

Flash Action Script 2.0

کدنویسی در Flash

ویرایش: آذرماه ۱۳۸۹

YousefAmiri.ir



کلاس String :

برای استفاده از توابع رشته‌ای، از این کلاس استفاده می‌شود. یک دکمه (Button) از پانل Component را روی صحنه قرار دهید. سپس یک TextInput هم به صحنه اضافه کرده و نام آنرا txtName بگذارید. دکمه را انتخاب و کد زیر را برای آن بنویسید: رویداد زیر از کلاس String () یک شیء به نام iStr ایجاد می‌کند. (نام شیء اختیاری است). متن txtName را در متغیر iStr قرار میدهد. همه حروف رشته را تبدیل به بزرگ نموده و آنرا در جای خود دوباره می‌نویسد.

```
on(click) {
    iStr=new String();

    iStr=_root.txtName.text;
    _root.txtName.text=iStr.toUpperCase();
}
```

رویداد زیر (برای دکمه)، رشته را در شیء iStr ذخیره می‌کند. کادر متن را خالی میکند. از انتهای رشته تا ابتدای آن، کاراکتر به کاراکتر، رشته را در کادر متن می‌نویسد. (رشته را معکوس می‌کند) charAt(i) یک کاراکتر از رشته را برمیگرداند. اندیس رشته از صفر شروع می‌شود.

```
on(click) {
    iStr=new String(_root.txtName.text);
    _root.txtName.text="";

    for(i=iStr.length; i>=0; i--)
        _root.txtName.text=_root.txtName.text+iStr.charAt(i);
}
```

دسترسی از عمق به بالا در MovieClip :

برای اینکه بتوان از داخل یک MovieClip کدی را در مورد آن اجرا نمود باید از _parent استفاده کنیم. در MovieClip یک دکمه وجود دارد، در داخل MovieClip و برای دکمه رویداد زیر را می‌نویسیم: این رویداد MovieClip و محتویات آنرا که شامل خود دکمه هم میشود، مخفی میکند. برای شیء Graphic همین نتیجه را نمی‌گیریم!

```
on(click) {
    this._parent._visible=false;
}
```

حرکت دادن شیء MovieClip :

دو رویداد زیر امکان جابجایی یک MovieClip را توسط ماوس می دهد:

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

چنانچه در MovieClip دکمه‌های قرار داشته باشد، آن دکمه نمی تواند رویداد کلیک را دریافت کند. برای حل این مشکل، یک MovieClip دیگر در MovieClip اصلی قرار داده و رویدادهای زیر را برای MovieClip داخلی می نویسیم:

```
on (press) {
    this._parent.startDrag();
}
on (release) {
    this._parent.stopDrag();
}
```

پخش یک فایل موسیقی خارجی :

رویداد زیر (برای دکمه) یک فایل MP3 خارج از فلش را پخش می کند: این رویداد ابتدا شیئی از کلاس Sound ایجاد میکند. یک فایل موسیقی را در این شیء قرار می دهد. موسیقی را پخش میکند.

```
on (click) {
    iSnd=new Sound();
    iSnd.loadSound("Itnee.mp3", true);
    iSnd.start();
}
```

کنترل روی صدا :

چند دکمه (Button) و یک کنترل شمارنده (NumericStepper) برای تنظیمات صدا روی صحنه قرار دهید: ابتدا مانند مثال قبیل یک فایل موسیقی را پخش می کنیم. اما در اینجا از یک متغیر سراسری برای شیء صدا استفاده می کنیم. تا در جاهای دیگر نیز در دسترس باشد. و بتوان صدا را کنترل نمود.

کد زیر آدرس نسبی فایل را در شیء سراسری `_global.iSnd` قرار میدهد:

```
On(click) {
    _global.iSnd=new Sound();
    _global.iSnd.loadSound("Music/Intro.mp3", true);
    _global.iSnd.start();
}
```

یک کنترل `NumericStepper` را برای تنظیم شدت صدا روی صحنه قرار داده و در سربرگ `Parameters` خصوصیات زیر را برای آن مقداردهی کنید:

```
maximum=100
minimum=0
stepSize=10
value=100
```

رویداد زیر را برای این کنترل (`NumericStepper`) بنویسید :
با تغییر مقدار کنترل شمارنده، شدت صدا نیز تغییر می یابد.

```
on(change) {
    _global.iSnd.setVolume(this.value);
}
```

رویداد زیر (برای یک دکمه) صدا را به ابتدا برگردانده و متوقف میکند:

```
on(click) {
    _global.iSnd.stop();
}
```

