

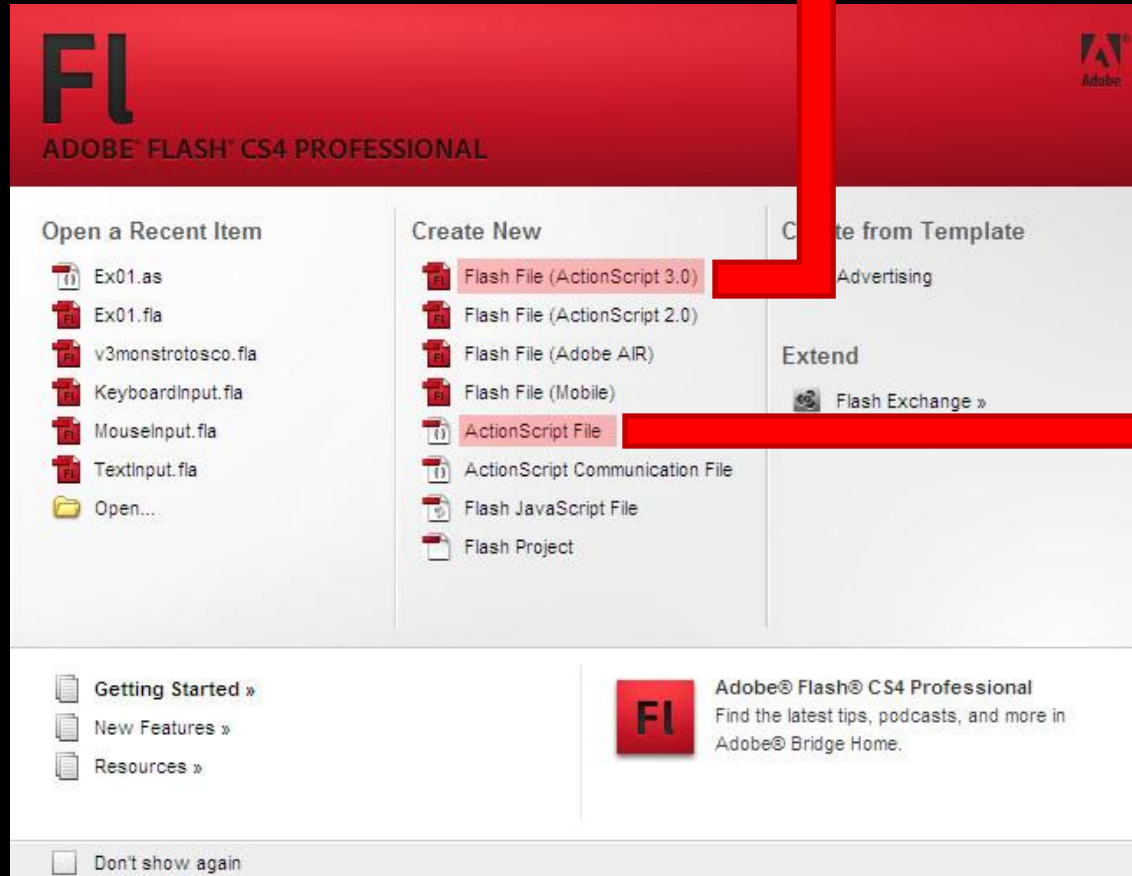
# Actionscript 3.0

Por: Vitor Maia

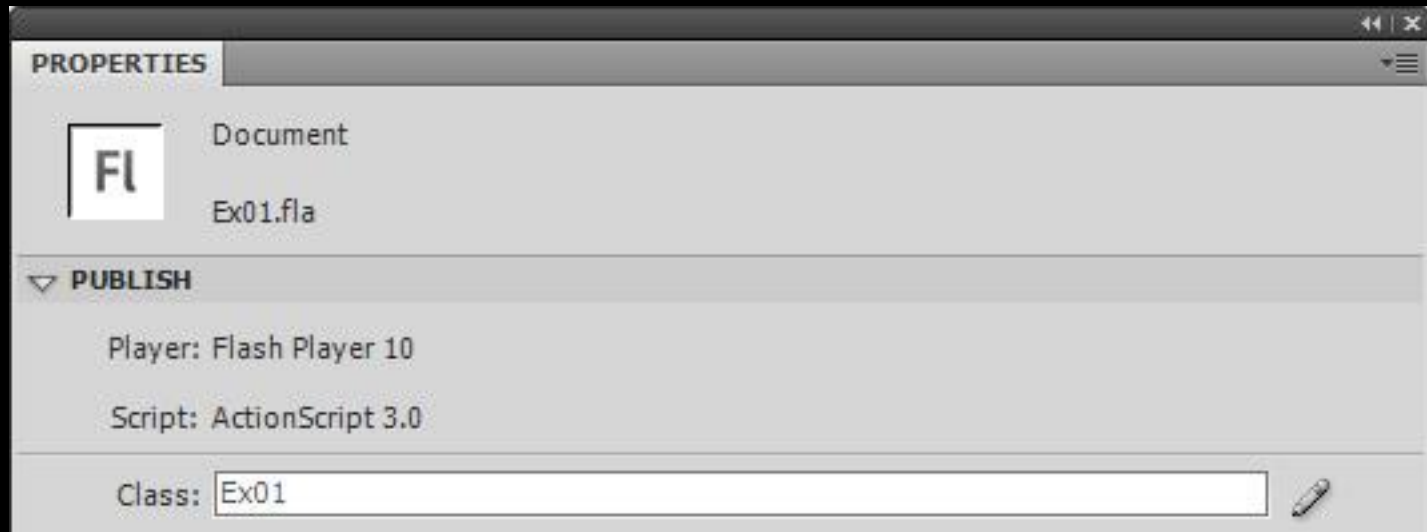
[www.dcc.ufrj.br/~vitormaia](http://www.dcc.ufrj.br/~vitormaia)



Crie um documento deste tipo, chamado ex01.fla...

