

# **PROGRAMAÇÃO PARA INTERNET RICA**

## **FUNÇÕES E EVENTOS EM ACTIONSCRIPT 3**

Prof. Dr. Daniel Caetano

2013 - 1

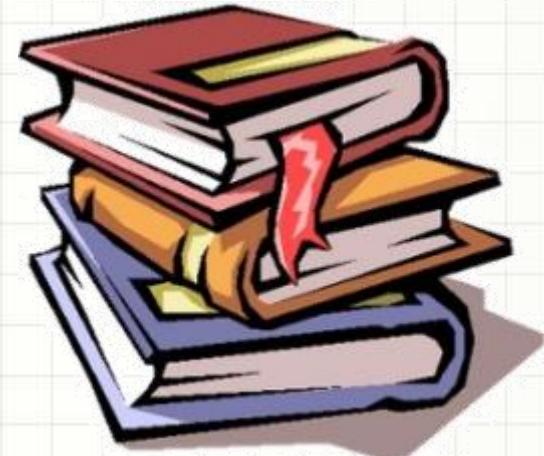
# Objetivos

- Entender funções em Flash
  - Compreender os eventos de interação
  - Compreender a contagem de tempo de quadros
  - Capacitar para criar animações interativas simples usando AS3 e o Flash CS4
- 
- **Trabalho B!**



# Material de Estudo

---



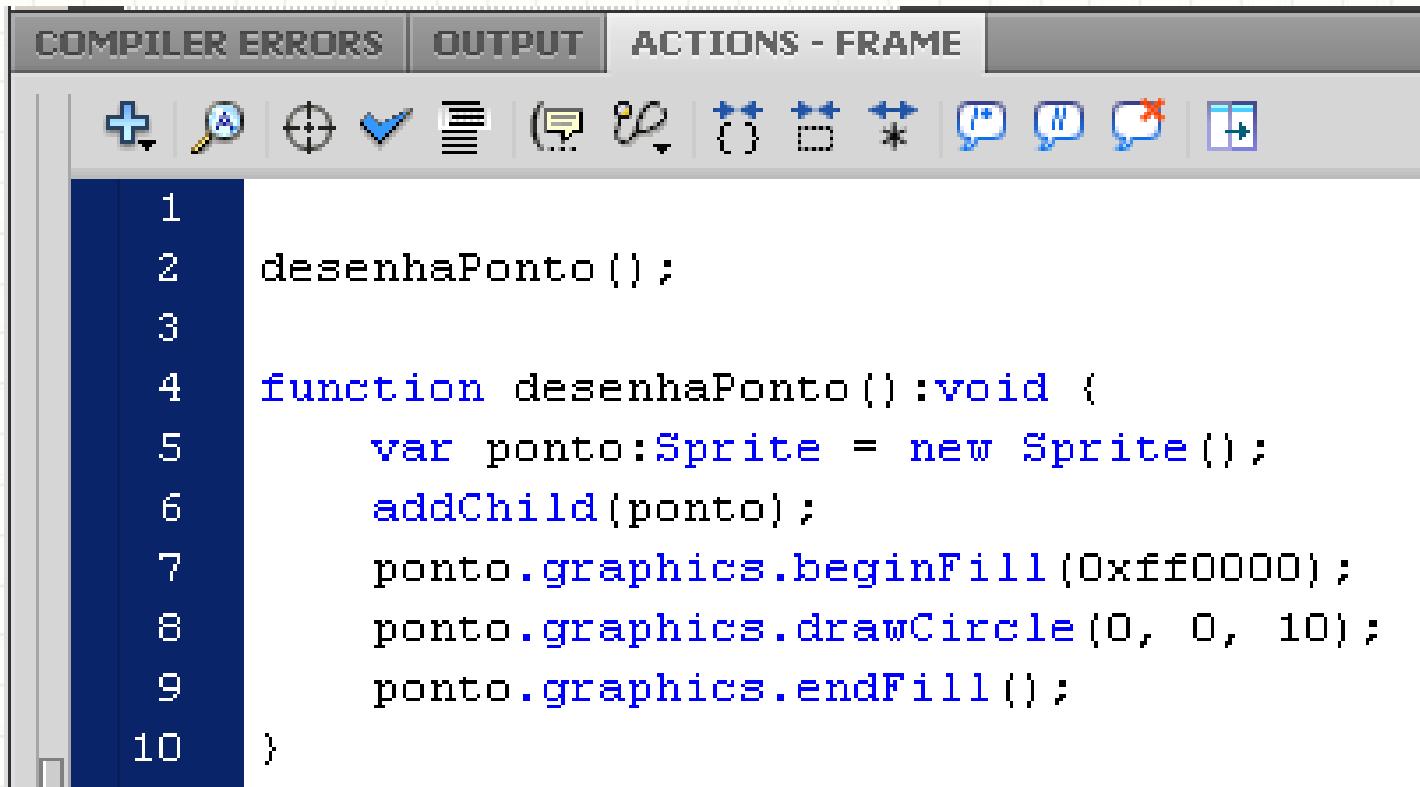
Material	Acesso ao Material
Tutorial Extra	<a href="http://www.caetano.eng.br/">http://www.caetano.eng.br/</a> Adobe Flash CS4
Apresentação	<a href="http://www.caetano.eng.br/">http://www.caetano.eng.br/</a> (Aula 13)
Internet	<a href="http://slekx.com/as3-intro/">http://slekx.com/as3-intro/</a> Free AS3 Tutorial
Google	ActionScript 3 AS3



# **FUNÇÕES EM ACTIONSCRIPT 3**

# Funções em ActionScript 3

- As funções são parecidas com C



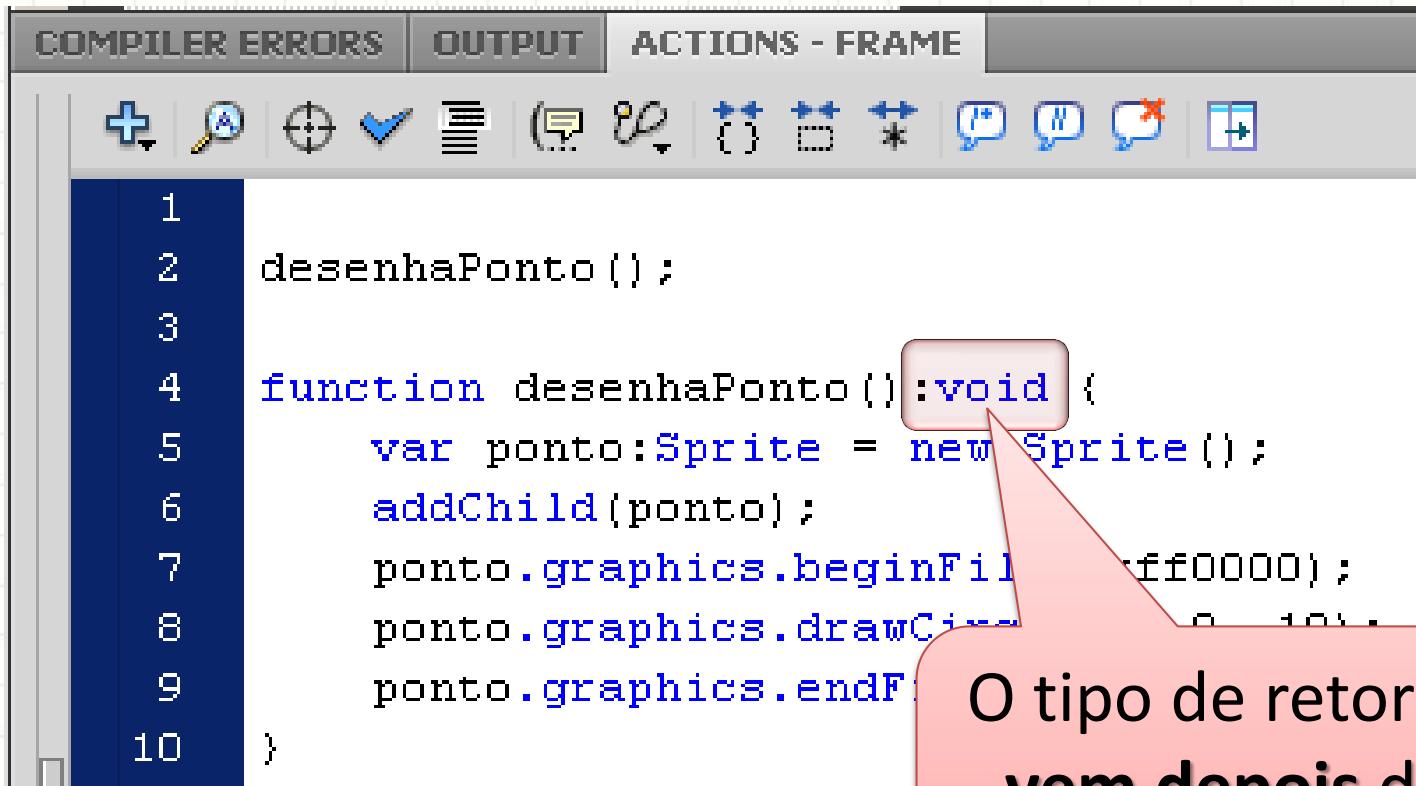
The screenshot shows the Flash IDE interface with the Actions - Frame panel selected. The panel has tabs for Compiler Errors, Output, and Actions - Frame. Below the tabs is a toolbar with various icons. The main area contains the following ActionScript code:

```
1
2 desenhaPonto();
3
4 function desenhaPonto():void {
5     var ponto:Sprite = new Sprite();
6     addChild(ponto);
7     ponto.graphics.beginFill(0xff0000);
8     ponto.graphics.drawCircle(0, 0, 10);
9     ponto.graphics.endFill();
10 }
```

- Experimente!

# Funções em ActionScript 3

- As funções são parecidas com JavaScript



The screenshot shows the Flash IDE interface with three tabs at the top: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'ACTIONS - FRAME' tab is selected. Below the tabs is a toolbar with various icons. The main area displays ActionScript code:

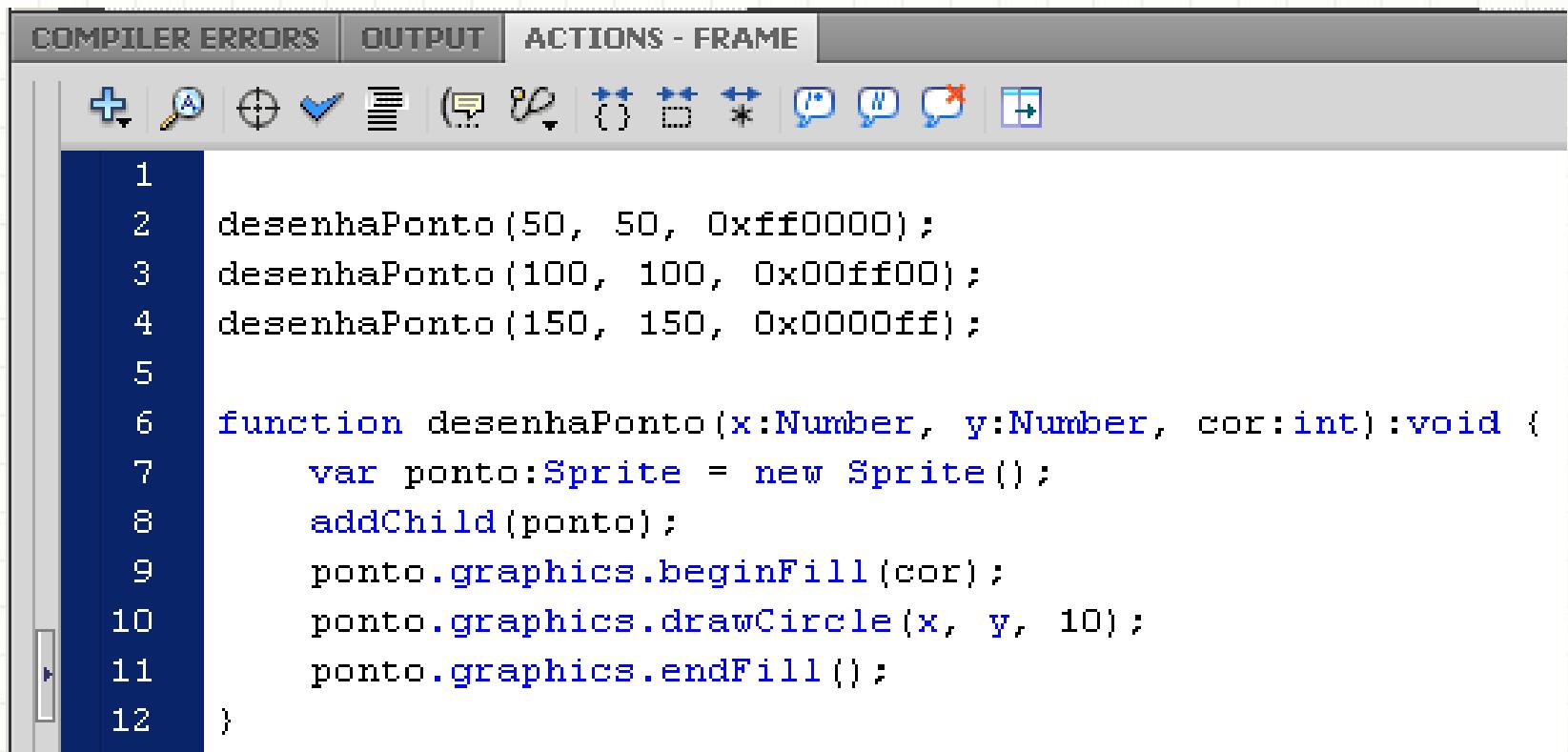
```
1
2  desenhaPonto();
3
4  function desenhaPonto():void {
5      var ponto:Sprite = new Sprite();
6      addChild(ponto);
7      ponto.graphics.beginFill(0xff0000);
8      ponto.graphics.drawCircle(0, 0, 10);
9      ponto.graphics.endFill();
10 }
```

A red callout bubble points to the ':void' keyword in the fourth line of the code. The text inside the bubble reads: 'O tipo de retorno vem depois do nome da função'.

- Experimente!

