمت چپ لوحه Actions وجود دارد استفاده کنید. در این حالت فقط باید ابزار مورد نظر را از جعبه ابزار انتخاب کنیم سپس روی آن کلیک راست کرده و گزینه Add to script را انتخاب کرد تا دستور به پنجره Script منتقل شود (و یا می‌توانیم به کمک ماوس آن را به ناحیه Script بکشیم). ولی در حالت حرفه‌های فلش کمک زیادی نمی‌کند و شما باید دستورات را خود در پنجره Script تایپ کنید. زمانی که در حالت حرفه‌های قرار دارید. فلش به طور خودکار درستی Syntax دستوراتی را که در پنجره Script تایپ می‌کنید را ارزیابی نمی‌کند و همین امر باعث می‌شود که در هنگام اجرا (در صورتی که Syntax دستور اشتباه باشد) با خطا رو به رو شوید. در این حالت می‌توانید برای اطمینان از درستی نگارش دستورات خود دکمه Check Syntax که در بالای لوح به شکل تیک قرار دارد را فشار دهید. حالت حرفه‌های در واقع برای اشخاص حرفه‌ای و متخصص طراحی شده است و افراد مبتدی بهتر است از حالت معمولی این لوح استفاده کنند. برای

[1] Action

تغییر دادن حالت Script نویسی از حالت معمولی به حرف‌های و بالعکس روی منوی تنظیم‌های لوحه که در شکل (۱-۸) مشخص شده کلیک کرده و حالت مورد نظر خود را انتخاب کنید.

۲-۸-نحوه قرارگیری Action Script در یک فیلم فلش

معمولاً در جاهای مختلفی از فایل فلش می‌توان یک Action Script اضافه کرد. به عنوان مثال شما می‌توانید برای یک قاب برنامه نویسی کنید و این برنامه زمانی اجرا خواهد شد که هد اجرا وارد قاب شود و زمانی که می‌خواهید برای یک رویداد مانند کلیک ماوس برنامه نویسی کنید، به گونه‌ای که وقتی کاربر نماد دکمه‌ای را کلیک کرد، آن برنامه اجرا گردد، باید برنامه مربوطه را در یک دکمه قرار دهید. بنابراین با توجه به عملی که قرار است صورت بگیرد محل قرارگیری یک Action Script متفاوت می‌باشد.

۱-۲-۸-قرار دادن Action Script در یک قاب

برای اضافه کردن Actions Script به یک قاب ابتدا باید در ستون مورد نظر از خط زمان یک قاب کلیدی درج کنید (برای این کار می‌توان از کلید F6 استفاده کرد) سپس از منوی Window→Actions را انتخاب کنید تا لوحه Actions - Frame باز شود.
اکنون باید Action مورد نظر خود را از جعبه ابزار واقع در سمت چپ لوح Actions - Frame انتخاب کنید.

۲-۲-۸-قرار دادن Action Script در یک دکمه

قرار دادن Action Script در یک دکمه باعث می‌شود تا فایل حالت محاورهای به خود بگیرد (یعنی بتواند با کاربر ارتباط برقرار کند). برای این منظور ابتدا باید دکمه مورد نظر خود را طراحی کرده و به کمک ماوس آن را به درون صفحه منتقل کنیم. سپس باید توسط ابزار انتخاب، دکمه را انتخاب کرده و فرمان Window→Actions را صادر کنیم تا لوحه Actions - Button باز شود. هنگامی‌که می‌خواهیم Action را به یک دکمه نسبت دهیم، از یک رویداد استفاده کنیم. در فلش تمامی رویدادها با کلمه on شروع می‌شوند و بعد در بین دو پرانتز نوع رویداد نوشته می‌شود. در زیر به معرفی تعدادی از رویدادهای پرکار برد می‌پردازیم:

- رویداد on (release): زمانی که با اشاره گر ماوس روی دکمه کلیک شده سپس رها شود، دستورات اجرا می‌گردند.
- رویداد on (press): در صورت فشردن دکمه دستورات اجرا می‌شود.
- رویداد on (roll over): دستورات زمانی اجرا می‌شود که اشاره‌گر ماوس روی دکمه برود.
- رویداد on (rollout): دستورات زمانی اجرا می‌شود که اشاره‌گر ماوس از روی دکمه بیرون رود.

زمانی که وارد لوحه Action Script می‌شویم در جعبه ابزار روی گزینه Global functions کلیک می‌کنیم سپس گزینه Movie Clip Control را انتخاب کرده و از بین گزینه‌های موجود، روی گزینه on کلیک دوتایی می‌کنیم تا بلوک on به پنجره script منتقل شود (بلوک به هر مجموعه دستوری گفته می‌شود که بین دو آکولاد باز و بسته نوشته می‌شوند) شکل (۲-۸)

شکل (۲-۸) ۲۸ نمایی از کدنویسی در پنل Action Script

اکنون از لیست کشویی باز شده، رویداد موردنظر خود را انتخاب می‌کنیم سپس در بین دو علامت آکولاد دستورات دلخواه خود را قرار می‌دهیم.

۳-۸-اضافه کردن توضیحات

اضافه کردن توضیحات در دو زمان بسیار مهم می‌باشد. زمانی که روی یک پروژه چندین نفر در حال کار کردن باشند و یا آن قدر کار شما پیچیدگی داشته باشد که بخواهید در هر لحظه توضیحاتی به آن اضافه کنید. البته اضافه کردن توضیحات از کارهای اصلی برنامه‌نویس می‌باشد. چرا که پس از گذشت مدت زمانی از ساخت پروژه، این توضیحات هستند که در یادآوری ساختارهای برنامه به برنامه‌نویس کمک می‌کنند.
ساختار توضیحات (در زبان Action Script) عبارت است از:

- /*Comment*/
- //Comment

برای تبدیل یک خط به توضیح کافی است به ابتدای آن // اضافه کنید و برای توضیحات چند خطی کافی است ابتدای آن را با */ شروع و در انتهای آن /* قرار دهید.
شما می‌توانید به Action scriptهای مربوط به هر دکمه، قاب، و یا فیلم توضیحاتی را درج کنید.


نکته اضافه کردن توصیفات علاوه بر این که باعث خواناتر شدن دستورات نوشته شده در لوحه Action می‌شوند، حجم فایل را نیز افزایش ن می‌دهند. به همین دلیل سرعت اجرای فایل کاهش ن می‌یابد.

به عنوان مثال می‌خواهیم توضیحاتی را درباره دستورات موجود در یک قاب به آن اضافه کنیم، برای این کار ابتدا روی قاب مورد نظر کلیک می‌کنیم سپس لوحه Action مربوط به آن را باز می‌کنیم. برای اضافه کردن توضیحات، ابتدا ۲ علامت اسلش را تایپ می‌کنیم سپس عبارت‌های خود را وارد می‌کنیم. هر چیزی که بعد از علامت // قرار بگیرد، به رنگ طوسی کم‌رنگ نوشته می‌شود و جز دستورات به حساب نمی‌آید.

```
on (release)
{
 // This is a test
}
```

همچنین می‌توانید توضیحاتی را به خط زمان اضافه کنیم، برای افزودن توضیحات به یک قاب در خط زمان بهتر است لایه جدیدی ایجاد کنیم و نام آن را Description بگذاریم. سپس روی قاب مورد نظر کلیک کرده و در لوحه خصوصیات در کادر Frame، توضیحات خود را وارد می‌کنیم. این کار باعث می‌شود تا کادر مربوط به Label type فعال شود. اکنون می‌توانیم از لیست کشویی Label type گزینه Comment را انتخاب کنیم.
بعد از انجام این عمل مشاهده خواهید کرد که در پشت توضیحات شما در کادر frame علامت // ظاهر شده است و همچنین قاب مورد نظرتان در خط زمان دارای همین علامت به رنگ سبز می‌باشد.


نکته در صورتی که بخواهیم برای یک قاب برچسب در نظر بگیریم، بعد از آن که عنوان برچسب را در کادر frame وارد کردیم باید از لیست کشویی Label type گزینه Name را انتخاب کنیم. با انجام این کار یک پرچم قرمز رنگ، روی قاب مورد نظر در خط زمان ظاهر می‌شود (شکل ۳۲۹-۸). در قسمت‌های بعدی به کاربرد برچسب‌گذاری روی قاب‌ها پی خواهید برد.

شکل ۲۹ نمایی از قاب برچسب‌گذاری شده

٤-٨-کد نویسی با Action Script

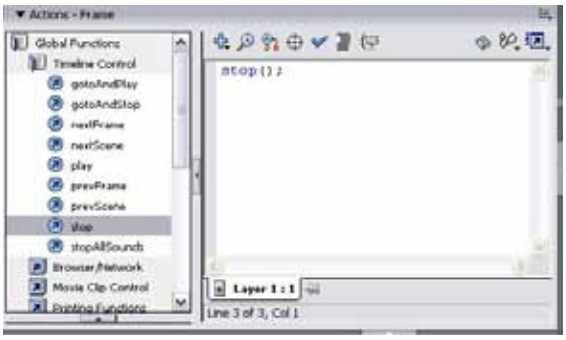
نرم‌افزار فلش برنامه های زیادی دارد که می‌توانیم از آن‌ها استفاده کنیم. در این قسمت به بررسی کدهای پای‌های که احتمال استفاده بیشتری را دارند می‌پردازیم:

١-٤-٨-متوقف کردن یک فیلم

زمانی که یک فیلم اجرا می‌شود تا زمانی که متوقف نشود، اجرای آن ادامه پیدا خواهد کرد. جهت متوقف کردن فیلم می‌توانید از تابع stop استفاده کنید.

```
stop();
```

تابع stop هیچ آرگومان ورودی ندارد. با استفاده از دستور stop می‌توان حرکت را در خط زمان متوقف کرد و یا به سادگی اجرای یک فیلم ویدئویی را متوقف ساخت. برای استفاده از تابع فوق ابتدا باید قاب، دکمه و یا فیلم ویدئویی مورد نظر خود را کلیک کنید سپس از فرمان Window→Actions را صادر کنید تا لوحه Actions باز شود. شکل (۴-۸).



شکل (٤-٨) نمایی از یک مثال که از تابع stop استفاده می‌کند

زمانی که فیلم خود را اجرا کنید مشاهده خواهید کرد در جایی که از دستور stop استفاده کرده‌اید فیلم متوقف خواهد شد.

