



# **PROGRAMAÇÃO PARA INTERNET RICA**

## **INTRODUÇÃO AO ACTIONSCRIPT 3**

Prof. Dr. Daniel Caetano

2012 - 2

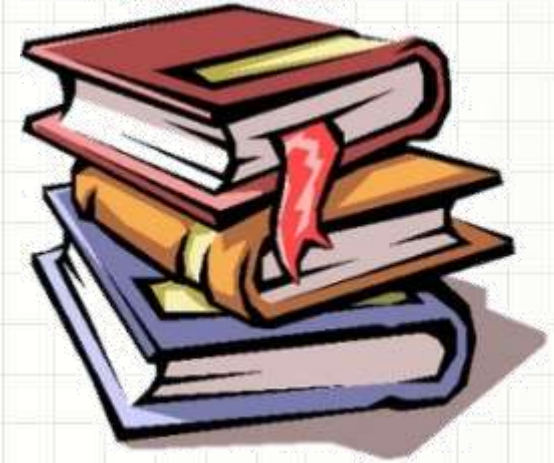
# Objetivos

- Apresentar os conceitos básicos da linguagem ActionScript 3.0
- Capacitar o aluno para criar programas simples usando ActionScript 3 e o Adobe Flash CS4

- **Trabalho B!**



# Material de Estudo



---

## Material

## Acesso ao Material

Tutorial Extra

<http://www.caetano.eng.br/>

Adobe Flash CS4 (Material de Apoio)

Apresentação

<http://www.caetano.eng.br/>

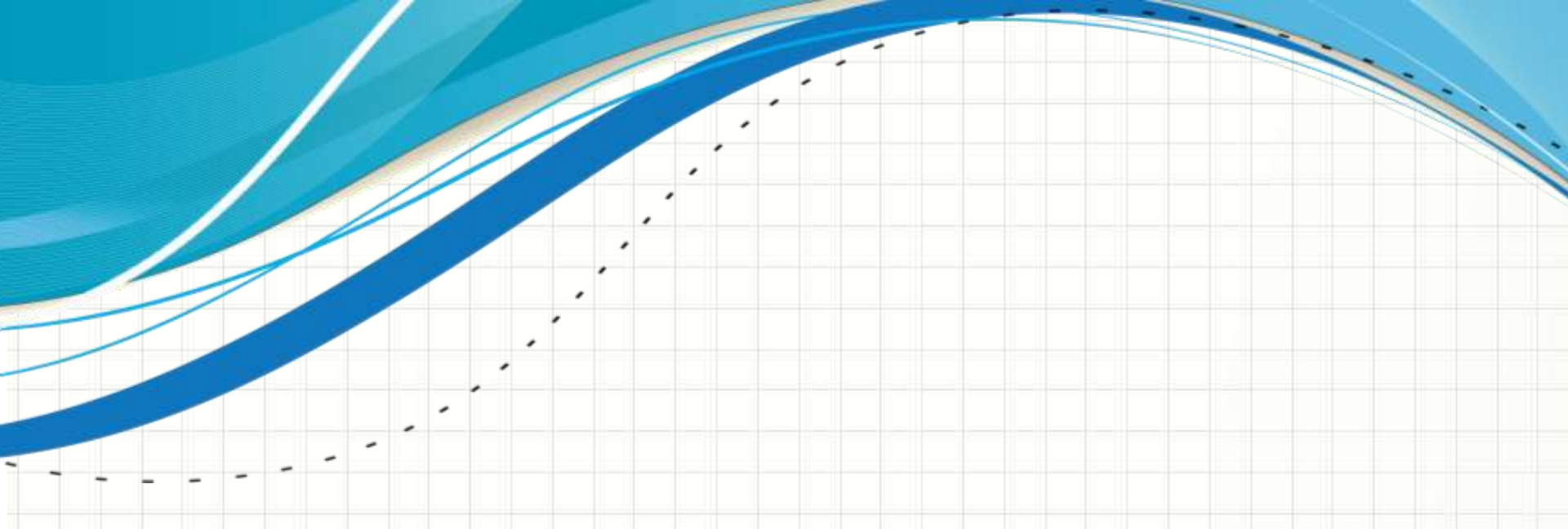
(Aula 12)

Google

ActionScript 3

AS3

---



# O QUE É ACTIONSCRIPT 3?

# Introdução

- ActionScript: evolução parecida com HTML
- Macromedia/Adobe
  - Flash Shockwave
  - Animações...
- Necessidade de interação
  - Formulários
  - Botões...
- Action Script



# Introdução

- ActionScript 1
  - Extremamente limitado
- ActionScript 2
  - Controle de animações
  - Recursos básicos de uma linguagem OO
- ActionScript 3
  - Linguagem OO completa
  - Similar ao JavaScript em muitos aspectos
  - Facilita a criação de animações!



# **ACTIONSCRIPT 3 NA PRÁTICA**

# ActionScript 3 na Prática

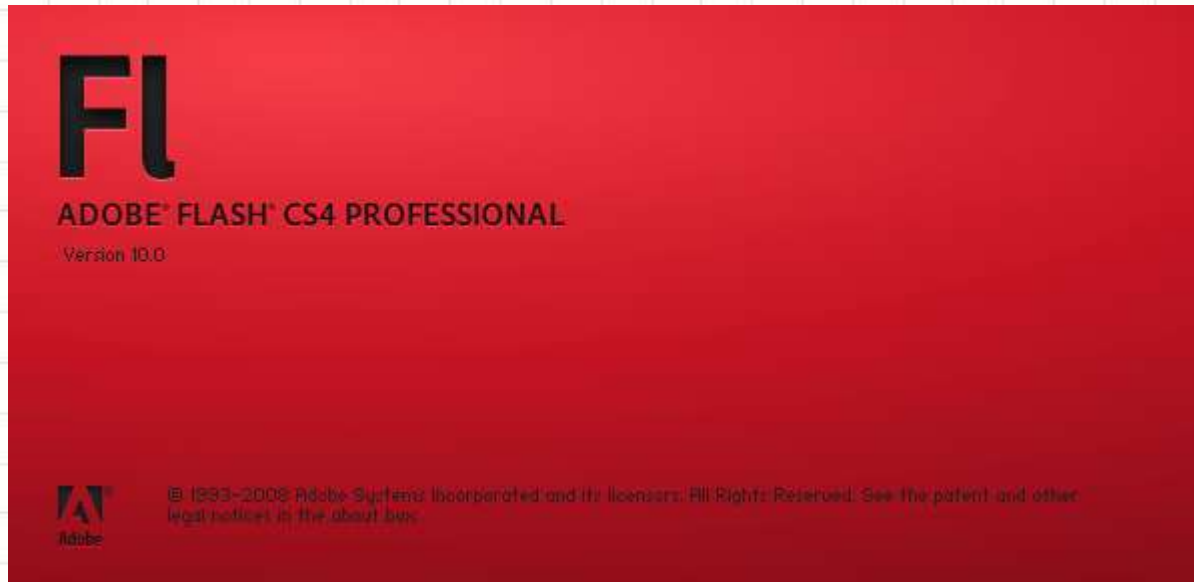
- AS3 como uma linguagem “independente”:
  - FlashDevelop
  - <http://www.flashdevelop.org/>
- Para usar AS3, programaremos no Flash
- Inicie o Flash CS4:





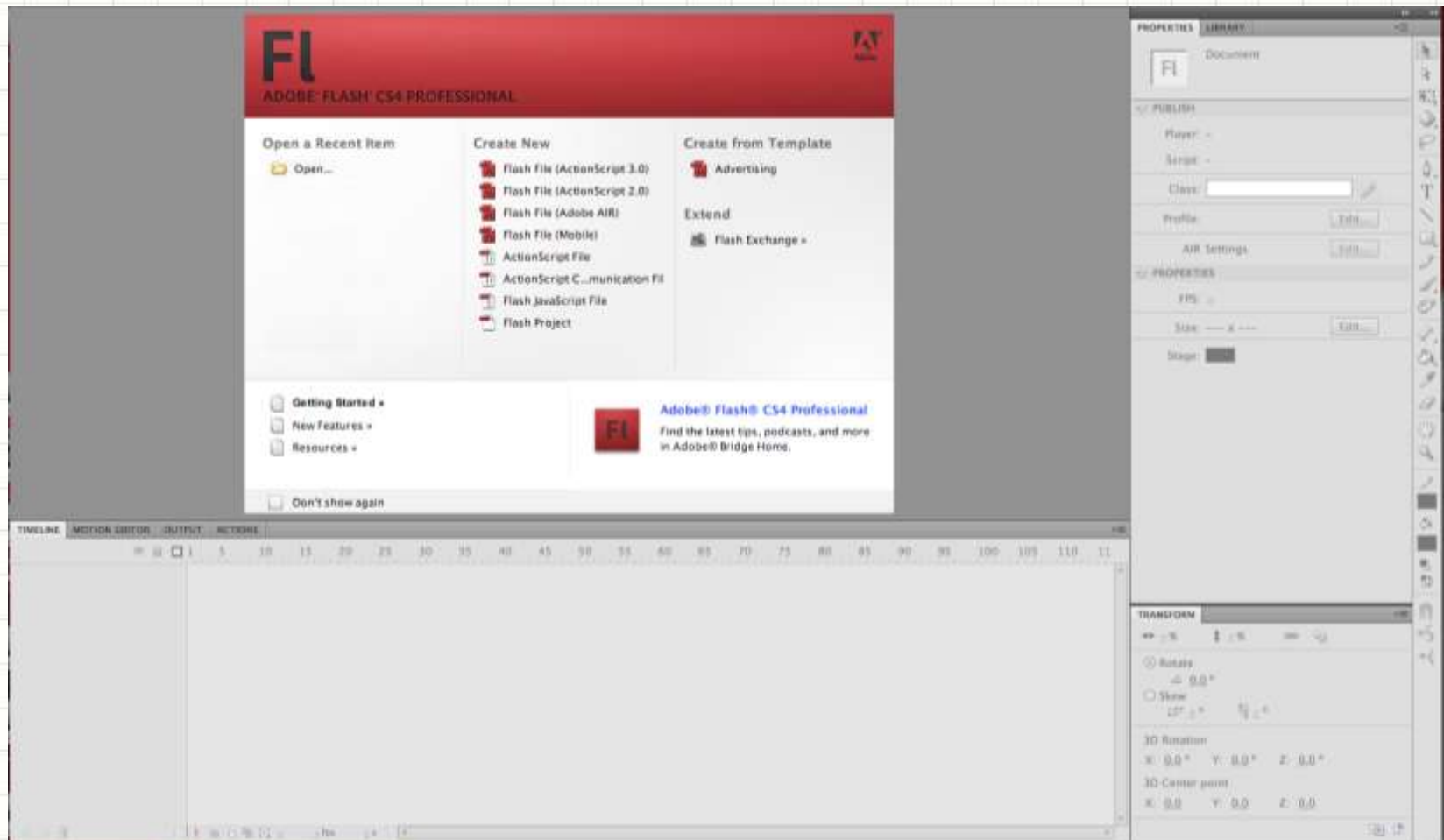
# ActionScript 3 na Prática

- Aguarde o carregamento...



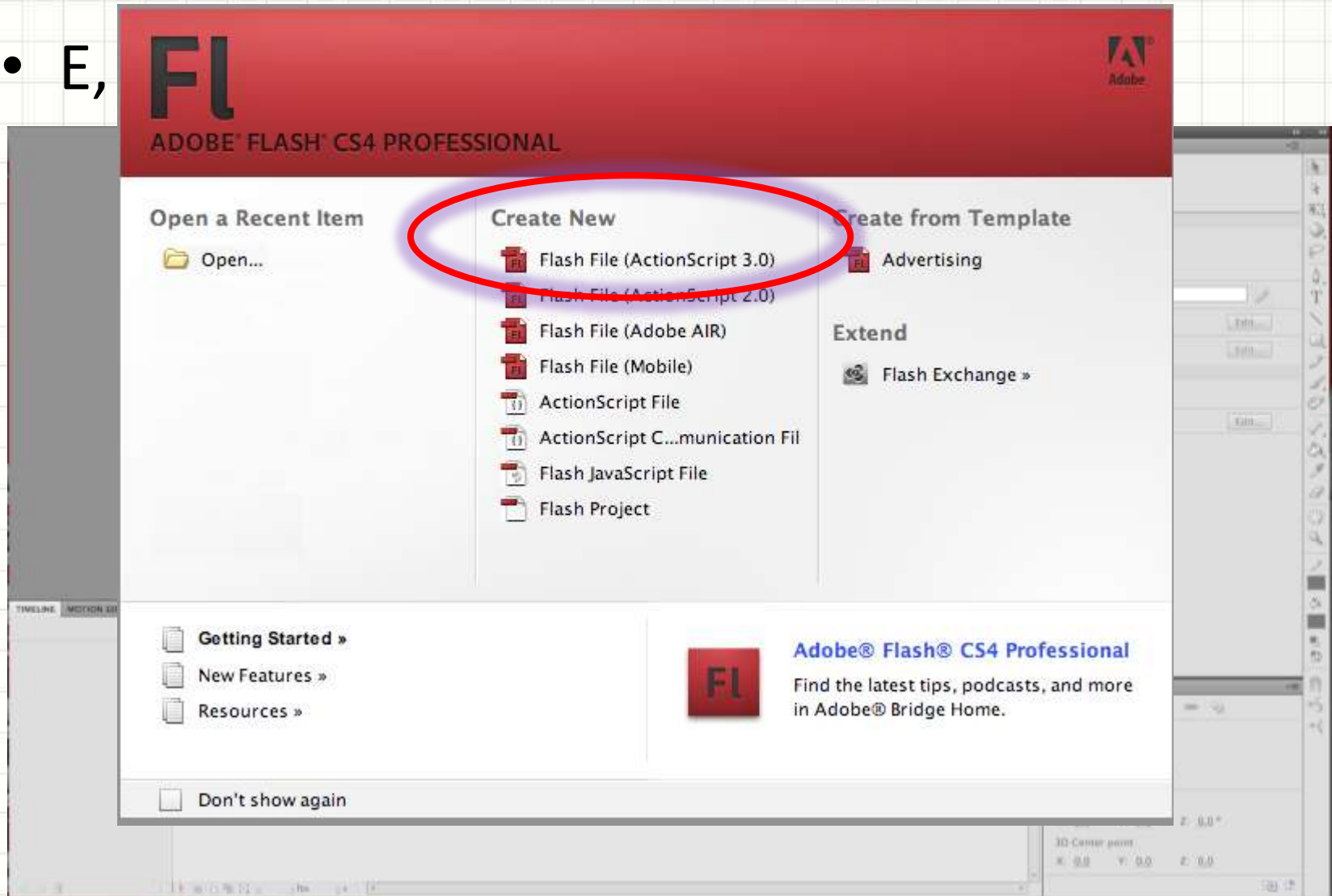
# ActionScript 3 na Prática

- E, finalmente, uma janela aparecerá...



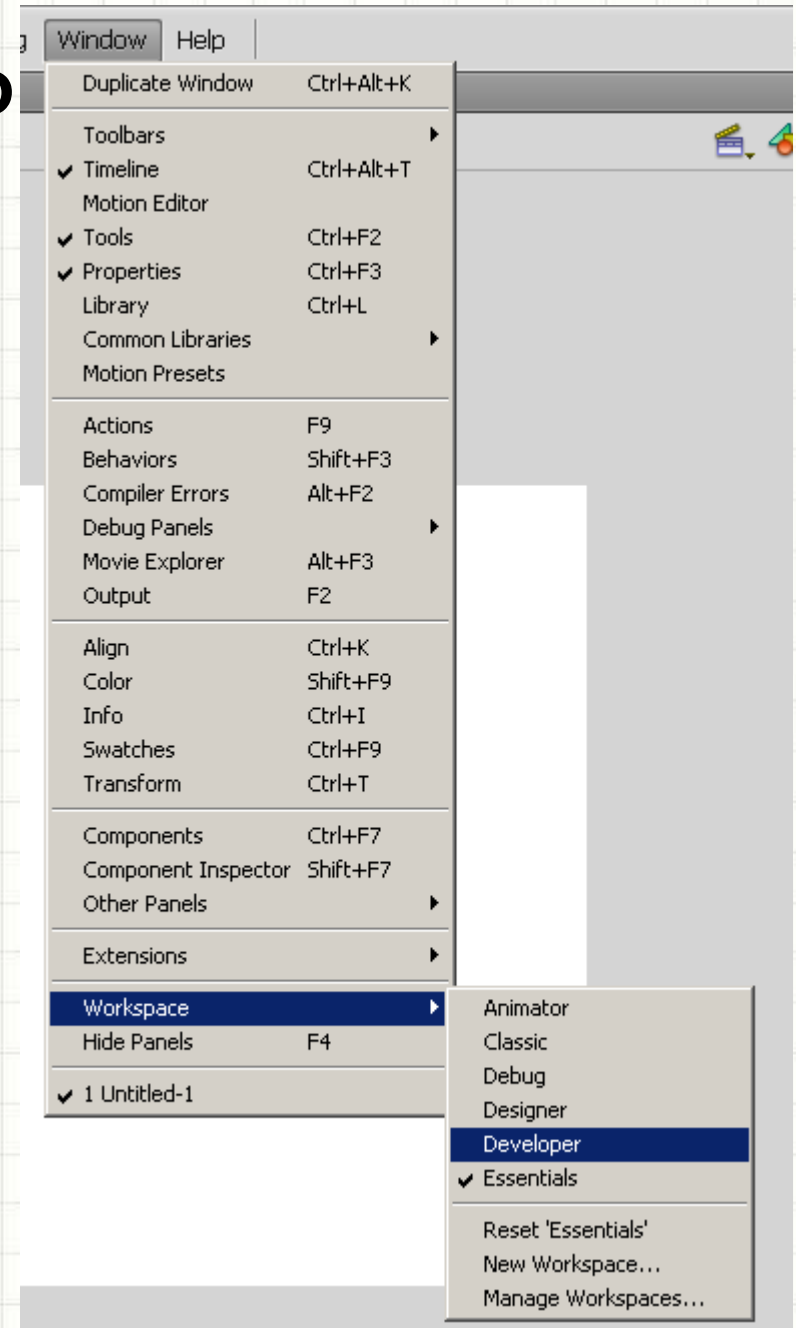
# ActionScript 3 na Prática

- E,



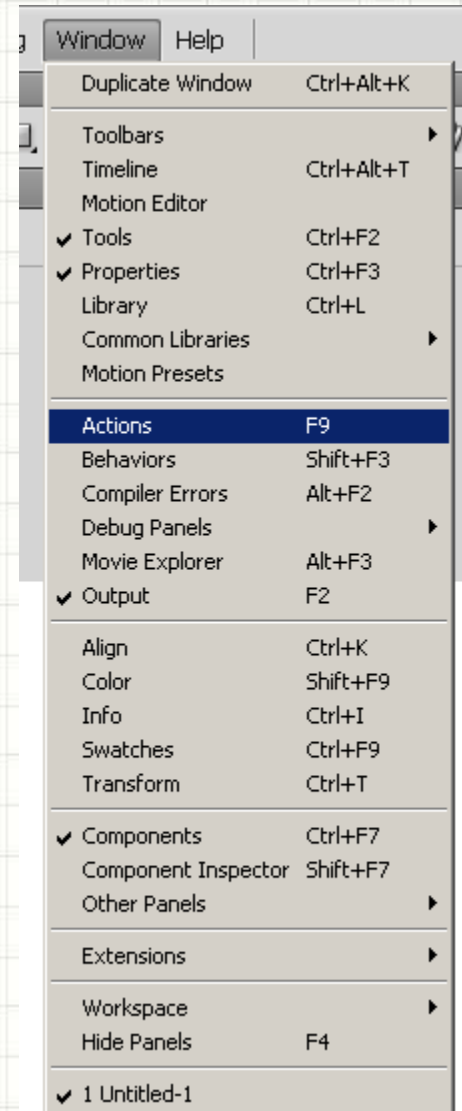
# ActionScript 3 na P

- Vamos, agora, preparar a área de trabalho
- Selecione **Window > Workspace > Developer**