پخش یک فایل موسیقی داخلی :

کد زیر (برای فریم) یک فایل MP3 وارد شده به فلش را پخش می کند:
ابتدا در `Library` روی صدا کلیک راست نموده و `Linkage...` را انتخاب نمایید.
سپس `Export for ActionScript` و `Export in First Frame` را تیک بزنید.
در قسمت `Identifier` نام `myMusic` را وارد کنید.
دو دکمه به نامهای `btnPlay` و `btnStop` روی صحنه قرار دهید.
در فریم اول صحنه اصلی کد زیر را وارد نمایید:
این رویداد ابتدا فیلم را در این فریم نگه میدارد.
شیئی از کلاس `Sound` ایجاد میکند.
یک فایل موسیقی داخلی را در این شیء قرار می دهد.
در فریم، یک رویداد برای هر کدام از دکمهها ایجاد می کند.

```
stop();

iSnd=new Sound();
iSnd.attachSound("myMusic");
```

```

btnPlay.onRelease = function() {
    iSnd.start();
}

btnStop.onRelease = function() {
    iSnd.stop();
}

```

پخش فایل MP3 توسط کنترل MediaPlayer :

توسط MediaPlayer هم میتوان یک فایل MP3 خارجی را پخش نمود. مزیت استفاده از این کنترل اینست که تمام اعمالی که بر روی صدا میتوان انجام داد را در خود دارد.

ابتدا این کنترل را روی صحنه قرار دهید:

در سربرگ Parameters دکمه Launch Component Inspector را کلیک کنید.

دکمه رادیویی MP3 را انتخاب کنید.

در قسمت URL مسیر و نام فایل MP3 را وارد کنید. (انتخاب Automatically Play سبب پخش اتوماتیک موسیقی می شود)

کد زیر (برای فریم) فایل Itnee.mp3 را در یک کنترل MediaPlayer به نام mpb قرار می دهد.

```

_root.mpb.contentPath="Music/Itnee.mp3";
_root.mpb.play();

```

رویدادهای کلیپ :

رویداد زیر برای یک کلیپ (MovieClip) سبب جابجایی آن با حرکت ماوس می شود:

```

onClipEvent(mouseMove) {
    this._x=_root._xmouse;
    this._y=_root._ymouse;
}

```

توابع داخلی :

```

_root.txtOut.text=random(5);
_root.txtOut.text=String(1385);
var x=Number(_root.txtDay.text);
Math.round(x);
getTimer();

```

یک عدد تصادفی صحیح بین 0 تا n-1 برمیگرداند.

تبدیل عدد به رشته

تبدیل رشته عددی به عدد و ذخیره در متغیر محلی

عدد x را گرد می کند

زمان اجرای فلش (میلی ثانیه)

ارسال متغیر به فایل‌های دیگر :

رویداد زیر page.asp را اجرا کرده و متغیرهای فایل swf را به آن می‌فرستد:

```
on(click) {
    getURL("page.asp", "_self", "POST");
}
```

توسط این دستور می‌توانید بین فایل‌های فلش و صفحات وب ارتباط برقرار نمایید. در دستور بالا متغیرهای تعریف شده در فایل swf با متد POST به صفحه page.asp ارسال می‌شوند. در فایل page.asp می‌توانید توسط request() متغیر را دریافت کنید.

ایجاد تابع :

دستورات زیر (برای فریم) تابعی را تعریف می‌کند که رشته‌های را دریافت کرده، در صورتیکه عدد باشد مقدار true و در غیر این صورت مقدار false برمیگرداند.

این تابع توسط رویدادی از دکمه btn در خط دوم آن فراخوانی می‌شود و مقدار موجود در txtOut را چک می‌کند.

```
function isDigit(str:String) {
    if(str==String(Number(str)))
        return true;
    else
        return false;
}

btn.onRelease=function() {
    if(isDigit(_root.txtOut.text))
        _root.txtOut.text="Digit";
    else
        _root.txtOut.text="Not Digit";
}
```

کنترل Timeline :

رویداد دکمه زیر سبب می‌شود که هر بار در خط زمانی یک فریم به جلو برویم و چنانچه در فریم آخر باشیم، به ابتدا برگردیم. از این کد می‌توان برای ایجاد آلبوم عکس استفاده نمود.

دو لایه در صحنه اصلی ایجاد کنید.

در هر فریم از لایه زیرین یک عکس قرار دهید.