اگر می‌خواهید یک فیلم ویدئویی را متوقف کنید باید بعد از آن که فیلم ویدئویی را روی محیط کاری منتقل کردید یک نام نمونه برای آن در نظر بگیرید. برای این منظور لوحه مربوط به فیلم ویدئویی در قسمت InstanceName یک نام برای آن وارد کنید (شکل ۵-۸).

نام نمونه فیلم ویدئویی ←

شکل (٥-٨) نمایی از لوحه Properties یک کلیپ ویدئویی

مثال ۱: می‌خواهیم زمانی که کاربر بر روی دکمه stop کلیک کرد، حرکت دورانی سیب متوقف شود. شکل (۶-۸).

شکل (٦-٨)

۱. یک فیلم ویدئویی ایجاد می‌کنیم و در آن تصویر سیب را رسم کرده و با روش Motion Tween یک حرکت دورانی برای آن ایجاد می‌کنیم.

۲. یک نماد دکمه همانند شکل (۶-۸) ایجاد می‌کنیم.

۳. نماد فیلم ویدئویی و نماد دکمه را از لوحه کتابخانه به محیط کاری منتقل می‌کنیم.

۴. روی نماد فیلم ویدئویی کلیک کرده و در لوحه Properties، نام نمونه آن را My_Mc می‌گذاریم.

۵. بر روی دکمه کلیک کرده و لوحه Action مربوط به آن را باز می‌کنیم. سپس خطوط زیر را در آن قرار می‌دهیم.

```
on (release)
{
 tellTarget ("my_MC")
 {
 stop () ;
 }
}
```

دستور tellTarget (“my_MC”) به منظور تعیین فیلم ویدئویی که قرار است روی آن تابعی (در این مثال تابع Stop) اعمال شود، به کار می‌رود. دستور stop باعث می‌شود، فیلم ویدئویی my_MC که شامل سیب با حرکت دورانی می‌باشد، بعد از فشردن دکمه متوقف شود.

۲-٤-۸-پخش کردن یک فیلم

زمانی که شما اجرای یک فیلم را متوقف می‌کنید، فیلم متوقف می‌ماند تا زمانی که دوباره فرمان اجرای آن صادر شود. برای اجرای فیلم از قاب جاری از تابع play استفاده می‌کنیم.

```
play () ;
```

تابع play هیچ آرگومانی ندارد. با استفاده از تابع play می‌توان پخش فیلم را ادامه داد و یا یک فیلم را از ابتدا به اجرا درآورد. برای استفاده از تابع فوق مانند تابع stop عمل می‌کنیم. یعنی ابتدا قاب یا دکمه مورد نظر خود را انتخاب می‌کنیم سپس لوحه Actions مربوط به آن‌ها را باز می‌کنیم. تابع play را در پنجره script وارد می‌کنیم. به عنوان مثال اگر تابع play را در رویداد on (release) یک دکمه قرار داده باشید، زمانی که فیلم را با استفاده از کلیدهای Ctrl+enter در Flash player اجرا می‌کنید، مشاهده خواهید کرد که در هنگام کلیک بر روی دکمه، اجرای فیلم شروع می‌شود.


نکته

برای اجرای یک فیلم ویدئویی نیز باید از نام نمونه آن استفاده کنید.

مثال ۲: به مثال قبل دکمه جدیدی اضافه می‌کنیم به طوری که با کلیک کاربر بر روی آن حرکت دورانی سیب، که با کلیک دکمه Stop متوقف شده بود، دوباره آغاز شود.

۱. نماد دکمه ای جدیدی ایجاد کرده سپس آن را به محیط کاری منتقل می‌کنیم.

۲. بر روی دکمه کلیک کرده و لوحه Action مربوط به آن را باز می‌کنیم، سپس خطوط زیر را در آن قرار می‌دهیم.

```
(on (release
{
 tellTarget (“My_MC”)
 {
 play () ;
 }
}
```


نکته

علاوه بر روش ذکر شده در قسمت قبل (استفاده از تابع play به همراه تابع tellTarget) می‌توانید از ساختار زیر برای اجرای یک فیلم ویدئویی استفاده کنید که به جای کلمه InstanceName باید نام نمونه فیلم ویدئویی را ذکر کنید.

```
InstanceName.play () ;
```

به مثال زیر توجه کنید.

My_MC.play () ;

۳-٤-۸پرش به یک قاب

با استفاده از توابع goto می‌توان حلقه‌های مختلفی را ایجاد کرد و یا هد اجرا را به یک نقطه خاص از فیلم انتقال داد. تابع goto دو شکل دارد که عبارتند از:

gotoAndStop (frame) ;

gotoAndPlay (frame) ;

آرگومان frame بیانگر شماره قابی است که می‌خواهید اجرا را به آن منتقل کنید. بهتر است به جای استفاده از شماره قاب از برچسب قاب استفاده کنید؛ (چگونگی ایجاد برچسب قاب از پنجره Frame در قسمت ۳-۸ توضیح داده شد.) زیرا شماره قاب‌ها با درج، حذف و یا تغییرات دیگر ممکن است عوض شوند ولی برچسب آن‌ها هرگز تغییر نخواهد کرد مگر آن که شما به طور صریح اقدام به تغییر نام قاب نمایید.

به عبارت زیر توجه کنید:

gotoAndstop (_currentframe) ;

این مثال اجرای فیلم را در قاب جاری متوقف می‌کند.

نکته

تابع _currentframe شماره قاب جاری را برمی‌گرداند.

همچنین با قراردادن توابع Goto در یک دکمه می‌توانید به یک قسمت خاص بپرید. برای این منظور کافی است در رویداد on (release) دکمه مورد نظر از یکی از توابع goto استفاده کنید.

٤-٤-۸-باز کردن یک صفحه وب

شما می‌توانید با استفاده از تابع getURL یک صفحه جدید در پنجره مرورگر باز کنید و متغیرهای موجود در فیلم را به یک برنامه انتقال دهید. تابع getURL زمانی مفید است که بخواهیم کاربر را از فیلم به یک صفحه وب دیگر انتقال دهیم. ساختار تابع getURL به شرح زیر است:

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

در پارامتر url آدرس صفحه وب مقصد را قرار می‌دهیم. در قسمت Window محل باز شدن صفحه را معین می‌کنیم که می‌تواند یکی از چهار مقدار زیر را داشته باشد.

- **self_**: صفحه در پنجره جاری باز می‌شود.
- **blank_**: یک صفحه جدید باز می‌شود.
- **parent_**: در parent فریم جاری باز می‌شود.
- **top_**: در بالاترین سطح فریم پنجره جاری باز می‌شود.

به کمک تابع getURL می‌توانیم متغیرها را نیز به آدرس مورد نظر انتقال دهیم. آرگومان variables اختیاری می‌باشد. برای اطلاعات بیشتر در مورد ارسال متغیرها به مراجع پیشرفته‌تر رجوع کنید.

مثال۳: می‌خواهیم زمانی که کاربر بر روی دکمه کلیک کرد صفحه خانگی Yahoo در پنجره جاری باز شود.

۱. یک نماد دکمه جدید ایجاد می‌کنیم و بعد از قرار دادن آن در محیط کاری در لوحه Action آن خطوط زیر را وارد می‌کنیم.

```
on (release)
{
 getURL ("http: //www.yahoo.com", "_self") ;
}
```

تابع getURL در لوحه Actions از طریق Global function→Browse network قابل دسترسی می‌باشد.

۵-٤-۸-ایجاد یک شیئی قابل کشیدن[1]

با استفاده از تابع startDrag می‌توانید اشیایی بسازید که توسط کاربران قابل کشیدن باشند. یک فیلم ویدئویی و یا یک دکمه می‌تواند قابل کشیدن باشند. ساختار تابع startDrag به شرح زیر است:

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

در آرگومان target آدرس فیلم ویدئویی یا دکمه‌ای که قرار است قابل کشیدن باشد را معین می‌کنیم (همچنین می‌توانید به جای نشانی از کلمه this استفاده کنید؛ قابل ذکر است که همواره this به شیئی اشاره می‌کند که در حال نوشتن رویداد برای آن هستیم).

در آرگومان lock یک مقدار منطقی[2] قرار می‌گیرد. اگر مقدار lock برابر true باشد، با کلیک کردن کاربر روی شی، مبدأ شی دقیقاً در زیر ماوس قرار خواهد گرفت و اگر این مقدار false باشد، از هر نقطه‌ای که کاربر کلیک کند، کشیدن شی از همان نقطه شروع می‌شود.

با استفاده از آرگومان‌های bottom, top, right, left می‌توانیم مستطیلی فرضی درست کنیم که کاربر فقط در داخل آن بتواند عمل کشیدن را انجام دهد. برای معین کردن محدوده باید از مقادیر عددی استفاده کنیم. اندازه‌ها نسبت به ابعاد فیلم ویدئویی می‌باشند در نتیجه گوشه سمت چپ دارای مقدار صفر است و هر چه به پایین برویم مقدار آن افزایش می‌یابد.

نکته شایان توجه آن است که در یک لحظه فقط یک شی می‌تواند کشیده شود. یک شی تا زمانی قابل کشیدن می‌ماند که از تابع stopDrag استفاده نشده باشد و یا عمل کشیدن یک شی دیگر توسط تابع startDrag آغاز نشده باشد.

دستور stopDrag برای متوقف کردن عمل کشیدن می‌باشد.

```
stopDrag () ;
```

مشاهده می‌کنید که این دستور هیچ آرگومانی ندارد.

مثال ۴: می‌خواهیم کاربر در هنگام کلیک روی یک دکمه و تا زمان رها کردن آن بتواند دکمه را بکشد و زمانی که دکمه ماوس را رها کرد عمل کشیدن متوقف گردد.

۱. ابتدا دکمه موردنظر را در محیط کاری قرار می‌دهیم.
۲. در لوحه Actions آن دستورات زیر را وارد می‌کنیم:

```
on (press)
{
 StartDrag (this, false, 500,500) ;
}
on (release)
{
 stopDrag () ;
}
```

توابع startDrag و stopDrag در لوحه Actions از طریق Global Function→Movie clip Control قابل دسترسی می‌باشند.

٦-٤-۸-Load و unload کردن یک فیلم

تابع loadMovie این امکان را فراهم می‌سازد تا یک فیلم جدید را بار گذاری کرده و آن را جایگزین فیلم جاری نماییم.

```
loadMovie ("url", target[, variables]) ;
```

با آرگومان url نشانی فایل SWF که قرار است اجرا شود را معین می‌کنیم. که این نشانی می‌تواند به صورت نسبی یا مطلق باشد.