# ActionScript 3 na Prática

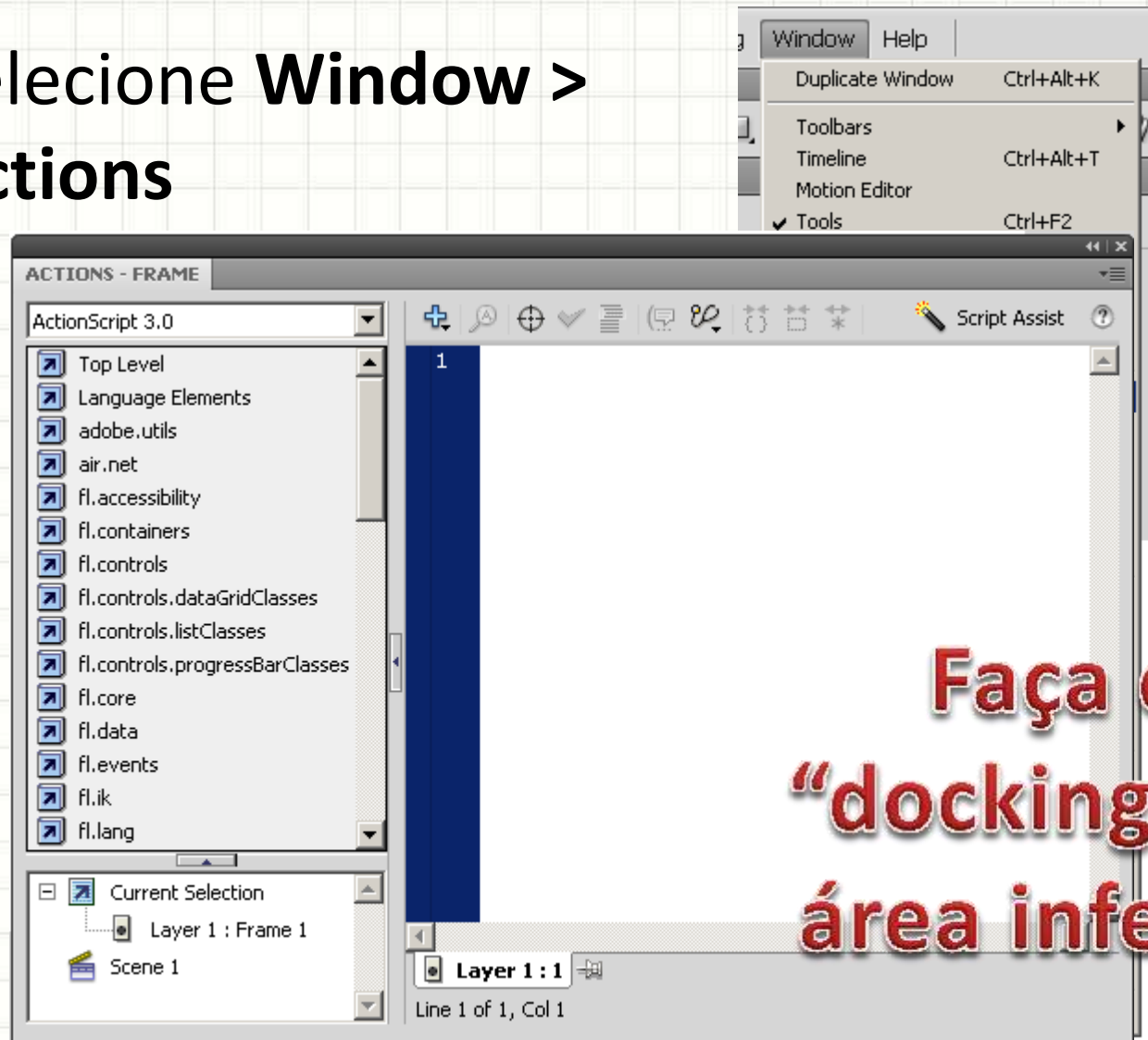
- Selecione **Window > Actions**





# ActionScript 3 na Prática

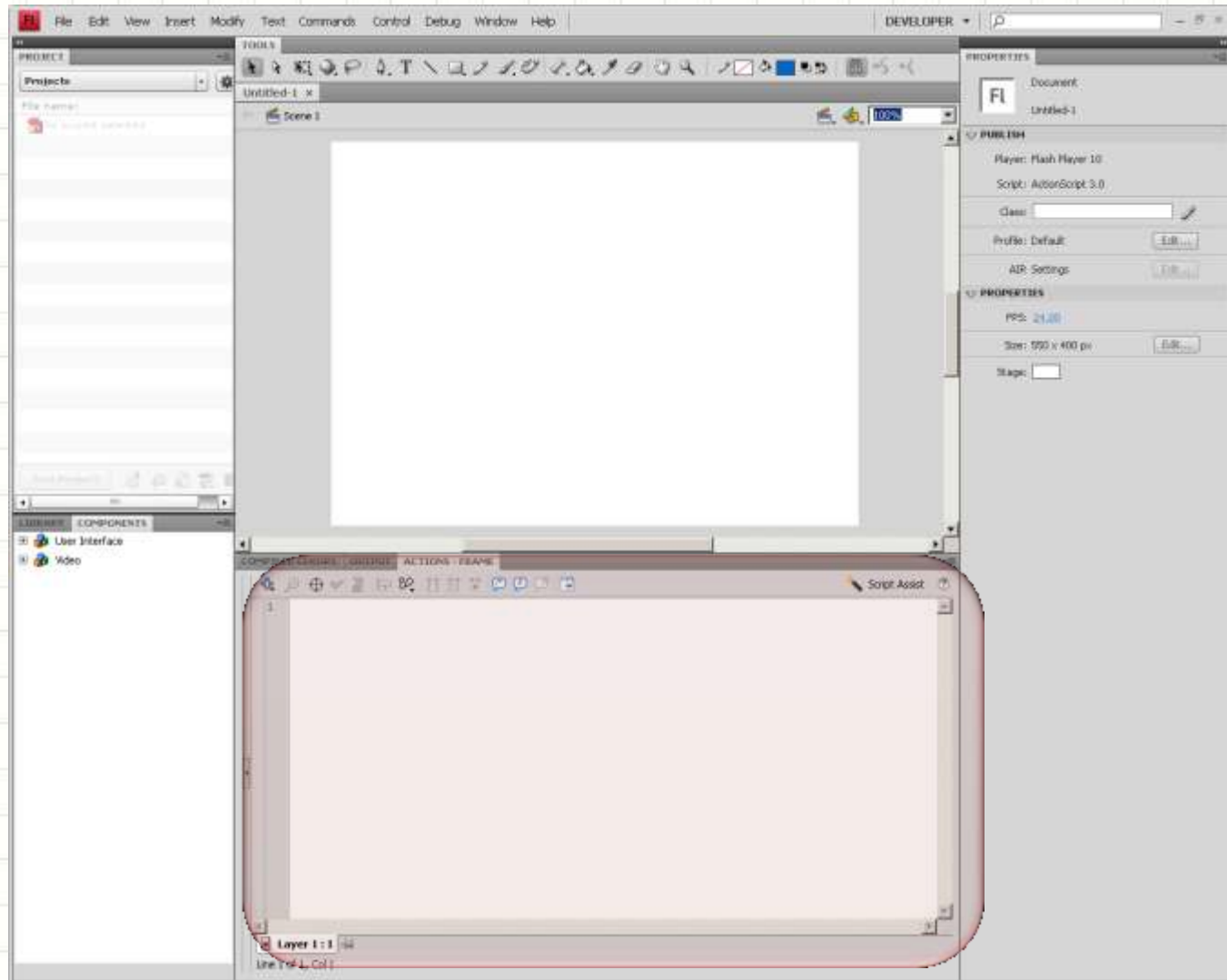
- Selecione **Window > Actions**



Faça o  
“docking” na  
área inferior

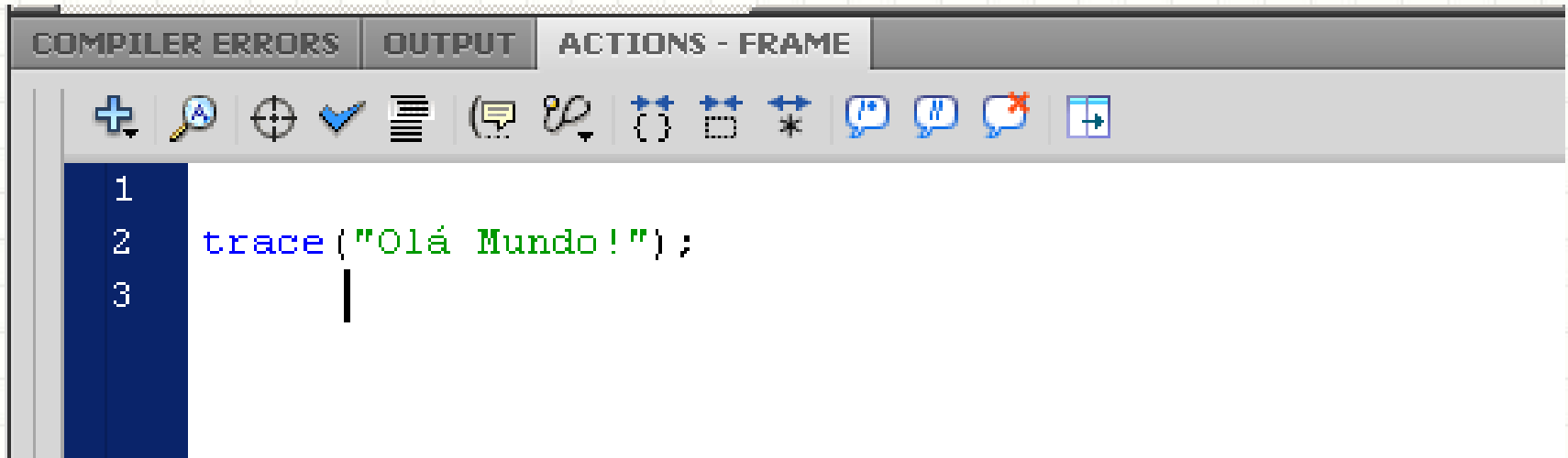
# ActionScript 3 na Prática

- Tela de trabalho



# ActionScript 3 na Prática

- Primeiro programa...

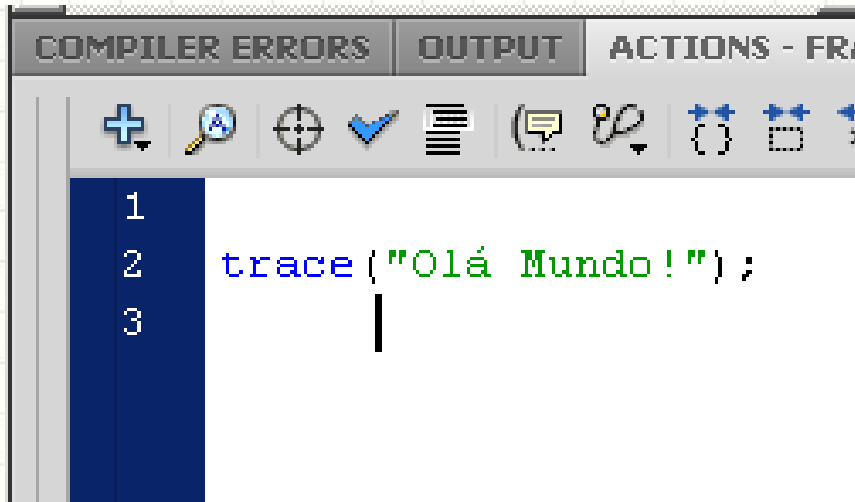


The screenshot shows a software interface with a grid background. At the top, there are three tabs: "COMPILER ERRORS", "OUTPUT", and "ACTIONS - FRAME". Below the tabs is a toolbar with various icons for editing and development. The main area is a code editor with a dark blue background. It contains three lines of code:

```
1  
2 trace("Olá Mundo!");  
3 |
```

# ActionScript 3 na Prática

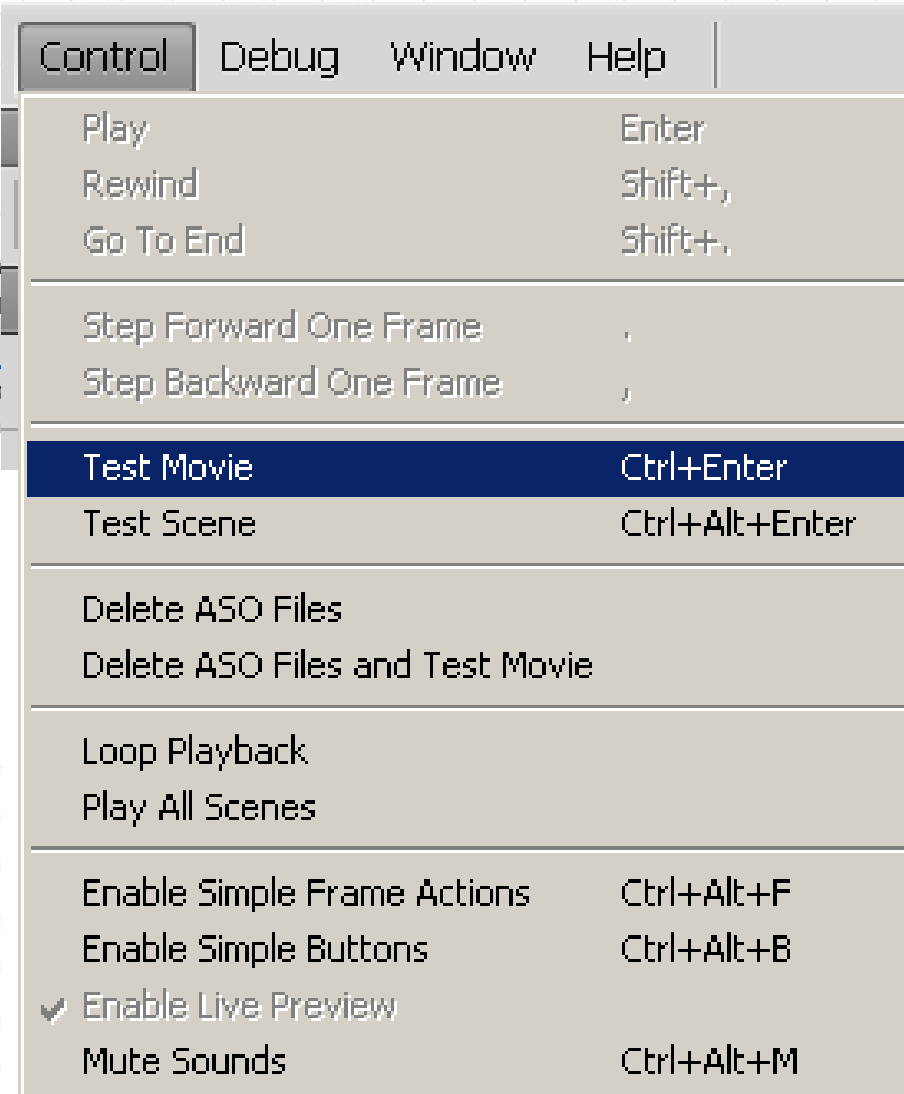
- Executando...  
**Control > Test Movie**



The screenshot shows the ActionScript IDE interface. At the top, there are three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FR...'. Below the tabs is a toolbar with various icons for editing and testing. The main area is a code editor with a dark blue background. It contains three lines of code:

```
1  
2 trace("Olá Mundo!");  
3 |
```

- **Ctrl + Enter** também funciona!

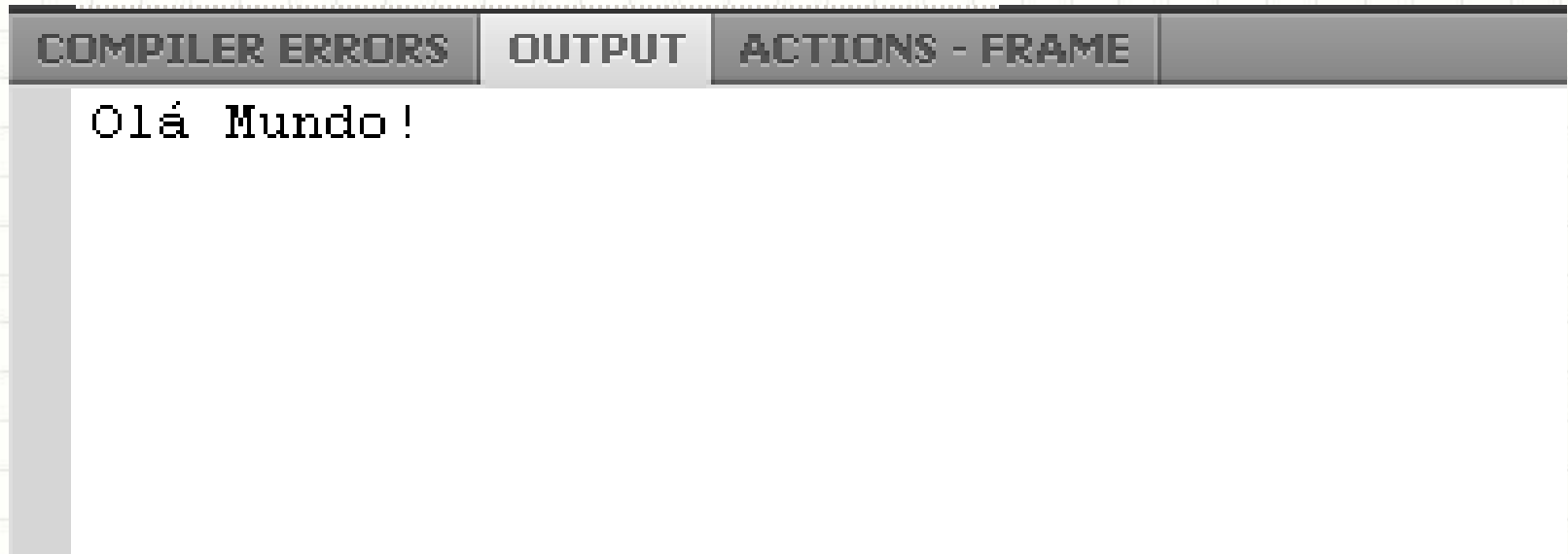


The screenshot shows the 'Control' menu in the ActionScript IDE. The menu items and their keyboard shortcuts are as follows:

Menu Item	Keyboard Shortcut
Play	Enter
Rewind	Shift+,
Go To End	Shift+,
Step Forward One Frame	,
Step Backward One Frame	.
<b>Test Movie</b>	<b>Ctrl+Enter</b>
Test Scene	Ctrl+Alt+Enter
Delete ASO Files	
Delete ASO Files and Test Movie	
Loop Playback	
Play All Scenes	
Enable Simple Frame Actions	Ctrl+Alt+F
Enable Simple Buttons	Ctrl+Alt+B
✓ Enable Live Preview	
Mute Sounds	Ctrl+Alt+M

# ActionScript 3 na Prática

- Resultado...



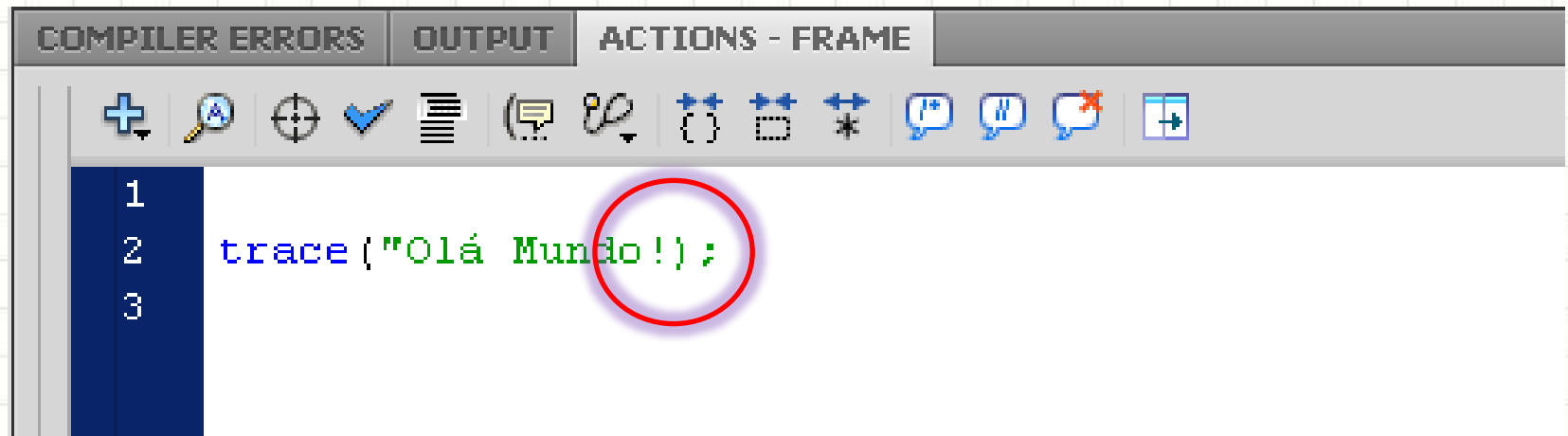
The screenshot shows a software interface with a dark grey header bar containing three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'OUTPUT' tab is selected and active. Below the tabs, the text 'Olá Mundo!' is displayed in a monospaced font on a white background.

- Não aparece no filme?
  - Não...
  - **trace** é um comando de teste!



# ActionScript 3 na Prática

- E se houver erro...?

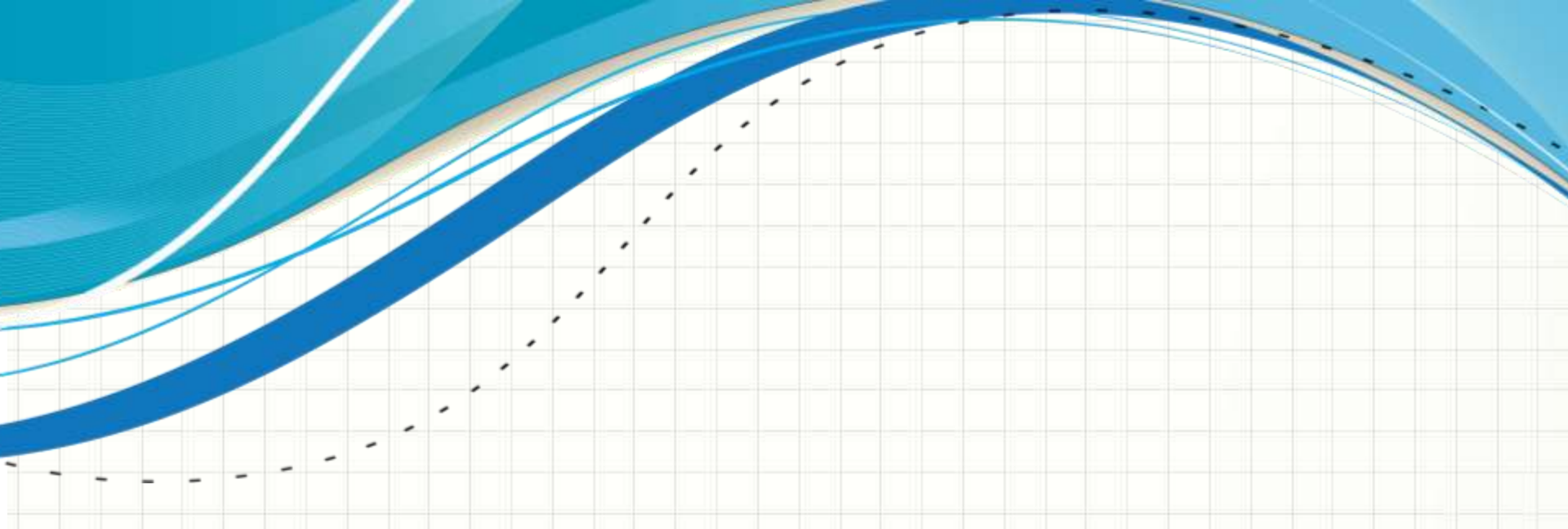


The screenshot shows an IDE window titled 'ACTIONS - FRAME'. The code editor contains three lines of code:

```
1  
2 trace ("Olá Mundo!");  
3
```

The closing parenthesis of the string literal in line 2 is circled in red, indicating a syntax error. The toolbar above the editor includes icons for zooming, undo, redo, and other editing functions.

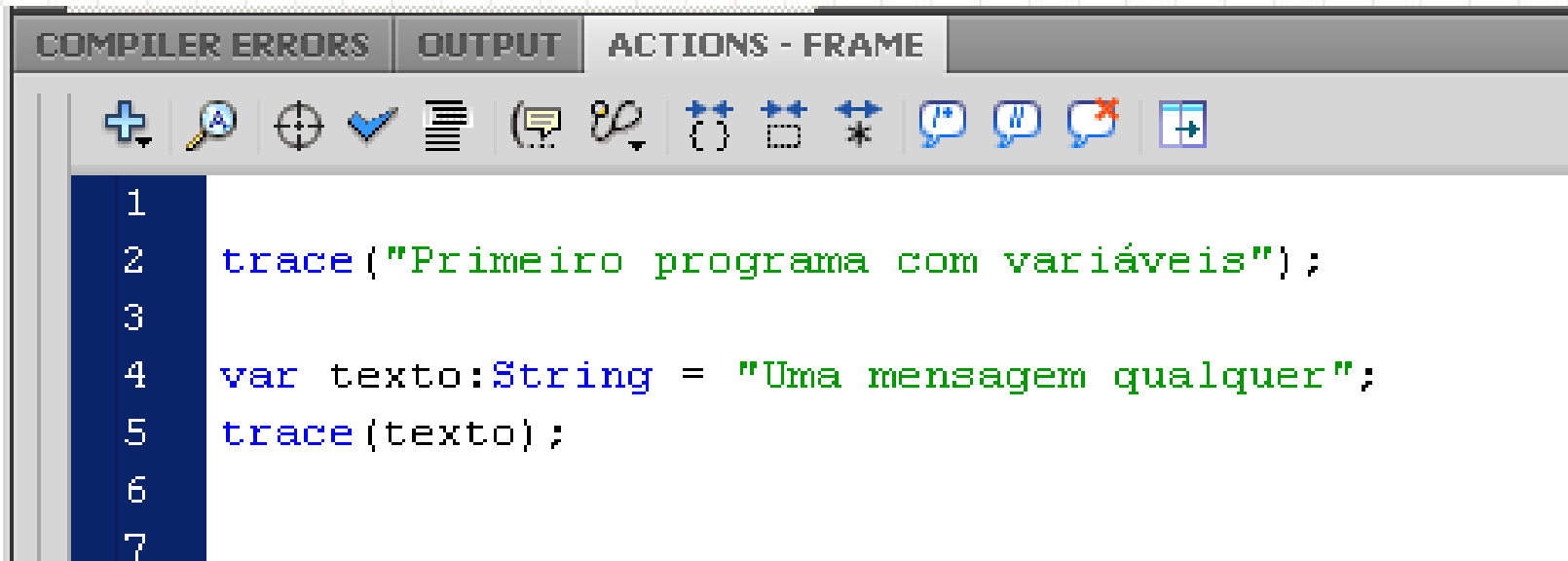
COMPILER ERRORS - 3 REPORTED	OUTPUT	ACTIONS - FRAME
Location	Description	Source
Scene 1, Layer 'Layer 1', Frame ...	1095: Syntax error: A string literal must be terminated before the line break.	trace("Olá Mundo!");
Scene 1, Layer 'Layer 1', Frame 1	1084: Syntax error: expecting identifier before end of program.	
Scene 1, Layer 'Layer 1', Frame 1	1084: Syntax error: expecting rightrightparen before end of program.	



# **VARIÁVEIS EM ACTIONSCRIPT 3**

# Variáveis em ActionScript 3

- Teste o programa...



```
1  
2 trace("Primeiro programa com variáveis");  
3  
4 var texto:String = "Uma mensagem qualquer";  
5 trace(texto);  
6  
7
```

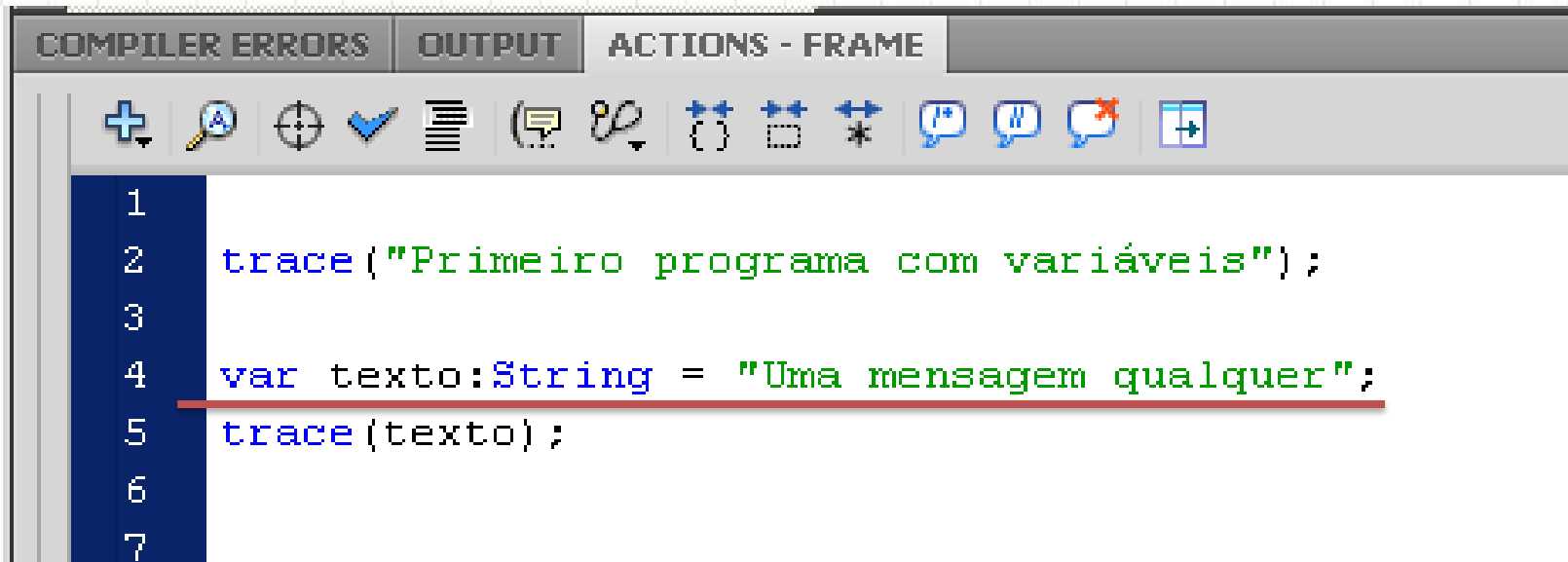
- **texto** é uma variável do tipo **String**

# Variáveis em ActionScript 3

- Declaramos variáveis sempre assim:

**var** *nome:Tipo* = *valor\_inicial*;

- Observe!