# Funções em ActionScript 3

- Função com parâmetros são mais úteis:



The screenshot shows the Flash IDE's Actions - FRAME panel. The tab bar at the top has three tabs: COMPILER ERRORS, OUTPUT, and ACTIONS - FRAME, with ACTIONS - FRAME selected. Below the tabs is a toolbar with various icons for file operations like new, open, save, and publish. The main area contains the following ActionScript code:

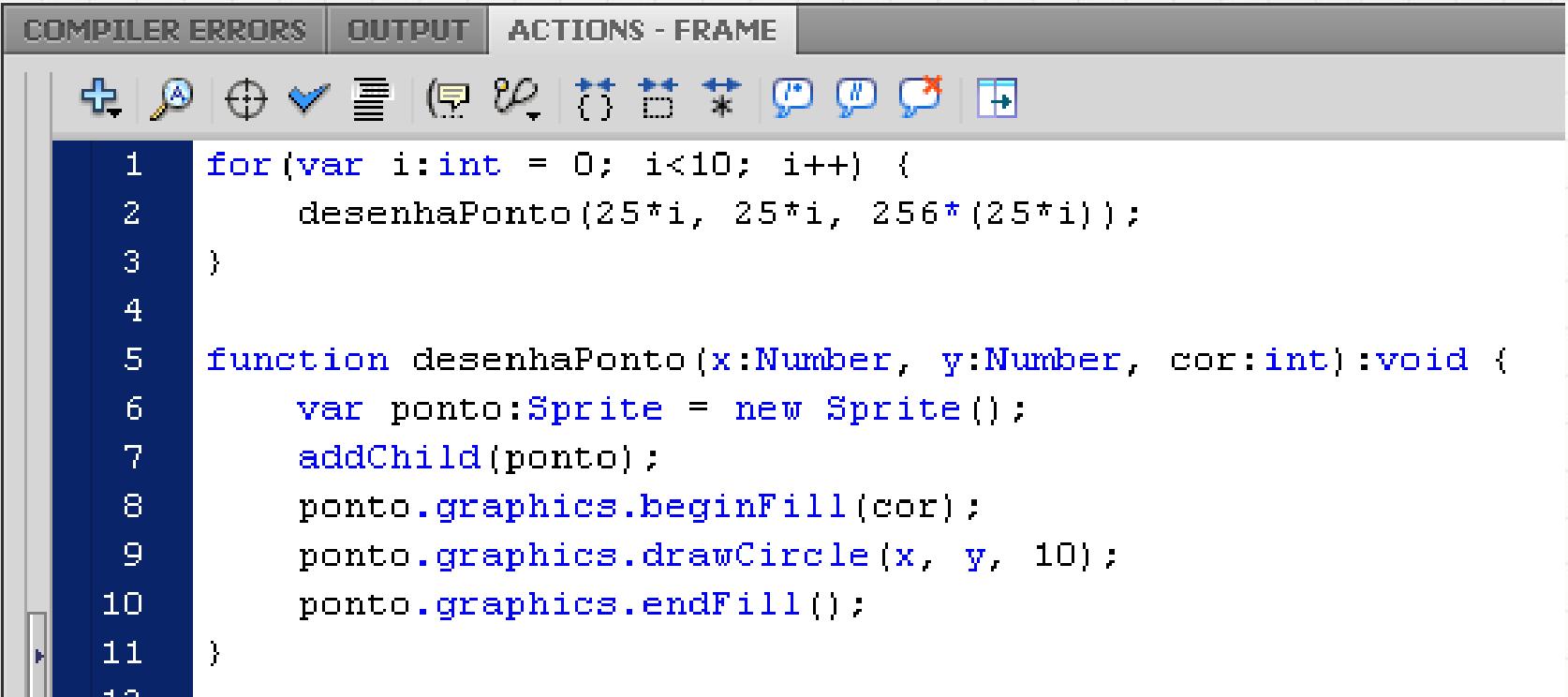
```
1
2 desenhaPonto(50, 50, 0xff0000);
3 desenhaPonto(100, 100, 0x00ff00);
4 desenhaPonto(150, 150, 0x0000ff);

6 function desenhaPonto(x:Number, y:Number, cor:int):void {
7     var ponto:Sprite = new Sprite();
8     addChild(ponto);
9     ponto.graphics.beginFill(cor);
10    ponto.graphics.drawCircle(x, y, 10);
11    ponto.graphics.endFill();
12 }
```

- Experimente!

# Funções em ActionScript 3

- Misturando tudo...



The screenshot shows the Flash IDE's Actions - Frame panel. The tab bar at the top has three tabs: COMPILER ERRORS, OUTPUT, and ACTIONS - FRAME, with ACTIONS - FRAME being the active tab. Below the tabs is a toolbar with various icons for editing and debugging. The main area contains the following ActionScript code:

```
1  for(var i:int = 0; i<10; i++) {
2      desenhaPonto(25*i, 25*i, 256^(25*i));
3  }
4
5  function desenhaPonto(x:Number, y:Number, cor:int):void {
6      var ponto:Sprite = new Sprite();
7      addChild(ponto);
8      ponto.graphics.beginFill(cor);
9      ponto.graphics.drawCircle(x, y, 10);
10     ponto.graphics.endFill();
11 }
```

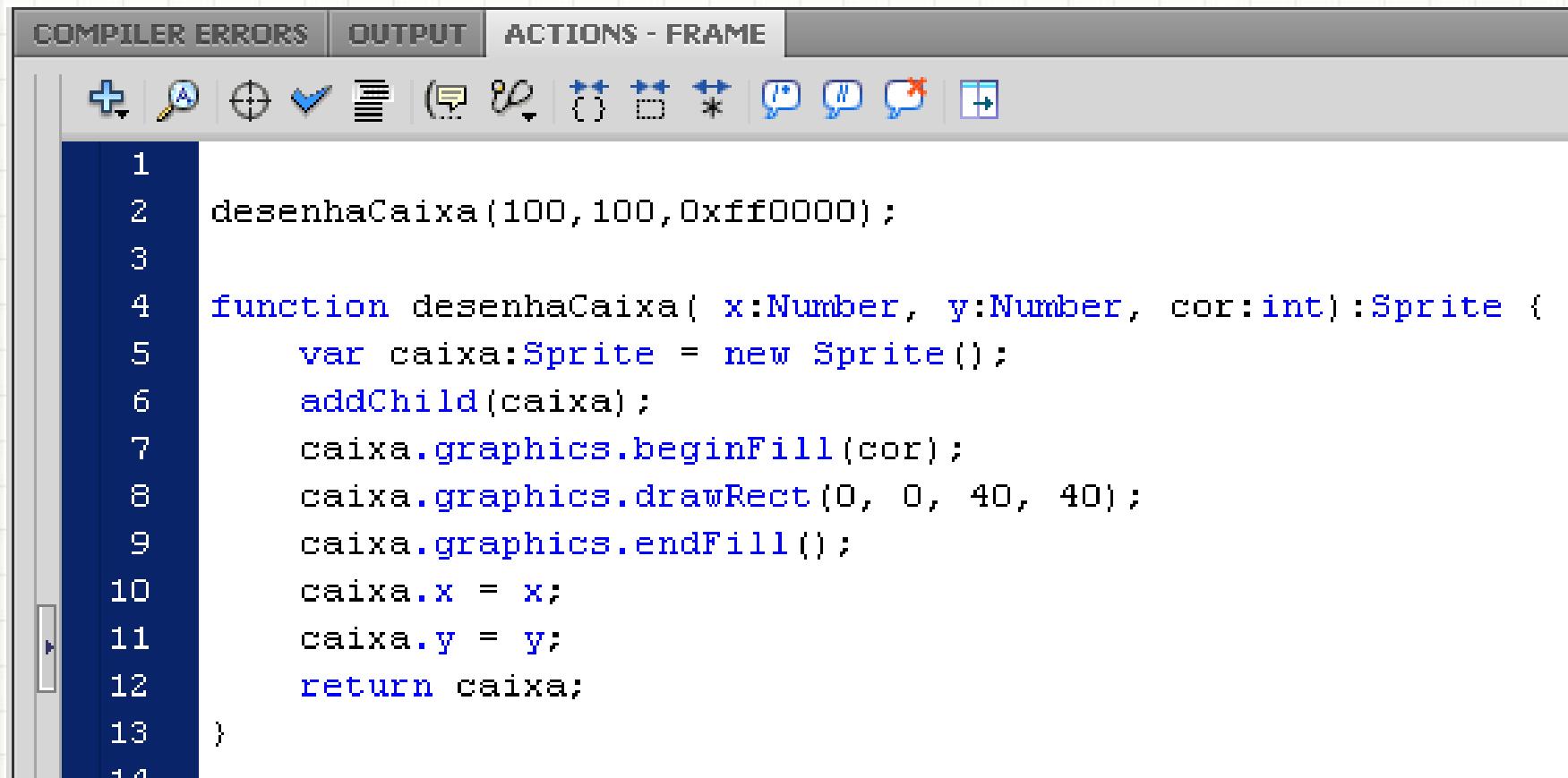
- Experimente!



# **EXERCITANDO O ACTIONSCRIPT 3**

# Introdução

- Vamos iniciar retomando a aula passada...

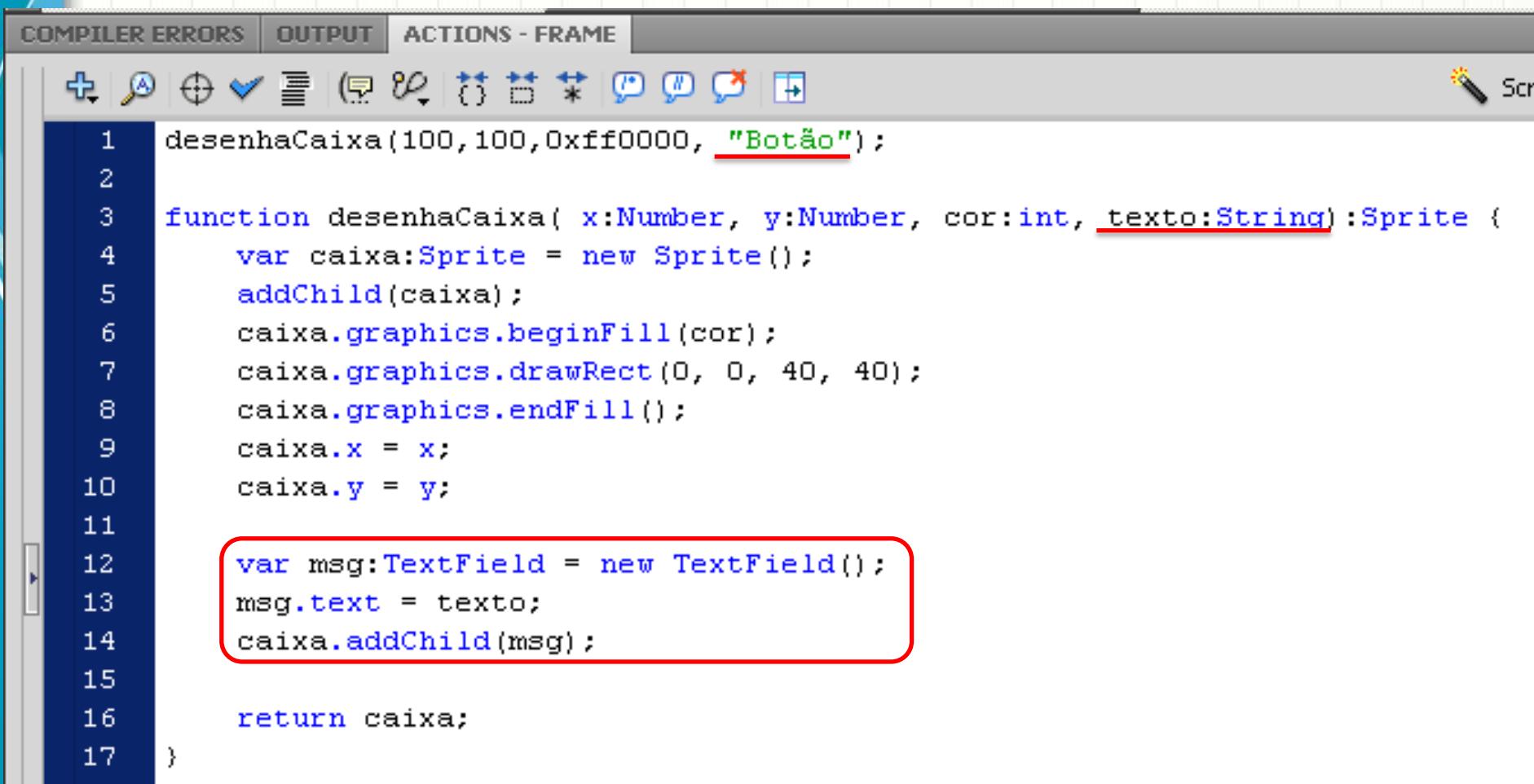


The screenshot shows a software interface with a dark-themed code editor. At the top, there are three tabs: "COMPILER ERRORS", "OUTPUT", and "ACTIONS - FRAME". Below the tabs is a toolbar with various icons, including a plus sign, a magnifying glass, a target, a checkmark, a list, a file, a person, a star, a speech bubble, and a square. The code editor itself has a dark background with white text. The code is as follows:

```
1 desenhaCaixa(100,100,0xff0000);
2
3
4 function desenhaCaixa( x:Number, y:Number, cor:int ):Sprite {
5     var caixa:Sprite = new Sprite();
6     addChild(caixa);
7     caixa.graphics.beginFill(cor);
8     caixa.graphics.drawRect(0, 0, 40, 40);
9     caixa.graphics.endFill();
10    caixa.x = x;
11    caixa.y = y;
12    return caixa;
13 }
14 }
```

# Introdução

- Adicionando texto no botão...



The screenshot shows a software interface with a toolbar at the top and a code editor below. The toolbar has tabs for 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The code editor displays the following AS3 code:

```
1 desenhaCaixa(100,100,0xff0000, "Botão");
2
3 function desenhaCaixa( x:Number, y:Number, cor:int, texto:String ):Sprite {
4     var caixa:Sprite = new Sprite();
5     addChild(caixa);
6     caixa.graphics.beginFill(cor);
7     caixa.graphics.drawRect(0, 0, 40, 40);
8     caixa.graphics.endFill();
9     caixa.x = x;
10    caixa.y = y;
11
12    var msg:TextField = new TextField();
13    msg.text = texto;
14    caixa.addChild(msg);
15
16    return caixa;
17 }
```

A red rectangular box highlights the code block from line 12 to line 16, which creates a new `TextField`, sets its `text` property to `texto`, and adds it as a child to the `caixa` sprite.