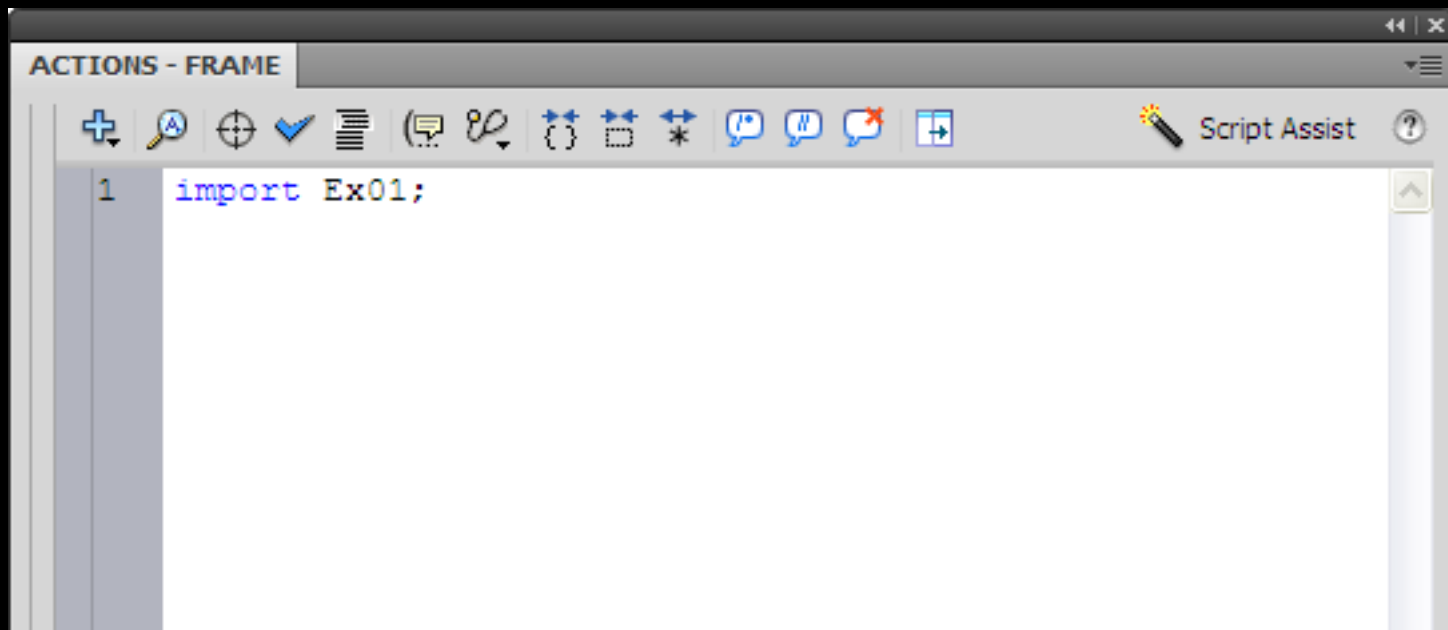


...e um deste tipo, chamado ex01.as, ambos na mesma pasta.



Nas propriedades do arquivo Ex01.fla, em “Class”, escreva Ex01.

Todo o código incluído no arquivo .as será “compilado” e executado pelo arquivo .fla



O mesmo acontece se o arquivo for importado através da aba Actions. Assim podem ser usados mais de um .as por programa, bastando apenas utilizar vários imports.

Para abrir esta aba, clique com o botão direito em um dos frames da linha do tempo e escolha Actions. Cada frame possui um Actions diferente.

```
//"Hello World"
package {
    // O primeiro import é obrigatório
    import flash.display.*;
    import flash.text.*;

    // Nome da classe = Nome do arquivo
    // A classe ligada ao arquivo .fla deve herdar do tipo MovieClip
    public class Ex01 extends MovieClip {
        // Construtor da classe, funciona como "main"
        public function Ex01() {
            var texto:String = "Actionscript";
            trace(texto);
        }
    }
}
```

Este programa apenas escreve "Actionscript" na tela. O comando trace() imprime numa tela de output, à parte da cena principal do arquivo .fla



```
package {
    import flash.display.*;
    import flash.text.*;

    public class Ex01 extends MovieClip{

        // Versão do código anterior para escrever
        // um texto na cena principal do .fla
        public function Ex01() {

            // Objeto campo de texto
            var texto:TextField = new TextField();

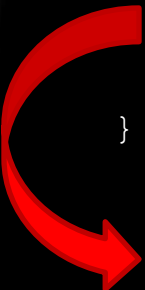
            // text é um atributo de TextField
            texto.text = "Actionscript";

            // Por enquanto, o objeto foi apenas criado
            // e editado. O comando addChild() adiciona
            // uma instância deste objeto na tela.
            addChild(texto);
        }
    }
}
```

OBS: Para poder ESCREVER no textField, adicione a linha:  
`texto.type = TextFieldType.INPUT;`




```
public function Ex01() {
    var formato:TextFormat = new TextFormat("Comic Sans", 50, 0xFF00FF)
    var texto:TextField = new TextField();
    texto.defaultTextFormat = formato;
    texto.text = "Actionscript";
    texto.width = 400;
    addChild(texto);
}
```