---

[1] Drag
[2] true or false


نکته

معمولاً دو نوع نشانی‌دهی وجود دارد، که عبارتند از: نسبی و مطلق. نشانی مطلق، نشانی کامل می‌باشد و همیشه یکسان است مگر این که محل فایل تغییر کند مانند C: \main\test اما نشانی نسبی نشانی است که نسبت به محل جاری (پوشه جاری) محل فایل را مشخص می‌کند بنابراین اگر پوشه جاری تغییر کند دیگر معتبر نخواهد بود مانند main\test همان نشانی مثال قبلی است به شرطی که پوشه جاری C: \ باشد.

در آرگومان target عدد صفر را وارد می‌کنیم. عدد صفر بیانگر آن است که فیلم اصلی و فیل می‌که قرار است توسط این تابع بار گذاری شود هر دو در سطح ریشه قرار دارند. اگر در آرگومان target، نام یک شی را وارد کنیم، فیلم موجود در url، در داخل آن شی بار گذاری خواهد شد. آرگومان variable دقیقاً مانند آرگومان variable در تابع getURL عمل می‌کند.

مثال ۵: می‌خواهیم زمانی که کاربر روی نماد دکمه کلیک کرد فیلم Sample.swf به نمایش درآید. برای این منظور دکمه را انتخاب می‌کنیم و در لوحه Actions آن خطوط زیر را وارد می‌کنیم.

```
on (release)
{
 loadMovie ("sample.Swf", 0) ;
}
```

همچنین دقت کنید که هر دو فایل (فایل اصلی و فایلی که قرار است بار گذاری شود) در یک پوشه ذخیره شده باشند زیرا همان طور که می‌بینید مسیر url به کار رفته در فراخوانی تابع LoadMovie نسبی است.

با تابع unloadMovie می‌توان فیل می‌را که توسط loadMovie بار گذاری شده است را آزاد کرد.

```
unloadMovie (target) ;
```

این تابع، فیل می‌را که در آرگومان target بار گذاری شده را از حافظه خارج می‌کند.

خط زیر باعث می‌شود فیل می‌که در سطح صفر بار گذاری شده از حافظه پاک شود.

```
unloadMovie (0) ;
```

۵-۸- کار با شی

۱-۵-۸-تغییر رنگ شی‌ها

می‌توان با شی Color تغییراتی روی صفحه اعمال کرد. با تابع Color می‌توان رنگ فیلم‌های ویدئویی در حال اجرا را تغییر داد. تغییر رنگ یک فیلم ویدئویی در دو فرآیند صورت می‌گیرد. ابتدا شما باید یک متغیر از نوع Color ایجاد کنید، سپس با تابع setRGB که تابعی از شی Color می‌باشد رنگ مورد نظر خود را تعیین کنید. به صورت زیر می‌توان یک شی از نوع Color ایجاد کرد.

```
objectName = new Color (target) ;
```

شناسه *objectname* نام متغیری است که از نوع color تعریف می‌شود و آرگومان target نام فیلم ویدئویی مورد نظر را مشخص می‌کند. شکل کلی تابع setRGB به شرح زیر است.

```
objectName.setRGB (0xrrggbb) ;
```

شناسه *objectName* نام متغیری از نوع Color می‌باشد که در مرحله قبل تعریف کرده‌ایم.

در قسمت rrggbb رنگ مورد نظر خود را به صورت هگزا دسیمال (مبنای ۱۶) معین می‌کنیم، می‌توانید برنامه رنگ دلخواه خود را از متن موجود در جعبه رنگ فلش کپی کنید.

مثال ۶: می‌خواهیم زمانی که کاربر بر روی دکمه کلیک می‌کند، دایره تغییر رنگ بدهد. (شکل۷-۸).

شکل (۷-۸) نمایی از دایره و دکمه که با کلیک روی دکمه رنگ دایره عوض می‌شود

۱. ابتدا یک نماد فیلم ویدئویی ایجاد می‌کنیم و نام آن را Circle می‌گذاریم و بعد داخل فیلم ویدئویی شکل یک دایره را رسم می‌کنیم و از آن خارج می‌شویم.
۲. اکنون فیلم ویدئویی را روی محیط کاری می‌کشیم و در لوحهProperties نام نمونه آن را circle_MC می‌گذاریم.
۳. یک دکمه ایجاد می‌کنیم.
۴. لایه جدیدی ایجاد می‌کنیم و در قاب اول آن دکمه را قرار می‌دهیم.
۵. اکنون دکمه را انتخاب کرده و در لوحه Action آن خطوط زیر را وارد می‌کنیم:

```
on (release)
{
 my_Color = new Color (circle_MC) ;
 my_Color.SetRGB (0xFF9900) ;
}
```

۶. اکنون فیلم را تست می‌کنیم. مشاهده خواهید کرد که با کلیک روی دکمه رنگ دایره نارنجی می‌شود.

۲-۵-۸-پنهان و آشکار کردن اشاره‌گر ماوس

اشاره‌گر ماوس معمولاً یک جهت نماد کوچک است. شما می‌توانید توسط آن ابزارهای مختلفی را بردارید (به این صورت که با کلیک روی هر ابزار شکل اشاره‌گر به نماد آن ابزار تغییر می‌کند). برداشتن یک ابزار به این معنی است با استفاده از اشاره‌گر ماوس می‌توانید توسط ابزار عملیات مورد نظر خود را روی اشیای فلش انجام دهید. با تابع Mouse می‌توانید معین کنید که اشاره‌گر ماوس به نمایش درآید یا خیر. با مخفی کردن آن شما می‌توانید اشاره‌گر را خودتان بسازید.

از تابع Mouse.hide برای مخفی کردن اشاره‌گر استفاده می‌شود.

```
Mouse.hide () ;
```

مشاهده می‌کنید که این تابع هیچ آرگومانی ندارد.

از تابع Mouse.show جهت نمایش دادن اشاره‌گر ماوس استفاده می‌شود. این تابع نیز هیچ آرگومانی ندارد.

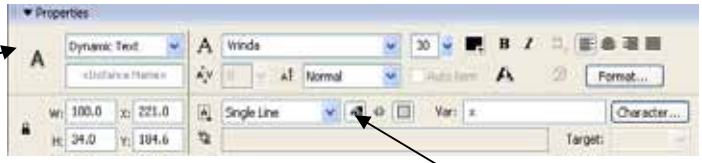
```
Mouse.show () ;
```

با خصوصیات Xmouse_ و Ymouse_ می‌توان موقعیت جاری اشاره‌گر را پیدا کرد. به یاد داشته باشید که این خصوصیات فقط خواندنی هستند، یعنی نمی‌توانید مقادیر آن‌ها را تغییر دهید.

مثال ۷: می‌خواهیم زمانی که کاربر اشاره‌گر ماوس را حرکت می‌دهد، موقعیت جاری آن نشان داده شود.

شکل (۸-۸) نمایی دو کادر متن x و y

۱. ابتدا یک فیلم ویدئویی ایجاد می‌کنیم و در داخل آن دو جعبه متن پویا قرار می‌دهیم. برای انجام این عمل روی ابزار متن کلیک می‌کنیم و در لوحه Properties مربوط به ابزار متن ابتدا از لیست کشویی گزینه Dynamic Text را انتخاب می‌کنیم سپس در کادر Var نام دلخواه را برای آن وارد می‌کنیم. شکل (۹-۸)



۹-۸) لوحه Properties ابزار متن

همان طور که در شکل مشاهده می‌کنید، یک جعبه متن پویا با نام x ایجاد شده است. اکنون یک جعبه متن پویای دیگر ایجاد می‌کنیم و نام انرا y می‌گذاریم

۲. بعد از ایجاد دو جعبه متن پویا، از فیلم ویدئویی خارج می‌شویم.

۳. در این مرحله فیلم ویدئویی را به درون محیط کاری می‌کشیم و لوحه Action مربوط به آن را باز کرده و خطوط زیر را وارد می‌کنیم. هنگامی‌که می‌خواهیم فعلی[1] را به یک فیلم ویدئویی نسبت دهیم، برای اجرای آن باید از فعل[2] onClipEvent استفاده کنیم. همانند فعل on، میان دو پرانتز باید نوع رویداد را قرار دهیم. مثلاً در این جا از رویداد mouseMovie استفاده کردیم یعنی در هنگام حرکت ماوس خطوط داخل دو علامت آکولاد اجرا شود. (در قسمت‌های بعدی به معرفی رویدادهای دیگری از این فعل می‌پردازیم).

```
onClipEvent (mouse Movie)
{
 x = _xmouse;
 y = _ymouse;
}
```

نکته: فعل فوق در مسیر Movie Clip Control در گزینه‌های سمت چپ لوحه Actions قرار دارد.

۴. اکنون فیلم را اجرا کنید. مشاهده خواهید کرد که با حرکت ماوس موقعیت جاری آن در روی صفحه نشان داده می‌شود.

۳-۵-۸-استفاده از تابع Date

قبل از آنکه بخواهیم به کمک تابع‌های موجود، روز و سال و ماه سیستم را فراخوانی کنیم، باید یک متغیر از نوع Date ایجاد کنیم.

objectName = new Date () ;

شناسه *objectName* نام متغیری است که از نوع Date تعریف می‌شود.

اکنون می‌توانیم از خطوط زیر برای به دست آوردن روز، ماه و سال سیستم استفاده کنیم:

objectName.getDay () ;

تابع فوق روز جاری را برمی‌گرداند. خروجی این تابع یک عدد صحیح بین ۱ تا ۳۱ به معنای روزهای ماه می‌باشد.

objectName.getMonth () ;

تابع فوق ماه جاری را برمی‌گرداند. خروجی این تابع مقداری بین ۰ تا ۱۱ می‌باشد که عدد ۰ نشان‌دهنده ماه ژانویه می‌باشد.

objectName.getFullYear () ;

با استفاده از تابع فوق یک عدد چهار رقمی به معنای سال حاضر برگردانده خواهد شد.

مثال ۸: می‌خواهیم زمانی که فیلم را اجرا می‌کنیم تاریخ سیستم روی صفحه نمایان شود (شکل ۱۰-۸).