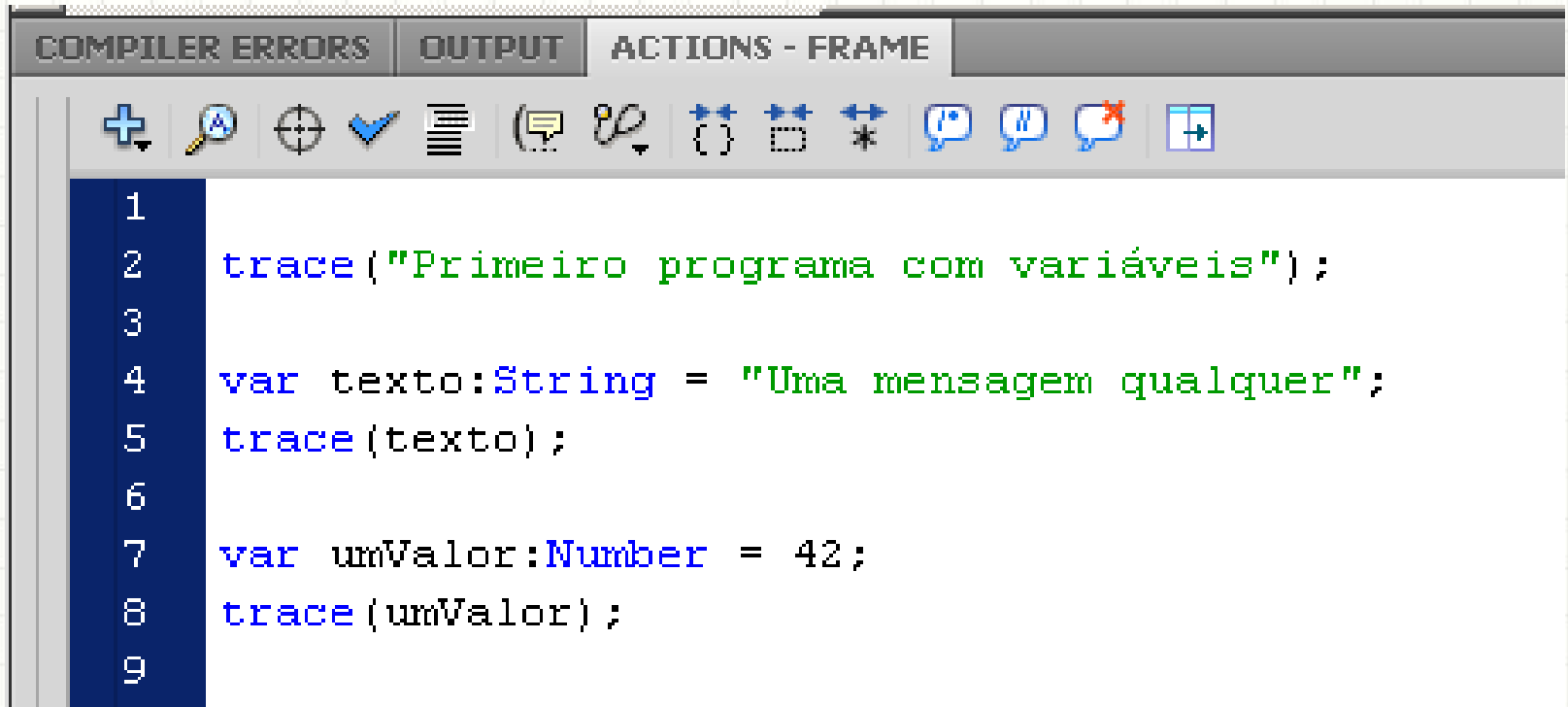
The screenshot shows an IDE window titled "ACTIONS - FRAME" with tabs for "COMPILER ERRORS", "OUTPUT", and "ACTIONS - FRAME". The code editor contains the following ActionScript code:

```
1  
2 trace("Primeiro programa com variáveis");  
3  
4 var texto:String = "Uma mensagem qualquer";  
5 trace(texto);  
6  
7
```

The line `var texto:String = "Uma mensagem qualquer";` is underlined in red.

# Variáveis em ActionScript 3

- Incrementando o programa



The screenshot shows an IDE window titled 'ACTIONS - FRAME'. The code editor contains the following ActionScript 3 code:

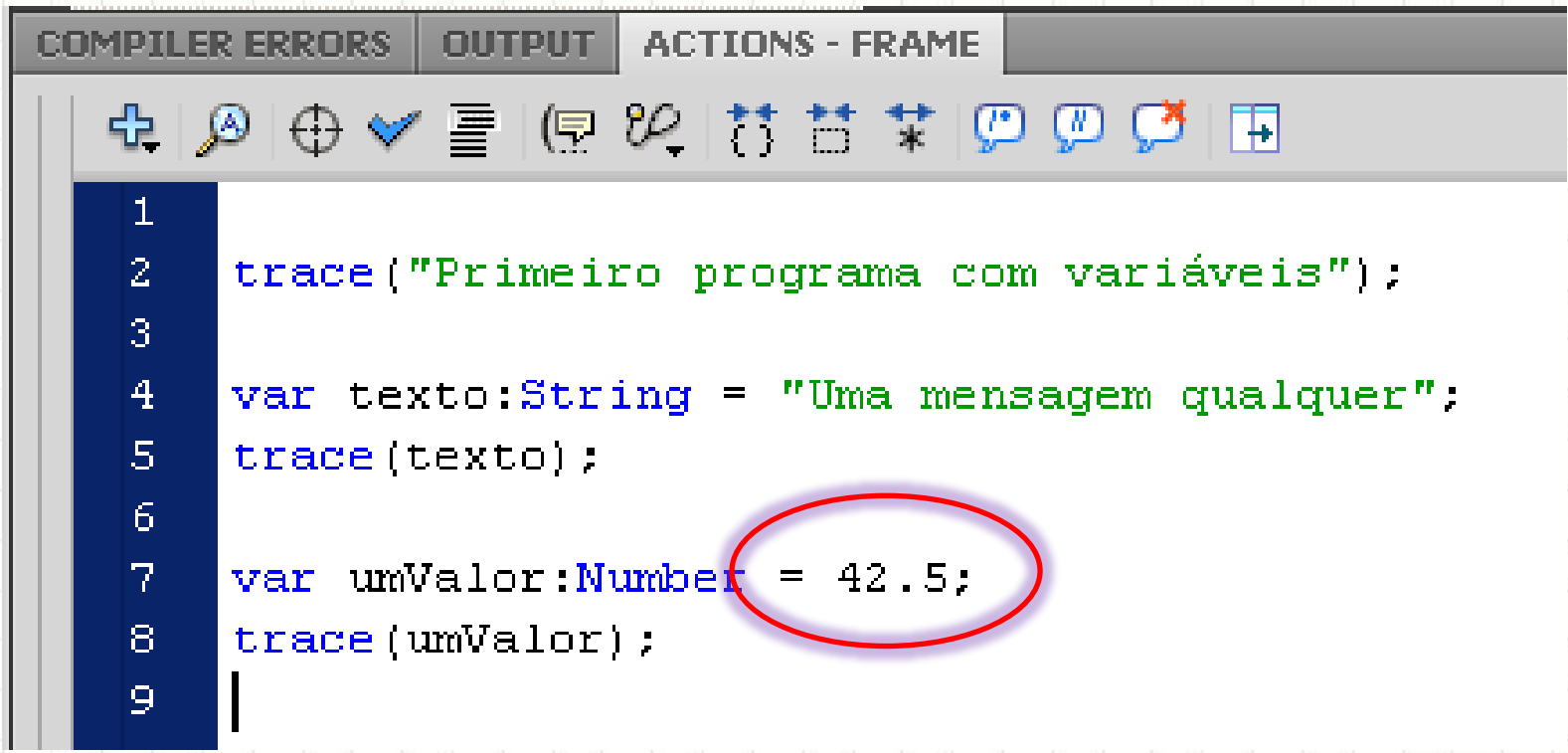
```
1  
2 trace("Primeiro programa com variáveis");  
3  
4 var texto:String = "Uma mensagem qualquer";  
5 trace(texto);  
6  
7 var umValor:Number = 42;  
8 trace(umValor);  
9
```

- **umValor** é uma variável do tipo **Number**



# Variáveis em ActionScript 3

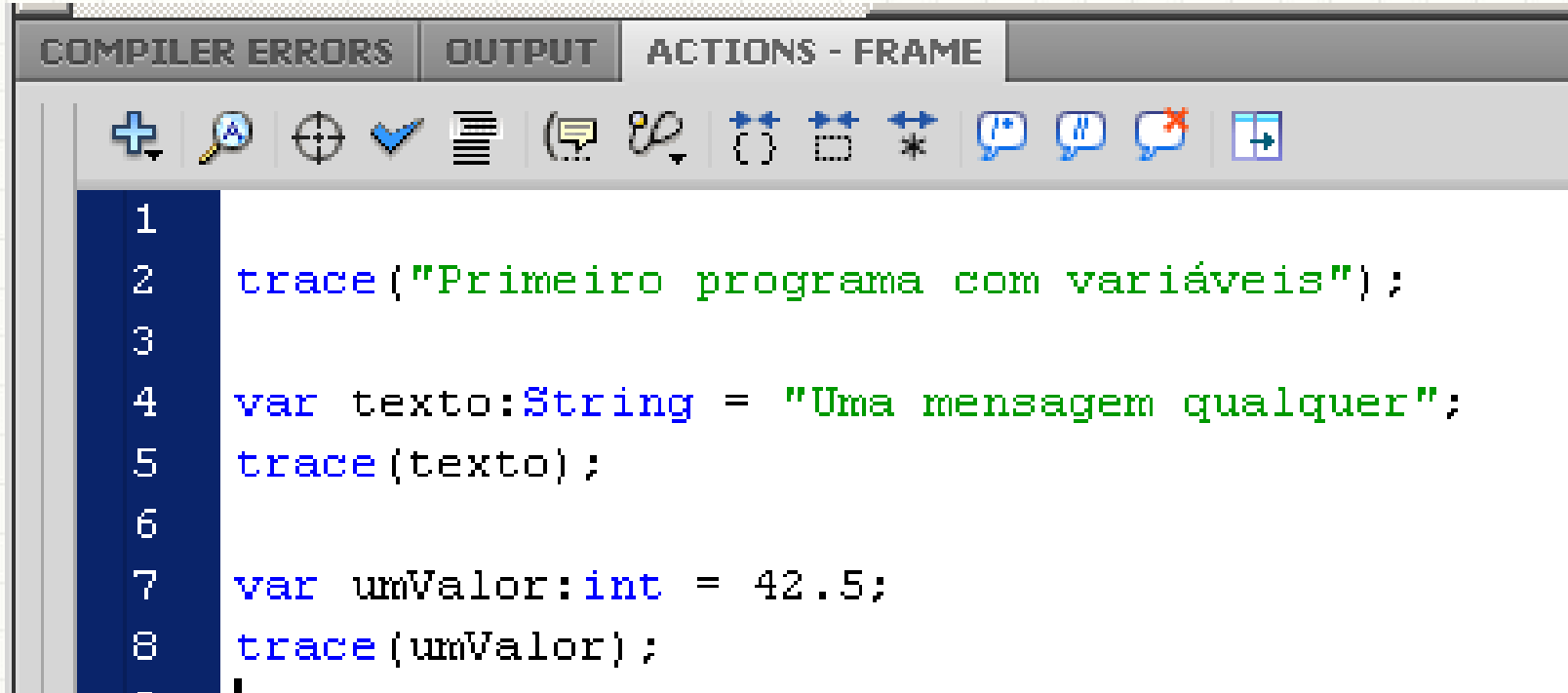
- Number aceita valores fracionários...



```
1  
2 trace("Primeiro programa com variáveis");  
3  
4 var texto:String = "Uma mensagem qualquer";  
5 trace(texto);  
6  
7 var umValor:Number = 42.5;  
8 trace(umValor);  
9
```

# Variáveis em ActionScript 3

- Há variáveis inteiras também...



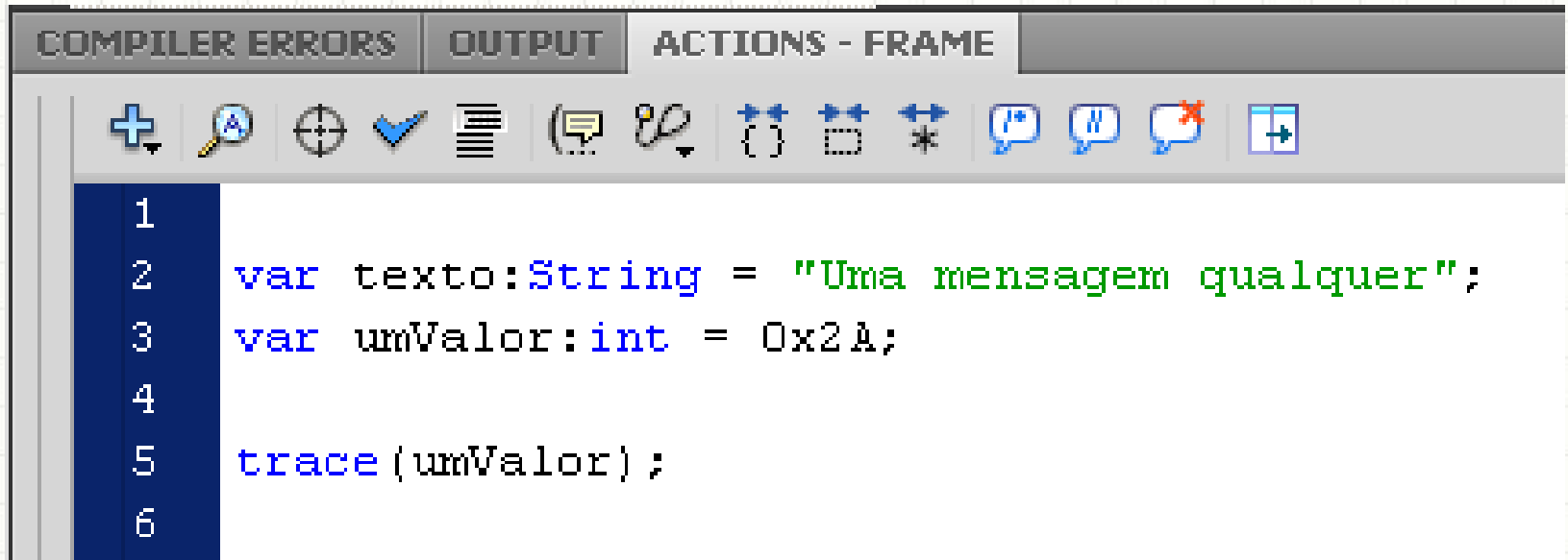
The screenshot shows an IDE window titled 'ACTIONS - FRAME'. The code editor contains the following ActionScript code:

```
1  
2 trace("Primeiro programa com variáveis");  
3  
4 var texto:String = "Uma mensagem qualquer";  
5 trace(texto);  
6  
7 var umValor:int = 42.5;  
8 trace(umValor);
```

- **umValor**, agora, é uma variável do tipo **int**
- O valor 42.5 será truncado!

# Variáveis em ActionScript 3

- Podemos usar valores em hexadecimal...



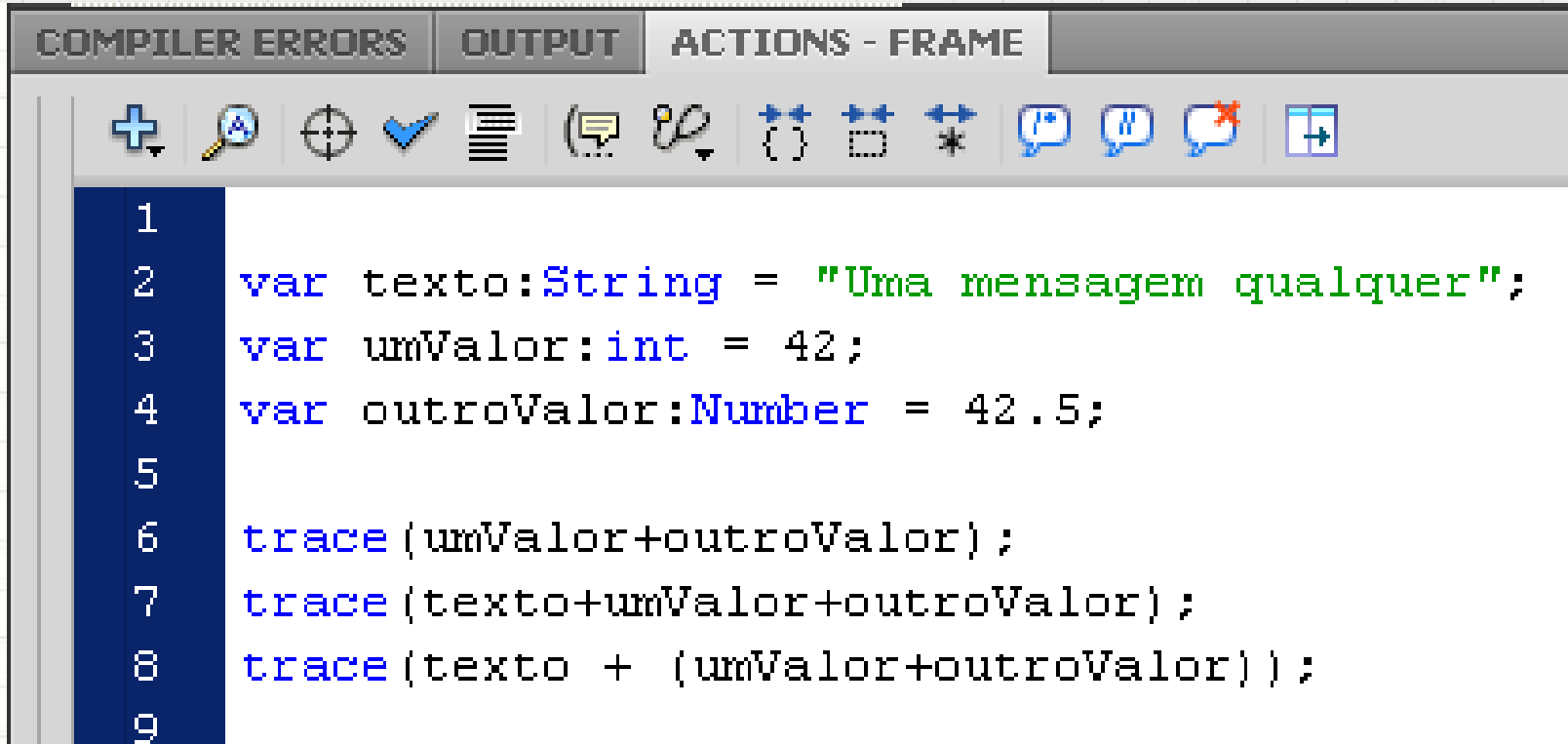
The screenshot shows an IDE window titled 'ACTIONS - FRAME'. The window has a toolbar with various icons for editing and debugging. Below the toolbar, there is a code editor with the following code:

```
1  
2 var texto:String = "Uma mensagem qualquer";  
3 var umValor:int = 0x2A;  
4  
5 trace(umValor);  
6
```

- Qual o valor impresso?

# Variáveis em ActionScript 3

- Podemos misturar variáveis no trace...



The screenshot shows an IDE window titled 'ACTIONS - FRAME'. The window has a toolbar with various icons for editing and debugging. Below the toolbar, there is a code editor with the following code:

```
1  
2 var texto:String = "Uma mensagem qualquer";  
3 var umValor:int = 42;  
4 var outroValor:Number = 42.5;  
5  
6 trace(umValor+outroValor);  
7 trace(texto+umValor+outroValor);  
8 trace(texto + (umValor+outroValor));  
9
```

- Execute e observe os resultados...
- Qual a diferença entre eles?

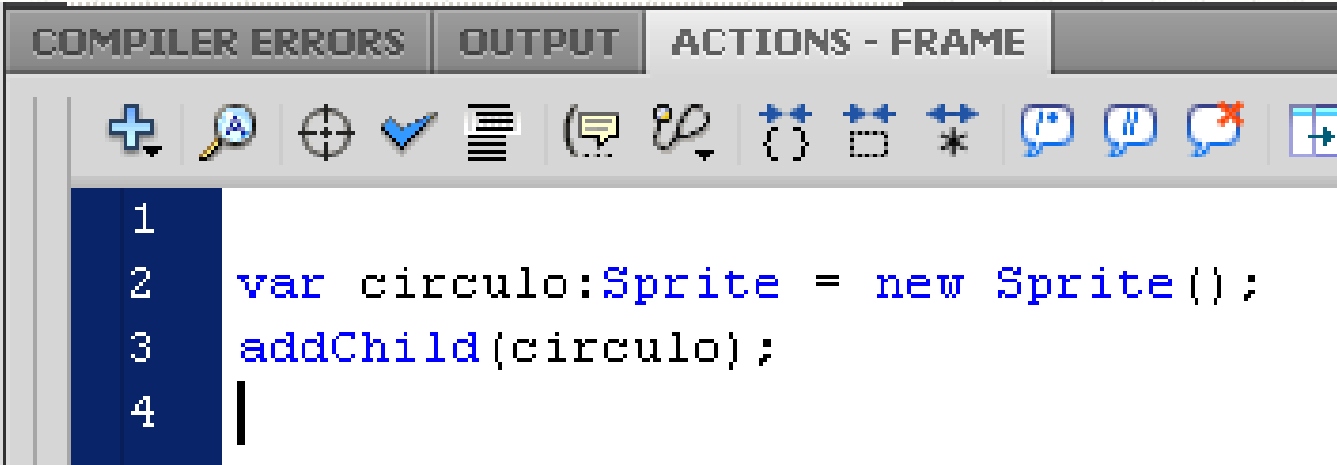


# **DESENHANDO NA TELA COM ACTIONSCRIPT 3**



# Desenhando em ActionScript 3

- Para desenhar na tela: criar um **Sprite**
- **Sprite** é um desenho que pode se mover
- Depois, precisamos adicionar o sprite como um filhote da animação corrente (**addChild**)
- Experimente o código abaixo... o que ocorre?