در هر فریم از لایه زیرین اکشن ; stop() را بنویسید.

در لایه بالا یک دکمه قرار داده و کد زیر را برای آن بنویسید:

```
on(click) {
    if(_root._currentframe != _root._totalframes)
        nextFrame();
}
```

```
else
    gotoAndStop(1);
}
```

نمایش پیام توسط کنترل Alert :

در جاوا اسکریپت و فلش از Alert برای نمایش پیام استفاده میشود. (همانند MsgBox در VB) دکمهای روی صحنه قرار دهید و کد زیر را برای آن بنویسید: ابتدا کنترل Alert فراخوانی می شود.

setStyle خصوصیات یک شیء (در اینجا Alert) را تغییر می دهد. و okLabel برچسب دکمه OK را تعیین میکند. (جدول زیر) سپس پیام نمایش داده می شود. (پارامترها به ترتیب متن پیام، عنوان و دکمه ها هستند)

```
on(click) {
    import mx.controls.Alert;

    _global.styles.Alert.setStyle("color", 0x0000ff);
    Alert.okLabel="تایید";

    Alert.show(" Welcome to Flash Programming World!",
              "Itnee.com",
              Alert.OK | Alert.CANCEL);
}
```

نکته: دستور ("Welcome!") trace را برای یک دکمه دیگر بنویسید. این بار پیام در پنجره Output نمایش داده میشود. خواص کنترل Alert:

Alert.okLabel="تایید";	برچسب دکمه OK را تغییر می دهد
Alert.cancelLabel="انصراف";	برچسب دکمه Cancel را تغییر می دهد

سبکهای کنترل Alert:

_global.styles.Alert.setStyle("color", 0x0000ff);	رنگ متن
_global.styles.Alert.setStyle("backgroundColor", 0xffcc00);	رنگ زمینه کادر پیام
_global.styles.Alert.setStyle("textAlign", "right");	تراز بندی متن
_global.styles.Alert.setStyle("fontFamily", "Tahoma");	فونت

کلاس TextFormat :

اگر توسط کد نویسی بخواهیم خواص مربوط به قالب بندی متن را تغییر دهیم از این کلاس استفاده می کنیم. یک textBox از نوع Dynamic و با نام txtName روی صحنه قرار دهید و کد زیر را در فریم اول بنویسید: ابتدا شیئی از نوع قالب بندی ایجاد می شود.

سپس خواص مربوط به این شیء تنظیم می شود.
در نهایت این خواص به txtName نسبت داده می شود.

```
iformat = new TextFormat();

iformat.font = "Tahoma";
iformat.color = "0xffcc00";
iformat.bold=true;

txtName.setTextFormat(iformat);
```

در خط سوم، یک عدد هگزا دسیمال برای رنگ اختصاص یافته است. قالب استفاده از رنگ در ActionScript بصورت مقابل است: 0xRGB

کنترل List :

در این مثال لیستی از نام صفحات یک وب سایت ایجاد کرده و رویدادی می نویسیم که با کلیک روی نام صفحه که فارسی نوشته شده است، آن صفحه باز شود.

در پانل Components یک کنترل List روی صحنه بکشید.

در پانل Properties نام آنرا lstLink قرار دهید.

در پانل Parametrs مقابل data دوبار کلیک کنید.

دکمه + را بزنید و آدرس صفحات را نسبت به این فایل swf بنویسید. (اگر مثلاً صفحه 1.htm و فایل swf که ایجاد می کنید هردو در یک پوشه اند کافیست به جای defaultValue بنویسید: 1.htm)

دو مرحله قبل را برای labels هم تکرار کنید با این تفاوت که به جای defaultValue نام فارسی هر صفحه را بنویسید. این مقادیر در لیست دیده می شوند. (مشکل فارسی را مطابق توضیحات کنترل CheckBox برطرف کنید.)

کد زیر را برای فریم اول بنویسید تا lstLink به حالت مناسب فارسی درآید.

```
Stop();
lstLink.setStyle("fontFamily","Tahoma");
lstLink.setStyle("textAlign","right");
```

لیست را انتخاب و رویداد زیر را برای آن بنویسید:

```
on (change) {
    getURL(this.value);
}
```

با کلیک روی یک گزینه از لیست، مقدار آن که در value قرار داده شده و getURL آن صفحه را باز می کند.

کنترل CheckBox :

کد زیر نحوه استفاده از عمومی ترین خواص کنترل CheckBox را بیان می کند.