---

[1] Action
[2] Action

شکل (۱۰-۸) نمایی از خروجی برنامه که تاریخ جاری را نشان می‌دهد

۱. ابتدا یک فیلم ویدئویی ایجاد می‌کنیم و در داخل آن یک جعبه متن پویا با نام my_Date ایجاد می‌کنیم.
۲. اکنون فیلم ویدئویی را درون صفحه می‌کشیم و لوحه اکشن مربوط به آن را باز کرده و خطوط زیر را در آن وارد می‌کنیم.

```
onClipEvent (enterFrame)
{
 current_Date = new Date () ;
 Day = current_Date.getDay () ;
 Month = current_Date.getMonth () ;
 Year = current_Date.getYear () ;
 my_Date = Day + "/" + Month + "/" + Year;
}
```

رویداد enterFrame باعث می‌شود زمانی که کنترل فیلم وارد قاب مربوطه شد، خطوط داخل آکولاد اجرا شوند. در واقع enterFrame یک حلقه دائمی است که تا زمانی که فیلم در حال اجرا می‌باشد، خطوط داخل این رویداد مکررا اجرا می‌شوند. سرعت تکرار این رویداد همان Frame Rate فیلم فلش می‌باشد.

۴-۵-۸-تنظیم مقدار تاریخ

می‌توان با کمک تابع‌های زیر، مقادیر تاریخ را تغییر داد.

ObjectName.setMonth (month[, day]) ;

آرگومان month مقداری بین ۰ تا ۱۱ خواهد بود و شماره ماه را مشخص می‌کند.

آرگومان day شماره روز را معین کند.

ObjectName.setDate (day) ;

آرگومان day روز را مشخص می‌کند و باید مقداری بین ۱ تا ۳۱ باشد.

objectName.setFullYear (year[, month[, day]]) ;

در تابع فوق می‌توانیم سال و ماه و روز را تنظیم کنیم.

یک فیلم ویدئویی ایجاد کنید و در آن سه جعبه متن پویا ایجاد کنید، سپس فیلم ویدئویی را بـه درون محیط کاری بکشید و درون لوحه Action از توابع تنظیم تاریخ در رویداد onClipEvent ((enterFrame استفاده کنید.

۶-۵-۸-استفاده از مجموعه Key

با تابع‌های مختلف مجموعه Key می‌توان آخرین کلید فشرده شده را شناسایی کرد یعنی می‌توانید کاری بکنید که فیلم توسط صفحه کلید کنترل شود. اهمیت مجموعه Key در برنامه‌نویسی بازی‌ها بسیار بالا می‌باشد. در Action script به هر کلید موجود در صفحه کلید یک برنامه اختصاص داده شده است. در واقع این کدها در میان تمام سیستم عامل‌ها مشترک هستند.

با تابع Key.getCode می‌توانیم برنامه آخرین کلید فشرده شده را به دست آوریم.

Key.getCode () ;

همان طور که مشاهده می‌کنید تابع Key.getCode ( ) فاقد آرگومان می‌باشد. برای مثال برنامه کلید k عدد 75 می‌باشد. یعنی اگر کاربر کلید k را فشار دهد تابع فوق عدد 75 را برمی‌گرداند.

x = key.getCode () ;

۱۲۷ کاراکتر اول به عنوان ASCII برنامه شناخته شده‌اند، تابع Key.getAscii برنامه اسکی آخرین کلید فشرده شده را برمی‌گرداند. یعنی برای حرف k عدد 107 را برمی‌گرداند. شکل کلی این تابع به صورت زیر است:

```
Key.getAscii () ;
```


نکته

در برنامه ASCII بین حروف بزرگ و کوچک تفاوت‌هایی وجود دارد. برای مثال برنامه حرف A با برنامه حرف a متفاوت است.

با تابع Key.isDown می‌توانیم نوع یک کلید خاص را کنترل کنیم که آیا فشار داده شده است یا خیر؟ اگر فشار داده شده باشد مقدار true و در غیر این صورت مقدار false برخواهد گشت.

```
Key.isDown (Keycode) ;
```

با آرگومان keycode برنامه کلید مورد نظر را معین می‌کنیم.

همچنین برای کنترل کردن کلیدهایی چون caps lock یا num lock باید از تابع Key.isToggled استفاده کنیم:

```
key.Istoggled (keycode) ;
```

مقداری که باید به آرگومان keycode برای چک کردن وضعیت Capslock بدهید، مقدار20 و مقداری که برای چک کردن وضعیت Numlock لازم است، عدد 144 می‌باشد.

مثال ۹: می‌خواهیم برنامه‌ای بنویسیم به طوری که هر بار که کاربر کلیدی را فشار می‌دهد، کد اسکی، برنامه مجازی، آن کلید برگردانده شود همچنین بررسی کند که آیا کلید فشرده شده، حروف L بوده است یا خیر؟ و آیا کلید cap lock فشرده شده است یا خیر. شکل (۱۱-۸).

شکل (۱۱-۸) نمایی از خروجی برنامه نمایش کدهای کلیدهای صفحه کلید

۱. ابتدا یک فیلم ویدئویی ایجاد می‌کنیم و در داخل آن ۴ متن پویا با نام‌های A_code، check,V_code و caps_lock قرار می‌دهیم.

۲. فیلم ویدئویی را روی محیط کاری قرار می‌دهیم و در لوحه اکشن آن خطوط زیر را وارد می‌کنیم.

```
onClipEvent (keyDown)
{
 A_code = Key.getAscii () ;
 V_code = Key.getCode () ;
 if (Key.isDown (76) )
 {
 Check = "ture";
 }
 else
 {
 Check = "false";
 }
 if (Key.isToggled (20) )
 {
 caps_lock = "on";
 }
 else
 {
 caps_lock = "off";
 }
}
```

رویداد keyDown باعث می‌شود زمانی که کلیدی از صفحه کلید فشار داده شد، خطوط مابین دو آکولاد اجرا شوند.
جدول (۱-۸) حاوی برنامه کلیدهای رزرو شده در مجموعه key است (به ترتیب حروف الفبا):

نام کلید	نحوه نمایش تابع	مقدار عددی
Backspace	Key.BACKSPACE	8
Capslock	Key.CAPSLOCK	20
ctrl	Key.CONTROL	17
Down (ˉ)	Key.DOWN	40
End	Key.END	35
Enter	Key.ENTER	13
Esc	Key.ESCAPE	27
Home	Key.HOME	36
Insert	Key.INSERT	45
Left (¬)	Key.LEFT	37
Page Down	Key.PADN	34
Page up	Key.PGUP	33
(Right (→	Key.Right	39
shift	Key.SHIFT	16
space	Key.SPACE	32
Tab	Key.TAB	9
Up (-)	Key.UP	38

جدول (۱-۸)

۷-۵-۸-استفاده از مجموعه Sound

کلاس Sound امکانات مختلفی را جهت کنترل پخش صدا در اختیار شما قرار می‌دهد. امکاناتی مثل قطع یا پخش صدا، افزایش یا کاهش صدا و یا تعیین بلندگوی پخش و ....
برای کار با توابع کلاس Sound ابتدا باید یک متغیر از نوع Sound تعریف کنیم:

Sound_name = new Sound () ;

بعد از انجام این مرحله، باید فایل صوتی مورد نظر خود را به این متغیر بچسبانید. تابع attachSound طبق ساختار زیر، فایل صوتی شما را به متغیر الحاق می‌کند.

Sound_name.attachSound ("*idName*") ;

شناسه *sound_Name* نام همان متغیری است که در مرحله قبل تعریف کرده‌ایم.
آرگومان *idName* نام فایل صوتی دلخواه است که می‌خواهیم آن را به متغیر *sound_Name* الحاق کنیم. دقت کنید که فایل صوتی مورد نظر را باید از قبل در کتابخانه قرار بدهیم.
بعد از این مرحله، از تابع start برای اجرای آن استفاده می‌کنیم.

Sound_name.start ([secondoffset, loop]) ;

آرگومان secondoffset زمان شروع فایل صوتی را مشخص کند (یعنی اگر فایل صوتی شما ۲۰ ثانیه باشد و شما آرگومان secondoffset را به ۱۰ تنظیم کرده باشید، فایل صوتی از وسط پخش می‌شود). در آرگومان loop تعداد دفعات تکرار پخش فایل صوتی را مشخص می‌کنیم.
همچنین برای متوقف کردن پخش فایل صوتی باید از تابع stop استفاده کرد.

Sound_name.stop () ;

در صورتی که بخواهیم کلیه صداهای در حال پخش را متوقف سازیم از تابع stopAllSound استفاده می‌کنیم.

مثال ۱۰: می‌خواهیم زمانی که کاربر روی دکمه کلیک کرد صدا پخش شود و زمانی که دوباره روی آن کلیک کرد اجرای صدا متوقف شود. شکل (۱۲-۸).

شکل (۱۲-۸) نمایی از خروجی برنامه پخش و قطع صدا

۱. ابتدا فایل صوتی مورد نظر را در کتابخانه قرار می‌دهیم.

۲. دکمه مورد نظر را روی محیط کاری قرار می‌دهیم و در لوحه اکشن آن خطوط زیر را وارد می‌کنیم:

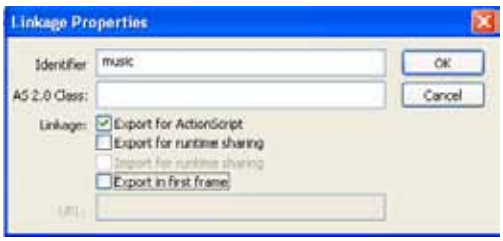
```
on (release)
{
 if (status == "false")
 {
 my_sound = new Sound () ;
 my_sound.attachSound ("music") ;
 my_Sound.start (0, 10) ;
 Status = "true";
 }
 else
 {
 my_sound.stop () ;
 status = "false";
 }
}
```

۳. در این مرحله می‌خواهیم مقدار اولیه متغیر status را تعیین کنیم. برای همین منظور روی قاب یک کلیک می‌کنیم و لوحه اکشن مربوط به آن را باز کرده و خط زیر را تایپ می‌کنیم:

```
Status = "false";
```

۴. اکنون لوحه اکشن را می‌بندیم.

۵. در داخل کتابخانه روی نماد صدا راست کلیک می‌کنیم و از منوی باز شده فرمان Linkage را صادر می‌کنیم تا کادر زیر ظاهر شود (شکل۱۳-۸).



شکل (۱۳-۸) نمایی از پنجره Linkage Properties

۶. در کادر باز شده فقط گزینه Export for ActionScript را تیک زده و OK را کلیک می‌کنیم تا کادر بسته شود.