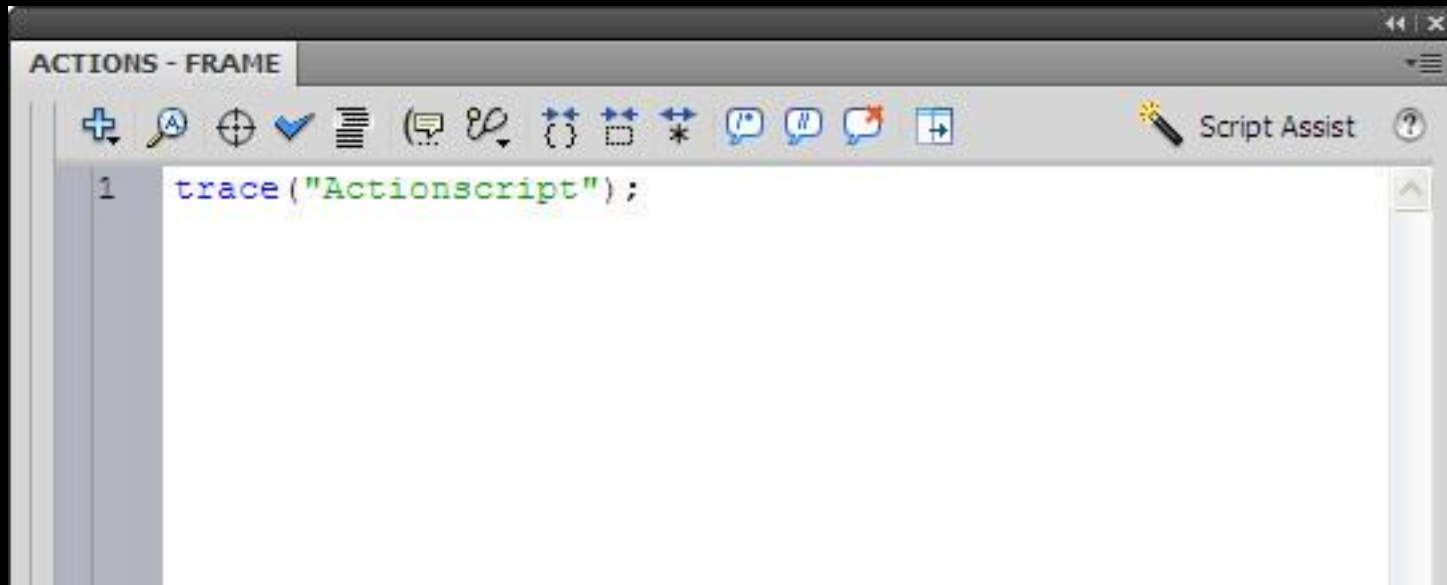
Para alterar fonte, faça estas modificações. Como o Flash possui auto-completar, não é preciso decorar a ordem dos parâmetros do construtor de TextFormat, e nem os atributos de TextField.

```
for(var i:int = 0 ; i < 10 ; i++) {
    if(i%2 == 0) {
        trace(i);
    }
}
```



Os comandos de repetição, de controle e os operadores são iguais aos de Java/C++.

Os comandos de Actionscript não precisam ser testados sempre dentro de uma classe. Eles podem ser chamados diretamente na aba Actions.



The screenshot shows a software interface window titled "ACTIONS - FRAME". The window has a toolbar with various icons for actions and a "Script Assist" button. The main area is a text editor containing the following code:

```
1 trace("Actionscript");
```

Exemplo de uso de função. Funções “soltas” não são colocadas em arquivo .as, apenas na aba de Ações.

```
function soma(i:int) {  
    var soma:int = 0;  
    for(var j:int = 0 ; j <= i ; j++) {  
        soma+=j;  
    }  
    return soma;  
}
```

```
var txt:TextField = new TextField();  
txt.text = soma(10);  
addChild(txt);
```



# Desenho de Primitivas

## Linha Reta:

```
graphics.strokeStyle(espessura, cor, alpha);  
graphics.moveTo(x_inicial, y_inicial);  
graphics.lineTo(x_final, y_final);
```

## Linha Curva:

```
graphics.strokeStyle(espessura, cor, alpha);  
graphics.moveTo(x_inicial, y_inicial);  
graphics.curveTo(curva_x, curva_y, x_final, y_final);
```



# Desenho de Primitivas

## Círculo:

```
graphics.strokeStyle(espessura, cor, alpha);  
graphics.beginFill(cor_interior, alpha);  
graphics.drawCircle(x, y, raio);  
graphics.endFill();
```

## Retângulo:

```
graphics.strokeStyle(espessura, cor, alpha);  
graphics.beginFill(cor_interior, alpha);  
graphics.drawRect(x, y, largura, altura);  
graphics.endFill();
```



# Desenho de Primitivas

## Elipse:

```
graphics.lineStyle(espessura, cor, alpha);  
graphics.beginFill(cor_interior, alpha);  
graphics.drawEllipse(x, y, largura, altura);  
graphics.endFill();
```

## Retângulo Arredondado:

```
graphics.lineStyle(espessura, cor, alpha);  
graphics.beginFill(cor_interior, alpha);  
graphics.drawRoundRect(x, y, largura, altura, curvatura);  
graphics.endFill();
```

Primitivas podem ser feitas também “na mão”, usando as ferramentas do Flash.