# Introdução

- Deixando o texto em negrito e... branco

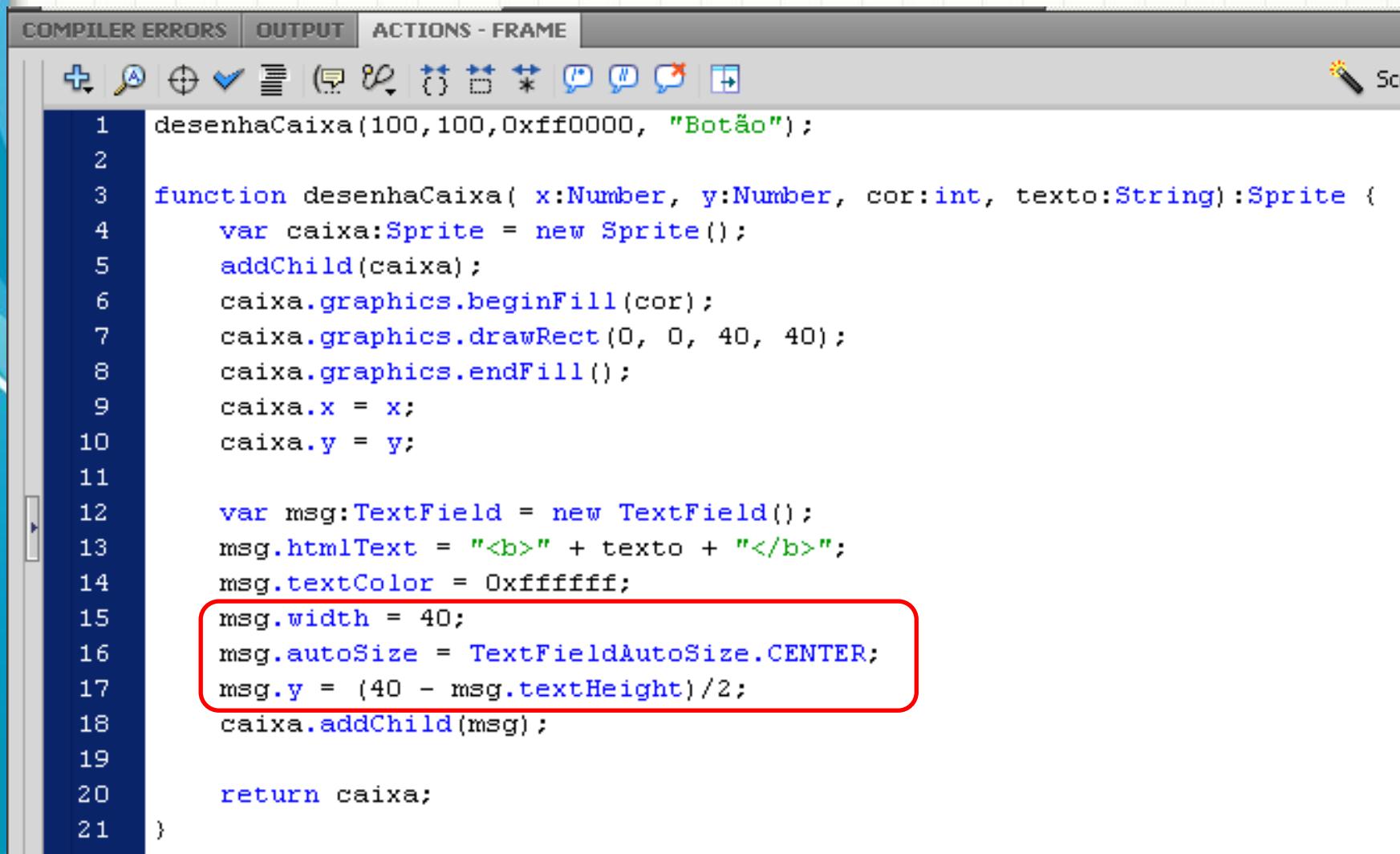
The screenshot shows a software interface with a toolbar at the top and a code editor below. The toolbar has tabs for 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The code editor displays the following ActionScript:

```
1 desenhaCaixa(100,100,0xff0000, "Botão");
2
3 function desenhaCaixa( x:Number, y:Number, cor:int, texto:String ):Sprite {
4     var caixa:Sprite = new Sprite();
5     addChild(caixa);
6     caixa.graphics.beginFill(cor);
7     caixa.graphics.drawRect(0, 0, 40, 40);
8     caixa.graphics.endFill();
9     caixa.x = x;
10    caixa.y = y;
11
12    var msg:TextField = new TextField();
13    msg.htmlText = "<b>" + texto + "</b>";
14    msg.textColor = 0xffffffff;
15    caixa.addChild(msg);
16
17    return caixa;
18 }
```

The lines of code from 13 to 14 are highlighted with a red rectangular box. The variable 'msg' is defined as a new instance of the 'TextField' class. The 'htmlText' property is set to the string '<b>' followed by the value of the 'texto' parameter, and then '</b>'. The 'textColor' property is set to the color 0xffffffff, which is white.

# Introdução

- Vamos centralizar o texto...



The screenshot shows a portion of the Adobe Flash IDE interface. The top menu bar includes tabs for "COMPILER ERRORS", "OUTPUT", and "ACTIONS - FRAME". Below the menu is a toolbar with various icons for file operations and selection. The main workspace displays the following ActionScript code:

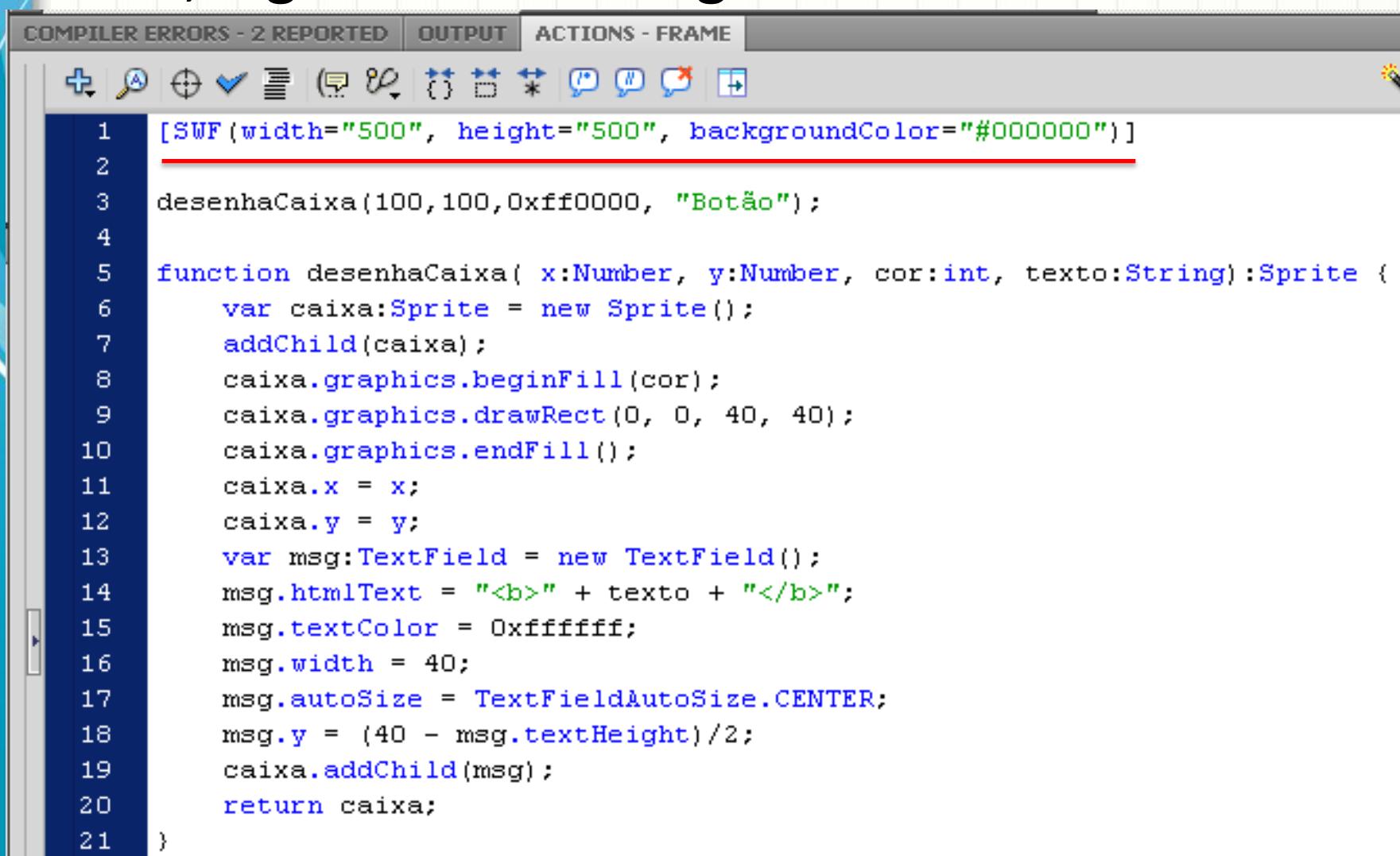
```
1 desenhaCaixa(100,100,0xff0000, "Botão");
2
3 function desenhaCaixa( x:Number, y:Number, cor:int, texto:String ):Sprite {
4     var caixa:Sprite = new Sprite();
5     addChild(caixa);
6     caixa.graphics.beginFill(cor);
7     caixa.graphics.drawRect(0, 0, 40, 40);
8     caixa.graphics.endFill();
9     caixa.x = x;
10    caixa.y = y;
11
12    var msg:TextField = new TextField();
13    msg.htmlText = "<b>" + texto + "</b>";
14    msg.textColor = 0xffffffff;
15    msg.width = 40;
16    msg.autoSize = TextFieldAutoSize.CENTER;
17    msg.y = (40 - msg.textHeight)/2;
18    caixa.addChild(msg);
19
20    return caixa;
21 }
```

A red rectangular box highlights the following lines of code, which are responsible for centering the text within the button's width:

```
msg.width = 40;
msg.autoSize = TextFieldAutoSize.CENTER;
msg.y = (40 - msg.textHeight)/2;
```

# Introdução

- E, algo novo... Configurar o cenário...



The screenshot shows a software interface with a toolbar at the top and a code editor below. The toolbar has tabs: COMPILER ERRORS - 2 REPORTED, OUTPUT, and ACTIONS - FRAME. The ACTIONS - FRAME tab is selected. The code editor displays the following AS3 code:

```
1 [SWF(width="500", height="500", backgroundColor="#000000")]
2
3 desenhaCaixa(100,100,0xff0000, "Botão");
4
5 function desenhaCaixa( x:Number, y:Number, cor:int, texto:String ):Sprite {
6     var caixa:Sprite = new Sprite();
7     addChild(caixa);
8     caixa.graphics.beginFill(cor);
9     caixa.graphics.drawRect(0, 0, 40, 40);
10    caixa.graphics.endFill();
11    caixa.x = x;
12    caixa.y = y;
13    var msg:TextField = new TextField();
14    msg.htmlText = "<b>" + texto + "</b>";
15    msg.textColor = 0xffffffff;
16    msg.width = 40;
17    msg.autoSize = TextFieldAutoSize.CENTER;
18    msg.y = (40 - msg.textHeight)/2;
19    caixa.addChild(msg);
20    return caixa;
21 }
```

The first line of code, which defines the SWF parameters, has a red underline, indicating a syntax error or warning.



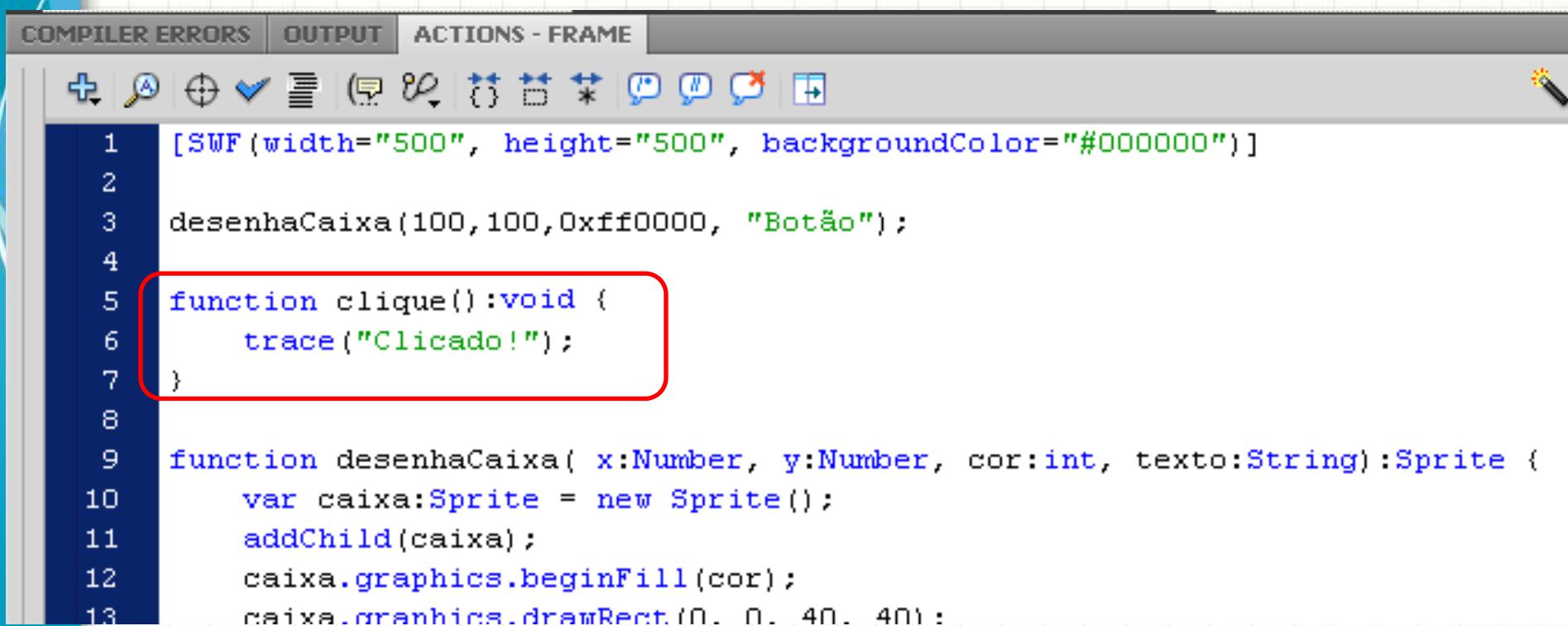
# **EVENTOS EM ACTIONSCRIPT 3**

# Introdução

- Interação JavaScript é baseada em eventos...
  - onclick, onmouseover, onmouseout...
- ActionScript também depende de eventos
  - Interação com o mouse
  - Interação com o teclado
  - Controle de tempo...