ابتدا یک کنترل CheckBox به نام ch1 روی صحنه قرار دهید.
یک کنترل TextInput نیز به صحنه اضافه نموده و نام آنرا txtEmail قرار دهید.

در فریم اول دستورات زیر را قرار دهید. در صورتیکه ch1 انتخاب شود، txtEmail فعال و در غیر این صورت غیرفعال خواهد شد:

ابتدا فونت چک باکس تعیین می شود.

سپس توسط خاصیت label یک برچسب فارسی برای آن قرار داده می شود. در صورتیکه متن فارسی شما به هم می ریزد یکی از دلایل آن کاراکتر «ی» است. برای حل این مشکل کد آن را وارد نمایید:

به جای «ی» کلید Alt را نگه داشته، 0236 را بزنید و Alt را رها کنید.

به جای «ی» کلید Alt را نگه داشته، 0237 را بزنید و Alt را رها کنید.

رویداد اول هنگامی رخ خواهد داد که یک شیء (در اینجا txtEmail) برای اولین بار ایجاد شود. و آنرا غیرفعال می کند.

رویداد دوم با کلیک کردن چک باکس اجرا می شود. اگر چک باکس تیک خورده باشد خاصیت value آن true و در غیر این صورت false می شود و این مقدار txtEmail را فعال یا غیرفعال می کند.

نکته: چون در لحظهای که رویداد Press اتفاق میافتد هنوز مقدار ch1 تغییر نکرده است بنابراین توسط not این اشکال رفع می شود.

```
ch1.setStyle("fontFamily", "Tahoma");
ch1.label="آدرس ایمیل";

txtEmail.onLoad=function() {
    txtEmail.enabled=false;
}

ch1.onPress=function() {
    txtEmail.enabled=not ch1.value;
}
```

کنترل Loader :

از این کنترل برای نمایش یک عکس یا فایل swf خارجی استفاده می شود.

طی این مثال به سادگی swf را که در SWiSH ایجاد کرده ایم توسط یک دکمه در فایل فلش خود پخش می کنیم. (حتماً با مشکلات استفاده از روشهای دیگر آشنا هستید!) بنابراین یک دکمه و یک کنترل Loader روی صحنه قرار دهید.

نام کنترل Loader را ldr قرار دهید. و اندازهی آنرا مناسب در نظر بگیرید.

در پانل Parametr خاصیت autoLoad آنرا false کنید.

کد زیر را برای دکمه وارد کنید.

```
on(release) {
    _root.ldr.contentPath="Tizer.swf";
    _root.ldr.load();
}
```

در contentPath نام و مسیر فایل jpeg یا swf را قرار دهید.

رزولوشن و سیستم عامل مقصد :

یکی از مشکلات طراحان، اطلاع از چگونگی پخش فایل آنها در کامپیوترهای دیگر است. رزولوشن کامپیوتر مقصد وقتی اهمیت پیدا میکند که از تصاویر Bitmap در swf خود استفاده کرده باشیم. اگر فایل شما برای رزولوشن 1024x768 طراحی شده باشد، برای رزولوشنهای بالاتر بجز مقداری افت کیفیت تصاویر بیت مپ، مشکل حادی وجود ندارد. اما در رزولوشنهای پایینتر اندازه آنها می تواند مشکل ساز باشد. کد زیر رزولوشن و سیستم عامل را کنترل می کند و در صورت سازگار نبودن، اجرا را پایان می دهد. در فریم اول کد زیر را وارد نمایید:

```
w=System.capabilities.screenResolutionX;
h=System.capabilities.screenResolutionY;
os=System.capabilities.os;
```

```
if(os!="Windows XP" || w!=1024 || h!=768)
fscommand("quit","true");
```

علامت || به معنای Or و علامت != به معنای not است.

تاریخ و زمان :

توسط کلاس Date میتوان به تاریخ و زمان کامپیوتری که فایل swf را اجرا میکند دسترسی داشت. مطابق معمول ابتدا شیئی از کلاس ایجاد می کنیم و سپس متدهای آنرا بکار می بریم.

کد زیر یک تایمر ایجاد می کند که ثانیه، دقیقه و ساعت را نشان میدهد. یک کلیپ (MovieClip) ایجاد کرده و داخل آن سه TextBox از نوع Dynamic قرار دهید. به آنها نامهای txtH, txtM, txtS بدهید. از کلیپ خارج شده و آنرا انتخاب کنید.