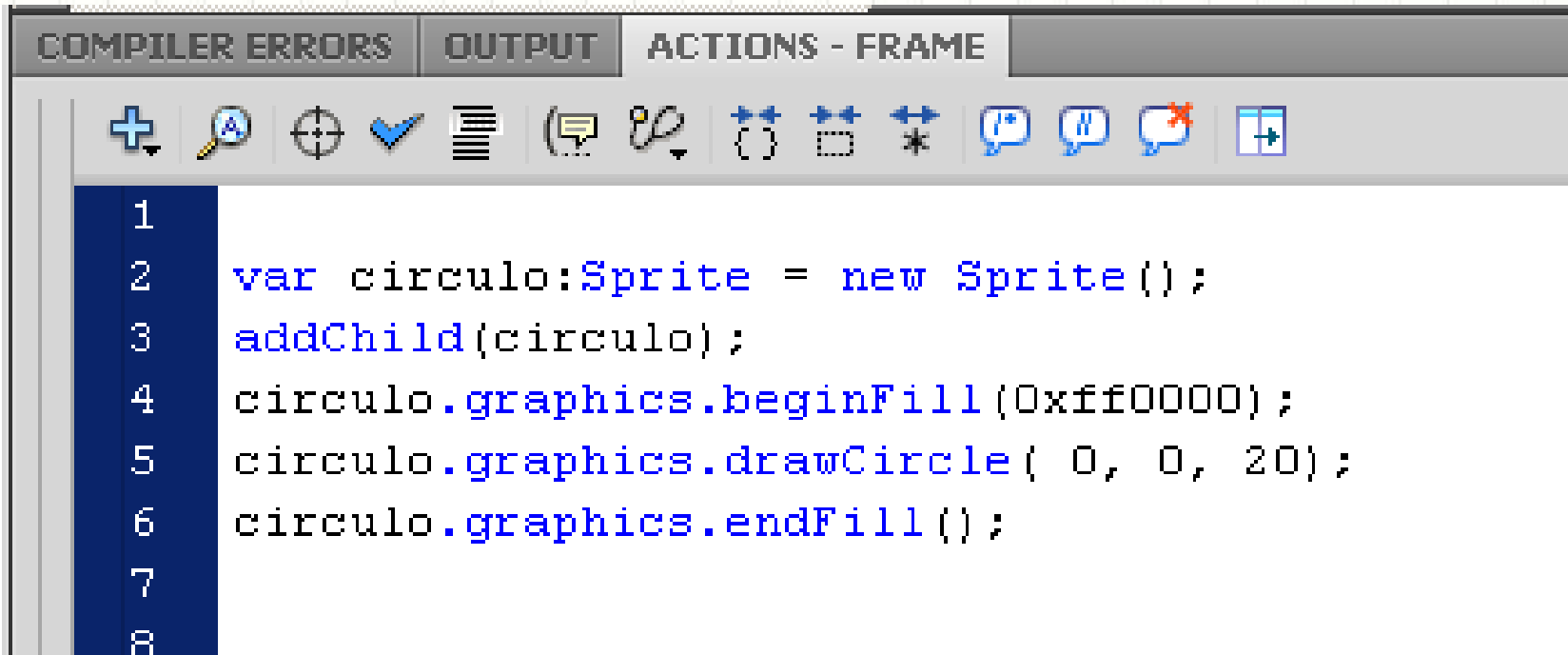


The screenshot shows a software interface with a tab labeled 'ACTIONS - FRAME'. Below the tab is a toolbar with various icons including a plus sign, a magnifying glass, a target, a checkmark, a list, a speech bubble, a hand, a code block, a star, a speech bubble with an asterisk, a speech bubble with a double slash, a speech bubble with a red X, and a window with a plus sign. Below the toolbar is a code editor with a dark blue background and white text. The code is as follows:

```
1  
2 var circulo:Sprite = new Sprite();  
3 addChild(circulo);  
4 |
```

# Desenhando em ActionScript 3

- O Sprite não tem figura ainda...
- Precisamos desenhar dentro do sprite



The screenshot shows a code editor window titled 'ACTIONS - FRAME'. The editor contains the following ActionScript 3 code:

```
1  
2 var circulo:Sprite = new Sprite();  
3 addChild(circulo);  
4 circulo.graphics.beginFill(0xff0000);  
5 circulo.graphics.drawCircle( 0, 0, 20);  
6 circulo.graphics.endFill();  
7  
8
```

- Experimente!

# o em ActionScript 3

tem figura ainda...

desenhar dentro do sprite

Indica que **iniciaremos o desenho** no sprite usando a cor de preenchimento vermelha (rrggbb)

```
2 var circulo:Sprite = new Sprite();
3 addChild(circulo);
4 circulo.graphics.beginFill(0xff0000);
5 circulo.graphics.drawCircle( 0, 0, 20);
6 circulo.graphics.endFill();
7
8
```

- Experimente!

# lo em Action

tem figura ainda

desenhar dentro

Indica que **iniciaremos o desenho** no sprite usando a cor de preenchimento vermelha (rrggbb)

Solicita o **desenho de um círculo** com centro em 0,0 e raio 20 pixels

```
2 var circulo:Sprite = new Sprite();
3 addChild(circulo);
4 circulo.graphics.beginFill(0xff0000);
5 circulo.graphics.drawCircle( 0, 0, 20);
6 circulo.graphics.endFill();
7
8
```

- Experimente!

# o em Action

tem figura ainda  
desenhar dentro

Indica que **iniciaremos o desenho** no sprite usando a cor de preenchimento vermelha (rrggbb)

Solicita o **desenho de um círculo** com centro em 0,0 e raio 20 pixels

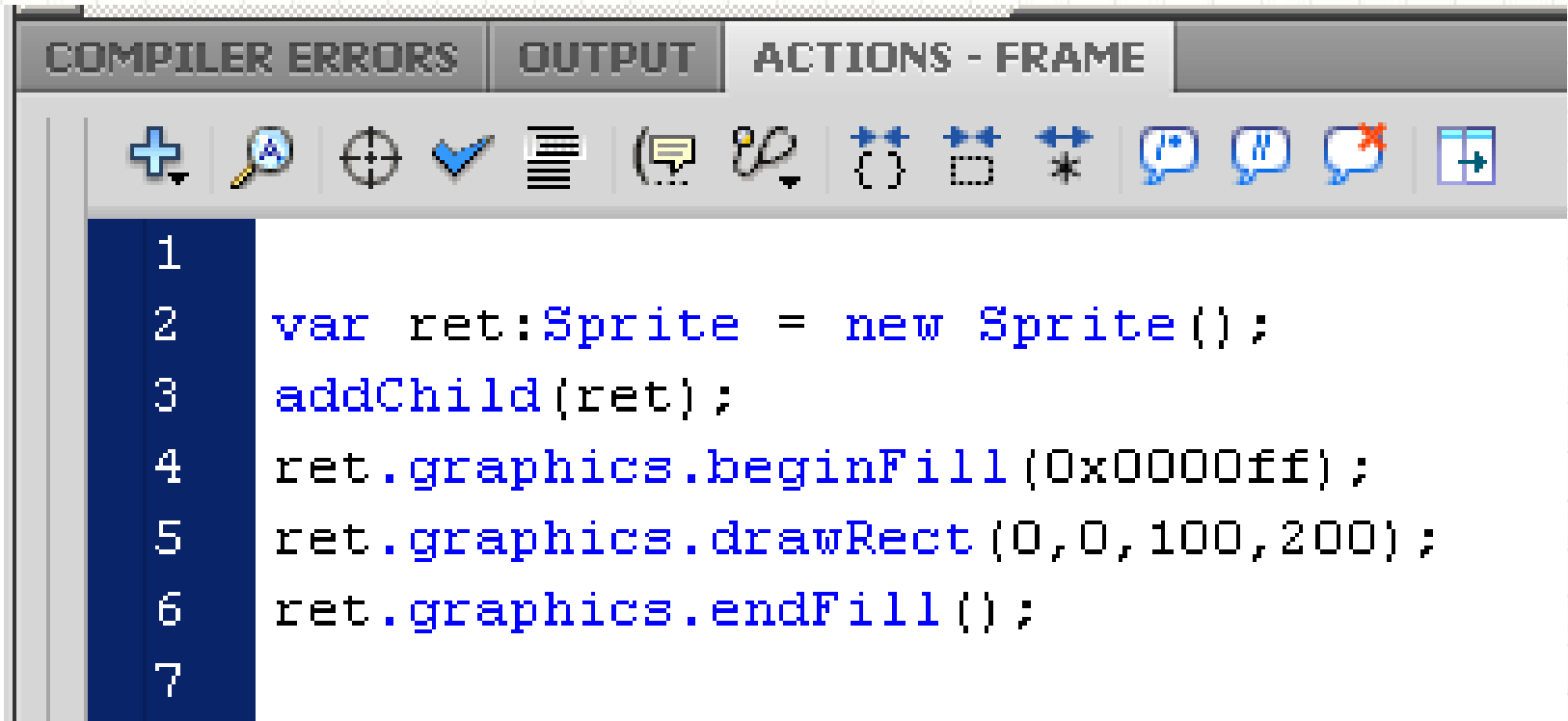
```
2 var circulo:Sprite = new Sprite();  
3 addChild(circulo);  
4 circulo.graphics.beginFill(0xff0000);  
5 circulo.graphics.drawCircle( 0, 0, 20);  
6 circulo.graphics.endFill();  
7  
8
```

Indica que o **desenho do sprite acabou**

- Experimente!

# Desenhando em ActionScript 3

- Desenhando um retângulo...



The screenshot shows a code editor window titled 'ACTIONS - FRAME'. The editor contains the following ActionScript 3 code:

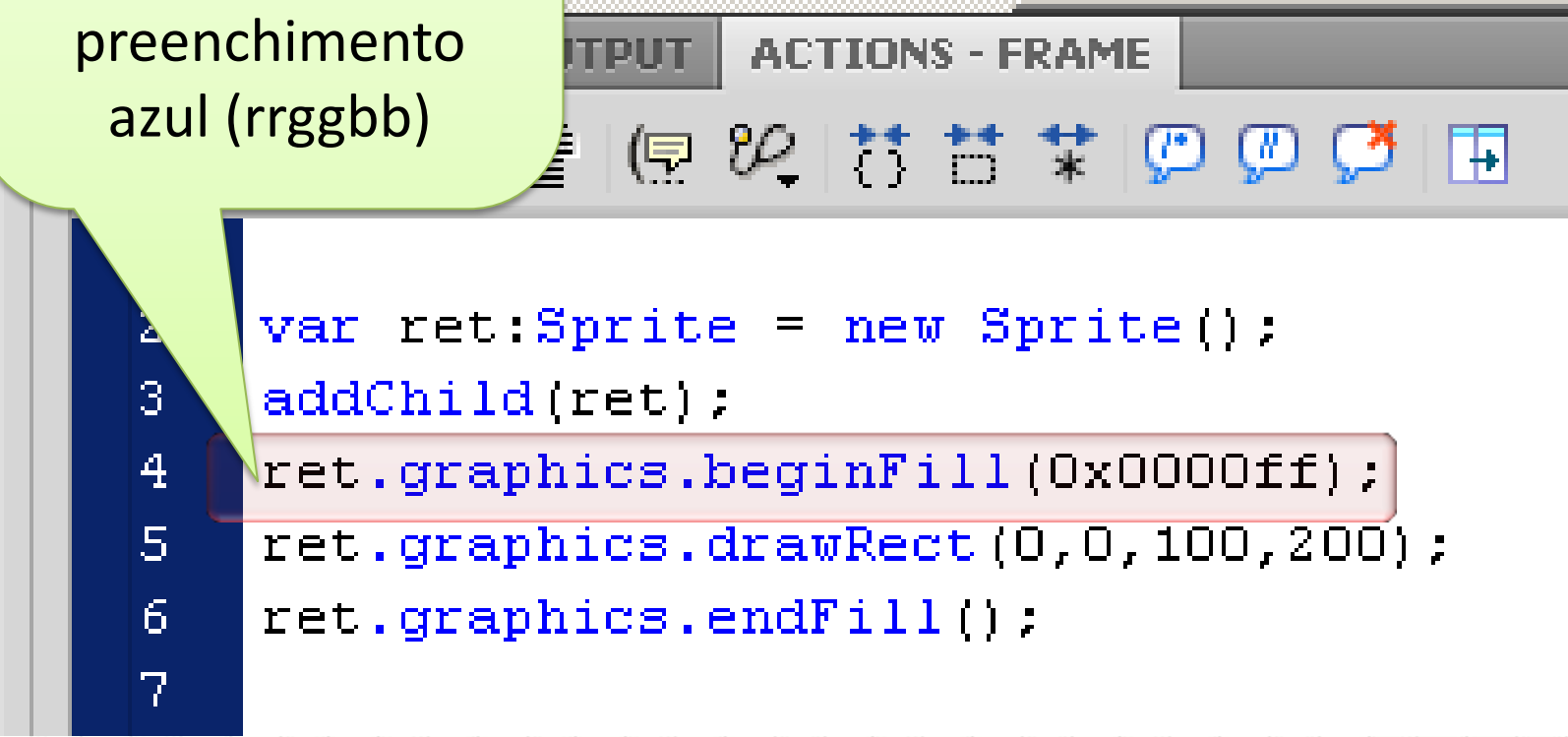
```
1  
2 var ret:Sprite = new Sprite();  
3 addChild(ret);  
4 ret.graphics.beginFill(0x0000ff);  
5 ret.graphics.drawRect(0,0,100,200);  
6 ret.graphics.endFill();  
7
```

- Experimente!



Indica que  
**iniciaremos o  
desenho** no sprite  
usando a cor de  
preenchimento  
azul (rrggbb)

# o em ActionScript 3 m retângulo...



```
2 var ret:Sprite = new Sprite();
3 addChild(ret);
4 ret.graphics.beginFill(0x0000ff);
5 ret.graphics.drawRect(0,0,100,200);
6 ret.graphics.endFill();
7
```

- Experimente!

Indica que **iniciaremos o desenho** no sprite usando a cor de preenchimento azul (rrggbb)

o em Act  
m retângulo

Solicita o **desenho de um retângulo** com canto superior esquerdo em 0,0, largura 100 pixels e altura 200 pixels

```
2 var ret:Sprite = new Sprite();  
3 addChild(ret);  
4 ret.graphics.beginFill(0x0000ff);  
5 ret.graphics.drawRect(0,0,100,200);  
6 ret.graphics.endFill();  
7
```

- Experimente!

Indica que **iniciaremos o desenho** no sprite usando a cor de preenchimento azul (rrggbb)

o em Act  
m retângulo

Solicita o **desenho de um retângulo** com canto superior esquerdo em 0,0, largura 100 pixels e altura 200 pixels

```
2 var ret:Sprite = new Sprite();  
3 addChild(ret);  
4 ret.graphics.beginFill(0x0000ff);  
5 ret.graphics.drawRect(0,0,100,200);  
6 ret.graphics.endFill();  
7
```

Indica que o **desenho do sprite acabou**

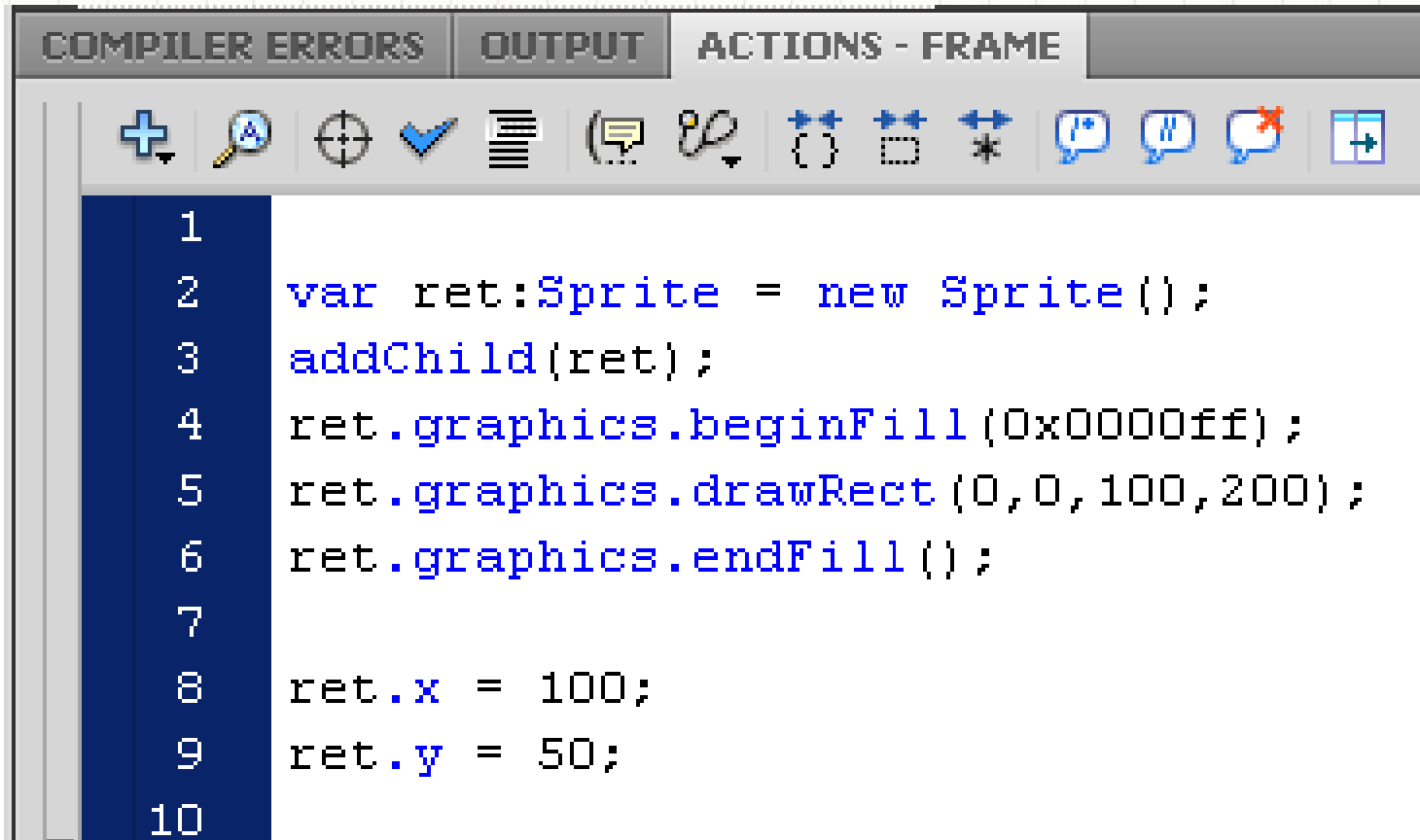
- Experimente!



# **PROPRIEDADES DOS SPRITES**

# Desenhando em ActionScript 3

- Podemos mover o nosso retângulo



The screenshot shows a code editor window titled 'ACTIONS - FRAME'. The editor contains the following ActionScript 3 code:

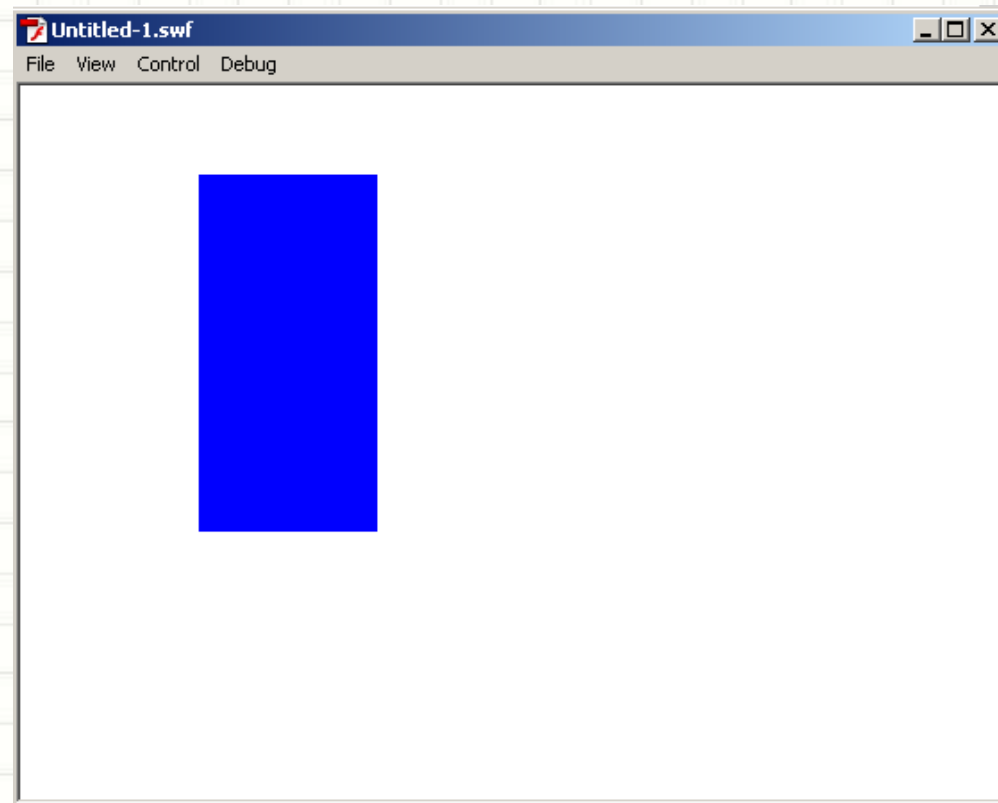
```
1  
2 var ret:Sprite = new Sprite();  
3 addChild(ret);  
4 ret.graphics.beginFill(0x0000ff);  
5 ret.graphics.drawRect(0,0,100,200);  
6 ret.graphics.endFill();  
7  
8 ret.x = 100;  
9 ret.y = 50;  
10
```

- Experimente!

# Desenhando em ActionScript 3

- Resultado

```
8 ret.x = 100;  
9 ret.y = 50;
```

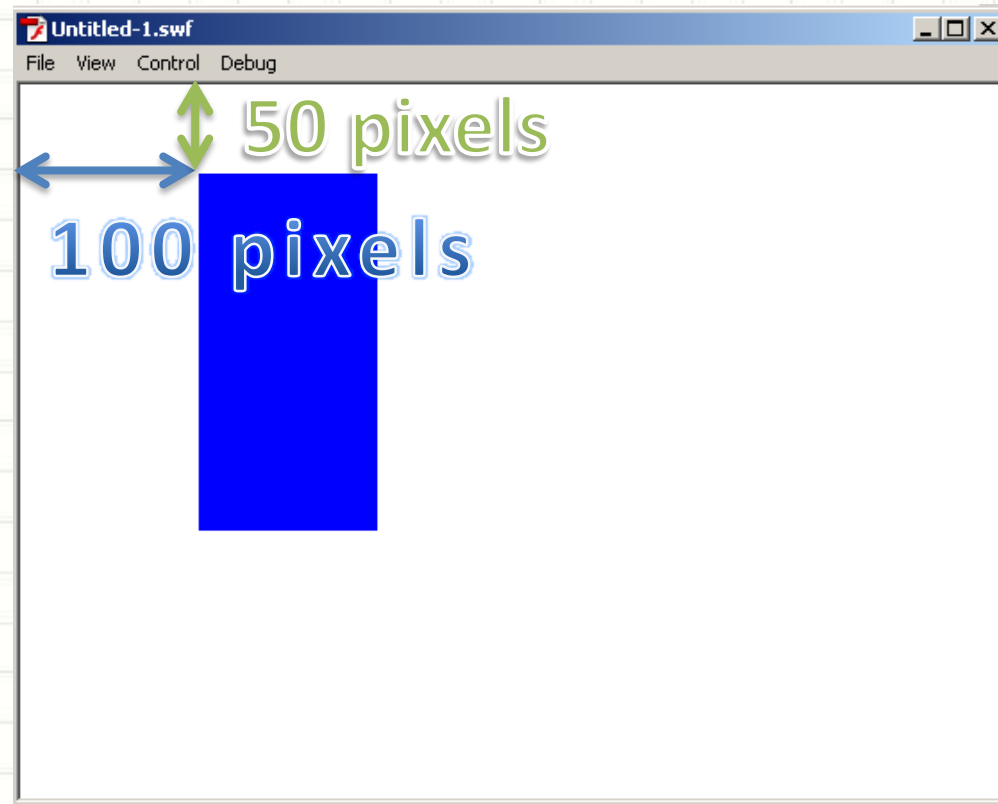




# Desenhando em ActionScript 3

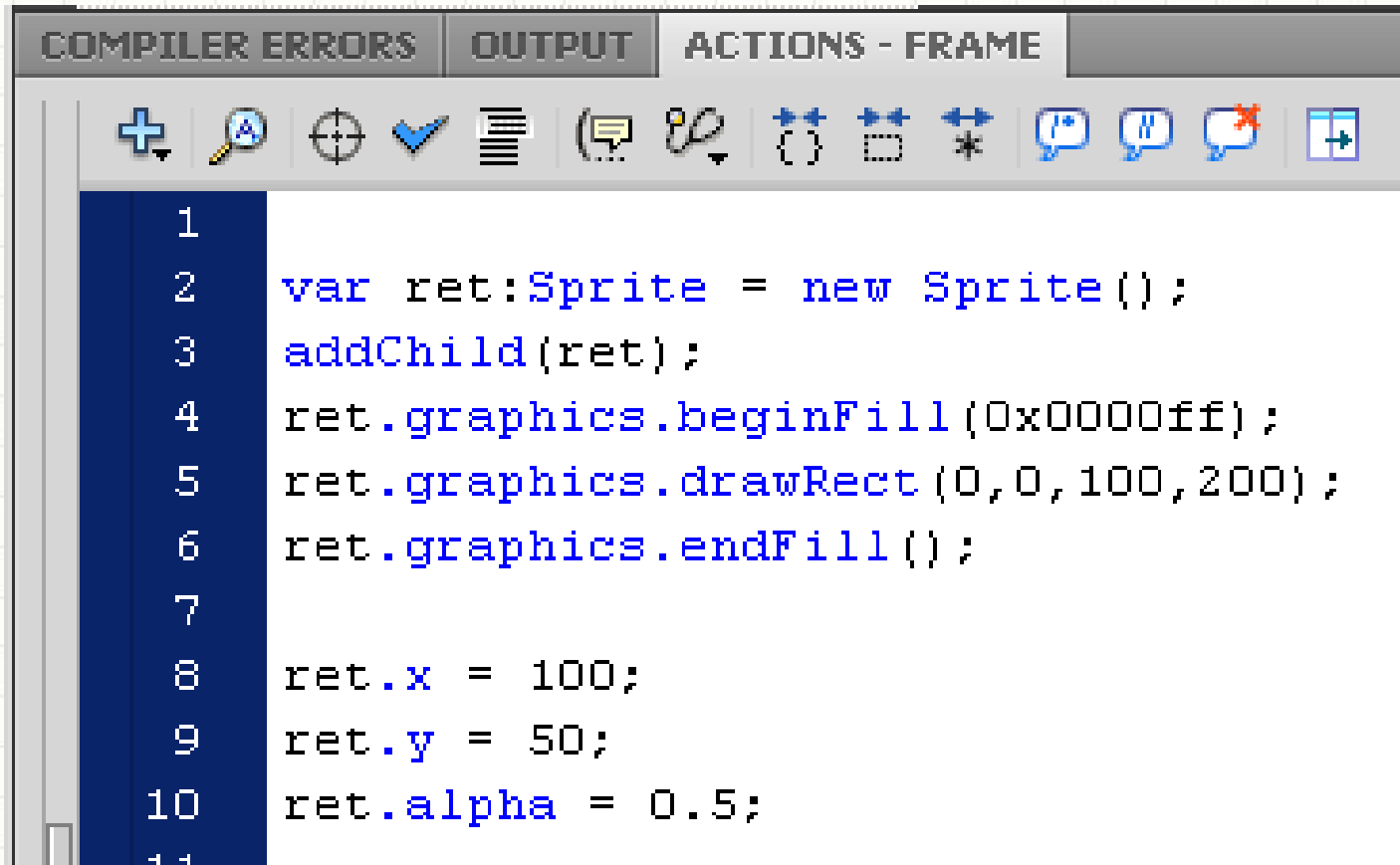
- Resultado

```
8 ret.x = 100;  
9 ret.y = 50;
```