# Eventos: Clique (Mouse)

- Criando função que será executada no clique



The screenshot shows a Flash IDE interface with three tabs at the top: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. Below the tabs is a toolbar with various icons. The main area contains ActionScript code. A red rectangle highlights the function definition from line 5 to line 7.

```
1 [SWF(width="500", height="500", backgroundColor="#000000")]
2
3 desenhaCaixa(100,100,0xff0000, "Botão");
4
5 function clique():void {
6     trace("Clicado!");
7 }
8
9 function desenhaCaixa( x:Number, y:Number, cor:int, texto:String):Sprite {
10    var caixa:Sprite = new Sprite();
11    addChild(caixa);
12    caixa.graphics.beginFill(cor);
13    caixa.graphics.drawRect(0, 0, 40, 40);
```

# Eventos: Clique (Mouse)

- Associando função ao clique:

The screenshot shows the Flash IDE interface with the 'ACTIONS - FRAME' tab selected. The code in the panel is as follows:

```
[SWF(width="500", height="500", backgroundColor="#000000")]

desenhaCaixa(100,100,0xff0000, "Botão");
stage.addEventListener(MouseEvent.CLICK, clique);

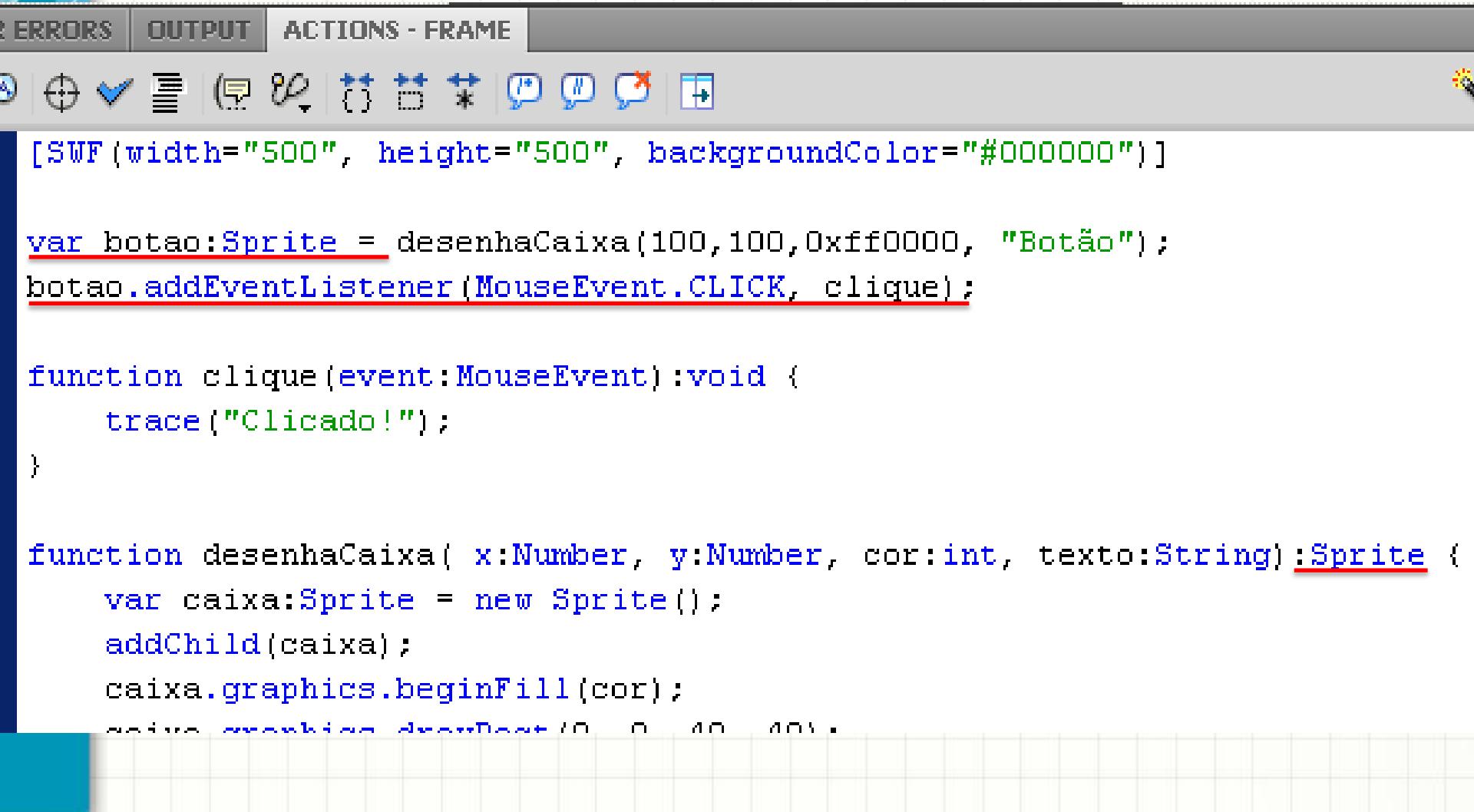
function clique(event:MouseEvent):void {
    trace("Clicado!");
}

function desenhaCaixa( x:Number, y:Number, cor:int, texto:String):Sprite {
    var caixa:Sprite = new Sprite();
    addChild(caixa);
    caixa.graphics.beginFill(cor);
    caixa.graphics.drawRect(0, 0, 40, 40);
```

The line `stage.addEventListener(MouseEvent.CLICK, clique);` is highlighted with a red underline.

# Eventos: Clique (Mouse)

- E para associar o clique ao botão?



The screenshot shows a Flash development environment with the following interface elements:

- Top bar: Includes tabs for "ERRORS", "OUTPUT", and "ACTIONS - FRAME".
- Toolbar: Contains icons for selection, zoom, and various actions.
- Code editor:

```
[SWF(width="500", height="500", backgroundColor="#000000")]

var botao:Sprite = desenhaCaixa(100,100,0xff0000, "Botão");
botao.addEventListener(MouseEvent.CLICK, clique);

function clique(event:MouseEvent):void {
    trace("Clicado!");
}

function desenhaCaixa( x:Number, y:Number, cor:int, texto:String):Sprite {
    var caixa:Sprite = new Sprite();
    addChild(caixa);
    caixa.graphics.beginFill(cor);
    caixa.graphics.drawRect(0, 0, 40, 40);
```

# Eventos: Clique (Mouse)

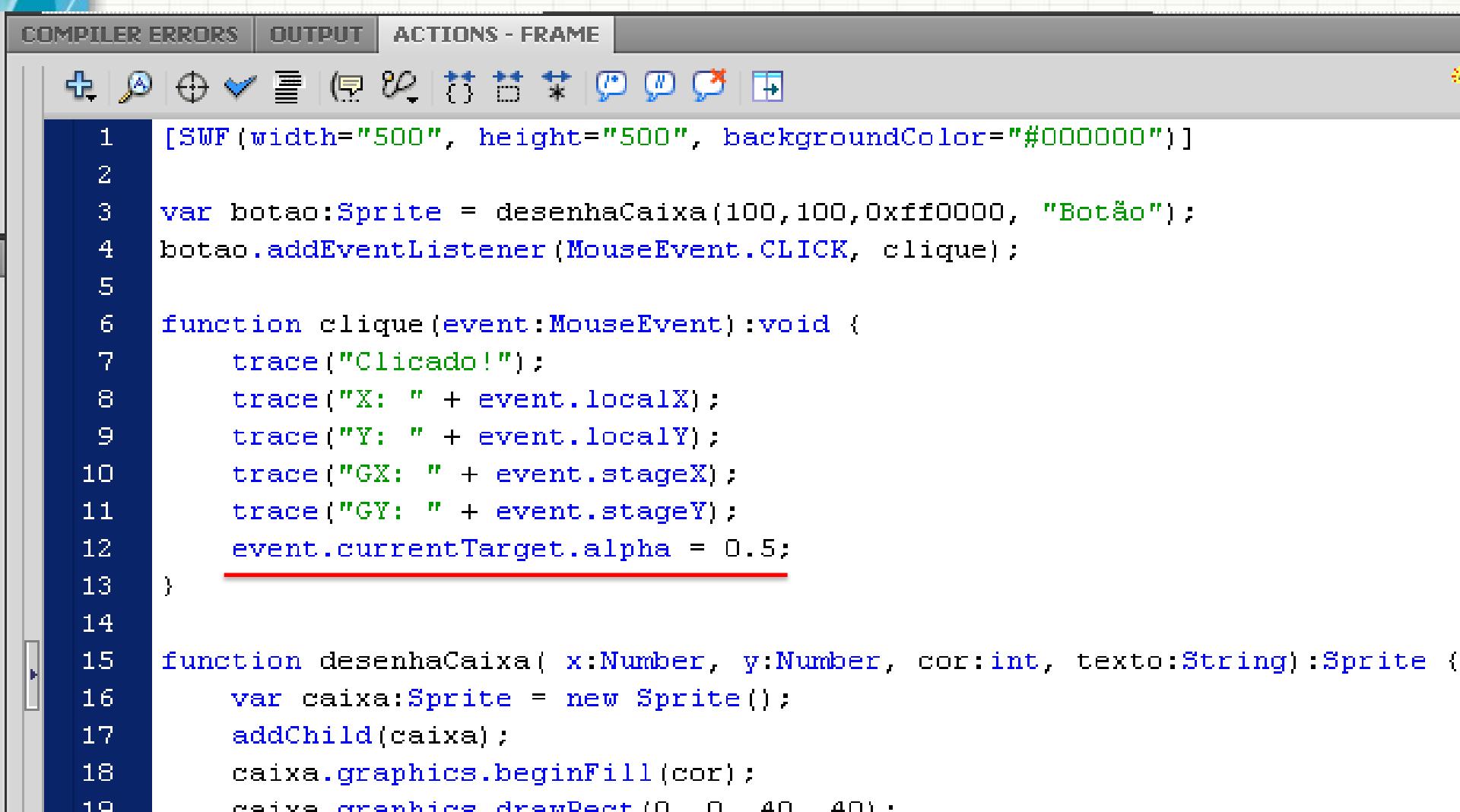
- Em que lugar foi o clique?

The screenshot shows a Flash IDE interface with the following tabs at the top: COMPILER ERRORS, OUTPUT, and ACTIONS - FRAME. Below the tabs is a toolbar with various icons. The main area contains ActionScript code. A red box highlights a section of code within a function named 'clique'.

```
1 [SWF(width="500", height="500", backgroundColor="#000000")]
2
3 var botao:Sprite = desenhaCaixa(100,100,0xff0000, "Botão");
4 botao.addEventListener(MouseEvent.CLICK, clique);
5
6 function clique(event:MouseEvent):void {
7     trace("Clicado!");
8     trace("X: " + event.localX);
9     trace("Y: " + event.localY);
10    trace("GX: " + event.stageX);
11    trace("GY: " + event.stageY);
12 }
13
14 function desenhaCaixa( x:Number, y:Number, cor:int, texto:String):Sprite {
15     var caixa:Sprite = new Sprite();
16     addChild(caixa);
17     caixa.graphics.beginFill(cor);
18     caixa.graphics.drawRect(x, y, 100, 40);
```

# Eventos: Clique (Mouse)

- Alterando transparência do Sprite...



The screenshot shows the Flash IDE interface with the Actions - Frame tab selected. The code editor displays the following ActionScript:

```
COMPILER ERRORS OUTPUT ACTIONS - FRAME

[SWF(width="500", height="500", backgroundColor="#000000")]

var botao:Sprite = desenhaCaixa(100,100,0xff0000, "Botão");
botao.addEventListener(MouseEvent.CLICK, clique);

function clique(event:MouseEvent):void {
    trace("Clicado!");
    trace("X: " + event.localX);
    trace("Y: " + event.localY);
    trace("GX: " + event.stageX);
    trace("GY: " + event.stageY);
    event.currentTarget.alpha = 0.5;
}

function desenhaCaixa( x:Number, y:Number, cor:int, texto:String ):Sprite {
    var caixa:Sprite = new Sprite();
    addChild(caixa);
    caixa.graphics.beginFill(cor);
    caixa.graphics.drawRect(0, 0, 40, 40);
}
```

The code defines a SWF file with width and height of 500 pixels and a black background color. It creates a button sprite at position (100, 100) with a red fill color and the text "Botão". The button's click event is listened to by the `clique` function. This function uses the `trace` command to output the coordinates of the click (localX, localY, stageX, stageY) and changes the button's alpha value to 0.5, making it semi-transparent.

# Eventos: Clique (Mouse)

- Aumentando o Sprite...

The screenshot shows a Flash IDE interface with tabs for 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. Below the tabs is a toolbar with various icons. The main area contains ActionScript code. A red box highlights the last four lines of the 'clique' function, which are responsible for increasing the size of the sprite when it is clicked.

```
1 [SWF(width="500", height="500", backgroundColor="#000000")]
2
3 var botao:Sprite = desenhaCaixa(100,100,0xff0000, "Botão");
4 botao.addEventListener(MouseEvent.CLICK, clique);
5
6 function clique(event:MouseEvent):void {
7     trace("Clicado!");
8     trace("X: " + event.localX);
9     trace("Y: " + event.localY);
10    trace("GX: " + event.stageX);
11    trace("GY: " + event.stageY);
12    event.currentTarget.alpha = 0.5;
13    event.currentTarget.width = event.currentTarget.width + 10;
14    event.currentTarget.height = event.currentTarget.height + 10;
15 }
16
17 function desenhaCaixa( x:Number, y:Number, cor:int, texto:String):Sprite {
18     var caixa:Sprite = new Sprite();
19     addChild(caixa);
20     caixa.graphics.beginFill(cor);
```

# Eventos: Clique (Mouse)

- Movendo o Sprite...

The screenshot shows a Flash IDE interface with three tabs at the top: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. Below the tabs is a toolbar with various icons. The main area contains ActionScript code. A red underline is placed under the line 'event.currentTarget.x = event.currentTarget.x + 50;'. The code is as follows:

```
1 [SWF(width="500", height="500", backgroundColor="#000000")]
2
3 var botao:Sprite = desenhaCaixa(100,100,0xff0000, "Botão");
4 botao.addEventListener(MouseEvent.CLICK, clique);
5
6 function clique(event:MouseEvent):void {
7     trace("Clicado!");
8     trace("X: " + event.localX);
9     trace("Y: " + event.localY);
10    trace("GX: " + event.stageX);
11    trace("GY: " + event.stageY);
12    event.currentTarget.alpha = 0.5;
13    event.currentTarget.width = event.currentTarget.width + 10;
14    event.currentTarget.height = event.currentTarget.height + 10;
15    event.currentTarget.x = event.currentTarget.x + 50;
16 }
17
18 function desenhaCaixa( x:Number, y:Number, cor:int, texto:String ):Sprite {
19     var caixa:Sprite = new Sprite();
20     addChild(caixa);
```



# ATIVIDADE

# Atividade

- Limpe o conteúdo da função **clique!**
- Altere o função para que girar o botão 10° a cada clique.
- O que acontece?