۷. اکنون فیلم را اجرا کنید.

۸-۵-۸-تنظیم صدا و تعادل پخش‌کننده‌ها

Action script مهیا کننده امکاناتی نظیر تنظیم صدا، ایجاد تعادل در بلندگوهای پخش‌کننده صدا می‌باشد. می‌توانیم از تابع setVolume جهت تنظیم صدا استفاده کنیم:

Sound_Name.setVolume (volume) ;

شناسه Sound_Name نام متغیری که از نوع Sound تعریف شده است. آرگومان volume مقداری است بین صفر تا ۱۰۰ که اندازه صدا را کنترل می‌کند، ۱۰۰ بلندترین صدا و صفر به معنای عدم پخش صدا می‌باشد (پیش‌فرض عدد ۱۰۰ می‌باشد).

همچنین برای بدست آوردن مقدار جاری اندازه صدا از تابع getVolume استفاده می‌کنیم:

Sound_Name.getVolume () ;

تابع فوق عددی بین ۰ تا ۱۰۰ را که به معنای سطح صدا می‌باشد، را برمی‌گرداند.

مثال ۱۱: در این قسمت می‌خواهیم به فیلم ویدئویی که توسط فلش در مثال قبلی ایجاد کردیم، دکمه‌هایی اضافه کنیم به گونه‌ای که وقتی کاربر روی این دکمه کلیک کرد اندازه صدای فایل صوتی روی صفحه ظاهر شود.

۱. فیلم فلش قبلی را باز کرده و یک جعبه متن پویا به نام my_Volume ایجاد می‌کنیم.
۲. یک دکمه جدید ایجاد کرده و آن را روی محیط کاری قرار می‌دهیم سپس در لوحه اکشن آن خطوط زیر را وارد می‌کنیم.

```
on (release)
{
 Vol = my_Sound.getVolume () ;
 if (vol < 95)
 {
 my_sound.setVolume (vol + 5) ;
 my_Volume = my_Sound.getVolume () ;
 }
}
```

قطعه برنامه بالا، اندازه صدای فایل صوتی را بررسی می‌کند. در صورتی که اندازه صدا عددی کوچکتر از ۹۵ باشد، ۵ واحد به آن اضافه می‌کند. در غیر اینصورت اندازه فایل صوتی را ۱۰۰ در نظر می‌گیرد.

با استفاده از تابع setPan می‌توانیم کنترل کنیم که صدا چگونه در پخش‌کننده تقسیم شود:

Sound_Name.setpan (pan) ;

آرگومان pan مقداری است بین ۱۰۰- و ۱۰۰، که مقدار ۱۰۰- به معنای استفاده از پخش‌کننده سمت چپ می‌باشد و نقطه مقابل آن، مقدار ۱۰۰ به معنای پخش‌کننده سمت راست می‌باشد و مقدار صفر به معنای استفاده از هر دو پخش‌کننده به یک میزان می‌باشد.

با استفاده از تابع getPan مقدار جاری آن را می‌توان به دست آورد:

Sound_Name.getPan () ;

در واقع عددی بین ۱۰۰ و ۱۰۰- را برمی‌گرداند.

مثال ۱۲: می‌خواهیم زمانی که کاربر روی دکمه کلیک کرد صدا فقط از پخش‌کننده سمت راست پخش گردد و اندازه آن نیز به ۱۰۰ افزایش پیدا کند.

۱. در قاب اول، کلیک کرده و لوحه اکشن مربوط به آن را باز می‌کنیم و خطوط زیر را در آن وارد می‌کنیم.

```
my_Sound = new sound () ;
my_sound.attachSound ("music") ;
my_sound.start (0, 10) ;
my_sound.setVolume (20) ;
my_sound.setPan (0) ;
```

دقت کنید که حتماً باید روی فایل صوتی واقع در کتابخانه کلیک راست کرده و گزینه Linkage را انتخاب کنید و عنوان "music" را به سمبل صوتی خود نسبت دهید.

۲. اکنون یک نماد دکمه ایجاد کرده و آن را روی محیط کاری قرار می‌دهیم سپس در لوحه اکشن آن خطوط زیر را وارد می‌کنیم.

```
on (release)
{
 my_sound.setVolume (100) ;
 my_sound.setPan (100) ;
}
```

۳. فیلم را اجرا کنید.

۶-۸-تنظیم‌های خصوصیات فیلم ویدئویی

۱-٦-۸حرکت دادن یک کلیپ ویدئویی در عرض صفحه

برای آن که بتوانیم شی‌ای را در میان صفحه به سمت چپ و یا راست حرکت دهیم باید از خصوصیت x_ استفاده کنیم.

قبل از آن که بتوانید از خصوصیت x_ استفاده کنید باید به فیلم ویدئویی خود یک نام نمونه اختصاص بدهید (برای نام‌گذاریکلیپ ویدئویی از لوحه properties مربوط به آن استفاده کنید).

InstanceName._x = value;

که شناسه InstanceName همان نام فیلم ویدئویی می‌باشد. با آرگومان value تعداد پیکسل‌های لازم را برای فاصله گرفتن از سمت چپ معین می‌کنیم. سمت چپ محیط کاری دارای مقدار صفر می‌باشد. همچنین می‌توانیم مقدار جاری خصوصیت x_ را بخوانیم. ساختار دسترسی به مقدار جاری خصوصیت x_ به صورت زیر است:

InstanceName._x;

که شناسه InstanceName برای معرفی کردن نمونه مورد نظر می‌باشد. به مثال زیر توجه کنید:

SampleMC._x = SampleMC._x + 10;

این مثال به موقعیت جاری فیلم ویدئویی ذکر شده، ۱۰ نقطه اضافه می‌کند.

مثال ۱۳: می‌خواهیم هر بار که کاربر روی دکمه go کلیک می‌کند ماشین ۵ نقطه به سمت جلو حرکت کند و هر بار که دکمه back را کلیک کند ۵ نقطه به عقب برگردد (شکل۱۴-۸).

شکل (۱٤-۸) نمایی از خروجی برنامه حرکت اتومبیل

۱. یک فیلم ویدئویی ایجاد می‌کنیم و در داخل آن تصویر یک ماشین را قرار می‌دهیم.

۲. فیلم ویدئویی را روی محیط کاری می‌کشیم و در لوحه Properties در کادر Instancename نام نمونه آن را Car می‌گذاریم.

۳. اکنون یک لایه جدید اضافه می‌کنیم و در قاب اول آن نماد دکمه‌های را که از قبل طراحی کرده‌ایم از لوحه کتابخانه روی محیط کاری می‌کشیم.

۴. دکمه را انتخاب کرده و لوحه اکشن مربوط به آن را باز می‌کنیم و خطوط زیرا را در آن قرار می‌دهیم (دکمه حرکت به سمت جلو).

```
on (release)
{
 Car._x = Car._x + 5;
}
```

۵. اکنون لایه سوم را ایجاد کرده و در آن نماد دکمه دیگری را قرار داده و لوحه اکشن مربوط به آن را باز می‌کنیم. خطوط زیر را در آن وارد می‌کنیم (دکمه حرکت به سمت عقب).

```
on (release)
{
 Car._x = car._x - 5;
}
```

۶. اکنون فیلم را اجرا می‌کنیم.

۲-۶-۸-بالا و پایین کردن کلیپ ویدئویی

با استفاده از خصوصیت y_ می‌توانیم موقعیت فیلم ویدئویی را در ارتفاع تغییر دهیم. با این خصوصیت می‌توان فیلم ویدئویی را به بالا و پایین حرکت داد.

همانند خصیصه x_ در قسمت InstanceName باید نام نمونه فیلم ویدئویی را در ساختار زیر بکار ببریم.

InstanceName._y = *value*;

ارگومان value میزان فاصله از بالای صفحه را به نقطه بیان می‌کند. بالای صفحه دارای مقدار صفر می‌باشد. در کامپیوتر بر خلاف سیستم مختصات ریاضی، هر چه از مبدأ y_ (که بالای صفحه است) پایین‌تر برویم به مقدار y_ افزوده می‌شود. همچنین می‌توانیم مانند خصیصه x_، در این جا نیز به مقدار جاری خصوصیت y_ دسترسی داشته باشیم.


تمرین

مثال قبل را بازنویسی کنید ولی این بار ماشین را به سمت بالا و پایین حرکت دهید.

۳-۶-۸-مخفی کردن یک کلیپ ویدئویی

با کمک خصوصیت visible_ می‌توان یک فیلم ویدئویی را مخفی کرد. این خصوصیت زمانی مفید خواهد بود که بخواهیم یک فیلم ویدئویی در ابتدای اجرا در صفحه ظاهر نباشد ولی به محض رسیدن هد اجرا به یک قاب خاص فیلم ویدئویی قابل مشاهده شود.

InstanceName._visible = *booleanvalue*;

همانند قبل، شناسه InstanceName نام فیلم ویدئویی می‌باشد. booleanvalue خصوصیت visible_ را مقدار دهی می‌کند، که این مقدار می‌تواند true و یا false باشد. اگر true باشد فیلم ویدئویی قابل مشاهده است و اگر false باشد مخفی خواهد بود.

همچنین می‌توان به مقدار visible_ نیز دسترسی پیدا کرد. برای مثال اگر بخواهیم دکمه‌ای بگذاریم تا برنامه قابل رویت بودن یا نبودن یک فیلم ویدئویی را چک کند و در صورت مخفی بودن فیلم را ظاهر کند و بالعکس، باید مطابق زیر عمل کنیم.

InstanceName._visible;

ابتدا باید اگر مقدار visible_ مساوی با true بود، تبدیل به false شود و در غیر این صورت باید true شود.

به قطعه برنامه زیر که روی اکشن یک دکمه قرار گرفته است توجه کنید:

```
on (release)
{
 x = getproperty ("sampleMC", _visible) ;
 if (x = true)
 {
 setproperty ("sampleMC", _visible, false) ;
 }
 else
 {
 Setproperty ("sampleMC", _visible, true) ;
```

```
}
}
```

زمانی که کاربر دکمه را رها می‌کند برنامه مقدار خصوصیت visible_ برای نمونه sampleMC بازگردانده شده و در متغیر x ذخیره می‌شود. چنانچه این مقدار true باشد (یعنی فیلم ویدئویی روی صفحه ظاهر است) و مقدار آن را false می‌کند و با این کار فیلم ویدئویی پنهان می‌شود و بالعکس.