# Desenhando em ActionScript 3

- Podemos mudar a “opacidade” do sprite



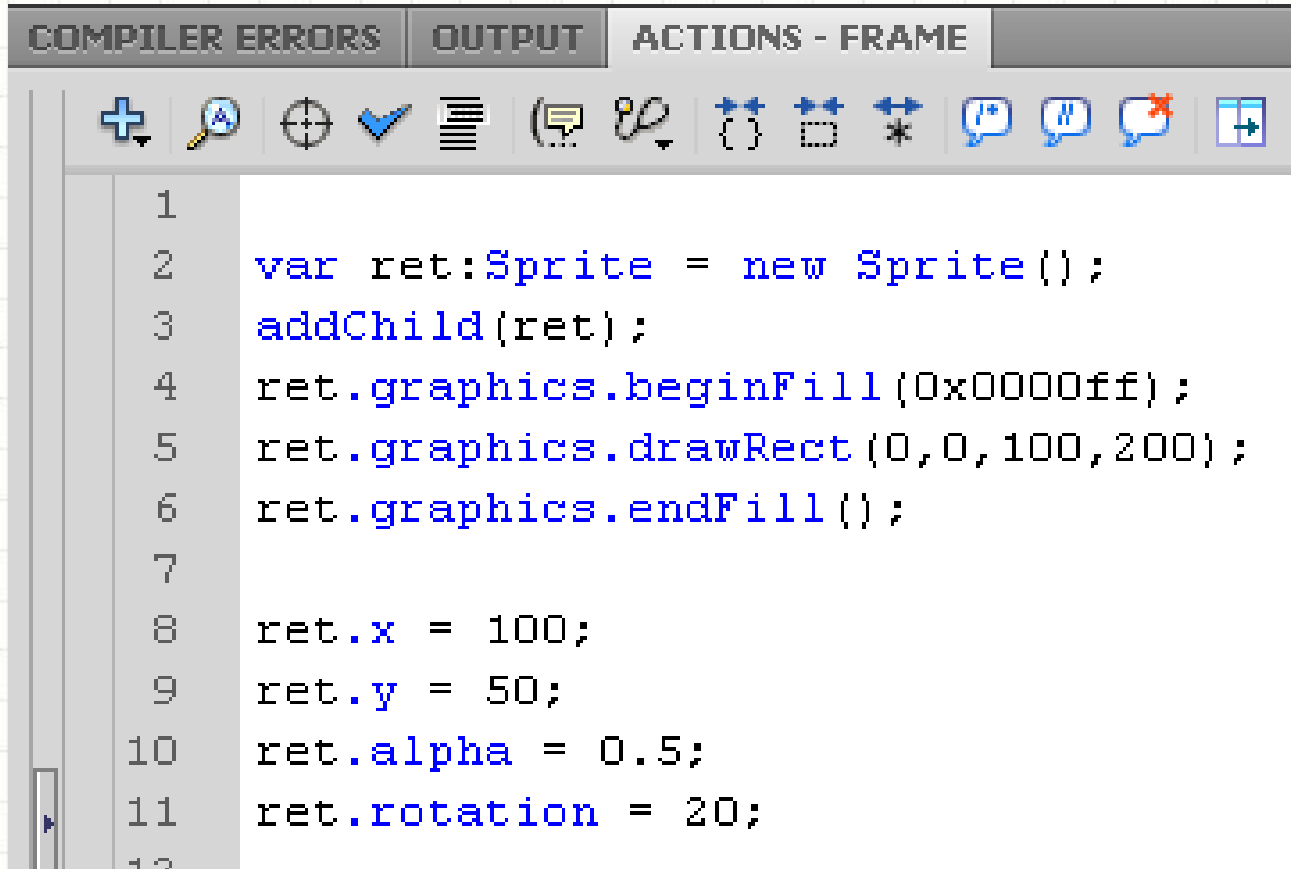
The screenshot shows an IDE window titled "ACTIONS - FRAME" with a toolbar containing icons for adding, searching, zooming, and other actions. The code editor displays the following ActionScript 3 code:

```
1  
2 var ret:Sprite = new Sprite();  
3 addChild(ret);  
4 ret.graphics.beginFill(0x0000ff);  
5 ret.graphics.drawRect(0,0,100,200);  
6 ret.graphics.endFill();  
7  
8 ret.x = 100;  
9 ret.y = 50;  
10 ret.alpha = 0.5;  
11
```

- Experimente!

# Desenhando em ActionScript 3

- Podemos “rodar” os sprite



The screenshot shows a code editor window with three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'ACTIONS - FRAME' tab is active, displaying the following ActionScript 3 code:

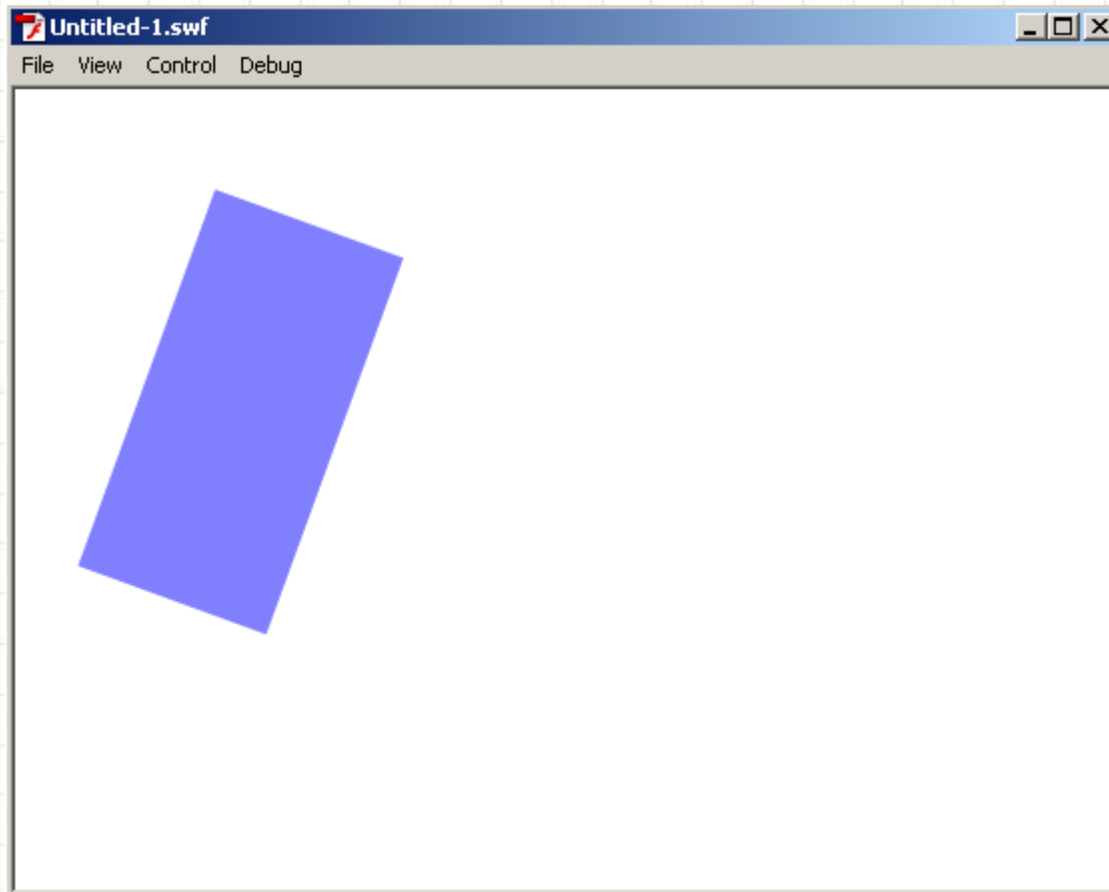
```
1  
2  var ret:Sprite = new Sprite();  
3  addChild(ret);  
4  ret.graphics.beginFill(0x0000ff);  
5  ret.graphics.drawRect(0,0,100,200);  
6  ret.graphics.endFill();  
7  
8  ret.x = 100;  
9  ret.y = 50;  
10 ret.alpha = 0.5;  
11 ret.rotation = 20;  
12
```

- Experimente!

# Desenhando em ActionScript 3

- **rotation** em graus... Sentido horário!

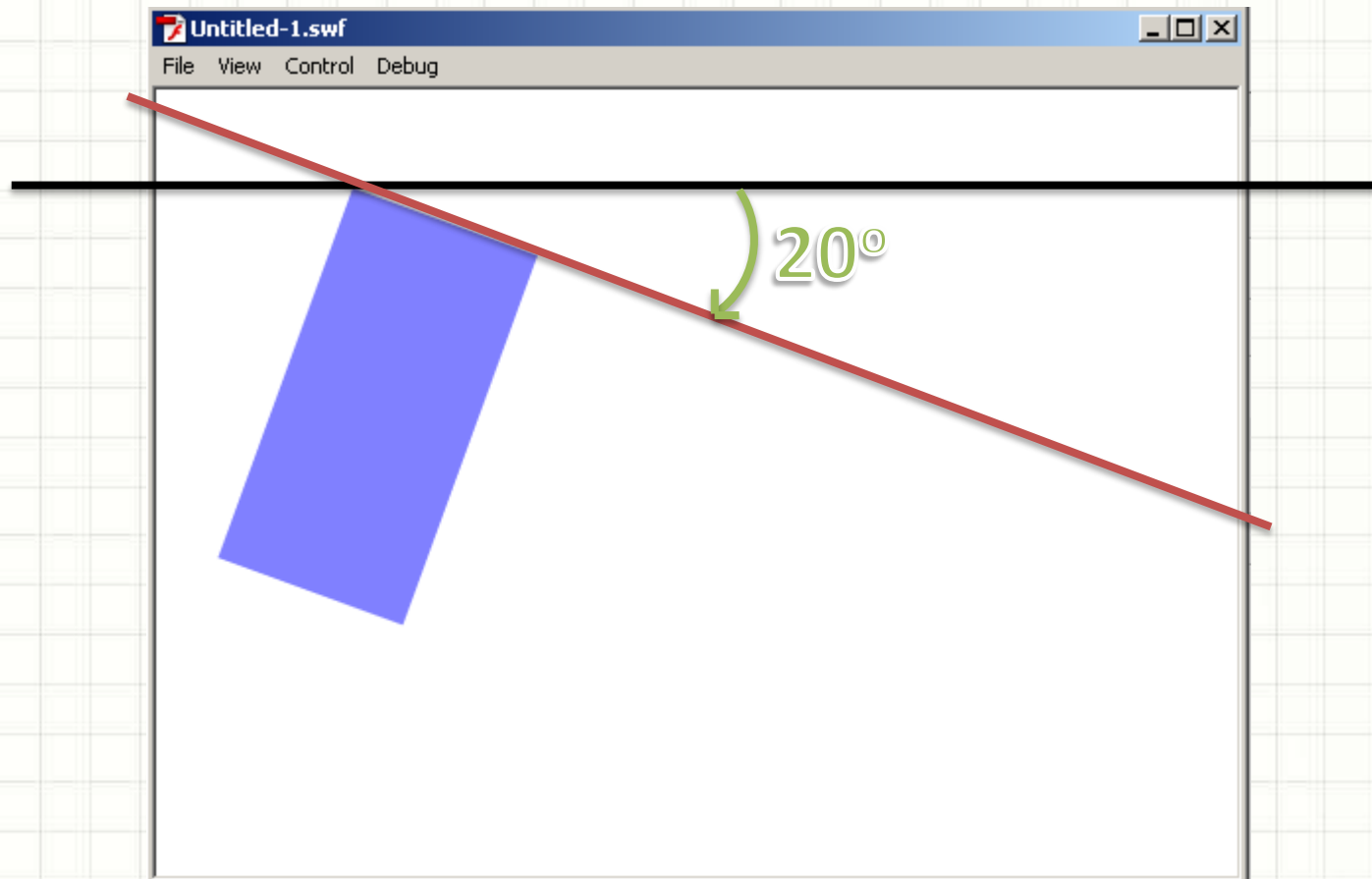
```
11 ret.rotation = 20;
```



# Desenhando em ActionScript 3

- **rotation** em graus... Sentido horário!

```
11 ret.rotation = 20;
```





# EIXOS DE COORDENADAS

# Coordenadas do AS3

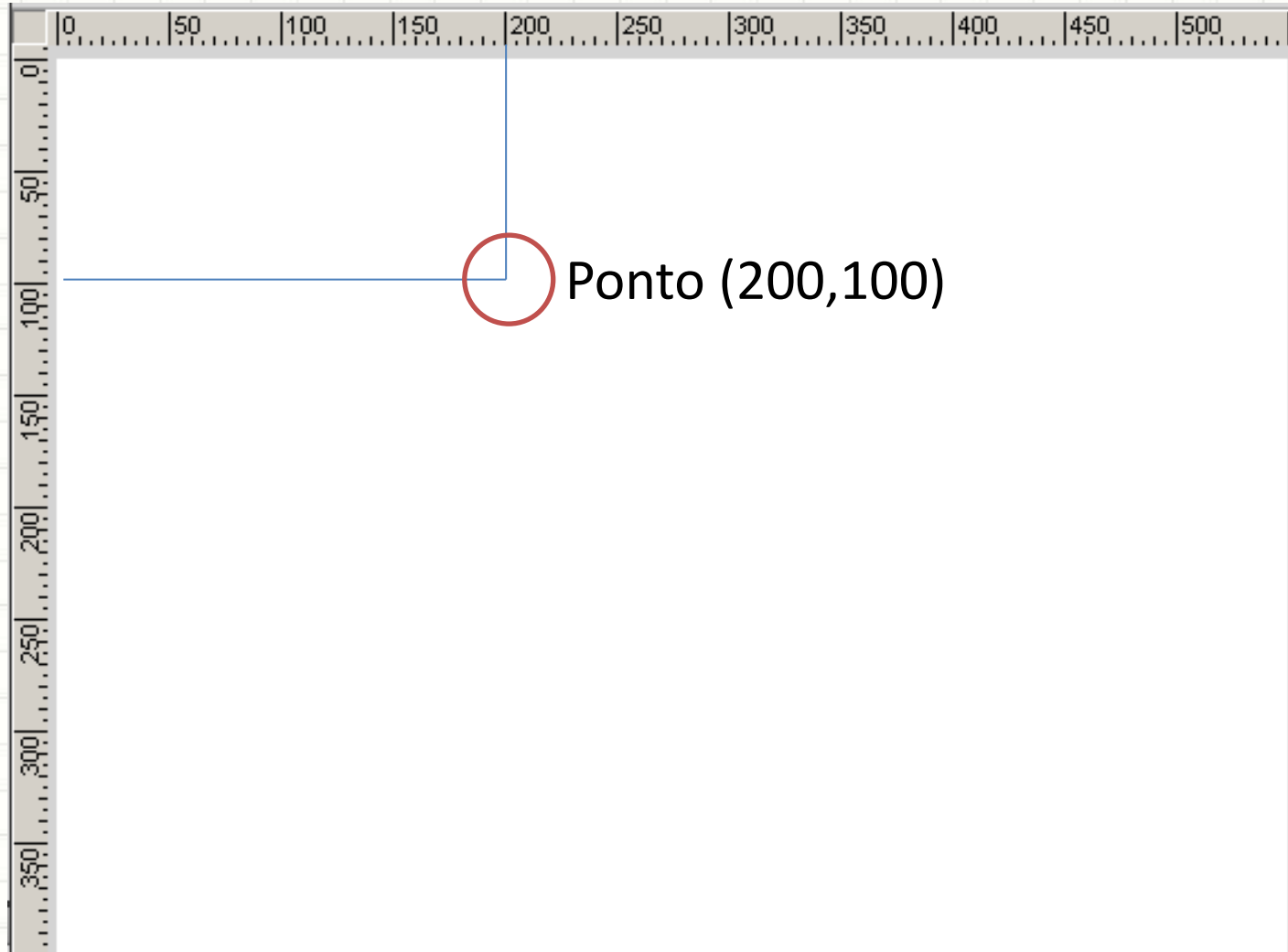
- Eixos de coordenadas do cenário





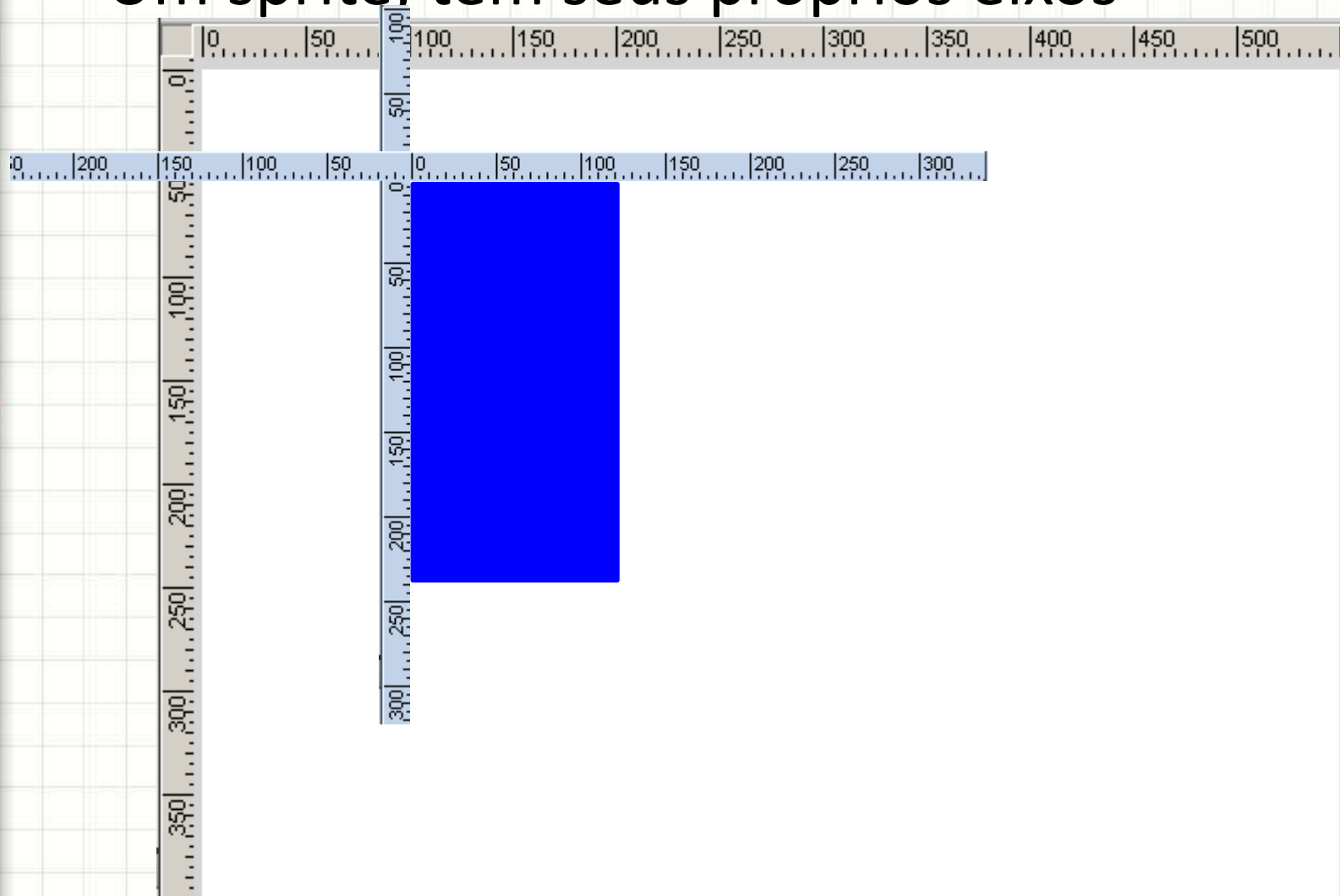
# Coordenadas do AS3

- Eixos de coordenadas do cenário



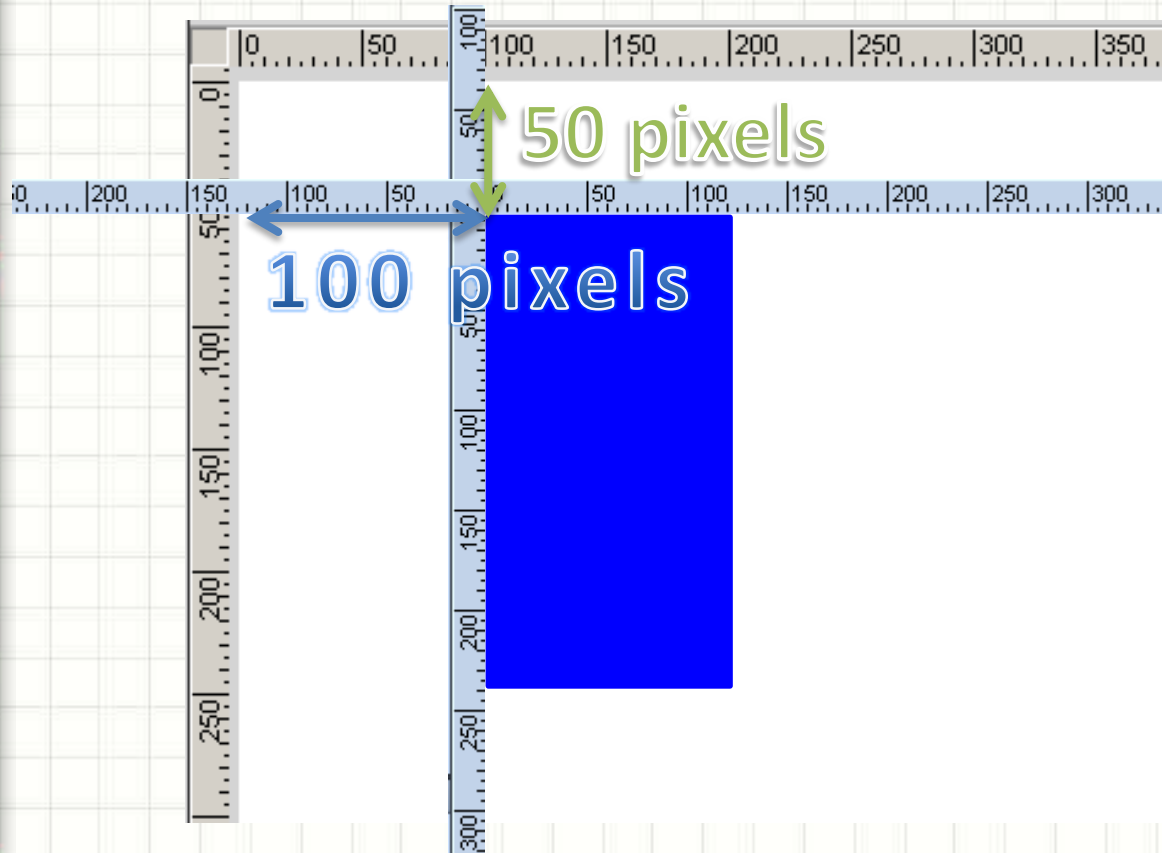
# Coordenadas do AS3

- Um sprite, tem seus próprios eixos



# Coordenadas do AS3

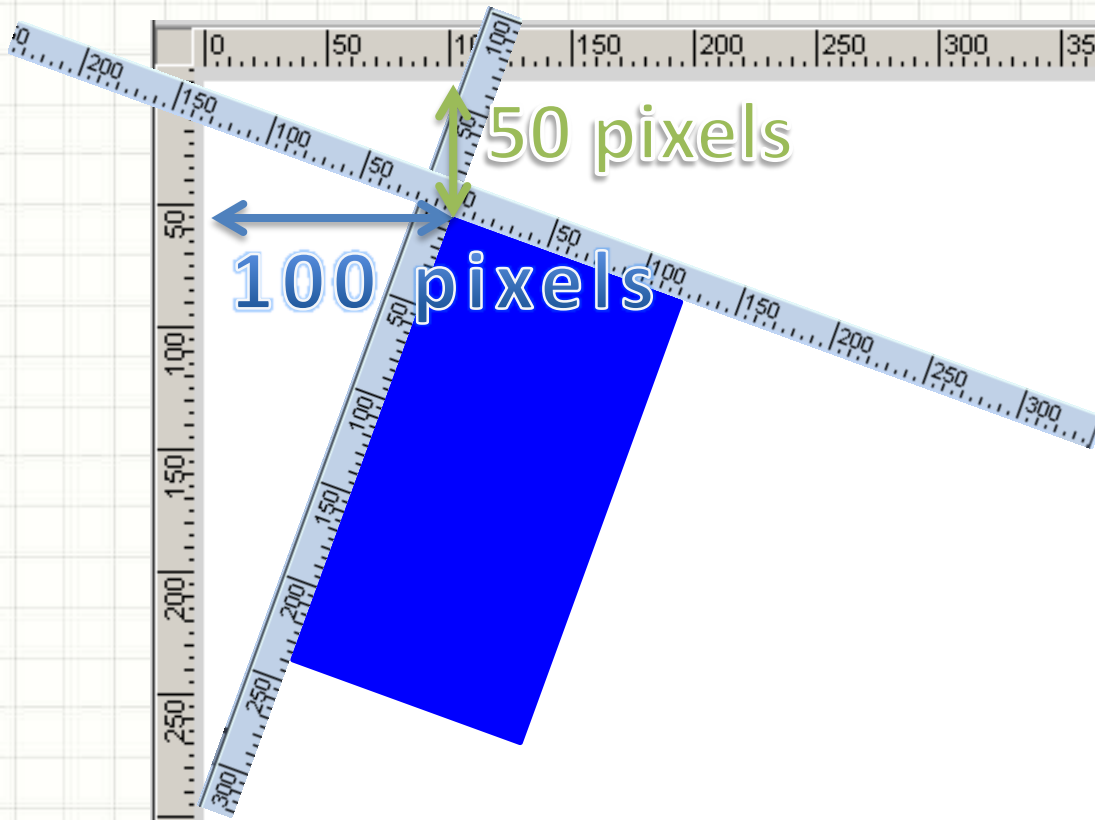
- Quando posicionamos os Sprites, posicionamos o 0,0 do sprite no cenário



```
ret.x = 100;  
ret.y = 50;
```