# Imagens da Biblioteca

Os arquivos importados para uso no flash ficam armazenados numa biblioteca de arquivos, no .fla. Para carregar uma imagem (ou qualquer outro tipo de arquivo), vá em: **Arquivo > Importar > Importar para a Biblioteca**



Para utilizar uma imagem da biblioteca num código Actionscript, é preciso avisar ao Flash que a imagem será usada.

No CS3: Clique no nome da imagem, e em Linkage, marque a opção “Exportar para Actionscript”

No CS4: A mesma coisa, mas em vez de Linkage, a opção estará em Propriedades.



# Imagens da Biblioteca

Bitmaps (normalmente) não são usados diretamente no Actionscript como objetos. São mais comuns dentro de MovieClips:



# Imagens da Biblioteca

---

Depois de criado o MovieClip com o bitmap, estes comandos o adicionam na tela:

---

```
var gdp:MovieClip = new GDP();  
gdp.x = 280;  
gdp.y = 200;  
addChild(gdp);
```

---

Exercício:

Desenhe na cena um quadrado transparente por primitiva, e coloque-o sobre um bitmap também transparente. Isso dentro de uma classe.

---



# Imagens da Biblioteca

## Solução:

```
package {
    import flash.display.*;

    public class Ex03 extends MovieClip {
        public function Ex03() {

            var gdp:MovieClip = new GDP();
            gdp.x = 100;
            gdp.y = 100;
            gdp.alpha = 0.6;
            addChild(gdp);

            var quadrado:MovieClip = new MovieClip();
            with(quadrado.graphics) {
               LineStyle(5,0x000000,0.7);
               beginFill(0xFA4434,0.9);
               drawRect(100,100,200,210);
               endFill();
            }
            addChild(quadrado);
        }
    }
}
```



# Eventos de Mouse

```
package {
    import flash.display.MovieClip;
    import flash.events.MouseEvent;

    public class Ex04 extends MovieClip {
        var n:MovieClip = new MovieClip();
        var m:MovieClip = new MovieClip();
        public function Ex04() {
            n.graphics.lineStyle(3,0x000000);
            n.graphics.beginFill(0x333333);
            n.graphics.drawCircle(40,40,10);

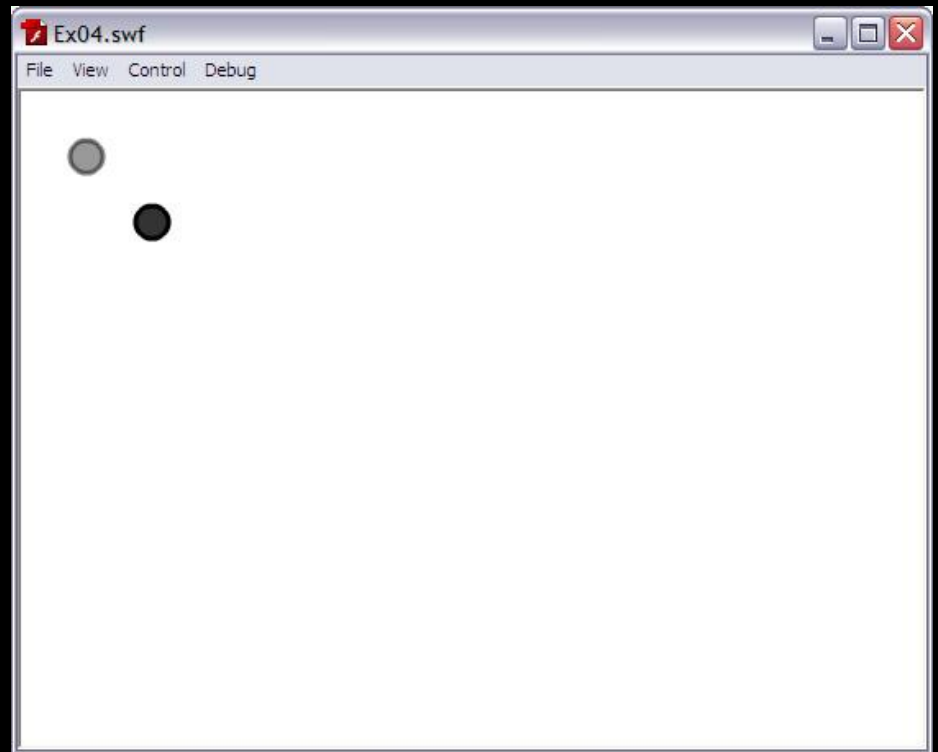
            m.graphics.lineStyle(3,0x000000);
            m.graphics.beginFill(0x333333);
            m.graphics.drawCircle(80,80,10);

            addChild(m);
            addChild(n);
            stage.addEventListener(MouseEvent.MOUSE_OVER,menosAlpha);
            stage.addEventListener(MouseEvent.MOUSE_OUT, maisAlpha);
        }
        function menosAlpha(evt:MouseEvent) {
            if(evt.target == n) {
                evt.target.alpha = 0.5;
            }
        }
        function maisAlpha(evt:MouseEvent) {
            evt.target.alpha = 1;
        }
    }
}
```



# Eventos de Mouse

Ao passar o mouse sobre o círculo n, ele fica semi transparente, mas o mesmo não ocorre com o círculo m.