نکته

با استفاده از تابع getproperty می‌توان مقدار خصوصیت _visible را درباره یک فیلم ویدئویی بدست آورد و با تابع setProperty می‌توان مقدار این خصوصیت را تنظیم کرد.

٤-٦-٨-چرخاندن کلیپ ویدئویی

با استفاده از خصوصیت rotation_ می‌توان یک فیلم ویدئویی را چرخاند. مقدار ورودی آن درجه‌های خواهد بود که فیلم ویدئویی شما نسبت به نقطه‌های در محیط کاری چرخانده خواهد شد. همانند سایر خصوصیات فیلم ویدئویی، در این جا هم باید از نام نمونه فیلم ویدئویی در ساختار این خصوصیت استفاده شود.

InstanceName._rotation = *value*;

value مقدار زاویه چرخش را معین می‌کند.

نکته قابل توجه آن است که اگر چرخش را ۹۰ فرض کنیم، نمونه شما ۹۰ درجه نسبت به مکان اصلی آن به مکان جاری می‌چرخد. قابل ذکر است که در این جا بر خلاف سیستم مختصات ریاضی هر چه جسم به سمت پایین میل کند در واقع زاویه آن بیشتر شده. برای استفاده مانند سیستم ریاضی، باید قرینه مقدار دلخواه خود را وارد خصوصیت کنید.

همچنین می‌توان به مقدار خصوصیت rotation _ به صورت زیر دسترسی داشت.

InstanceName._rotatoin;

برای مثال، اگر بخواهیم به موقعیت جاری فیلم ویدئویی ۹۰ درجه چرخش بدهیم به صورت زیر عمل می‌کنیم:

SampleMC._rotation = sampleMC._rotation + 90;

مثال ۱۴: می‌خواهیم هر بار که کاربر روی دکمه کلیک کرد، فیلم ویدئویی ۳۰ درجه چرخش داشته باشد (شکل (۱۵-۸) ۱۵۳۰-۸).

شکل (۱۵-۸) ۳۰نمایی از خروجی برنامه چرخش کلیپ ویدئویی

۱. یک فیلم ویدئویی ایجاد می‌کنیم و داخل آن شکل مثلث را رسم می‌کنیم.
۲. فیلم ویدئویی را روی محیط کاری قرار می‌دهیم و نام نمونه آن را myrotate می‌گذاریم.
۳. لایه جدیدی ایجاد می‌کنیم و در آن نماد دکمه‌ای را قرار می‌دهیم. لوحه اکشن مربوط به آن را باز کرده و خطوط زیر را وارد می‌کنیم.

```
on (release)
{
 my_rotate._rotation = my_rotate._rotation + 30;
}
```

۴. فیلم را اجرا می‌کنیم.

۵-۶-۸-تغییر پهنای کلیپ ویدئویی

با استفاده از خصوصیت _width می‌توان پهنای فیلم ویدئویی را به صورت نقطه بیان کرد و به آن مقدار داد.

InstanceName._width = *value*;

شناسه InstanceName همان نام نمونه فیلم ویدئویی مورد نظر می‌باشد و value مقدار را به نقطه بیان می‌کند.

می‌توان از آرگومان فوق برای به دست آوردن مقدار جاری یک فیلم ویدئویی استفاده کرد. خط زیر طریقه بدست آوردن مقدار خصوصیت Width را نشان می‌دهد.

Instancename._width;

البته می‌توانید خروجی این تابع را به یک جعبه متن پویا اختصاص دهید تا به این وسیله روی صفحه نمایان شود.


نکته

به یاد داشته باشید که برای به دست آوردن و تنظیم تمام خصوصیات یک فیلم ویدئویی می‌توانید از تابع getproperty و تابع setproperty استفاده کنید.

مثال ۱۵: می‌خواهیم هر بار که کاربر روی دکمه کلیک می‌کنید ۵ نقطه به پهنا اضافه شود. شکل (۱۶-۸)

شکل (۱۶-۸) نمایی از خروجی برنامه افزایش پهنای کلیپ ویدئویی

۱. ابتدا یک فیلم ویدئویی را ایجاد کرده و داخل آن شکل یک مستطیل را رسم می‌کنیم. سپس آن را روی محیط کاری قرار می‌دهیم و نام نمونه آن را Rec می‌گذاریم.

۲. لایه جدیدی ایجاد کرده و در قاب اول آن را انتخاب می‌کنیم. سپس یک دکمه را روی محیط کاری قرار داده و لوحه اکشن مربوط به آن را باز کرده و خطوط زیر را در آن وارد می‌کنیم:

```
on (release)
{
 Rec._width = Rec._ width + 5;
}
```

۳. فیلم را اجرا می‌کنیم.


تمرین

روی اکشن یک دکمه قطعه برنامه زیر را قرار دهید. زمانی که فیلم ویدئویی را اجرا می‌کنید چه عملی اتفاق می‌افتد؟

```
on (keyPress "< left>")
{
 Widthv = getproperty ("sampleMC", _width) ;
 setProperty ("sampleMC", _width, widthv + 10) ;
}
on (keyPress "< right>")
{
 Widthv = getproperty ("sampleMC, _width) ;
 Setproperty ("sampleMC", _width, widthv - 10) ;
}
```

راهنمایی:

۱. به وسیله **on (key press "key")** می‌توان خطوط مابین آکولاد را در صورت فشردن کلید خاصی اجرا کرد.
۲. **sampleMC** نام نمونه یک فیلم ویدئویی است که داخل آن شکل یک مربع رسم شده است.
۳. **widthv** یک متغیر ساده می‌باشد.

۶-۶-۸-تغییر ارتفاع کلیپ ویدئویی

با کمک خصوصیت _height می‌توان ارتفاع یک فیلم ویدئویی را معین کرد. می‌توانیم با استفاده از این خصیصه ارتفاع فیلم ویدئویی را افزایش یا کاهش دهیم.

InstanceName._height = *value*;

شناسه InstanceName همان نام نمونه فیلم ویدئویی مورد نظر می‌باشد و value مقدار ارتفاع را بر حسب نقطه بیان می‌کند.

تمرین: مانند مثال قبل فیلم‌ای توسط فلش ایجاد کنید که هر بار کاربر روی دکمه کلیک کرد ۵ نقطه به ارتفاع آن اضافه شود.

۷-۶-۸-تغییر شفافیت کلیپ ویدئویی

خصوصیت _alpha در هر فیلم ویدئویی درصد شفافیت آن فیلم را بیان می‌کند:

InstanceName._alpha = *value*;

value مقداری است بین صفر تا ۱۰۰ که مقدار صفر فیلم را کاملاً شفاف و در واقع غیرقابل مشاهده می‌کند. مقدار پیش فرض این خصوصیت در هر فیلم ویدئویی ۱۰۰ می‌باشد که شی به طور کامل نمایش داده می‌شود. اگر با استفاده از این خصوصیت شفافیت فیلم ویدئویی را تغییر دهید، به میزان شفافیت آن فیلم ویدئویی، فیلم‌ها و اشیای زیر آن شی قابل رویت خواهند بود برای درک بهتر این خصوصیت به مثال زیر توجه کنید:

مثال ۱۶: می‌خواهیم فیلمی داشته باشیم که با هر بار فشردن دکمه، درصد شفافیت آن ۱۰ واحد کاهش پیدا کند.

۱. یک فیلم ویدئویی ایجاد کرده و در آن تصویر دلخواهی را قرار می‌دهیم.
۲. نماد فیلم ویدئویی را روی محیط کاری قرار می‌دهیم.
۳. نام نمونه آن را myobject قرار می‌دهیم.
۴. در کنار آن یک دکمه قرار داده و قطعه برنامه زیر را در لوحه اکشن آن وارد می‌کنیم.

```
on (release)
{
 myobject._alpha = myobject._alpha - 10;
}
```

۵. فیلم را اجرا می‌کنیم.

به عنوان تمرین دکمه دیگری در کنار این دکمه قرار دهید و در لوحه اکشن آن کدی وارد کنید که درصد شفافیت جسم را مجدداً افزایش دهد و با استفاده از ساختار if نگذارد مقدار _alpha بیشتر یا کمتر از حد تعریف شده، شود.

مثال ۱۷: می‌خواهیم زمانی که کاربر بر روی دکمه کلیک کرد هواپیما شروع به پرواز کند. شکل (۱۷-۸)

شکل (۱۷-۸)

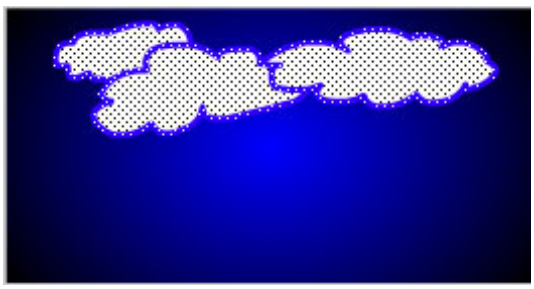
۱. نام لایه را Backgroud می‌گذاریم و در قاب اول آن تصویر یک مستطیل آبی رنگ را رسم می‌کنیم. شکل (۱۸-۶). سپس ستون ۶۰ را به عنوان مقصد در نظر گرفته و درآن یک قاب کلیدی (F6) ایجاد می‌کنیم.

شکل (۱۸-٦)

۲. لایه جدیدی ایجاد می‌کنیم و نام آن را Black clouds می‌گذاریم. سپس در ستون اول آن یک قاب کلیدی درج کرده و تصویر ابرها را رسم می‌کنیم. شکل (۱۹-۸)

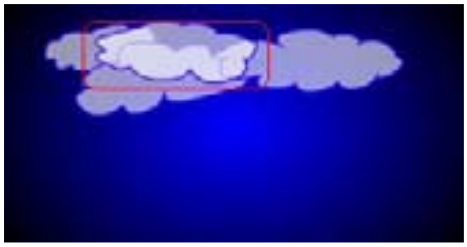
شکل (۱۹-۸)

۳. ستون ۶۰ را به عنوان مقصد در نظر می‌گیریم و در آن کلید F6 را می‌فشاریم. سپس با ابزار انتقال اندازه ابرها را ک می‌بزرگ تر کرده و به سمت راست می‌بریم. شکل (۲۰-۸). در این مرحله قاب‌های میانی، قاب مبدا و قاب مقصد لایه Back clouds را به حالت انتخاب در آورده و از لوحه Properties مربوط به آن Shapetween را انتخاب می‌کنیم.