# Coordenadas do AS3

- Quando o sprite é rodado, ele roda ao redor de seu próprio 0,0

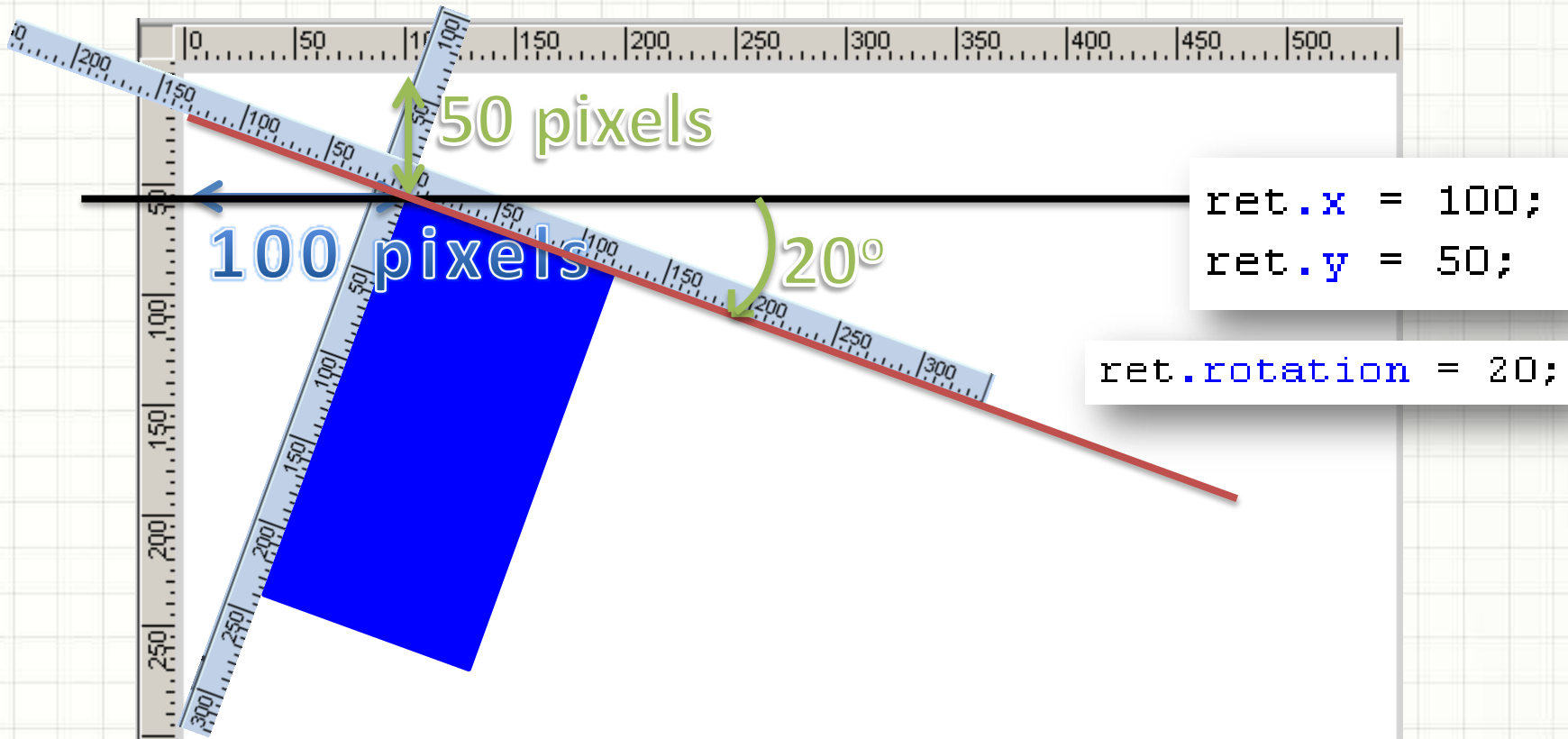


```
ret.x = 100;  
ret.y = 50;
```

```
ret.rotation = 20;
```

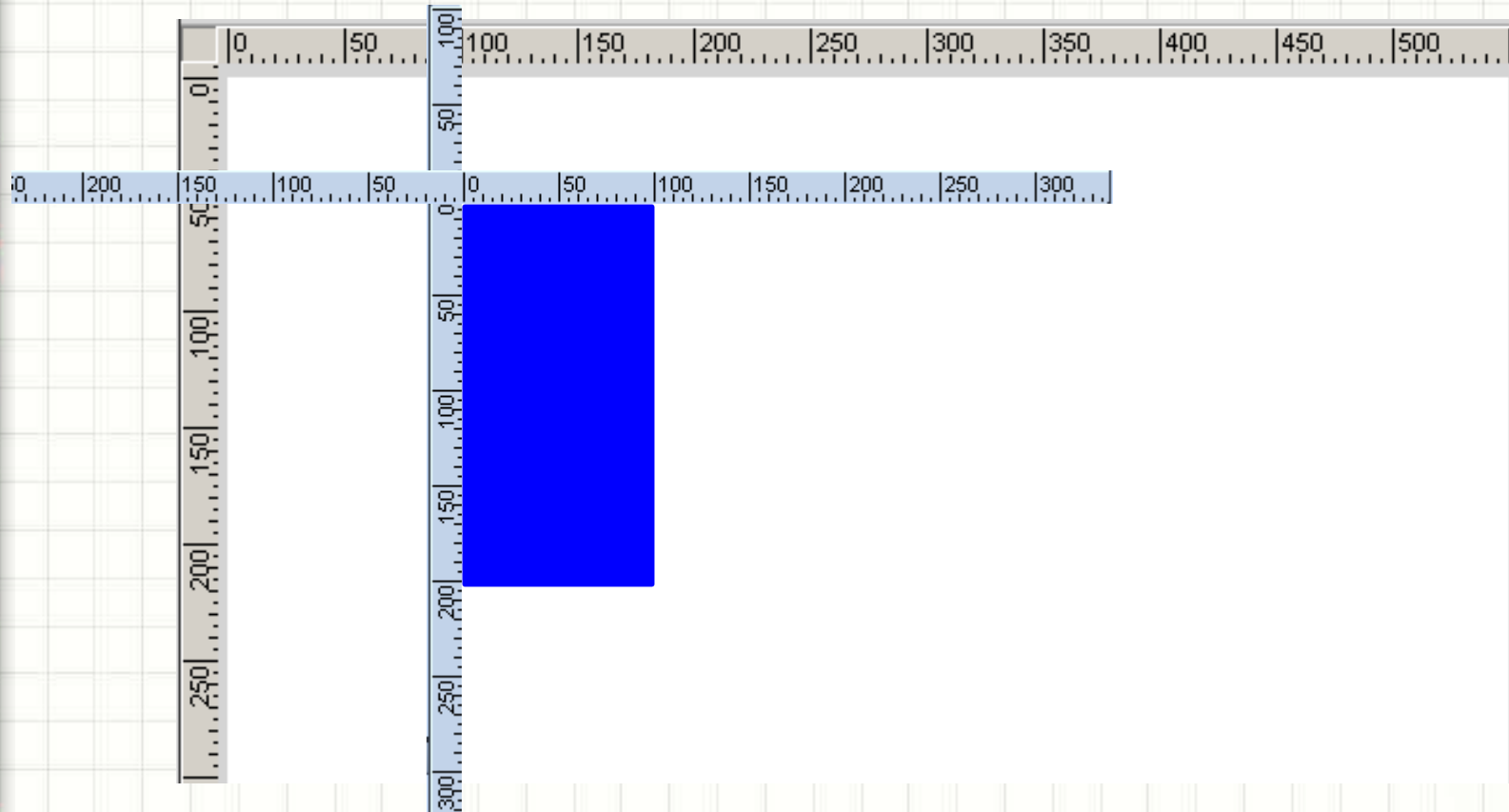
# Coordenadas do AS3

- Quando o sprite é rodado, ele roda ao redor de seu próprio 0,0



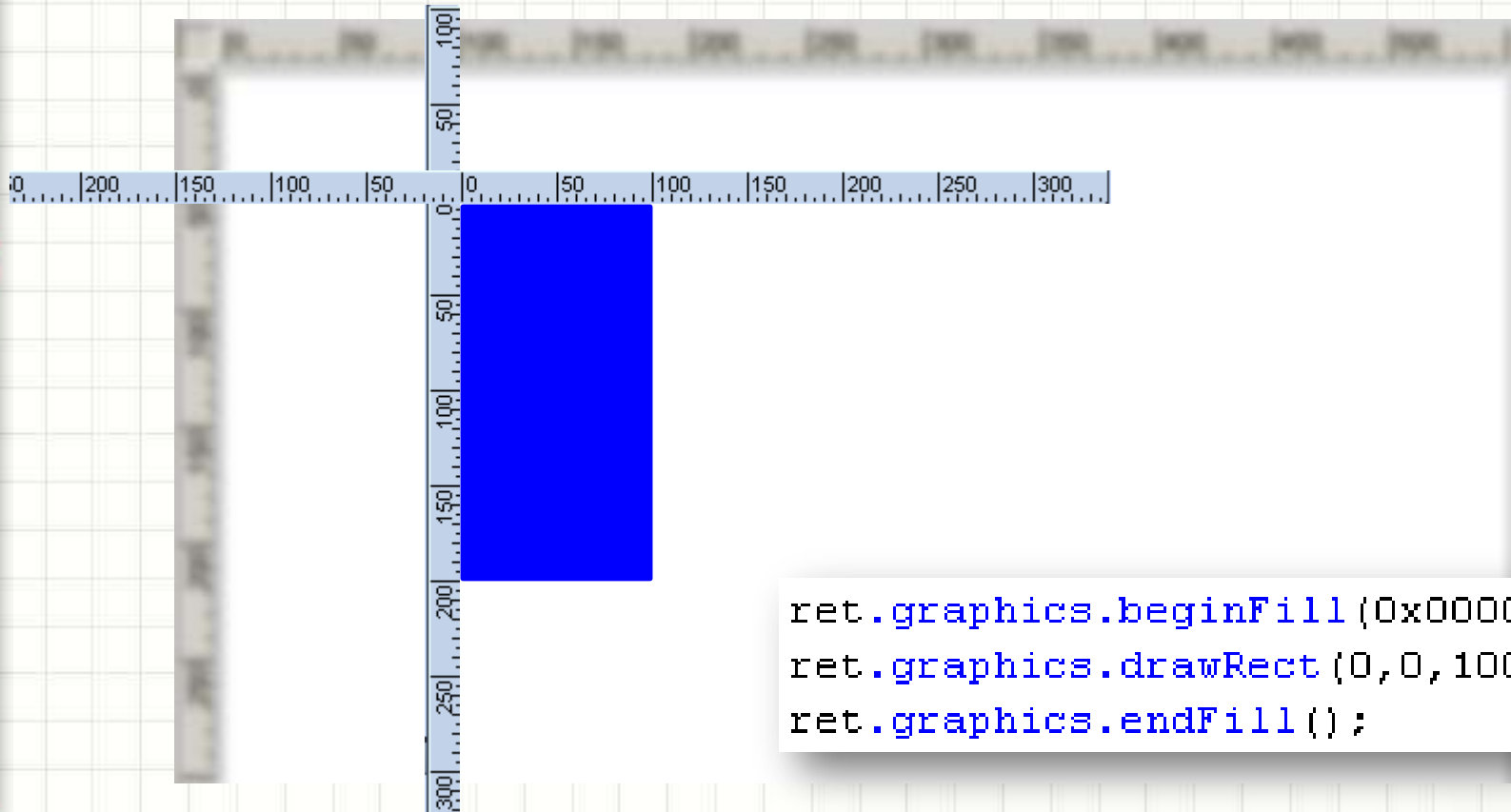
# Coordenadas do AS3

- Para que servem os eixos do sprite?



# Coordenadas do AS3

- Para que servem os eixos do sprite?
- Ao **desenhar** o sprite, são eles que valem!

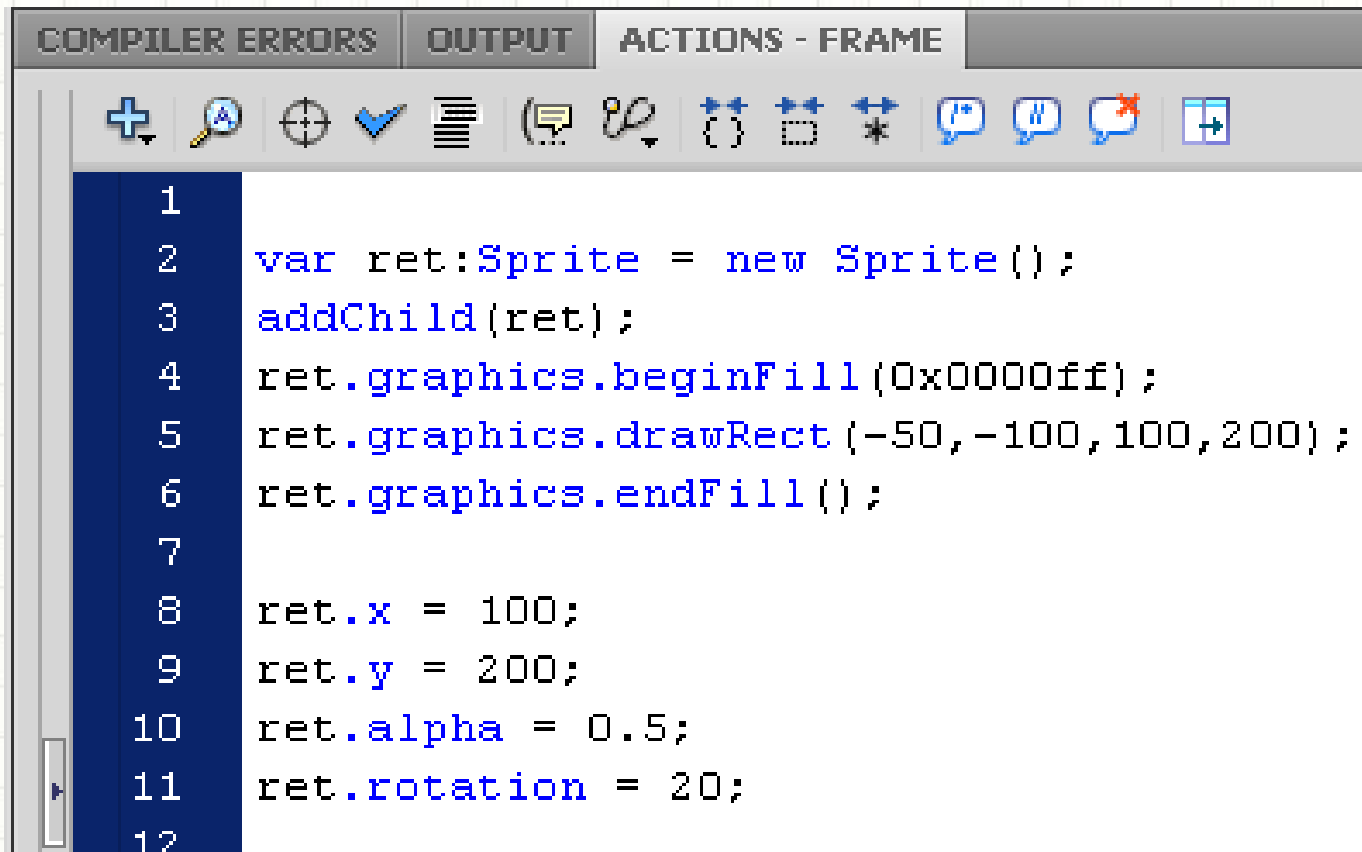


```
ret.graphics.beginFill(0x0000ff);  
ret.graphics.drawRect(0,0,100,200);  
ret.graphics.endFill();
```



# Desenhando em ActionScript 3

- Onde está o centro de rotação agora?



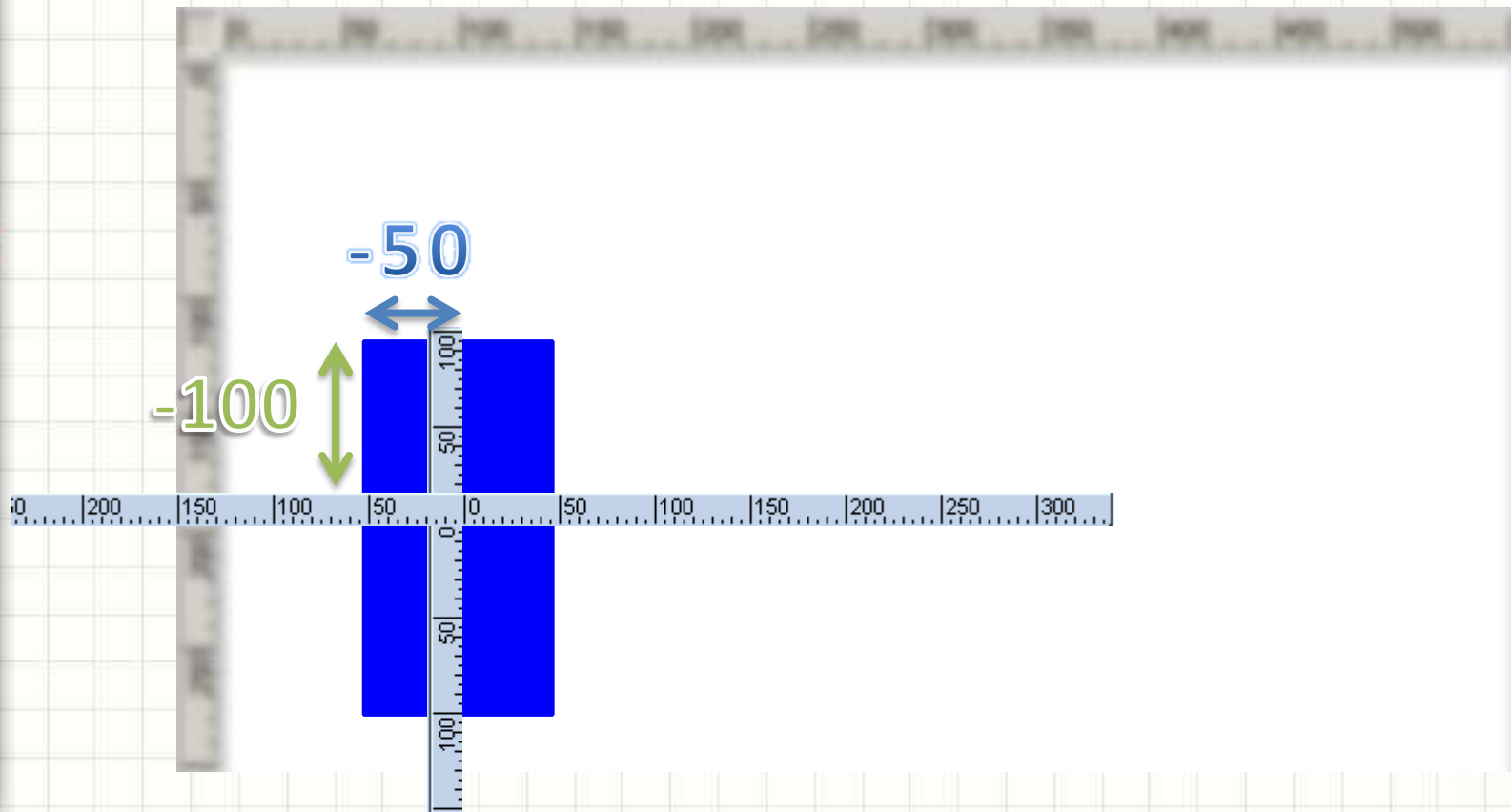
```
1  
2 var ret:Sprite = new Sprite();  
3 addChild(ret);  
4 ret.graphics.beginFill(0x0000ff);  
5 ret.graphics.drawRect(-50,-100,100,200);  
6 ret.graphics.endFill();  
7  
8 ret.x = 100;  
9 ret.y = 200;  
10 ret.alpha = 0.5;  
11 ret.rotation = 20;  
12
```

- Experimente!

# Desenhando em ActionScript 3

- Observe como desenhamos...

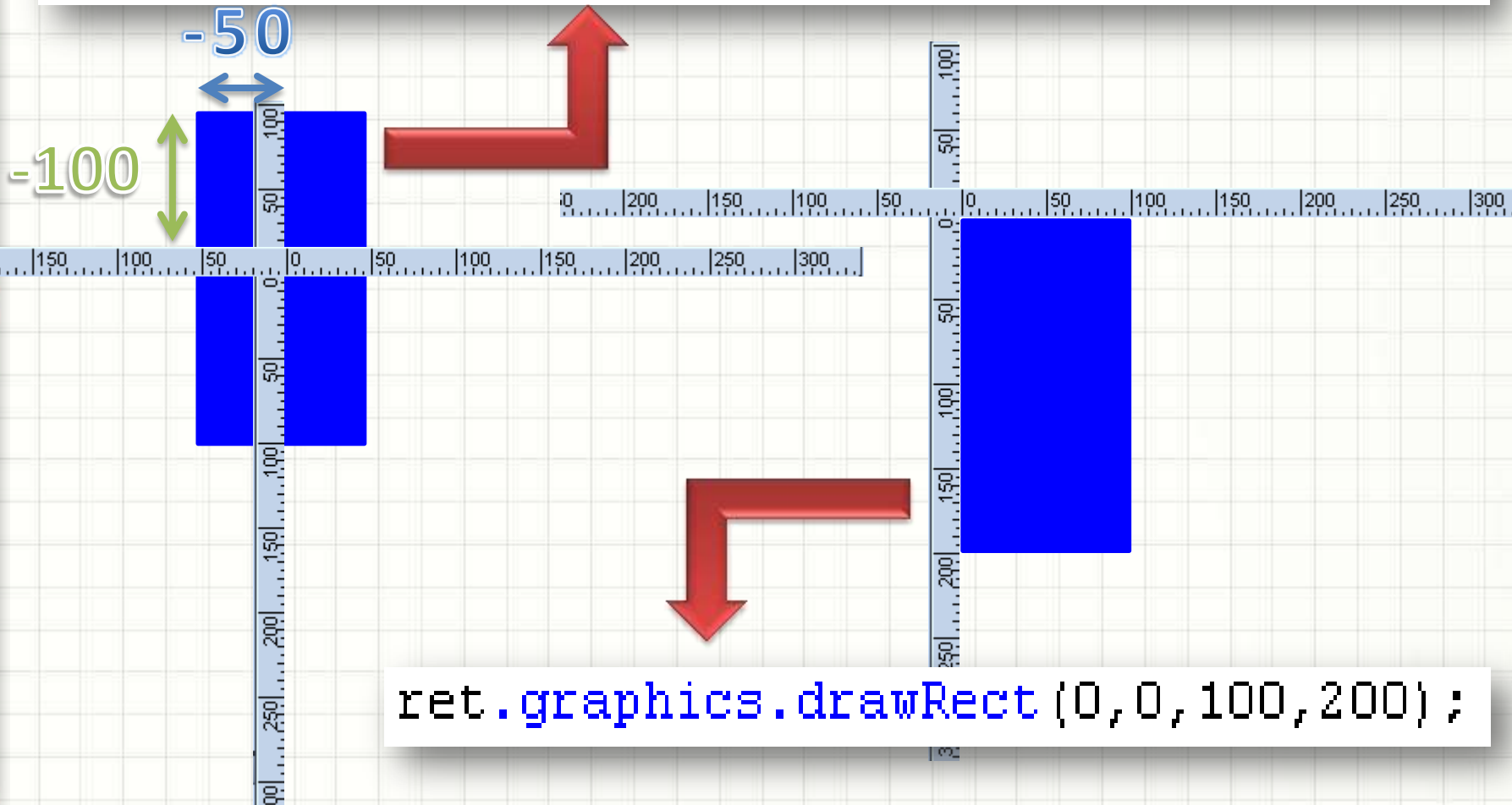
```
ret.graphics.drawRect (-50, -100, 100, 200) ;
```



# Desenhando em ActionScript 3

- Compare

```
ret.graphics.drawRect (-50, -100, 100, 200) ;
```



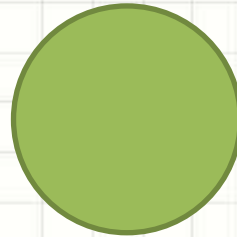
```
ret.graphics.drawRect (0,0,100,200) ;
```

The image features a background with a light gray grid. In the upper left corner, there are several overlapping, wavy lines in shades of blue and white, creating a sense of motion and depth. A prominent, thick blue curve arches across the top of the frame. Below this, a dashed black line follows a similar path, slightly lower and more curved. The overall aesthetic is clean and modern, typical of a corporate or educational presentation slide.