رویداد زیر را برای کلیپ بنویسید. این رویداد هر لحظه برای MovieClip اتفاق می افتد و عددها را به روز می کند.

```
onClipEvent (enterFrame) {
    iDate=new Date ();

    this.txtS.text=iDate.getSeconds ();
    this.txtM.text=iDate.getMinutes ();
    this.txtH.text=iDate.getHours ();
}
```

سایر توابع:

```
txtTest.text=iDate.getDate ();
txtTest.text=iDate.getDay ();
```

روز nام ماه

روز nام هفته (یکشنبه برابر ۰)

```
txtTest.text=iDate.getFullYear();
txtTest.text=iDate.getMonth();
txtTest.text=iDate.getMilliseconds();
```

سال ۴ رقمی
ماه nام سال (ژانویه برابر ۰)
میلی ثانیه

اجرای یک فایل دیگر :

از تابع `loadMovie()` برای اجرای یک فایل `swf`، `jpeg`، `gif` غیر انیمیشنی و یا `png` در فایل `swf` فعلی استفاده می شود. می توانید فایل دوم را به ریشه (سطح ۰) یا سطوح بالاتر لود کنید. همچنین می توان یکی از این فایلها را به داخل یک کلیپ هم بارگذاری کرد؛ تنها نکته های که در مورد حالت اخیر لازم است، اینست که نقطه رجیستریشن کلیپ را گوشه چپ بالا در نظر بگیرید تا فایل دوم مطابق انتظار در محلی که می خواهید قرار گیرد. تابع `unloadMovie()` هم اجرای فایل دوم را پایان می دهد.

ابتدا در فایل `1.swf`، دو دکمه روی صحنه قرار دهید.
فایل دیگری به نام `2.swf` در کنار فایل بالا قرار دهید.

رویداد زیر را برای دکمه اول بنویسید:

```
on(click){ loadMovie("2.swf","_root");}
```

این رویداد فایل دوم را به ریشه می آورد و فایل اول را از ریشه برمیدارد. اگر بخواهید دوباره فایل `1.swf` را به ریشه برگردانید، باید در فایل `2.swf` هم یک دکمه قرار داده و رویداد زیر را برای آن بنویسید:

```
on(click){
loadMovie("1.swf","_root");
}
```

`_level0` نام دیگر `_root` است. رویداد زیر همان عمل بالا را انجام می دهد:

```
on(click){
loadMovie("2.swf","_level0"); // عدد صفر
}
```

حال یک کلیپ روی صحنه قرار دهید و نام آنرا `iMC` بگذارید. (دقت کنید رجیستریشن آن گوشه چپ بالا باشد). این بار رویداد زیر را برای دکمه اول بنویسید. با این عمل فایل دوم به داخل کلیپ بارگذاری میشود و سایر اشیاء صحنه باقی می ماند:

```
on(click){
loadMovie("2.swf","_root.iMC");
}
```

رویداد زیر را برای دکمه دوم بنویسید تا اجرای فایل `2.swf` را در داخل کلیپ پایان دهد:

```
on(click){
unloadMovie("_root.iMC");
}
```

این تابع دارای یک پارامتر است. پارامتر آن محل بارگذاری فایل دوم است.

و سرانجام روش زیر باعث لود شدن فایل دوم به سطوح دیگر فایل اول شده و آبجکتهای ریشه را از بین نمی برد:

```
on(click) {
    loadMovie("2.swf", "_level1");
}
```

اگر فایل دوم همه صحنه را نپوشاند و بتوان دکمه دوم را دید، کد زیر را برای دکمه دوم بنویسید تا سطح ۱ را پاک کند:

```
on(click) {
    unloadMovie("_level1");    // عدد یک
}
```

ایجاد رابط نرم افزار (UI) توسط فلش :

با استفاده از loadMovie، unloadMovie و _level ها، می توان یک رابط نرم افزار MDI را توسط فلش طراحی کرد. کاری که لازم است انجام دهید اینست که یک فایل swf برای منوی اصلی برنامه خود می سازید. و آنرا به حالت تمام صفحه درمی آورید: در فریم اول این فایل بنویسید:

```
fscommand("fullscreen", "true");
```

حال برای هر کدام از پنجرههای فرزند (Child) یک فایل جداگانه swf بسازید.