# Atividade - Solução

```
1 [SWF(width="500", height="500", backgroundColor="#000000")]
2
3 var botao:Sprite = desenhaCaixa(100,100,0xff0000, "Botão");
4 botao.addEventListener(MouseEvent.CLICK, clique);
5
6 function clique(event:MouseEvent):void {
7     event.target.rotation += 10;
8 }
9
10 function desenhaCaixa( x:Number, y:Number, cor:int, texto:String):Sprite {
11     var caixa:Sprite = new Sprite();
12     addChild(caixa);
13     caixa.graphics.beginFill(cor);
14     caixa.graphics.drawRect(0, 0, 40, 40);
15     caixa.graphics.endFill();
16     caixa.x = x;
17     caixa.y = y;
18     var msg:TextField = new TextField();
19     msg.htmlText = "<b>" + texto + "</b>";
20     msg.textColor = 0xffffffff;
21     msg.width = 40;
22     msg.autoSize = TextFieldAutoSize.CENTER;
23     msg.y = (40 - msg.textHeight)/2;
24     caixa.addChild(msg);
25
26     return caixa;
27 }
```

# Atividade – Sol. – Texto Rodando

```
function desenhaCaixa( x:Number, y:Number, cor:int, texto:String):Sprite {
    var caixa:Sprite = new Sprite();
    addChild(caixa);
    caixa.graphics.beginFill(cor);
    caixa.graphics.drawRect(0, 0, 40, 40);
    caixa.graphics.endFill();
    caixa.x = x;
    caixa.y = y;
    var msg:TextField = new TextField();
    msg.htmlText = "<b>" + texto + "</b>";
    msg.textColor = 0xffffffff;
    msg.width = 40;
    msg.autoSize = TextFieldAutoSize.CENTER;
    //msg.y = (40 - msg.textHeight)/2;
    //caixa.addChild(msg);

    var bmp:Bitmap = new Bitmap( new BitmapData(msg.width,msg.height,true,0x000000));
    bmp.bitmapData.draw(msg);
    bmp.y = (40 - msg.height)/2;
    caixa.addChild(bmp);

    return caixa;
}
```

# Atividade

- Altere o programa para mover o sprite para uma posição aleatória, dentro do cenário, quando for clicado.
- Dica: o tamanho máximo do cenário:  
**stage.stageWidth** e **stage.stageHeight**
- Dica: **Math.random()** sempre retorna um valor entre 0 e 1.
- Dica: **stage.stageWidth\*Math.random() = ???**

# Atividade - Solução

COMPILER ERRORS | OUTPUT | ACTIONS - FRAME

The screenshot shows a Flash IDE interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save), undo/redo, and various project management tools.
- Code Editor:** Displays ActionScript code. Lines 8 and 9 are highlighted with a red rectangle.
- Code Content:**

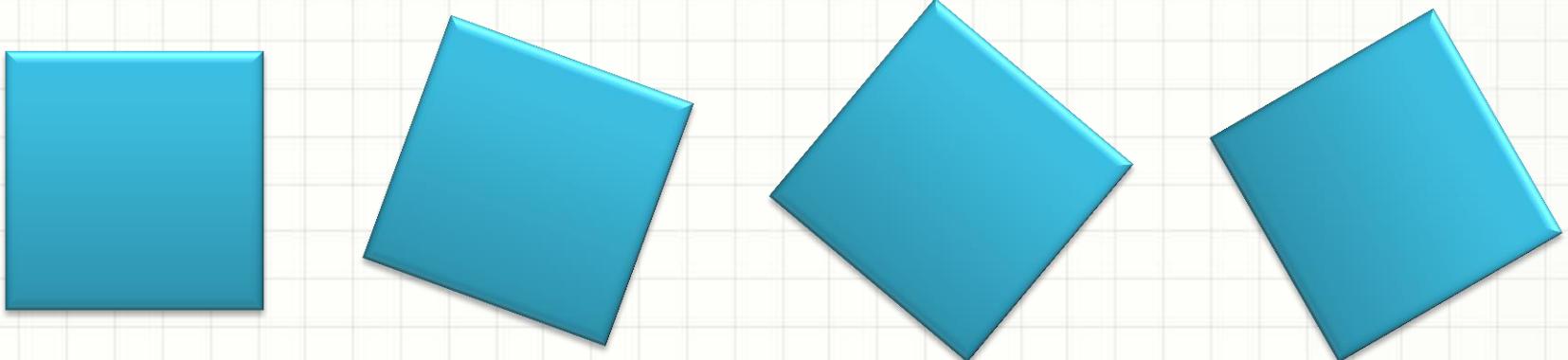
```
1 [SWF(width="500", height="500", backgroundColor="#000000")]
2
3 var botao:Sprite = desenhaCaixa(100,100,0xff0000, "Botão");
4 botao.addEventListener(MouseEvent.CLICK, clique);
5
6 function clique(event:MouseEvent):void {
7     event.target.rotation += 10;
8     event.target.x = stage.stageWidth * Math.random();
9     event.target.y = stage.stageHeight * Math.random();
10 }
11
12 function desenhaCaixa( x:Number, y:Number, cor:int, texto:String):Sprite {
13     var caixa:Sprite = new Sprite();
14     addChild(caixa);
15     caixa.graphics.beginFill(cor);
16     caixa.graphics.drawRect(0, 0, 40, 40);
```



**ANIMAÇÃO**

# Animação em ActionScript 3

- O que é uma animação?
- Uma animação é uma sucessão de imagens estáticas, chamadas **quadros** (ou *frames*)



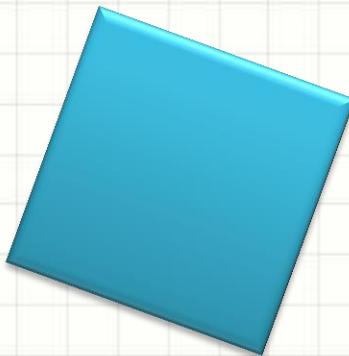
# Animação em ActionScript 3

- Exemplo....



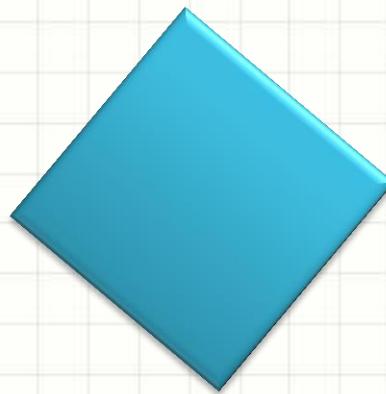
# Animação em ActionScript 3

- Exemplo....



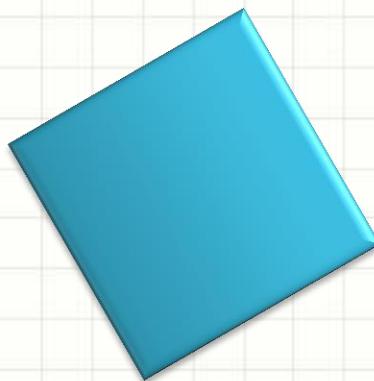
# Animação em ActionScript 3

- Exemplo....



# Animação em ActionScript 3

- Exemplo....



# Animação em ActionScript 3

- Quanto mais quadros de animação por segundo, mais *suave* é a animação
- Uma animação de 10 segundos precisa de quantos quadros?
- Depende da taxa de quadros por segundo...
  - *frame rate*
  - Medida em *fps* (frames por segundo)
  - *Quanto maior o fps, mais suave a animação*

# Animação em ActionScript 3

- Se a animação tiver **10 fps**, terá quantos frames em 10 segundos?

$$10\text{s} * 10\text{fps} = 100 \text{ frames}$$

- Se a animação tiver **24 fps**, terá quantos frames em 10 segundos?

$$10\text{s} * 24\text{fps} = 240 \text{ frames}$$

- O que acontece se configurar 24fps e usar apenas 100 frames?
  - Animação acaba em menos de 10 segundos!

# Animação em ActionScript 3

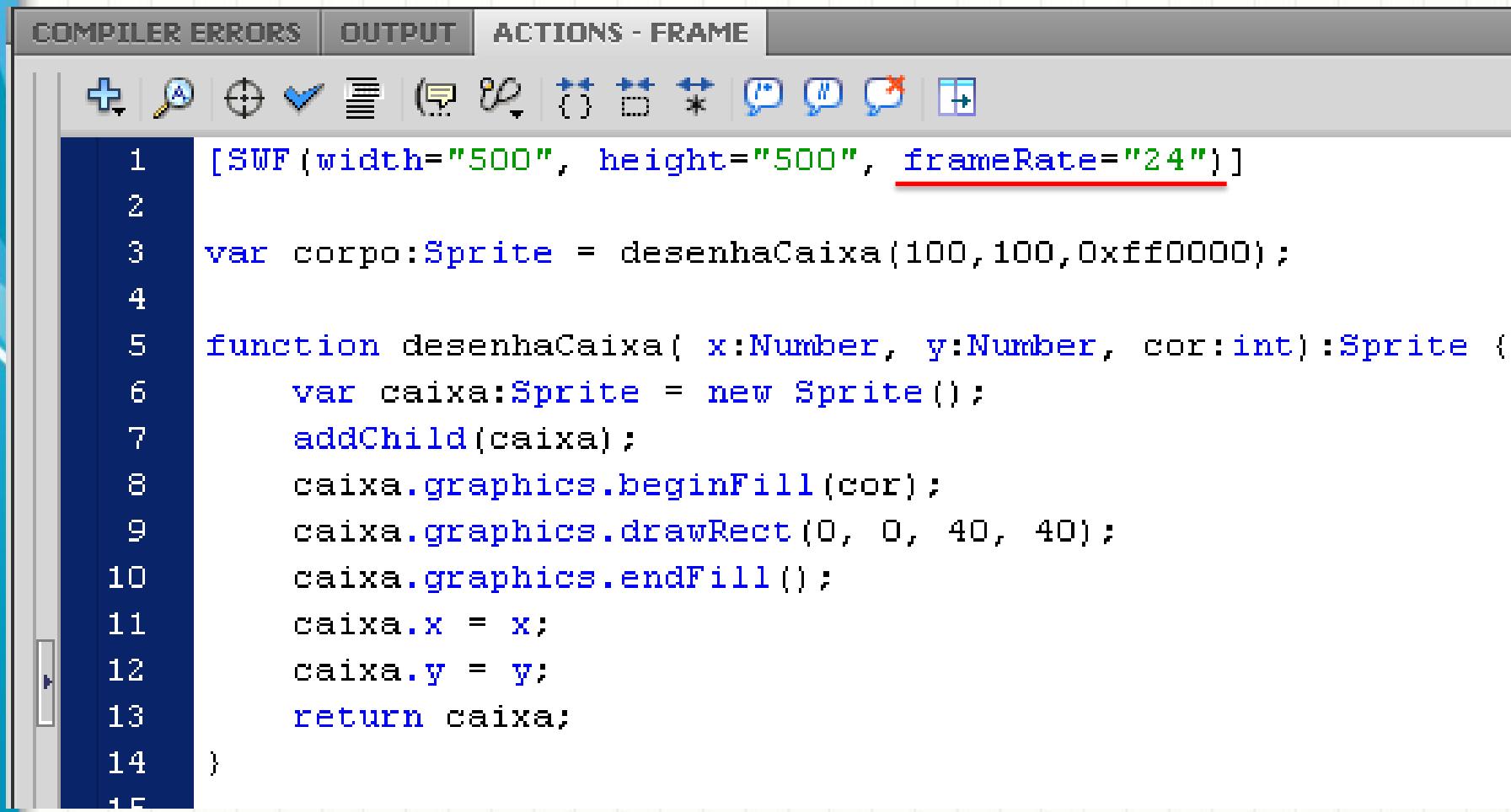
- Exemplo



**24 fps x 48 fps**

# Animação em ActionScript 3

- Como configurar a taxa de frames?



The screenshot shows the Flash IDE interface with the "ACTIONS - FRAME" tab selected. The code in the panel is as follows:

```
1 [SWF(width="500", height="500", frameRate="24")]
2
3 var corpo:Sprite = desenhaCaixa(100,100,0xff0000);
4
5 function desenhaCaixa( x:Number, y:Number, cor:int ):Sprite {
6     var caixa:Sprite = new Sprite();
7     addChild(caixa);
8     caixa.graphics.beginFill(cor);
9     caixa.graphics.drawRect(0, 0, 40, 40);
10    caixa.graphics.endFill();
11    caixa.x = x;
12    caixa.y = y;
13    return caixa;
14 }
15
```

The line `frameRate="24"` is highlighted with a red underline, indicating it is being edited or is the focus of the current discussion.

# Animação em ActionScript 3

- Como

## Como Animar o Corpo?

The screenshot shows the Adobe Flash CS3 IDE interface. The title bar says "COMPILER ERROR". The toolbar has various icons for file operations, selection, and search. The code editor window contains the following ActionScript:

```
1 [SWF(width="500", height="500", frameRate="24")]
2
3 var corpo:Sprite = desenhaCaixa(100,100,0xff0000);
4
5 function desenhaCaixa( x:Number, y:Number, cor:int ):Sprite {
6     var caixa:Sprite = new Sprite();
7     addChild(caixa);
8     caixa.graphics.beginFill(cor);
9     caixa.graphics.drawRect(0, 0, 40, 40);
10    caixa.graphics.endFill();
11    caixa.x = x;
12    caixa.y = y;
13    return caixa;
14 }
15 }
```

The line `frameRate="24"` is underlined in red, indicating a compiler error.



# **EVENTO DE ANIMAÇÃO EM ACTIONSCRIPT 3**

# Evento de Animação ActionScript 3

- Momento de um novo quadro?

- Evento: ENTER\_FRAME

```
1 [SWF(width="500", height="500", frameRate="24")]
2
3 var corpo:Sprite = desenhaCaixa(100,100,0xff0000);
4 addEventListener(Event.ENTER_FRAME, novoQuadro);
5
6 function novoQuadro( evento:Event ):void {
7     trace("Novo quadro!");
8 }
9
10 function desenhaCaixa( x:Number, y:Number, cor:int ):Sprite {
11     var caixa:Sprite = new Sprite();
12     addChild(caixa);
13     caixa.graphics.beginFill(cor);
14     caixa.graphics.drawRect(0, 0, 40, 40);
15     caixa.graphics.endFill();
16     caixa.x = x;
17     caixa.y = y;
18     return caixa;
19 }
```

# Evento de Animação ActionScript 3

- Vamos mover o corpo a cada quadro...

```
1 [SWF(width="500", height="500", frameRate="24")]
2
3 var corpo:Sprite = desenhaCaixa(100,100,0xff0000);
4 addEventListener(Event.ENTER_FRAME, novoQuadro);
5
6 function novoQuadro( evento:Event ):void {
7     corpo.x += 2;
8 }
9
10 function desenhaCaixa( x:Number, y:Number, cor:int ):Sprite {
11     var caixa:Sprite = new Sprite();
12     addChild(caixa);
13     caixa.graphics.beginFill(cor);
14     caixa.graphics.drawRect(0, 0, 40, 40);
15     caixa.graphics.endFill();
16     caixa.x = x;
17     caixa.y = y;
18     return caixa;
19 }
```



# ATIVIDADE

# Atividade

- Modifique a animação desenvolvida para que o corpo comece a girar 5 graus por quadro. Ele deve continuar se deslocando lateralmente, como estava antes.