**ATIVIDADE**

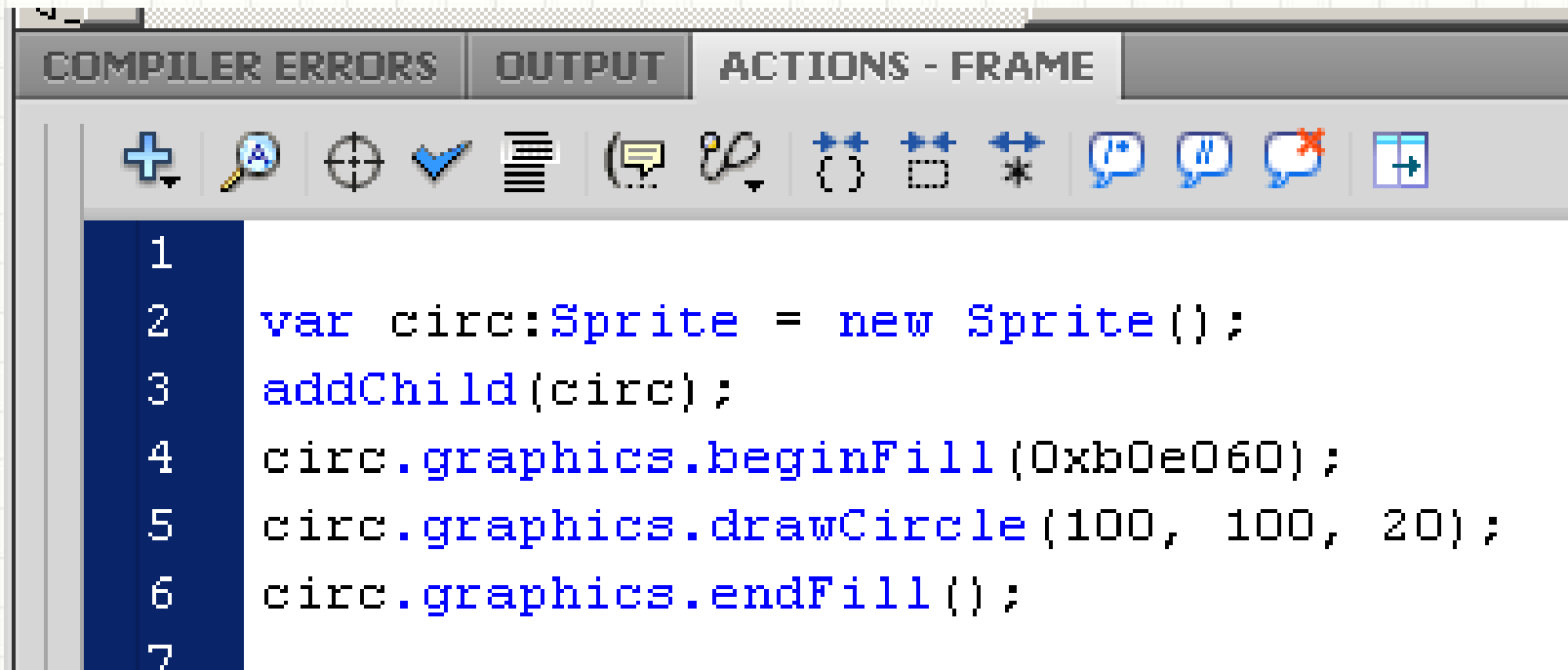
# Atividade

- Faça um programa que mostre 1 círculo verde de raio 20 pixels



# Atividade - Solução

- Faça um programa que mostre 1 círculo verde de raio 20 pixels

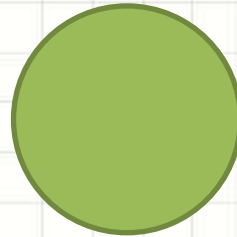


The image shows a screenshot of an IDE window with three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'ACTIONS - FRAME' tab is active. Below the tabs is a toolbar with various icons for editing and development. The main area displays the following code:

```
1  
2 var circ:Sprite = new Sprite();  
3 addChild(circ);  
4 circ.graphics.beginFill(0xb0e060);  
5 circ.graphics.drawCircle(100, 100, 20);  
6 circ.graphics.endFill();  
7
```

# Atividade

- Faça um programa que mostre 1 círculo verde de raio 20 pixels

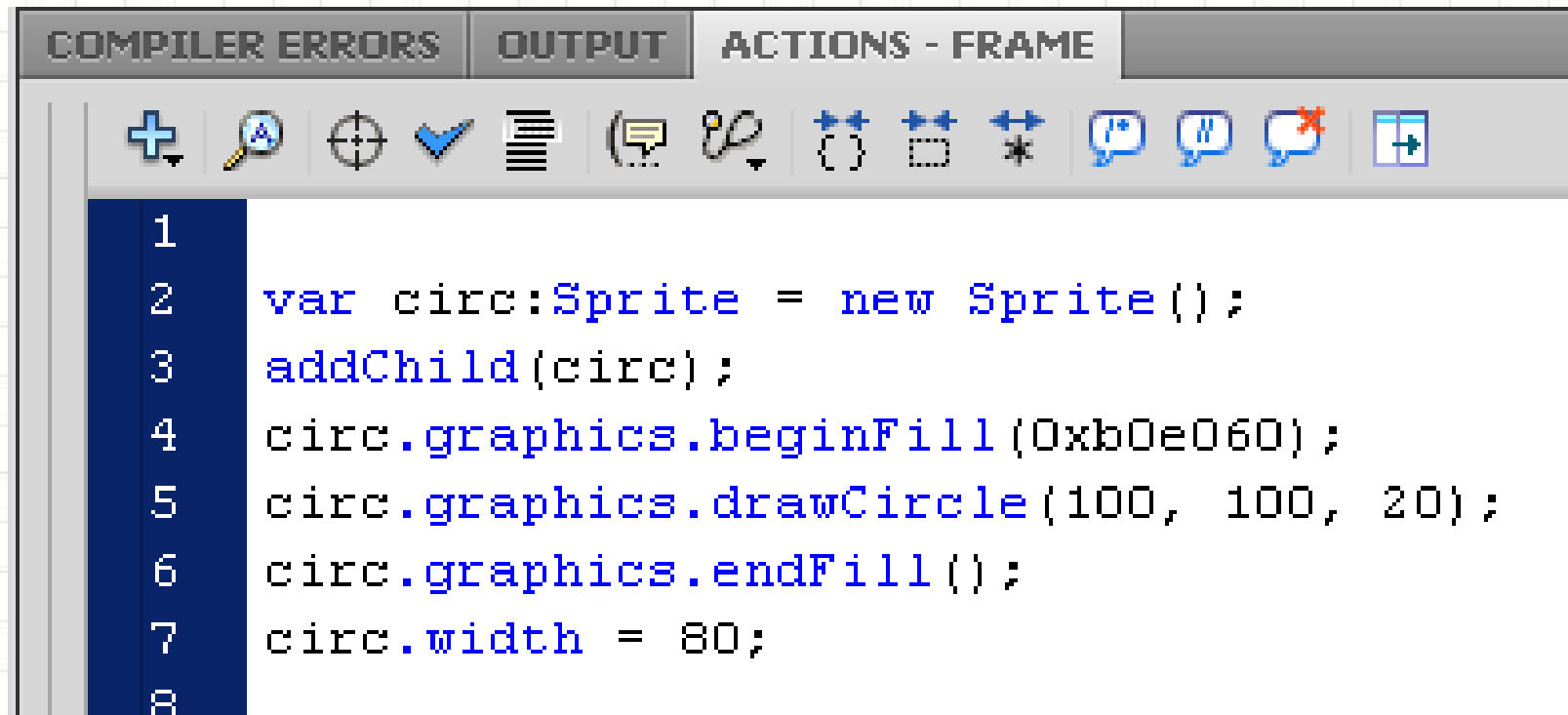


- Experimente alterar os atributos **width** e **height** do círculo... Veja o que acontece!



# Atividade - Solução

- Experimente alterar os atributos **width** e **height** do círculo... Veja o que acontece!

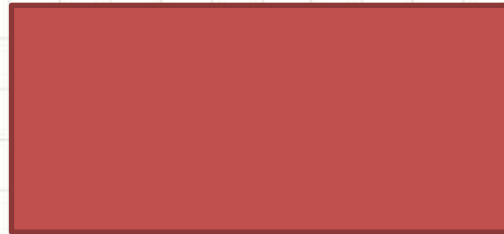


The image shows a screenshot of an IDE window titled "ACTIONS - FRAME". The window has a toolbar with various icons for editing and debugging. The code editor displays the following code:

```
1  
2 var circ:Sprite = new Sprite();  
3 addChild(circ);  
4 circ.graphics.beginFill(0xb0e060);  
5 circ.graphics.drawCircle(100, 100, 20);  
6 circ.graphics.endFill();  
7 circ.width = 80;  
8
```

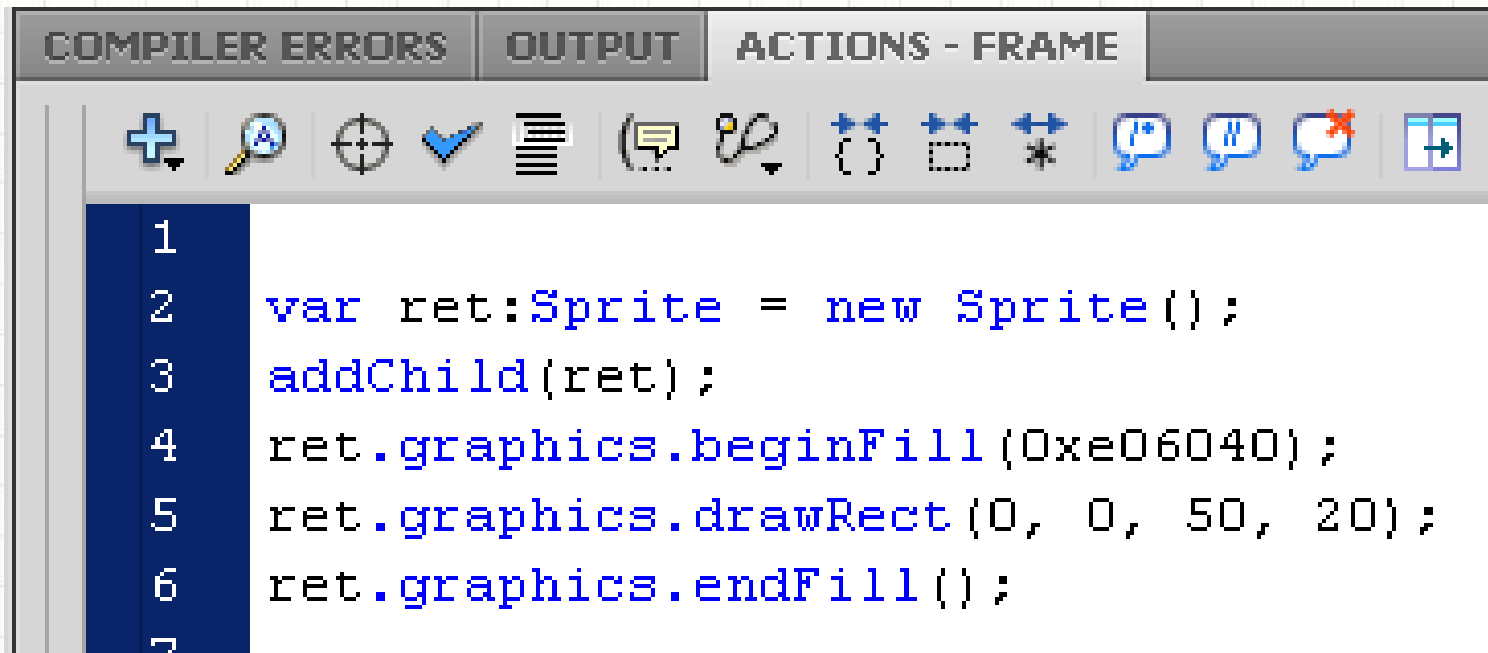
# Atividade

- Faça um programa que mostre 1 retângulo vermelho com 50 x 20 pixels



# Atividade - Solução

- Faça um programa que mostre 1 retângulo vermelho com 50 x 20 pixels

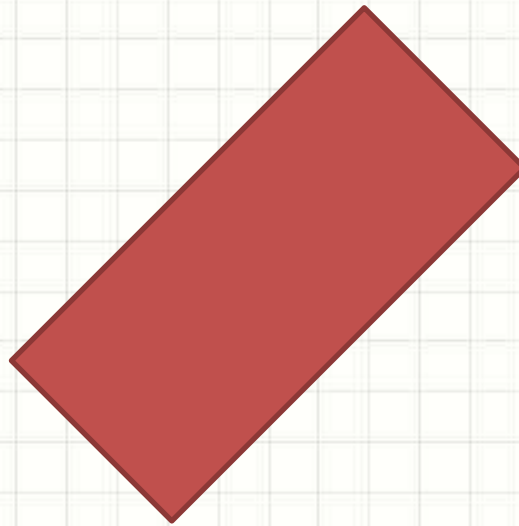


The image shows a screenshot of an IDE's console or output window. The window has three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'OUTPUT' tab is active. Below the tabs is a toolbar with various icons for actions like adding, searching, zooming, and running. The main area of the window displays a code snippet with line numbers 1 through 7. The code is as follows:

```
1  
2 var ret:Sprite = new Sprite();  
3 addChild(ret);  
4 ret.graphics.beginFill(0xe06040);  
5 ret.graphics.drawRect(0, 0, 50, 20);  
6 ret.graphics.endFill();  
7
```

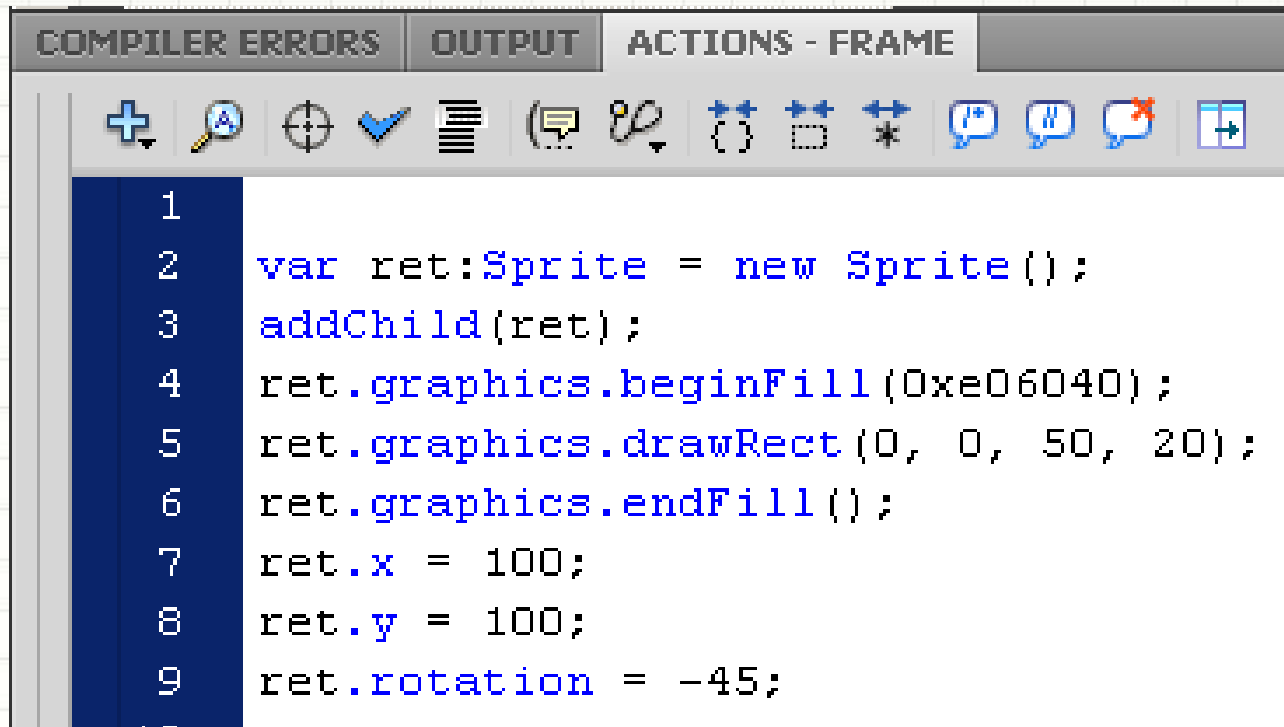
# Atividade

- Faça um programa que mostre 1 retângulo vermelho com 50 x 20 pixels e o incline em 45 graus



# Atividade - Solução

- Faça um programa que mostre 1 retângulo vermelho com 50 x 20 pixels e o incline em 45 graus



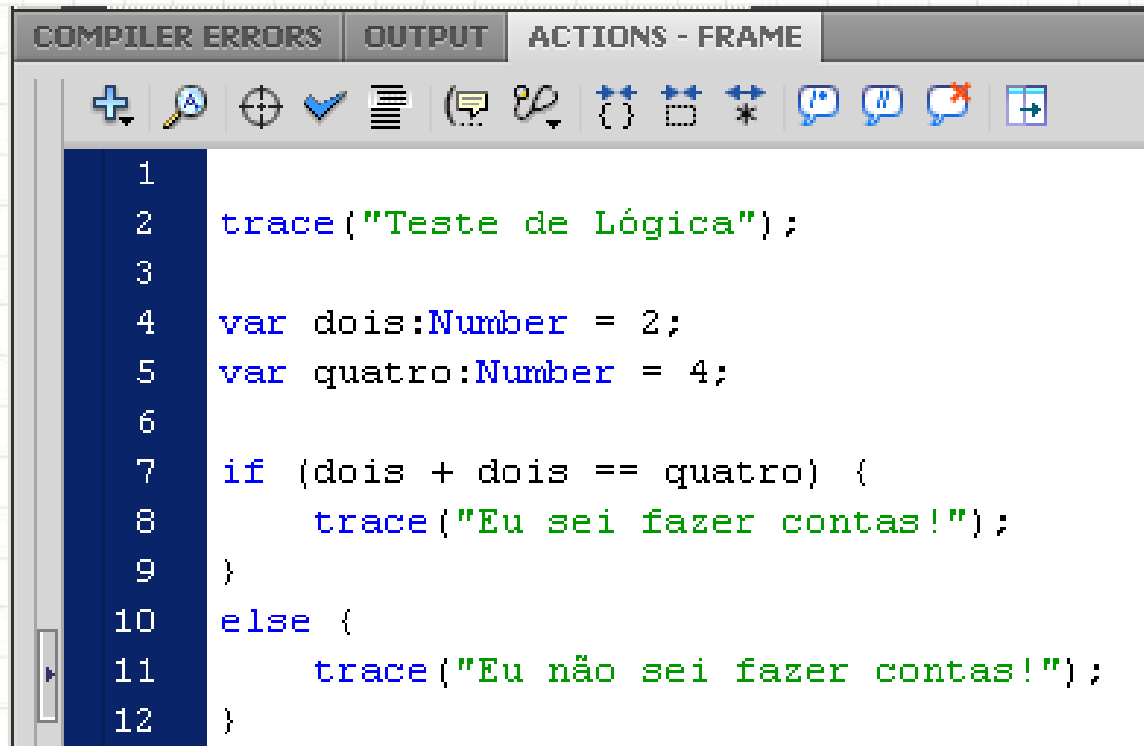
```
COMPILER ERRORS | OUTPUT | ACTIONS - FRAME  
+ 🔍 📏 ✓ ☰ (💬) ✍️ {} ☐ * 💬 💬 ✖️ 📄  
1  
2 var ret:Sprite = new Sprite();  
3 addChild(ret);  
4 ret.graphics.beginFill(0xe06040);  
5 ret.graphics.drawRect(0, 0, 50, 20);  
6 ret.graphics.endFill();  
7 ret.x = 100;  
8 ret.y = 100;  
9 ret.rotation = -45;
```



# LÓGICA E LAÇOS

# Logica e Laços em ActionScript 3

- As estruturas de decisão são idênticas ao C



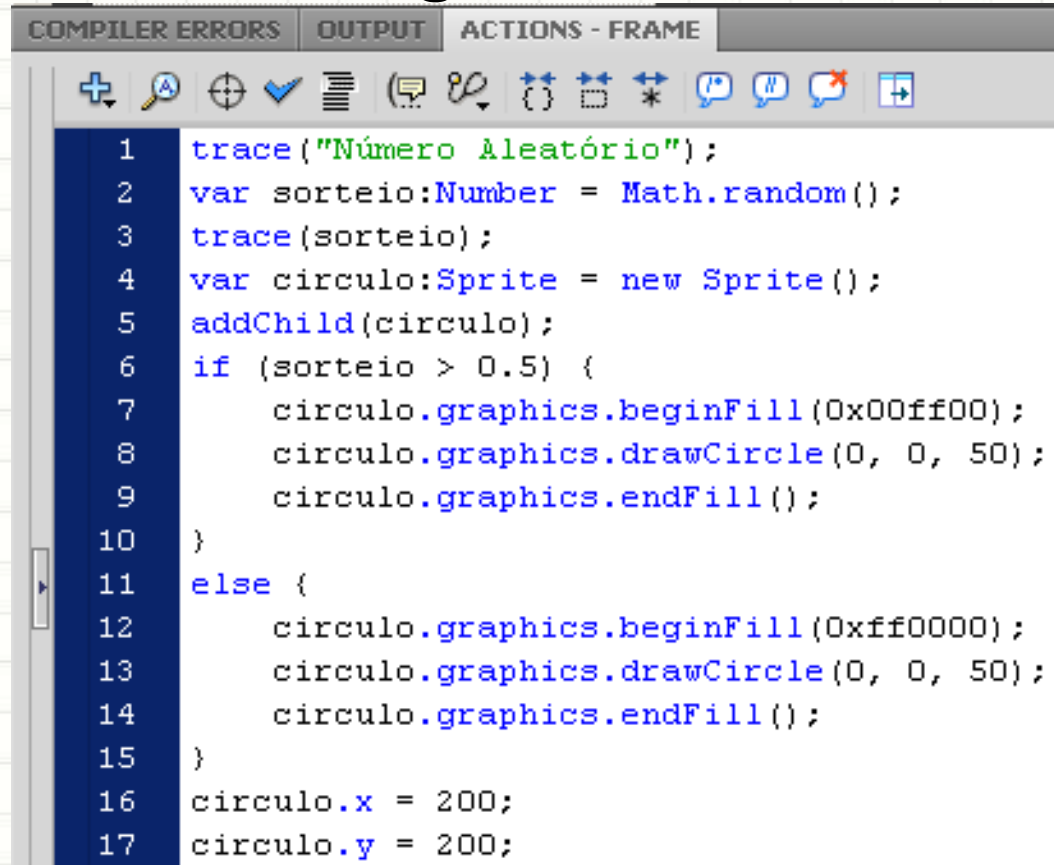
```
COMPILER ERRORS OUTPUT ACTIONS - FRAME
+ 🔍 📏 ✓ ☰ (💬) 🔄 {} 📅 * 💬 💬 💬 ➕
1
2 trace("Teste de Lógica");
3
4 var dois:Number = 2;
5 var quatro:Number = 4;
6
7 if (dois + dois == quatro) {
8     trace("Eu sei fazer contas!");
9 }
10 else {
11     trace("Eu não sei fazer contas!");
12 }
```

- Experimente!