# Eventos de Mouse

Eventos disparados quando:

- `MOUSE_DOWN`: Um botão é clicado;
- `MOUSE_MOVE`: A seta se movimenta;
- `MOUSE_OUT`: A seta está fora de uma área;
- `MOUSE_OVER`: A seta está sobre uma área;
- `MOUSE_UP`: Após o clique, quando a seta continua acima;
- `MOUSE_WHEEL`: A roda é movida;
- `CLICK`: É dado um clique;
- `DOUBLE_CLICK`: São dados dois cliques;
- `ROLL_OUT`: A seta passa sobre uma área;
- `ROLL_OVER`: A seta sai de uma área;



# Eventos de Teclado

```
package {
    import flash.display.MovieClip;
    import flash.events.KeyboardEvent;
    import flash.ui.Keyboard;

    public class Ex05 extends MovieClip {

        var n:MovieClip = new MovieClip();
        public function Ex05() {
            n.graphics.lineStyle(3,0x000000);
            n.graphics.beginFill(0x333333);
            n.graphics.drawCircle(100,100,10);
            addChild(n);
            // Existem apenas KEY_UP e KEY_DOWN
            stage.addEventListener(KeyboardEvent.KEY_DOWN,andar);
        }

        function andar(evt:KeyboardEvent) {
            if(evt.keyCode == Keyboard.LEFT) { n.x-- }
            if(evt.keyCode == Keyboard.RIGHT) { n.x++ }
            if(evt.keyCode == Keyboard.UP) { n.y-- }
            if(evt.keyCode == Keyboard.DOWN) { n.y++ }
        }
    }
}
```

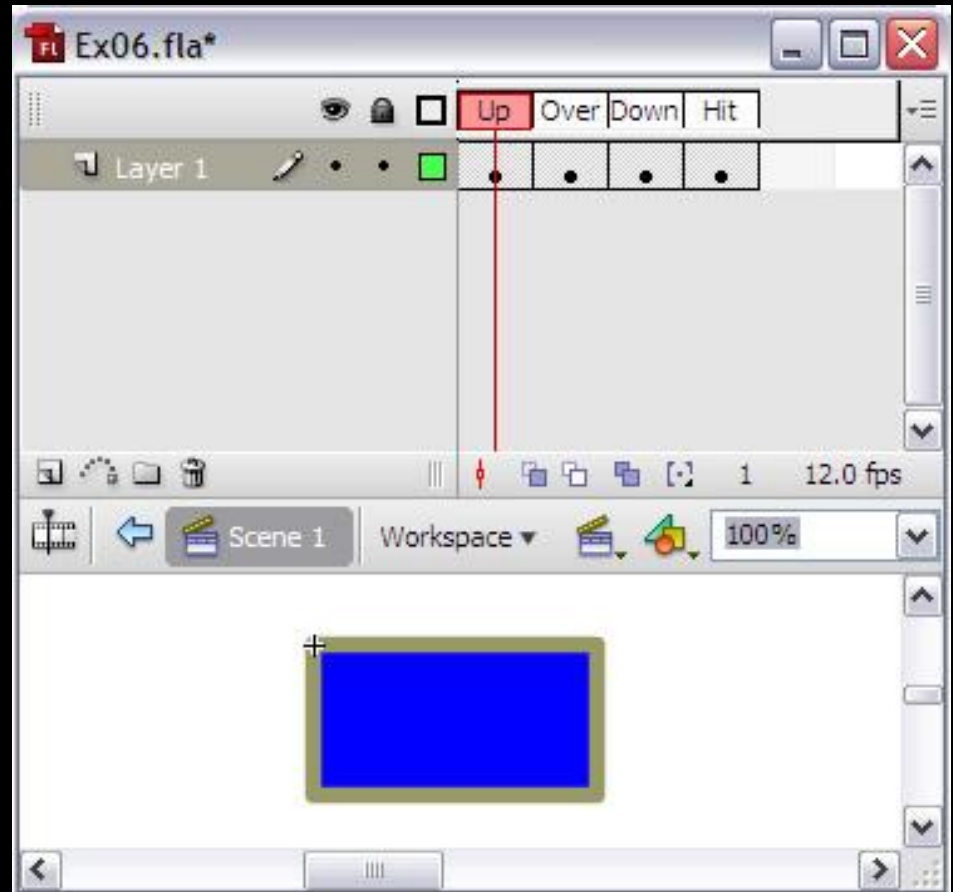


# Botão

Modo 1: Botões podem ser criados “na mão”, e adicionados com o `addChild`

Modo 2: Podem receber quatro MovieClips por parâmetro:

```
new SimpleButton(  
new ButtonUp(),  
new ButtonOver(),  
new ButtonDown(),  
new ButtonHit());
```



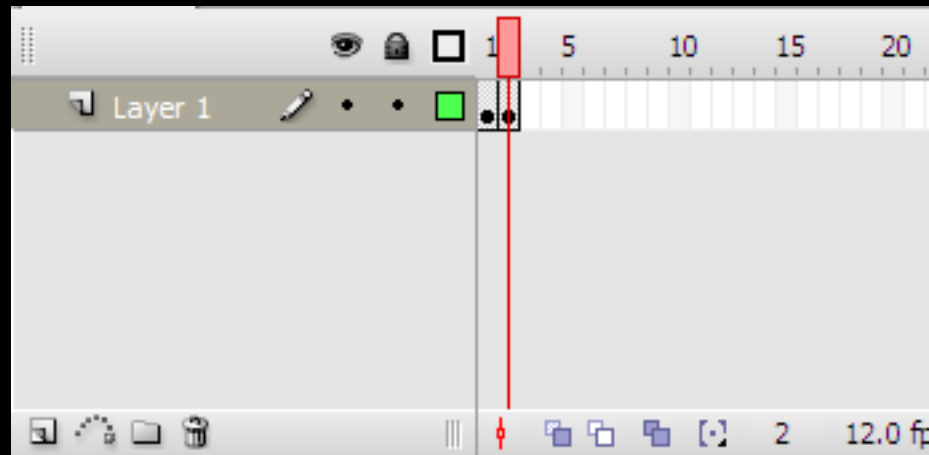
# Botão com Evento de clique

```
var but1:SimpleButton = new Bt1();  
addChild(but1);  
  
addEventListener(MouseEvent.CLICK, clique);  
  
function clique(evt:MouseEvent) {  
    trace("Você me clicou");  
}
```