# Atividade - Solução

```
1 [SWF(width="500", height="500", frameRate="24")]
2
3 var corpo:Sprite = desenhaCaixa(100,100,0xff0000);
4 addEventListener(Event.ENTER_FRAME, novoQuadro);
5
6 function novoQuadro( evento:Event ):void {
7     corpo.x += 2;
8     corpo.rotation += 5;
9 }
10
11 function desenhaCaixa( x:Number, y:Number, cor:int ):Sprite {
12     var caixa:Sprite = new Sprite();
13     addChild(caixa);
14     caixa.graphics.beginFill(cor);
15     caixa.graphics.drawRect(0, 0, 40, 40);
16     caixa.graphics.endFill();
17     caixa.x = x;
18     caixa.y = y;
19     return caixa;
20 }
```

# Atividade

- Corrija o programa para que o quadrado gire em torno de seu centro!

# Atividade - Solução

```
1 [SWF(width="500", height="500", frameRate="24")]
2
3 var corpo:Sprite = desenhaCaixa(100,100,0xff0000);
4 addEventListener(Event.ENTER_FRAME, novoQuadro);
5
6 function novoQuadro( evento:Event ):void {
7     corpo.x += 2;
8     corpo.rotation += 5;
9 }
10
11 function desenhaCaixa( x:Number, y:Number, cor:int ):Sprite {
12     var caixa:Sprite = new Sprite();
13     addChild(caixa);
14     caixa.graphics.beginFill(cor);
15     caixa.graphics.drawRect(-20, -20, 40, 40);
16     caixa.graphics.endFill();
17     caixa.x = x;
18     caixa.y = y;
19     return caixa;
20 }
```

# Atividade

- Modifique o programa para que o corpo só comece a rodar quando ele for clicado

# Atividade - Solução

```
1  [SWF(width="500", height="500", frameRate="24")]
2  var rotacao:int = 0;
3
4  var corpo:Sprite = desenhaCaixa(100,100,0xff0000);
5  addEventListener(Event.ENTER_FRAME, novoQuadro);
6  corpo.addEventListener(MouseEvent.CLICK, mudaRotacao);
7
8  function mudaRotacao( evento:MouseEvent ):void {
9      rotacao = 5;
10 }
11
12 function novoQuadro( evento:Event ):void {
13     corpo.x += 2;
14     corpo.rotation += rotacao;
15 }
16
17 function desenhaCaixa( x:Number, y:Number, cor:int ):Sprite {
18     var caixa:Sprite = new Sprite();
19     addChild(caixa);
20     caixa.graphics.beginFill(cor);
```

# Atividade

- Modifique o programa para que o corpo só comece a rodar quando ele for clicado e...
- A cada clique subsequente, inverta o giro

# Atividade - Solução

```
1 [SWF(width="500", height="500", frameRate="24")]
2 var rotacao:int = 0;
3
4 var corpo:Sprite = desenhaCaixa(100,100,0xff0000);
5 addEventListener(Event.ENTER_FRAME, novoQuadro);
6 corpo.addEventListener(MouseEvent.CLICK, mudaRotacao);
7
8 function mudaRotacao( evento:MouseEvent ):void {
9     if (rotacao == 0) rotacao = 5;
10    else rotacao *= -1;
11 }
12
13 function novoQuadro( evento:Event ):void {
14     corpo.x += 2;
15     corpo.rotation += rotacao;
16 }
17
18 function desenhaCaixa( x:Number, y:Number, cor:int ):Sprite {
19     var caixa:Sprite = new Sprite();
20     addChild(caixa);
```

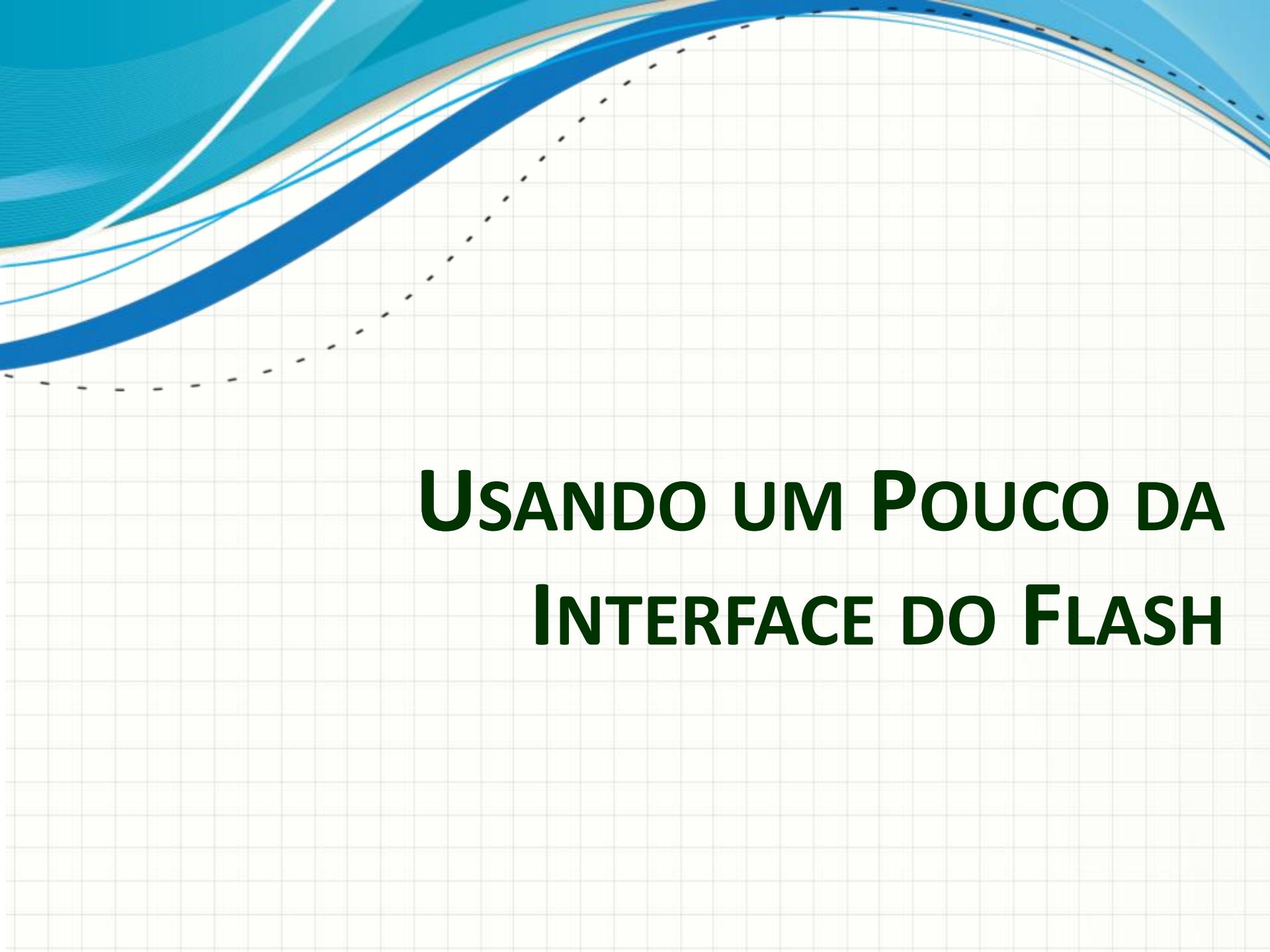


# **DISCUSSÃO: VÍDEO x JOGOS**

# Vídeo x Jogos

- Vídeo e jogos são iguais?
- 24 fps é suficiente para um jogo?
- Jogo envolve simulação física
  - Processamento quadro a quadro.
  - Se a bala se move 10m/s...
  - Bala pula personagem!
- Em simulações...
  - 24 fps não são suficientes!
  - Para jogos também não!





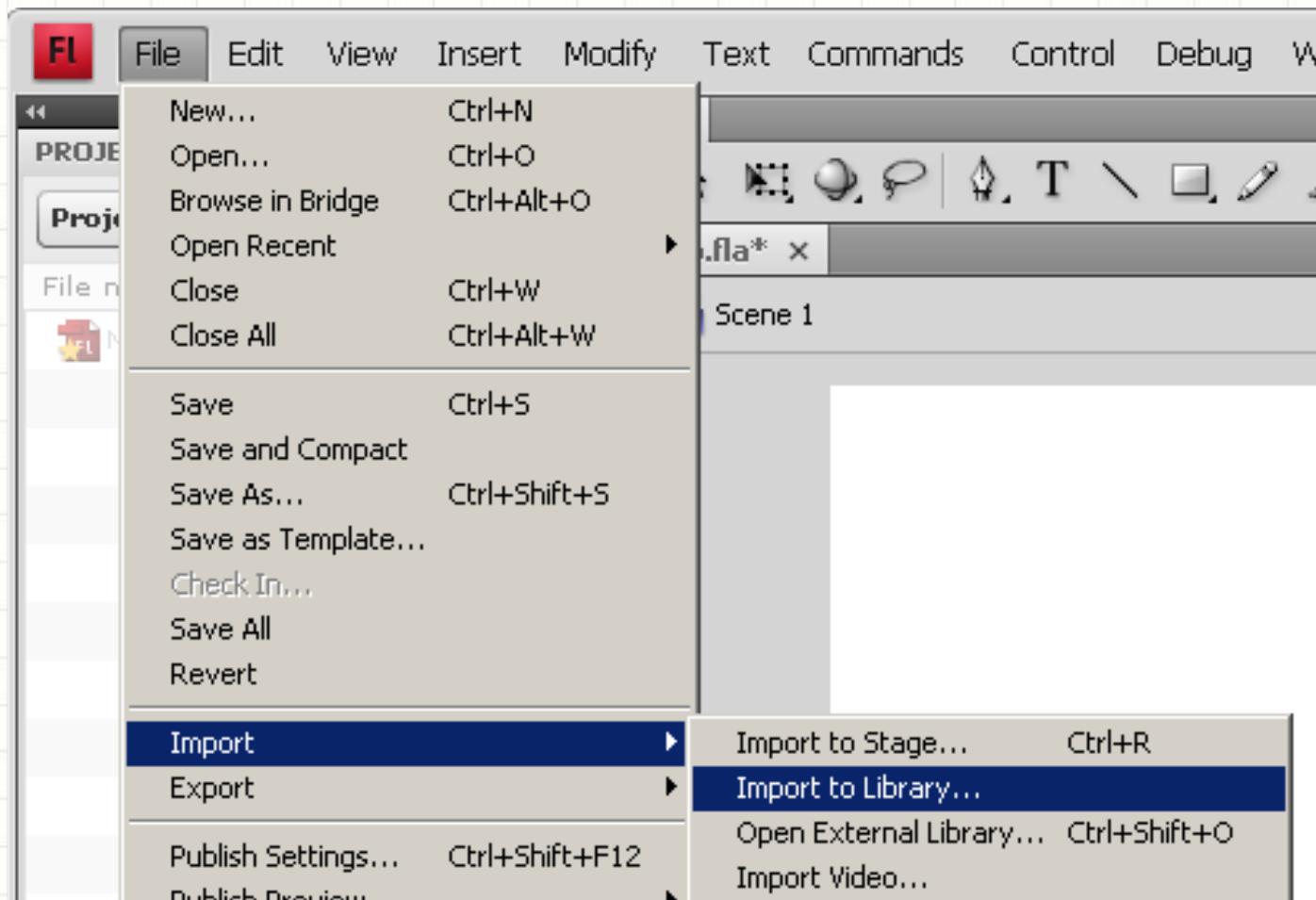
# **USANDO UM POUCO DA INTERFACE DO FLASH**

# Usando um Pouco o Flash

- É claro que podemos colocar um fundo criado como um sprite...
- Mas que tal usarmos algumas vantagens do Flash CS4?

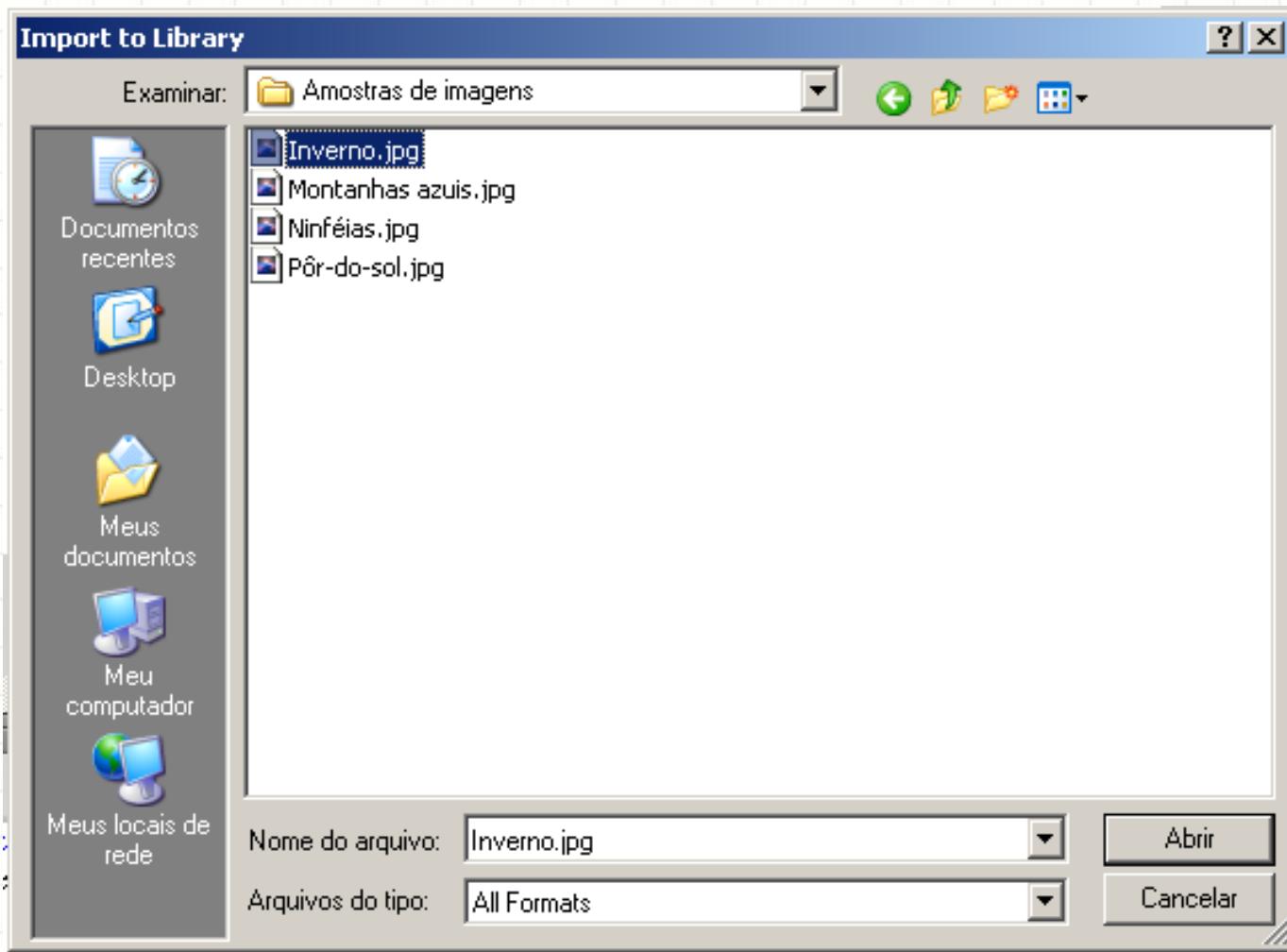
# Usando um Pouco o Flash

- Para adicionar uma imagem de fundo, clique em **File > Import > Import to Library...**



# Usando um Pouco o Flash

- Escolha uma imagem...