شکل (۲۰-۸)

۴. لایه جدیدی ایجاد کرده و نام آن را cloud می‌گذاریم. سپس تصویر یک ابر را در میان ابر های قبلی با رنگی ملایم تر رسم می‌کنیم. شکل (۲۱-۸)

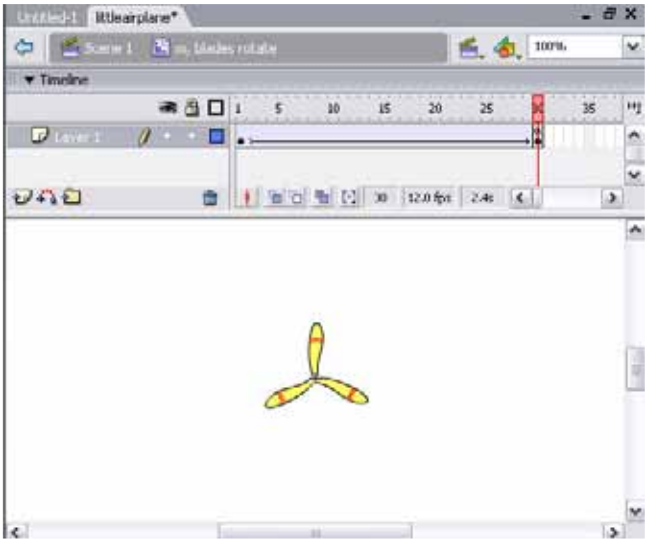


نمایی از ابر در لایه Cloud

شکل (۲۱-۸)

۵. در ستون ۶۰ از لایه Cloud یک قاب کلیدی درج می‌کنیم. سپس ابر را ک می‌به سمت چپ منتقل می‌کنیم. در این مرحله قاب‌های میانی، قاب مبدا و قاب مقصد لایه Cloud را به حالت انتخاب در آورده و از لوحه Properties مربوط به آن Shapetween را انتخاب می‌کنیم.

۶. یک فیلم ویدئویی ایجاد کرده و نام آن را Blades rotate می‌گذاریم. سپس در آن تصویر پروانه هواپیما را رسم کرده و بدون آنکه مکان آن را تغییر دهیم، با تکنیک Motion Tween به پروانه هواپیما یک حرکت دورانی می‌دهیم. شکل (۲۲-۸).



شکل (۲۲-۸)

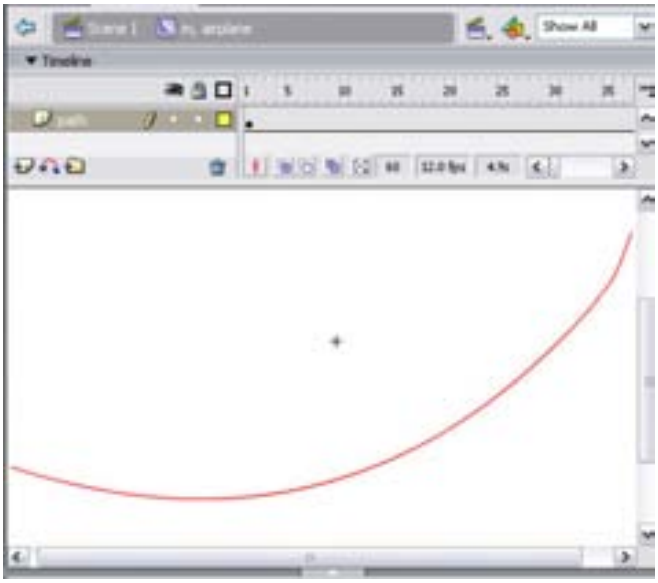
۷. در قاب ۳۰ کلیک کرده و در لوحه Action مربوط به آن خط زیر را قرار می‌دهیم.

```
GotoAndPlay (1) ;
```

۸. بر روی عبارت Scene 1 کلیک می‌کنیم تا به محیط کاری اصلی برگردیم.

۹. فیلم ویدئویی جدیدی ایجاد کرده و نام را airplane می‌گذاریم.

۱۰. نام لایه اول را Path می‌گذاریم و مسیر پرواز هواپیما را رسم می‌کنیم. شکل (۲۳-۸)



شکل (۲۳-۸)

۱۱. ستون ۶۰ لایه path را به عنوان مقصد در نظر گرفته و برای درج قاب کلیدی از کلید F6 را استفاده می‌کنیم.

۱۲. لایه جدیدی ایجاد کرده و در ستون اول تصویر هواپیما را رسم می‌کنیم. شکل (۲۴-۸)

شکل ۲۴-۸)

۱۳. فیلم ویدئویی مربوط به پروانه هواپیما را به تصویر اضافه می‌کنیم. شکل (۲۵-۸).

شکل (۲۵-۸)

۱۴. در ستون ۶۰ لایه فوق، یک قاب کلیدی درج می‌کنیم. سپس مکان قرار گیری تصویر را تغییر داده و به انت‌های مسیر منتقل می‌کنیم.

یادآوری: برای حرکت تصویر در یک مسیر منحنی، با کمک ابزار snap to Object مرکز تصویر را به ابتدا و انت‌های مسیر بچسبانید.

۱۵. قاب‌های میانی را انتخاب می‌کنیم، سپس در قاب مقصد راست کلیک کرده و از منوی باز شده گزینه Create motion tween را استفاده می‌کنیم.

۱۶. مراحل ذکر شده در قسمت ۵-۶ را دنبال کنید تا هواپیما روی مسیر منحنی حرکت کند.

۱۷. در ستون اول لایه مربوط به تصویر هواپیما، کلیک کرده و در لوحه Actionآن خط زیر را قرار می‌دهیم.

```
Stop () ;
```

۱۸. بر روی عبارت Scene 1کلیک می‌کنیم تا به محیط کاری برگردیم.

۱۹. یک کپی از فیلم ویدئویی airplane به محیط کاری منتقل کرده و نام نمونه آن را fly می‌گذاریم. (فیلم ویدئویی را در سمت چپ، پایین محیط کاری قرار دهید.)

۲۰. نماد دکمه را طراحی کرده و آن را روی محیط کاری قرار می‌دهیم. سپس در لوحه Action مربوط به آن خطوط زیر را قرار می‌دهیم.

```
on (release) {
        tellTarget ("fly") {
Play () ;
}
}
```

۲۱. با استفاده از کلید های ترکیبی Ctrl+Enter فیلم را اجرا می‌کنیم.

تمرین

به هواپیما ی در حال حرکت صدای مناسبی را اضافه کنید.

۷-۸-معرفی خطوط داخلی کلیپ ویدئویی

خطوط داخلی فیلم ویدئویی توانایی کنترل یک فیلم ویدئویی را فراهم می‌سازند.

```
Movieclipname.method (arguments) ;
```

خطوط داخلی (method)، فیلم ویدئویی‌ها در جدول (۲-۸) آمده است:

تابع داخلی (**method**)	عملکرد
۱) duplicateMovieClips	یک نسخه جدید از فیلم ویدئویی مورد نظر می‌سازد.
۲) getURL	یک صفحه وب در مرورگر باز می‌کند.
۳) gotoAndPlay	به قابی خاص از محیط کاری رفته و آن را اجرا می‌کند.
۴) gotoAndStop	به قابی خاص از محیط کاری رفته و آن را متوقف می‌کند.
۵) loadMovie	یک فیلم خاص را به فیلم ویدئویی هدایت می‌کند.
۶) loadVariables	متغیرهای یک فایل خارجی را به داخل فیلم ویدئویی می‌برد.
۷) nextFrame	هد اجرا را به قاب بعدی می‌فرستد.
۸) Play	فیلم ویدئویی خاصی را اجرا می‌کند.
۹) prevFrame	هد اجرا را به قاب قبلی می‌فرستد.
۱۰) removeMovieClip	فیلم ویدئویی ساخته شده توسط تابع dublicateMovieClip را پاک می‌کند.
۱۱) startDray	فیلم ویدئویی خاصی را قابل کشیدن می‌کند.
۱۲) Stop	فیلم ویدئویی خاصی را متوقف می‌سازد.
۱۳) stop Dray	قابل Drag بودن فیلم ویدئویی را متوقف می‌کند.

جدول (۲-۸)

در جدول (۳-۸) مثال‌های کوچکی از هر یک از خطوط داخلی جدول بالا ارائه شده است.

مثال	تابع
SampleMC.duplicateMovieClip ("newsampleMC", 1) ;	Movieclip.duplicateMovieClip
SampleMC.getURL ("http: //www.Yahoo.com", "blank") ;	MovieClip.GetURL
SampleMC.gotoAndPlay (1) ;	MovieClip.gotoAndPlay
SampleMC.gotoAndStop (1) ;	MovieClip.gotoAndStop
SampleMC.loadMovie ("load.Swf", 1) ;	MovieClip.loadMovie
SampleMC.loadVariable ("sales.Txt", 0);	MovieClip.loadVariable
SampleMC.nextFrame () ;	MovieClip.nextFrame
SampleMC.Play () ;	MovieClip.Play
SampleMC.PrevFrame () ;	MovieClip.PrevFrame
SampleMC.removeMovieClip () ;	MovieClip.RemoveMovieClip
SampleMC.startDray () ;	MovieClip.StartDray
SampleMC.stopDray () ;	MovieClip.StopDray

جدول (۳-۸)


نکته

دقت کنید که sampleMC نام نمونه فیلم ویدئویی است.

٦-١- خلاصه فصل

لوحه Action در فلش در دو مد معمولی و حرف‌های کار می‌کند. Scriptها یا در قاب یا در دکمه و یا در یک فیلم ویدئویی قرار می‌گیرند. برای درج خطوط در یک دکمه باید لوحه Action Button را روی یک Button باز کرد و برای اجرای خطوط آن حتماً باید از بلوک on استفاده کنیم. همچنین برای اضافه کردن تابع به یک فیلم ویدئویی لوح Action را روی Movieclip باز می‌کنیم و جهت اجرای خطوط آن از بلوک onClipEvent به همراه یکی از رویدادهای آن بهره می‌گیریم. برخی از رویدادها عبارتند از:

enterFrame: پس از هر بازه زمانی که با Framerate مشخص می‌شود خطوط موجود در این رویداد اجرا می‌شوند.

mouseMove: با هر گونه حرکت ماوس خطوط اجرا خواهند شد.

mousedown: زمانی که دکمه سمت چپ ماوس فشرده شود خطوط مربوطه اجرا خواهند شد.

mouseUp: زمانی که دکمه سمت چپ ماوس رها شود خطوط مربوطه اجرا خواهند شد.

keyDown: زمانی که یک کلید فشرده شود خطوط مربوطه اجرا خواهند شد.

keyUp: زمانی که دست را از روی کلید برمی‌داریم خطوط مربوطه اجرا خواهند شد.