# Programando Animações

Uma animação é feita adicionando-se mais de um quadro na linha do tempo:



# Programando Animações

É também possível adicionar quadros num MovieClip através do evento ENTER\_FRAME. Crie uma imagem, marque para exportar para Actionscript e adicione o seguinte código:

```
var treco:MovieClip = new Treco();
treco.x = 550;
treco.y = 150;
addChild(treco);

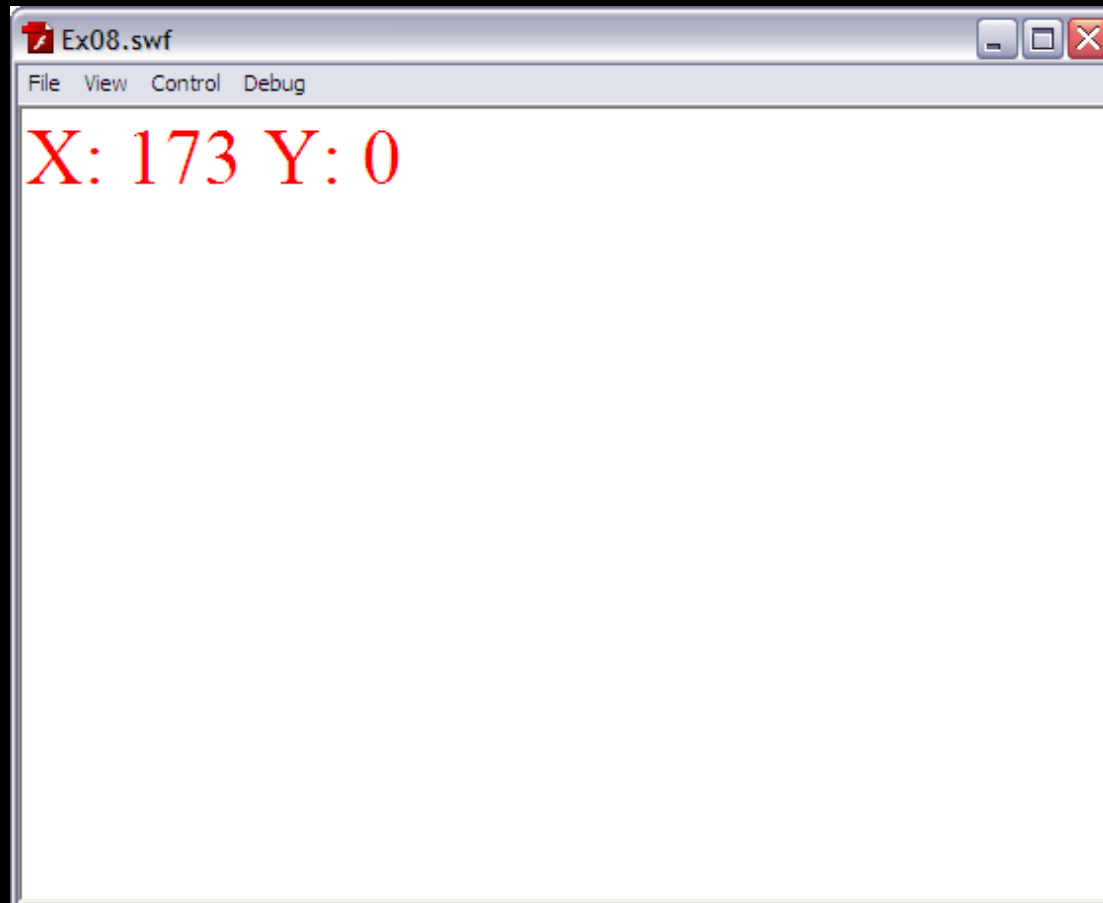
treco.addEventListener(Event.ENTER_FRAME,mover);

function mover(evt:Event) {
    if(treco.x > 0) {
        treco.x-=5;
    }
}
```



# Programando Animações

Exercício: Faça um programa que escreva num campo de texto os valores atuais de x e de y do mouse.