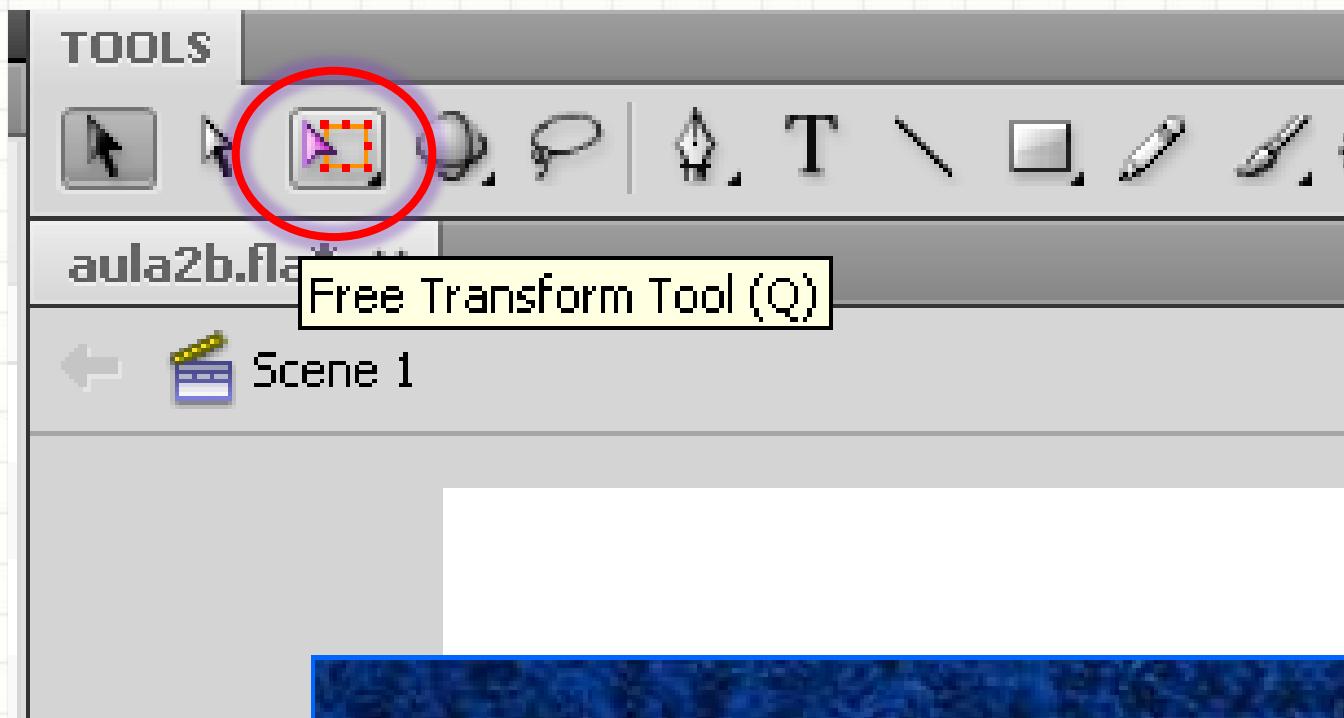
# Usando um Pouco o Flash

- Observe que ela aparecerá na Library
  - Olhe no canto inferior esquerdo!



# Usando um Pouco o Flash

- Arraste a imagem para o “stage” e use a ferramenta FreeTransform para redimensioná-la para o tamanho correto.



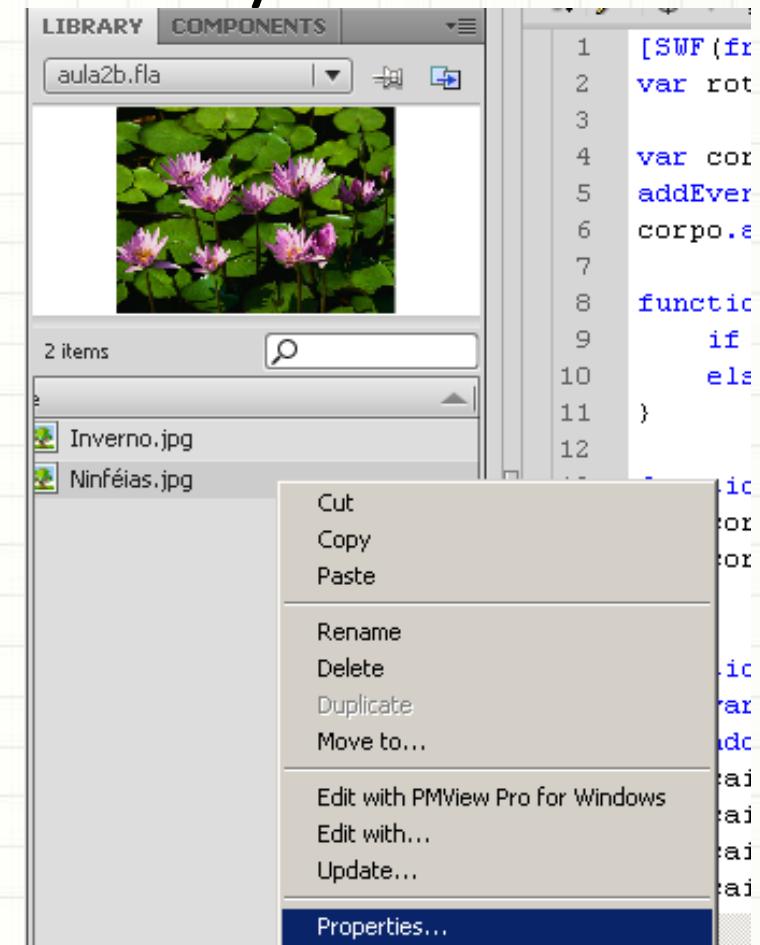
# Usando um Pouco o Flash

- Execute... O que houve?
- Ajuste para não redimensionar o “stage”

```
1 [SWF(frameRate="24")]
2 var rotacao:int = 0;
3
4 var corpo:Sprite = desenhaCaixa(100,100,0xff0000);
5 addEventListener(Event.ENTER_FRAME, novoQuadro);
6 corpo.addEventListener(MouseEvent.CLICK, mudaRotacao);
7
8 function mudaRotacao( evento:MouseEvent ):void {
9     if (rotacao == 0) rotacao = 5;
10    else rotacao *= -1;
11 }
12
13 function novoQuadro( evento:Event ):void {
14     corpo.x += 2;
15     corpo.rotation += rotacao;
```

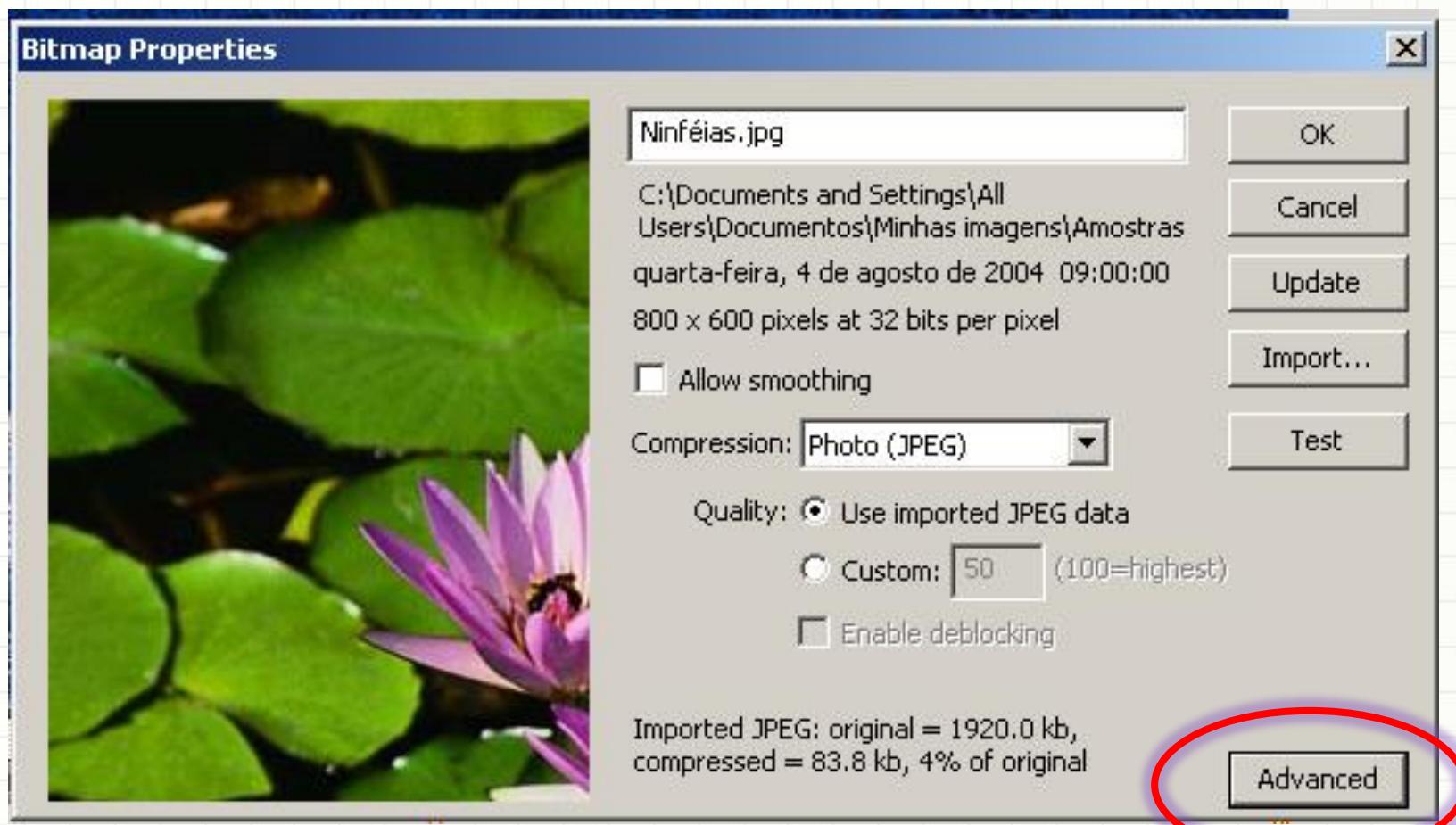
# Usando um Pouco o Flash

- Que tal usar uma imagem como sprite?
- Importe a imagem para a Library...
- E abra as propriedades...



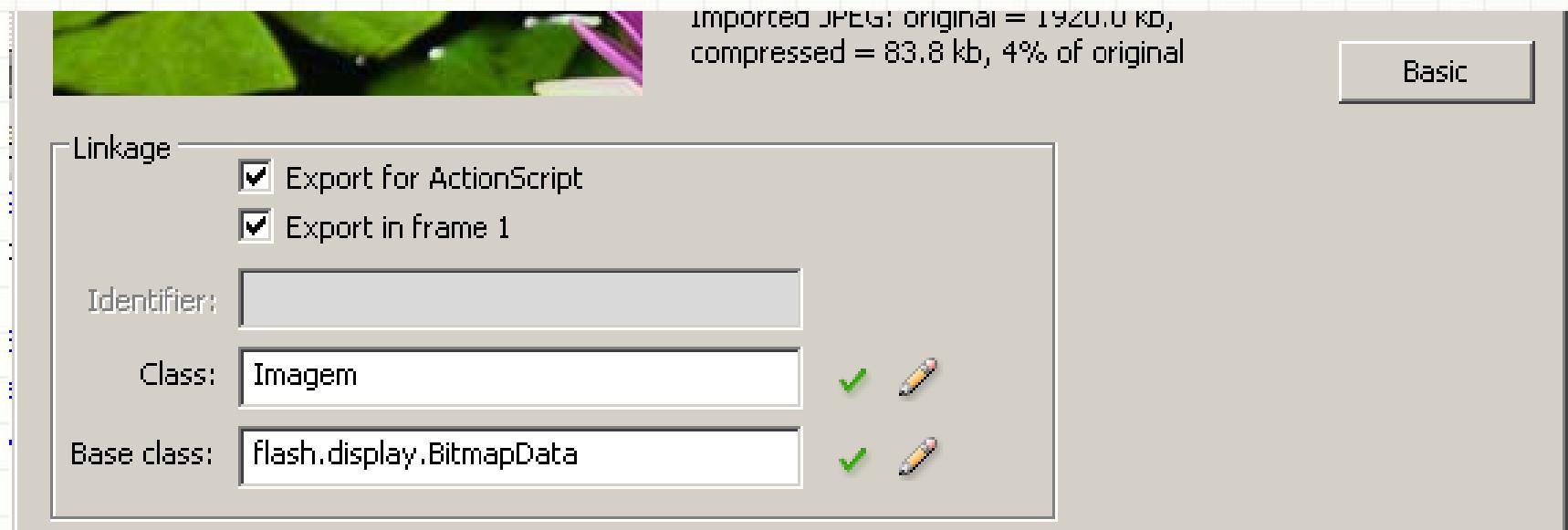
# Usando um Pouco o Flash

- Clique no botão “Advanced”



# Usando um Pouco o Flash

- Ajuste para exportar para o ActionScript...
- E indique o nome da classe: **Imagen**



# Usando um Pouco o Flash

- Vamos usar essa nova classe no código!

```
18 function desenhaCaixa( x:Number, y:Number, cor:int ):Sprite {
19     var caixa:Sprite = new Sprite();
20     addChild(caixa);
21     //caixa.graphics.beginFill(cor);
22     //caixa.graphics.drawRect(-20, -20, 40, 40);
23     //caixa.graphics.endFill();
24     var bmp:Bitmap = new Bitmap(new Imagem(0,0));
25     bmp.x = -20;
26     bmp.y = -20;
27     bmp.width = 40;
28     bmp.height = 40;
29     caixa.addChild(bmp);
30
31     caixa.x = x;
32     caixa.y = y;
33     return caixa;
34 }
```



# ATIVIDADE

# Atividade

- Modifique o programa para que:
  - Transparência do sprite varie de 1.0 a 0.0
  - De 0.05 em 0.05...
- Extras:
  - Quando chegar em 0.0, aumente até 1.0...
  - Quando chegar em 1.0, diminua até 0.0...