در جدول (۴-۸) زیر لیست خطوطی که کاربرد زیادی دارند را به همراه یک توضیح مختصر می‌دهیم.

نام تابع	عملکرد
Play ()	اجرای فیلم را در قاب جاری ادامه می‌دهد.
Stop ()	برعکس تابع بالا عمل کرده و اجرای فیلم را متوقف می‌سازد.
gotoAndPlay (x)	هد اجرا را به قابی که مشخص کرده‌ایم، انتقال داده و اجرای فیلم را از آنجا آغاز می‌کند.
gotoAndstop (x)	این تابع هد اجرا را به قاب مورد نظر منتقل می‌کند ولی اجرای فیلم را در آنجا متوقف می‌گرداند.
nextFrame ()	هد اجرا را به اندازه یک قاب به جلو منتقل کرده و در آنجا اجرای فیلم را متوقف می‌کند.
prevFrame ()	هد اجرا را به اندازه یک قاب به عقب می‌برد و در آنجا اجرای فیلم را متوقف می‌سازد.
loadMovie ()	این تابع یک فیلم جدید را بار گذاری کرده و آن را جایگزین فیلم جاری می‌کند.
unLoadMovie ()	این تابع فیل می‌را که مشخص کرده‌ایم از حافظه پاک می‌کند.
Trace (نام متغیر)	هنگامی‌که یک فیلم را آزمایش می‌کنیم تا متوجه شویم که در هر مرحله چه اتفاقی می‌افتد، این تابع اطلاعات را در یک پنجره خروجی نشان می‌دهد. به وسیله این تابع می‌توانیم هر عبارت یا متغیری را در هر لحظه در آن پنجره چاپ کنیم.

getURL ()	سندی را از یک URL معین بار گذاری می‌کند.
loadVariables ()	از یک URL مشخص اطلاعاتی را درون فیلم جاری بار گذاری می‌کند.
startDrag ()	نمونه فیلم ویدئویی را قابل کشیده شدن در هنگام اجرای فیلم می‌کند.
stopDrag ()	از کشیده شدن نمونه فیلم ویدئویی جلوگیری می‌کند.
telltarget ()	از این تابع جهت ارسال دستورات به فیلم‌های ویدئویی استفاده می‌شود.
setRGB ()	این تابع رنگ فیلم های ویدئویی در حال اجرا را تغییر می‌دهد.
mouse.hide ()	اشاره‌گر ماوس مخفی می‌شود.
mouse.show ()	اشاره‌گر ماوس ظاهر می‌شود.
getDate ()	روز جاری را برمی‌گرداند. خروجی عددی بین ۱ تا ۳۱ می‌باشد.
getMonth ()	ماه جاری را برمی‌گرداند. خروجی عددی بین ۰ تا ۱۱ می‌باشد.
getFullYear ()	یک عدد چهار رقمی به معنای سال را برمی‌گرداند.
setMonth ()	تابع فوق دو آرگومان month , day دارد که با آن‌ها می‌توانیم ماه و روز را تنظیم کنیم.
setDate ()	این تابع تنها یک آرگومان day دارد و روز سیستم را می‌توان با آن تنظیم کرد.
Math.cos (x)	cos زاویه را برمی‌گرداند. (ورودی این تابع برحسب رادیان می‌باشد نه درجه).
Math.sin (x)	Sin زاویه را برمی‌گرداند. (ورودی این تابع نیز برحسب رادیان می‌باشد)
Math.log (x)	لگاریتم عدد ورودی را محاسبه می‌کند.
Math.pow ()	عدد اول را به توان عدد دوم می‌رساند.
Math.sqrt ()	ریشه دوم عدد را برمی‌گرداند.
Math.round ()	اعداد را به عدد صحیح گرد می‌کند.
Math.ceil ()	سقف عدد را برمی‌گرداند.
Math.floor ()	کف عدد را برمی‌گرداند.
Math.random ()	عدد تصادفی بین ۰ و ۱ برمی‌گرداند.
Math.abs ()	قدر مطلق عدد را محاسبه می‌کند.
key.getcode ()	کد کلید فشرده شده را برمی‌گرداند.
key.getAscii ()	کد اسکی کلید فشرده شده را برمی‌گرداند.
key.isDown ()	این تابع یک مقدار true یا false را برمی‌گرداند مبنی بر این که آیا کلید خاصی فشرده شده است یا خیر.
sound_name.attachsound ()	فایل صوتی مورد نظر را به متغیر الحاق می‌کند. (متغیر از نوع Sound می‌باشد).
sound_name.start ()	نمونه صوتی را پخش می‌کند.
sound_name.stop ()	اجرای نمونه صوتی را متوقف می‌سازد.
stopAllsound ()	کلیه صداهای در حال پخش را متوقف می‌سازد.
setVolume ()	اندازه صدای یک نمونه صوتی را تنظیم می‌کند.
getVolume ()	اندازه صدای یک نمونه صوتی را برمی‌گرداند.
setPan ()	نحوه پخش صدا در پخش‌کننده‌ها را تنظیم می‌کند.
getPan ()	چگونگی پخش صدا از پخش‌کننده‌ها را برمی‌گرداند.
خصیصه _x	یک فیلم ویدئویی را در عرض حرکت می‌دهد.
خصیصه _y	یک فیلم ویدئویی را در طول حرکت می‌دهد.
خصیصه _visible	دو مقدار true یا false می‌گیرد و برحسب آن فیلم ویدئویی را مخفی یا آشکار می‌کند.
خصیصه _rotation	فیلم ویدئویی را به اندازه دلخواه می‌چرخاند.
خصیصه _width	پهنای فیلم ویدئویی را برحسب نقطه تنظیم می‌کند.
خصیصه _alpha	درصد شفافیت فیلم ویدئویی را برمی‌گرداند.
خصیصه _height	ارتفاع فیلم ویدئویی را برحسب نقطه تنظیم می‌کند.
setProperty ()	مقدار یک خصوصیت معین از فیلم ویدئویی را تغییر می‌دهد.
getProperty ()	مقدار یک خصوصیت معین از فیلم ویدئویی را برمی‌گرداند.

onClipEvent	در زمانیکه یک رویداد مربوط به نمونه فیلم ویدئویی واقع می‌شود مجموع‌های از عبارات تابعی را اجرا می‌کند.
On	هنگام اجرا شدن یک رویداد ماوس مشخص، یک سری دستورات را اجرا می‌کند.

جدول (٤-٨)

خودآزمایی

۱) ضرورت استفاده از Action script در فیلم‌های فلش چیست؟
۲) مدهای لوحه Action را نام برده و توضیح دهید.
۳) تفاوت بین تابع (gotoAndPlay) و (Play) را شرح دهید.
۴) تابع (prevFrame) و (nextFrame) را توضیح دهید.
۵) جهت متوقف کردن فیلم از چه تابعی استفاده می‌شود. با ذکر مثال شرح دهید.
۶) کدهای Action script در چه قسمت‌هایی از فیلم قرار می‌گیرند.
۷) جهت قراردادن Action برای یک دکمه چه اقدامی باید انجام داد.
۸) کاربرد بلوک on را بیان کنید و پارامترهای آن را شرح دهید.
۹) کاربرد بلوک onClipEvent را بیان کرده و پارامترهای آن را شرح دهید.
۱۰) خصیصه‌های visible_، y_، و x_ در یک فیلم را شرح دهید.

پروژه:

با استفاده از فلش، فیل می‌ایجاد کنید که ۶ گونه از گل‌های مختلف را معرفی کند. برای انجام این کار ابتدا برای هر کدام از گل‌ها یک فیلم جداگانه ایجاد کرده و با نام همان گل در پوشه ای با نام Flowers (این پوشه را از قبل در محل دلخواه خود ایجاد کنید) ذخیره کنید. سپس فیلم اصلی را با نام Introduce ایجاد کنید و در آن ۶ نماد دکمه قرار دهید. دکمه‌ها که هر کدام تصویر کوچکی از گل مورد نظر می‌باشد، به گونه ای طراحی شوند که وقتی نشانگر ماوس از روی آن‌ها عبور می‌کنید تصویر گل ک می‌بزرگ شود و وقتی از روی تصویر عبور کرد به اندازه اول برگردد.
زمانی که کاربر بر روی هر کدام از دکمه‌ها کلیک کرد، فیلم مخصوص آن گل توسط دستور Loadmovie به نمایش در بیاید.
در ساخت فیلم هر کدام از گل‌ها موارد زیر را رعایت کنید:

۱. در فیلم تعدادی عکس از گل مورد نظر در نظر بگیرید، به گونه ای که عکس اول از حالت محو پیدا و نمایان شود، سپس عکس دوم از حالت نمایان به تدریج محو شود و به همین ترتیب تا عکس آخر.
۲. در عکس آخر حروف نام گل هر کدام به طور همزمان از سمت بالا (بیرون محیط کاری) به سمت پایین (داخل محیط کاری،) حرکت کرده و در مرکز عکس متوقف شوند. (برای انجام این کار، هر کدام از حروف گل را با ابزار متن نوشته و جداگانه با استفاده از روش Motion tween آن را حرکت دهید. برای توقف در مقصد، از دستور stop در قاب مقصد استفاده کنید.

نکته: در صورتی که بخواهید در فلش متن فارسی تایپ کنید باید از یک فارسی ساز (مانند فارسی ساز مریم) استفاده کنید، به طوری که متن خود را در فارسی ساز تایپ کرده سپس آن را کپی کرده و در فلش برگردانید (عمل paste کردن).

۳. در فیلم یک نماد دکمه با عنوان «بازگشت» قرار دهید. زمانی که کاربر بر روی دکمه کلیک کرد با ید به فیلم اصلی بر گردد. (برای این کار از دستور Loadmovie استفاده کنید).

فهرست منابع:

www.flashkit.com
www.flash-creations.com
www.bestflashanimationsite.com
www.macromedia.com
www.informit.com
www.flash2be.com