# Programando Animações

## Solução:

```
var texto:TextField = new TextField();
texto.defaultTextFormat = new TextFormat("Comic Sans",40,0xFF0000);
texto.width = 250;
addChild(texto);

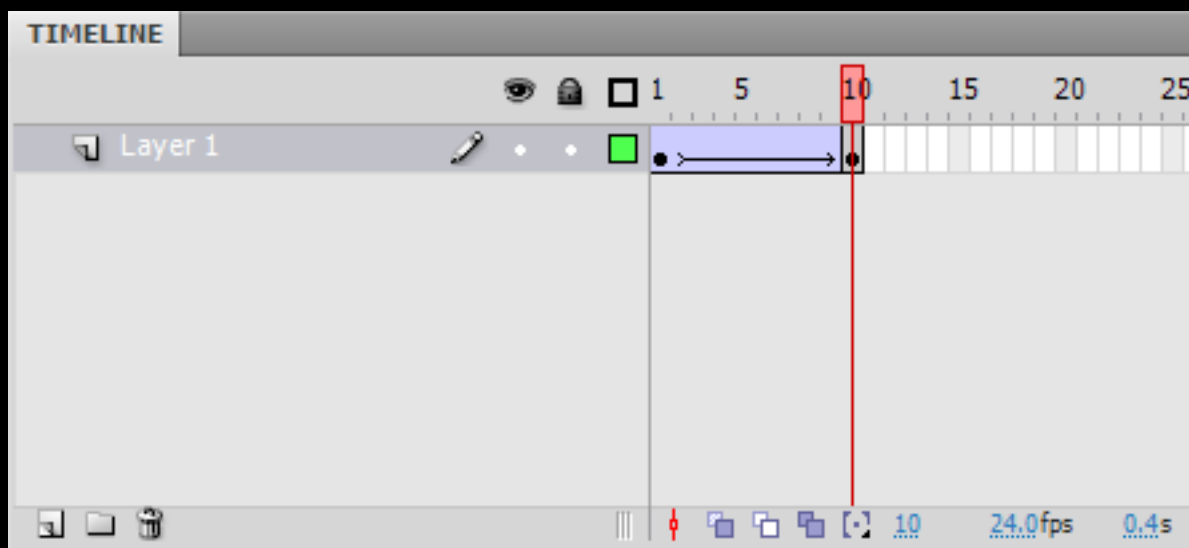
addEventListener(Event.ENTER_FRAME,posit);

function posit(evt:Event) {
    texto.text = "X: " + mouseX + " " + "Y: " + mouseY;
}
```



# Programando Animações

O Flash permite fazer interpolações entre quadros. Num novo arquivo, desenhe um círculo no quadro 1, e o mesmo círculo no quadro 10, em uma posição diferente. Os quadros 2-9 devem ser prolongamentos do quadro 1. Para fazer isso, selecione o quadro 10 e pressione F6 quando o círculo já estiver desenhado no primeiro quadro. Em seguida, clique em qualquer um dos quadros 2-9 com o botão direito do mouse e escolha **Motion Tween** no Flash CS3 ou **Classic Tween** no Flash CS4. O círculo moverá da posição que se encontra no Frame 1 até a posição que se encontra no Frame 10.



# Programando Animações

Para fazer Interpolações de Movimento com Actionscript, existe a classe Tween:

```
new Tween(objeto, propriedade, função, inicio, fim, duração, emSegundo?)
```

Exemplo de uso:

```
new Tween(bola, "x", Strong.easeOut, 100, 200, 1, true)
```

Este exemplo faz com que a bola ande horizontalmente de 100 até 200. Strong.easeOut é uma das funções de efeito de movimento que podem ser usadas como terceiro parâmetro. Pesquise sobre as funções presentes em **Strong, Elastic e Bounce**.

Para alterar duas propriedades simultaneamente:

```
new Tween(bola, "x", Elastic.easeOut, 100, 200, 5, true)  
new Tween(bola, "alpha", Strong.easeOut, 0.1, 1, 5, true)
```



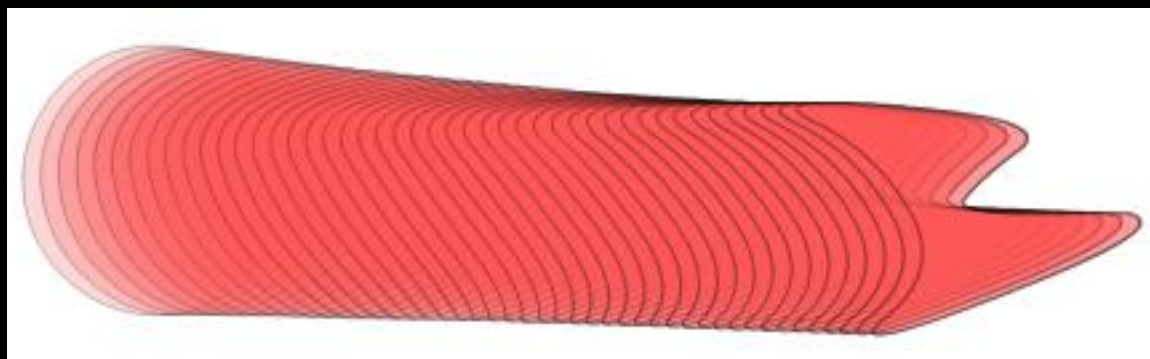
# Programando Animações

Strong, Elastic e Tween só podem ser utilizados se forem importados, mesmo na aba Actions. Para importá-los:

```
import fl.transitions.Tween;  
import fl.transitions.easing.*;
```

Teste também, através do Flash, a Interpolação de Forma (Shape Tween) e o Motion Tween do Flash CS4. O primeiro transforma uma imagem em outra totalmente diferente, e o segundo é uma versão melhorada do Motion antigo.

Opção Onion Skin marcada numa Interpolação de Forma:



# Temporização

Para controlar a frequência de certos movimentos durante uma animação, existe a classe Timer. Objetos desta classe podem sozinhos animar um objeto, seguindo a temporização definida.