# Atividade - Solução

```
1 [SWF(frameRate="24")]
2 var rotacao:int = 0;
3 var transp = 1.0;
4 var dtransp = 0.05;
5
6 var corpo:Sprite = desenhaCaixa(100,100,0xff0000);
7 addEventListener(Event.ENTER_FRAME, novoQuadro);
8 corpo.addEventListener(MouseEvent.CLICK, mudaRotacao);
9
10 function mudaRotacao( evento:MouseEvent ):void {
11     if (rotacao == 0) rotacao = 5;
12     else rotacao *= -1;
13 }
14
15 function novoQuadro( evento:Event ):void {
16     corpo.x += 2;
17     corpo.rotation += rotacao;
18     if (transp >= 1.0 || transp <= 0.0) dtransp *= -1;
19     transp = transp + dtransp;
20     corpo.alpha = transp;
21 }
22 }
```

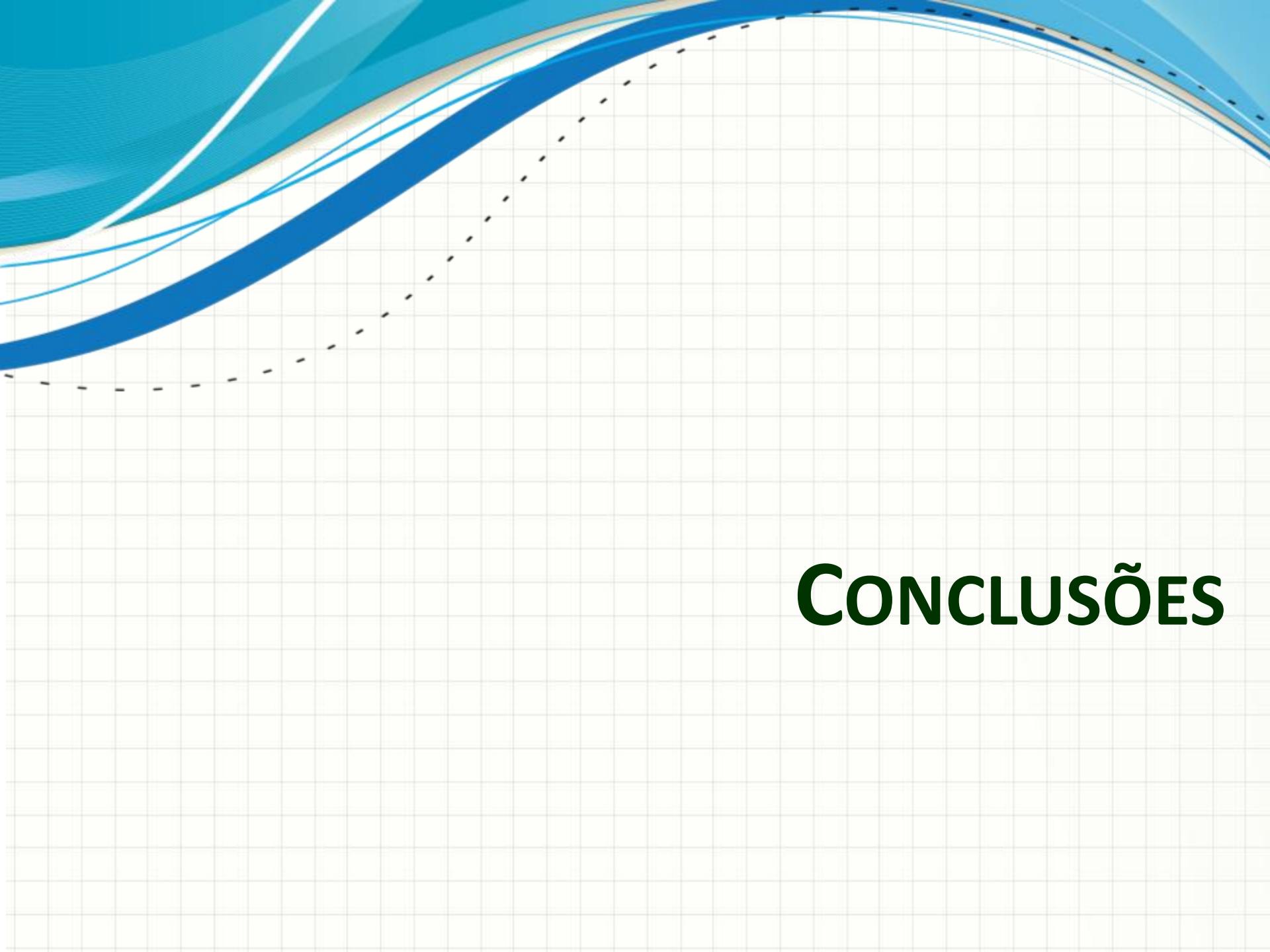


# **EXTRA: EVENTOS DE TECLADO**

# Eventos de Teclado em AS3

- Experimente isso...

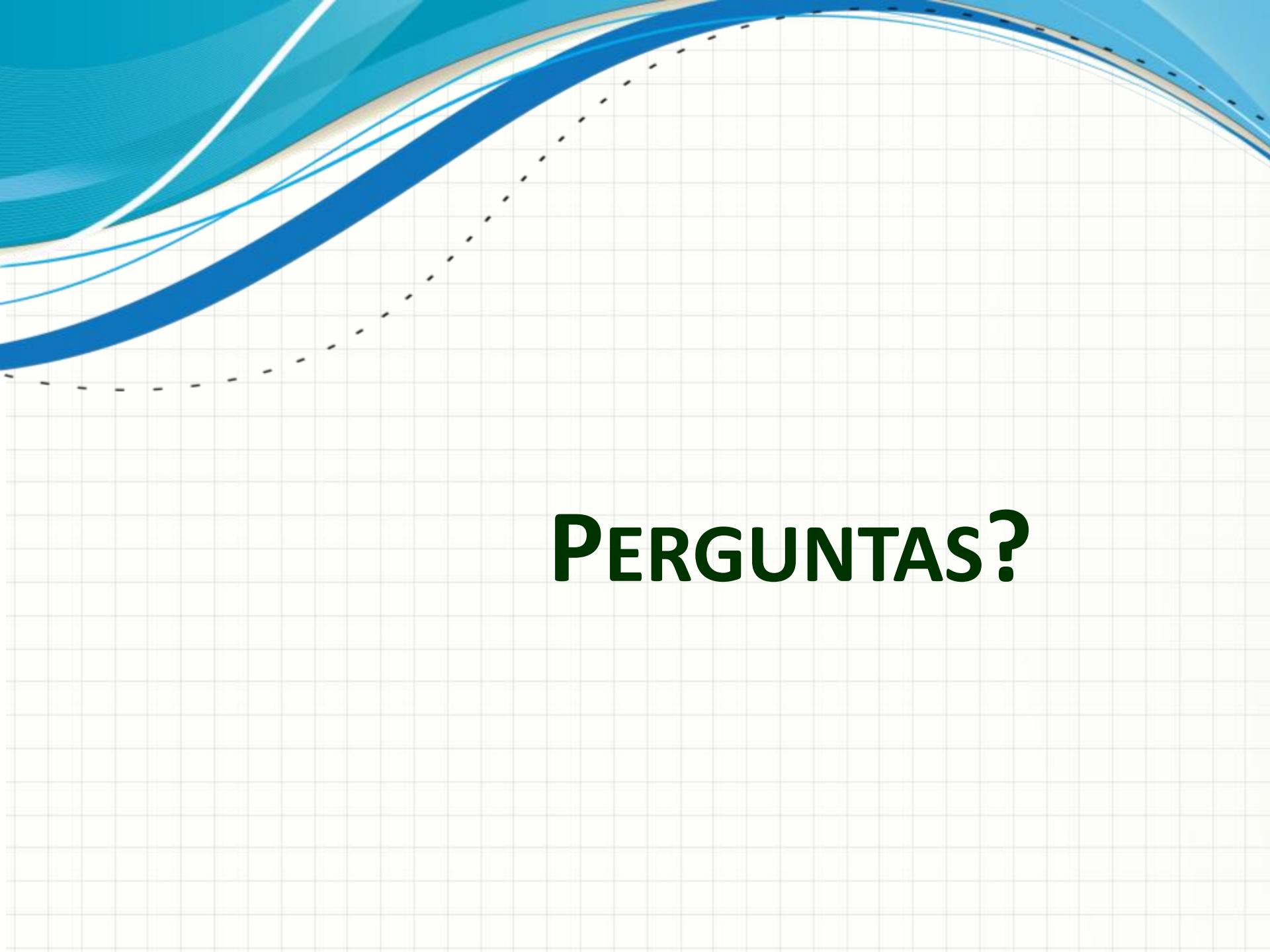
```
10 stage.addEventListener (KeyboardEvent.KEY_DOWN, apertouTecla);
11 stage.addEventListener (KeyboardEvent.KEY_UP, soltouTecla);
12
13 function apertouTecla(evento:KeyboardEvent):void {
14     trace("Key code: " + evento.keyCode);
15     trace("Ctrl status: " + evento.ctrlKey);
16     trace("Key location: " + evento.keyLocation);
17     trace("Shift key: " + evento.shiftKey);
18 }
19
20 function soltouTecla(evento:KeyboardEvent):void {
21     trace("Key code: " + evento.keyCode);
22 }
23
```



# CONCLUSÕES

# Resumo

- ActionScript 3: ótimo para animações
- Controle de interação e tempo: **eventos**
- Criando funções de maneira inteligente...
  - Simples fazer um aplicativo/jogo completo
- ActionScript 3: pode ser integrado ao Flash
  - Elementos criados pela interface do Flash
- **TAREFA**
  - Trabalho B!

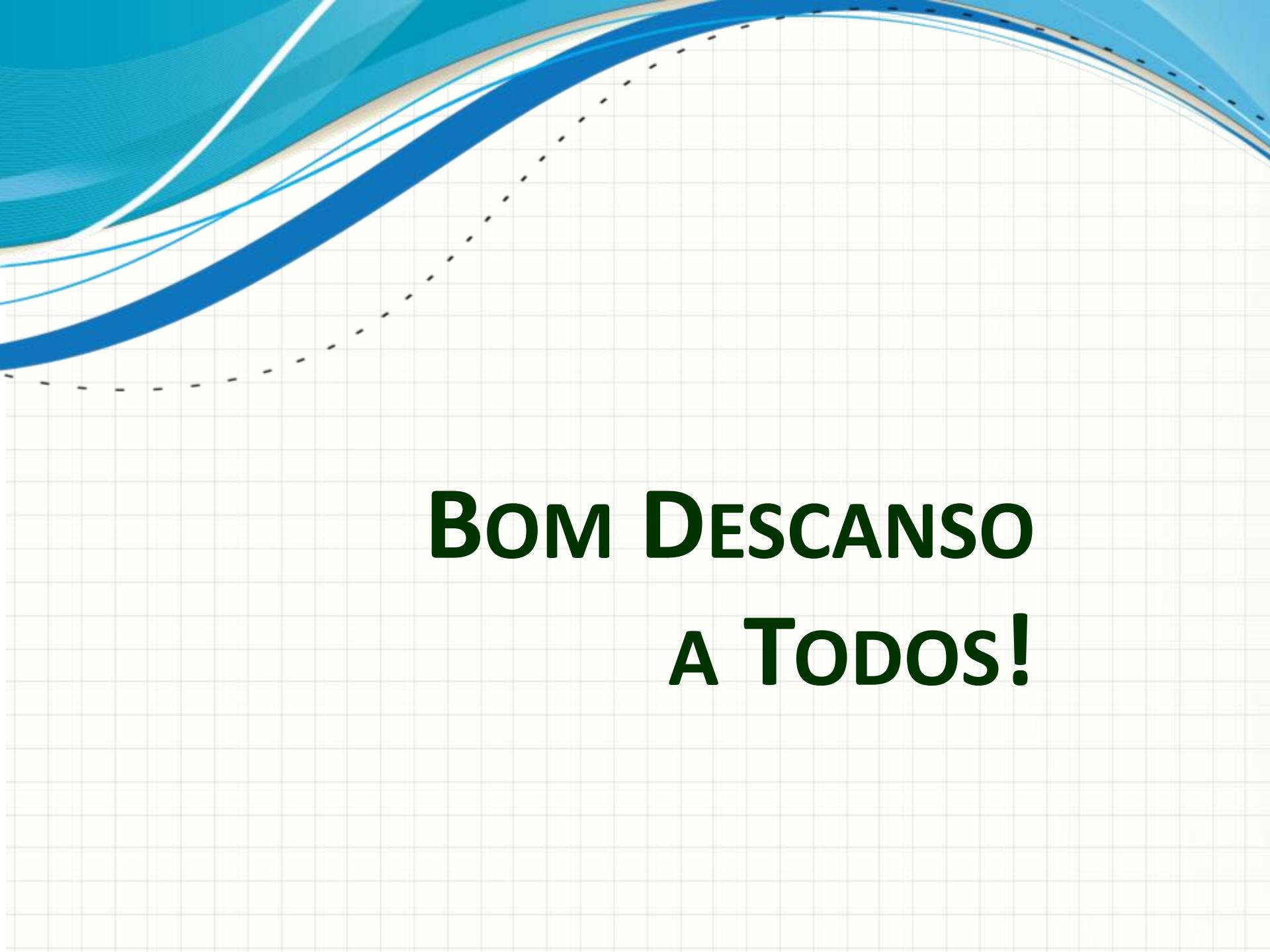


**PERGUNTAS?**

# Próxima Aula



- AS3 é bom para animações...
  - Mas e se tiver de fazer um aplicativo/página mais tradicional?
  - Que tal um pouco de ASP.Net?



**BOM DESCANSO  
A TODOS!**