یک میله منو برای فایل اصلی طراحی کنید و مطابق مثالهای بالا، هر کدام از فایلها swf فرعی را به یک سطح از فایل اصلی لود کنید. این فایلها دارای دکمه های تایید یا انصراف هستند که توسط آنها می توانید از () unloadMovie استفاده کنید. برای دسترسی به اشیاء هر سطح میتوانید از رویداد زیر که برای یک دکمه نوشته شده است استفاده کنید:

دکمه ای در ریشه قرار دارد، میخواهیم توسط آن به فریم دوم فایل لود شده در سطح ۲ برویم:

```
On (press) {
    _level2.gotoAndStop(2);
}
```

نرم افزار XeroniaPanel نمونههای از این نوع است که میتوانید آنرا از Itnee.com دریافت کنید.

اجرای فایلها exe :

برای اجرای فایلها اجرایی دیگر برنامه ها و فایلها exe فلش از fscommand استفاده می شود. با این شرط که این فایلها باید در پوشه ای به نام fscommand قرار داشته باشند. بنابراین :
پوشه ای به نام fscommand در کنار فایل فلش ایجاد کنید.
در داخل این پوشه فایلی به نام setup.exe قرار دهید.

یک دکمه در فایل فلش قرار داده و کد زیر را برای آن بنویسید:

```
On (Release) {
    fscommand ("exec", "setup.exe")
}
```

اضافه کردن گزینه به منوی کلیک راست

یک TextBox به نام txt به صحنه اضافه کنید و دستورات زیر را به فریم اول اضافه کنید:

```
var m=new ContextMenu();
m.customItems.push(new ContextMenuItem("About Itnee",menuaction));
function menuaction(){
    txt.text="Itnee.com"
}

m.builtInItems.quality=false; //Remove Items from Menu
m.builtInItems.print=false;
m.builtInItems.loop=false;
m.builtInItems.zoom=false;
m.builtInItems.save=false;
m.builtInItems.forward_back=false;
m.builtInItems.rewind=false;
m.builtInItems.play=false;

_root.menu=m;
```

پیمایش TextBox:

برای اسکرول کردن یک جعبه متن، کافیست دو دکمه و یک TextBox از نوع Dynamic یا Input روی صحنه قرار دهید نام این جعبه متن را txt قرار دهید. و برای هر کدام از دکمه ها کد زیر را بنویسید:

توجه داشته باشید که متن داخل جعبه متنی باید بیشتر از ابعاد آن باشد تا نتیجه را ببینید.

```
on(release) //first button
{
    _root.txt.scroll++;
}

on(release) //second button
{
    _root.txt.scroll--;
}
```

متن روان

برای نمایش کاراکتر به کاراکتر یک رشته داخل یک فیلد متنی به روش زیر عمل کنید:

یک فیلد متنی از نوع Dynamic و با نام txt روی صحنه قرار دهید. و در صورتیکه متن روان شما فارسی باشد جهت آنرا راست به چپ در نظر بگیرید (برای نوشتن متن فارسی به توضیحات کنترل CheckBox در Part1 نگاه کنید)

در فریم اول کد زیر را بنویسید:

```
iStr=new String("به آی تی نی خوش آمدید");
txt.textColor="0x0000ff";
var i=0;

id=setInterval(ItneeText,80);

function ItneeText()
{
    if(i<iStr.length)
        txt.text=txt.text + iStr.charAt(i++);
    else
        clearInterval(id);
}
```

کنترل متن ورودی

یک فیلد متنی از نوع Input و با نام x روی صحنه قرار دهید. در فریم اول یکی از دستورات زیر را بنویسید. داخل کوتیشن بازه ای از کاراکترها را که می خواهید فیلد متنی قبول کند بنویسید. (فونت Tahoma با اندازه ۹ را برای متون فارسی انتخاب کنید و جهت فیلد متنی را راست به چپ انتخاب کنید. خواهید دید که دستور اول فقط اجازه ورود متون فارسی را به فیلد متنی می دهد)

```
x.restrict="آ-ی";
x.restrict="0-9^78"; // تا ۹ بجز ۷ و ۸
x.restrict="\u0020 a-z"; // کاراکتر فاصله و حروف کوچک انگلیسی. به کد فاصله در یونیکد دقت کنید
x.restrict="0-9 a-z A-Z @. _"; // مثالی برای دریافت آدرس ایمیل
```

کنترل صحنه توسط صفحه کلید

در فریم اول کد زیر را بنویسید. با فشردن کلیدهای کنترل + فاصله فایل فیلم بسته خواهد شد.

```
_root.onEnterFrame=function()
{
    if(Key.isDown(Key.CONTROL) && Key.isDown(Key.SPACE))
        fscommand("quit",true);
}
```