Imports necessários quando utilizado em classe:

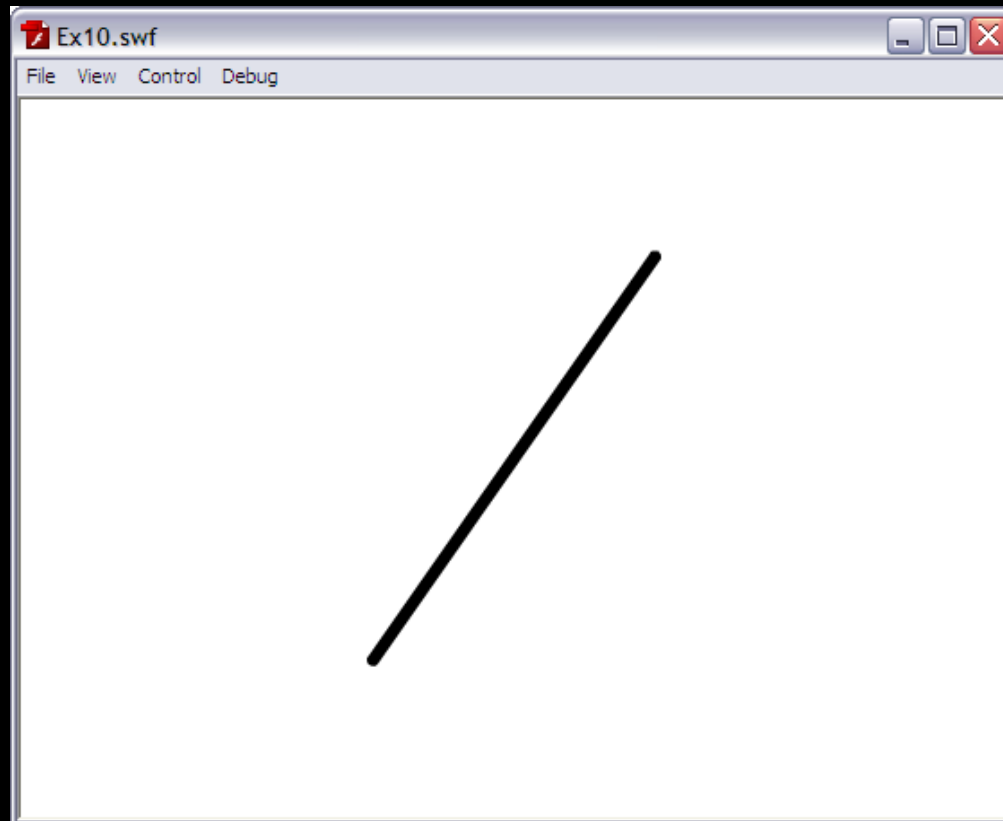
```
flash.events.TimerEvent;  
flash.utils.Timer;
```

```
// 1000 = 1 segundo  
var tempo:Timer = new Timer(200);  
var mov:MovieClip = new Mov();  
addChild(mov);  
mov.x = 550;  
mov.y = 200;  
  
tempo.addEventListener(TimerEvent.TIMER, anima);  
tempo.start();  
function anima(evt:TimerEvent) {  
    mov.x-=5;  
}
```



# Temporização

Exercício: Para rotacionar um objeto, usa-se a propriedade `rotation`, presente nos `MovieClips`, que armazena um valor entre `-180` e `180`. Faça uma animação de uma linha reta vertical que gire no sentido horário, de 5 em 5 graus a cada segundo.



# Temporização

Solução:

```
var linha:MovieClip = new Linha();  
linha.x = 275;  
linha.y = 200;  
addChild(linha);
```

```
var tempo:Timer = new Timer(1000);  
tempo.addEventListener(TimerEvent.TIMER, gira);  
tempo.start();
```

```
function gira(evt:TimerEvent) {  
    linha.rotation += 5;  
}
```



# Colisão de Imagens

O programador de jogos não precisa se preocupar com algoritmos de colisão, pois o Actionscript possui o seu próprio, que analisa o menor quadrado ao redor das imagens.

```
var img1:MovieClip = new Img01();
img1.x = 200;
img1.y = 200;
addChild(img1);
var img2:MovieClip = new Img02();
addChild(img2);
var txt:TextField = new TextField();
addChild(txt);

addEventListener(Event.ENTER_FRAME, colisao);
function colisao(evt:Event) {
    img2.x = mouseX;
    img2.y = mouseY;

    if(img2.hitTestObject(img1)) {
        txt.text = "Colidiu";
    } else {
        txt.text = "Não Colidiu";
    }
}
```



# Colisão de Imagens

É possível conferir a colisão de um ponto com uma imagem. Basta modificar o método `hitTestObject`. Neste caso, a colisão é perfeita.

```
if(img2.hitTestPoint(VALOR_X,VALOR_Y)) {  
    txt.text = "Colidiu";  
} else {  
    txt.text = "Não Colidiu";  
}  
}
```



# Som

Para tocar um arquivo de música, é preciso primeiro importá-lo para a biblioteca, e marcar a opção de exportação para ActionScript, assim como já foi feito com imagens.

```
var som:Som = new Som();  
var channel:SoundChannel = som.play();
```

Caso o arquivo esteja num site, é possível carregá-lo de uma url:

```
var som:Som = new Som();  
var url:URLRequest = new URLRequest("Som.mp3");  
som.load(url);
```



# Lista Encadeada

O tipo Array funciona como uma lista encadeada, e pode carregar itens de tipos diferentes ao mesmo tempo. Abaixo, os métodos principais:

Nome:	Exemplo:	Descrição:
[]	vetor[3]	Retorna o valor na posição.
push	vetor.push(35);	Inclui um valor no final.
pop	vetor.pop()	Remove e retorna o último valor.
unshift	vetor.unshift("a")	Adiciona no início do vetor.
shift	vetor.shift()	Remove e retorna o primeiro valor.
indexOf	vetor.indexOf("b")	Retorna a posição de um item.
sort	vetor.sort()	Ordena os elementos.