# Logica e Laços em ActionScript 3

- Para testar “if”: gerar números aleatórios?



```
COMPILER ERRORS OUTPUT ACTIONS - FRAME
+ 🔍 📏 ✓ ☰ ☹️ 🔄 🔄 🔄 🗨️ 🗨️ 🗨️ 🗨️
1 trace("Número Aleatório");
2 var sorteio:Number = Math.random();
3 trace(sorteio);
4 var circulo:Sprite = new Sprite();
5 addChild(circulo);
6 if (sorteio > 0.5) {
7     circulo.graphics.beginFill(0x00ff00);
8     circulo.graphics.drawCircle(0, 0, 50);
9     circulo.graphics.endFill();
10 }
11 else {
12     circulo.graphics.beginFill(0xff0000);
13     circulo.graphics.drawCircle(0, 0, 50);
14     circulo.graphics.endFill();
15 }
16 circulo.x = 200;
17 circulo.y = 200;
```

- Execute várias vezes... O que acontece?

COMPILER ERRORS

OUTPUT

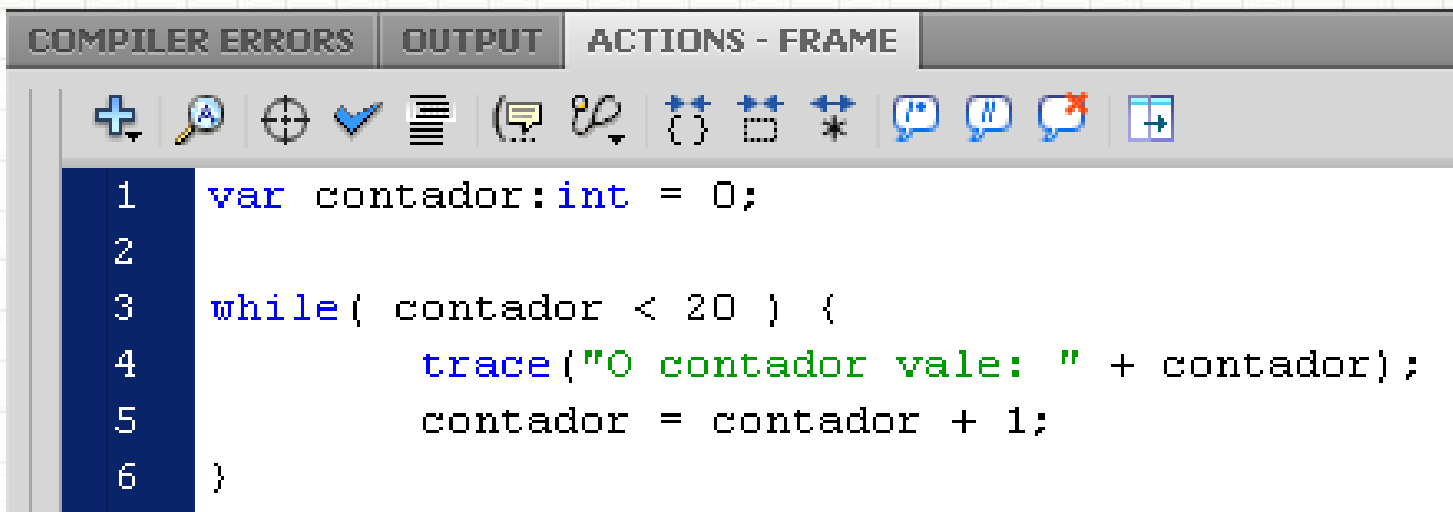
ACTIONS - FRAME



```
1  trace("Número Aleatório");
2  var sorteio:Number = Math.random();
3  trace(sorteio);
4  var circulo:Sprite = new Sprite();
5  addChild(circulo);
6  if (sorteio > 0.5) {
7      circulo.graphics.beginFill(0x00ff00);
8      circulo.graphics.drawCircle(0, 0, 50);
9      circulo.graphics.endFill();
10 }
11 else {
12     circulo.graphics.beginFill(0xff0000);
13     circulo.graphics.drawCircle(0, 0, 50);
14     circulo.graphics.endFill();
15 }
16 circulo.x = 200;
17 circulo.y = 200;
```

# Logica e Laços em ActionScript 3

- Estruturas de repetição são iguais às em C



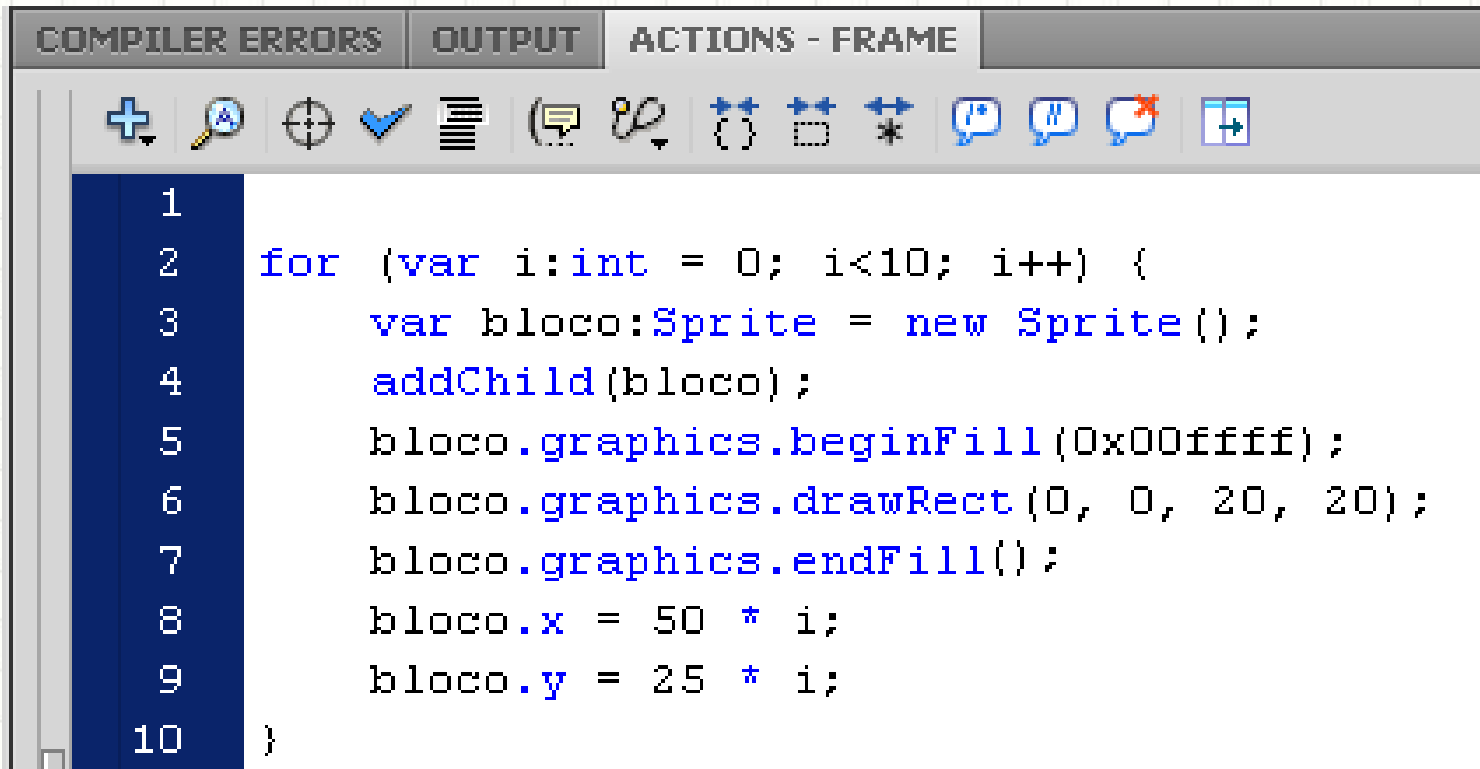
The screenshot shows a code editor window with three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'ACTIONS - FRAME' tab is active, displaying the following ActionScript 3 code:

```
1 var contador:int = 0;
2
3 while( contador < 20 ) {
4     trace("O contador vale: " + contador);
5     contador = contador + 1;
6 }
```

- Experimente!

# Logica e Laços em ActionScript 3

- Usando para desenho...



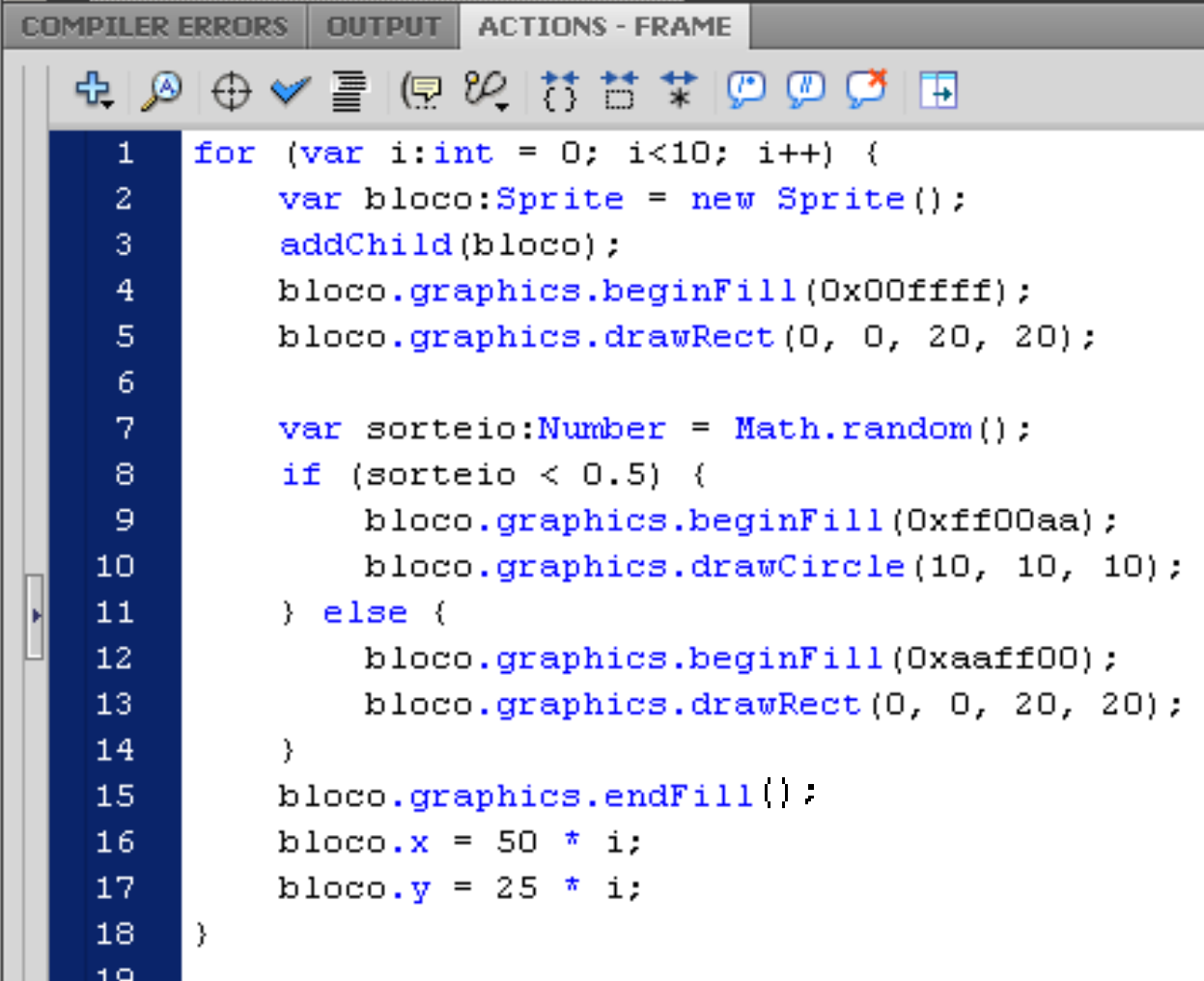
The screenshot shows an IDE window titled "ACTIONS - FRAME" with a toolbar and a code editor. The code editor contains the following ActionScript 3 code:

```
1  
2 for (var i:int = 0; i<10; i++) {  
3     var bloco:Sprite = new Sprite();  
4     addChild(bloco);  
5     bloco.graphics.beginFill(0x00ffff);  
6     bloco.graphics.drawRect(0, 0, 20, 20);  
7     bloco.graphics.endFill();  
8     bloco.x = 50 * i;  
9     bloco.y = 25 * i;  
10 }
```

- Experimente!

# Logica e Laços em ActionScript 3

- Vamos juntar tudo, agora...



```
COMPILER ERRORS OUTPUT ACTIONS - FRAME  
+ 🔍 📏 ✓ ☰ (🗨️) 🔄 ⏪ ⏩ ⚙️ 🗨️ 🗨️ 🗨️ ✖️ 📄  
1 for (var i:int = 0; i<10; i++) {  
2     var bloco:Sprite = new Sprite();  
3     addChild(bloco);  
4     bloco.graphics.beginFill(0x00ffff);  
5     bloco.graphics.drawRect(0, 0, 20, 20);  
6  
7     var sorteio:Number = Math.random();  
8     if (sorteio < 0.5) {  
9         bloco.graphics.beginFill(0xff00aa);  
10        bloco.graphics.drawCircle(10, 10, 10);  
11    } else {  
12        bloco.graphics.beginFill(0xaaff00);  
13        bloco.graphics.drawRect(0, 0, 20, 20);  
14    }  
15    bloco.graphics.endFill();  
16    bloco.x = 50 * i;  
17    bloco.y = 25 * i;  
18 }  
19
```



```
1  for (var i:int = 0; i<10; i++) {
2      var bloco:Sprite = new Sprite();
3      addChild(bloco);
4      bloco.graphics.beginFill(0x00ffff);
5      bloco.graphics.drawRect(0, 0, 20, 20);
6
7      var sorteio:Number = Math.random();
8      if (sorteio < 0.5) {
9          bloco.graphics.beginFill(0xff00aa);
10         bloco.graphics.drawCircle(10, 10, 10);
11     } else {
12         bloco.graphics.beginFill(0xaaaff00);
13         bloco.graphics.drawRect(0, 0, 20, 20);
14     }
15     bloco.graphics.endFill;
16     bloco.x = 50 * i;
17     bloco.y = 25 * i;
18 }
19
```



**ATIVIDADE**

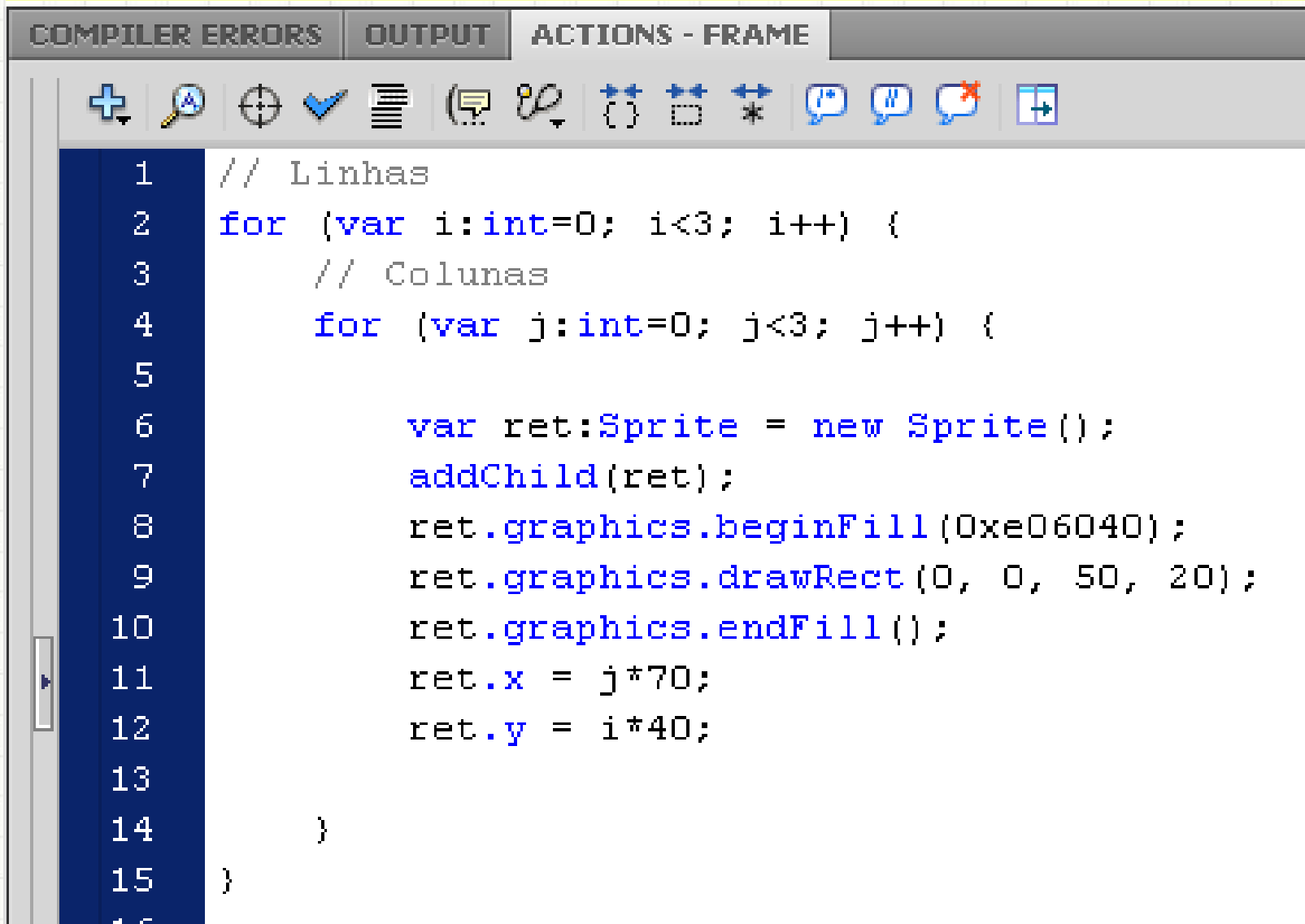


# Atividade

- Faça um programa que mostre 9 retângulos vermelhos 50 x 20 pixels da seguinte forma



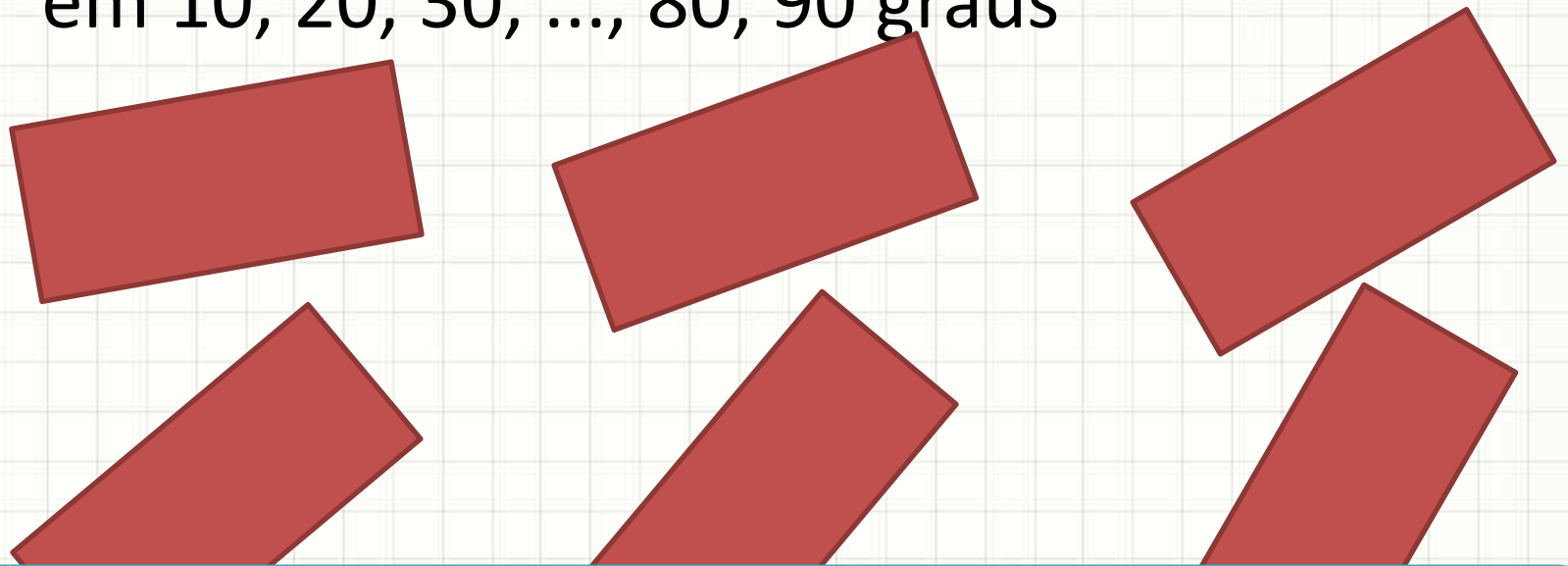
# Atividade - Solução



```
1 // Linhas
2 for (var i:int=0; i<3; i++) {
3     // Colunas
4     for (var j:int=0; j<3; j++) {
5
6         var ret:Sprite = new Sprite();
7         addChild(ret);
8         ret.graphics.beginFill(0xe06040);
9         ret.graphics.drawRect(0, 0, 50, 20);
10        ret.graphics.endFill();
11        ret.x = j*70;
12        ret.y = i*40;
13
14    }
15 }
```

# Atividade

- Faça com que cada retângulo esteja inclinado em 10, 20, 30, ..., 80, 90 graus



Dica: se usar 2 FORs, i de 1 a 3 e j, interno, de 1 a 3, a fórmula da rotação fica:

$$\text{rot} = -10 * (j + 3 * (i - 1))$$

# Atividade - Solução

lo

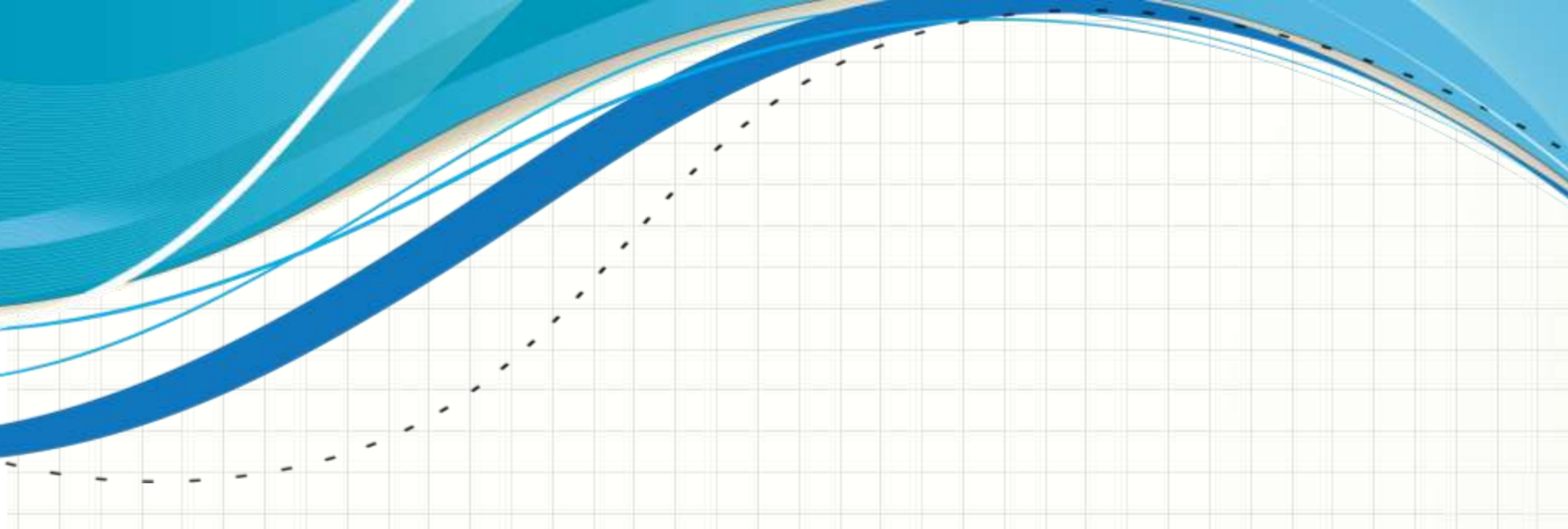
COMPILER ERRORS

OUTPUT

ACTIONS - FRAME



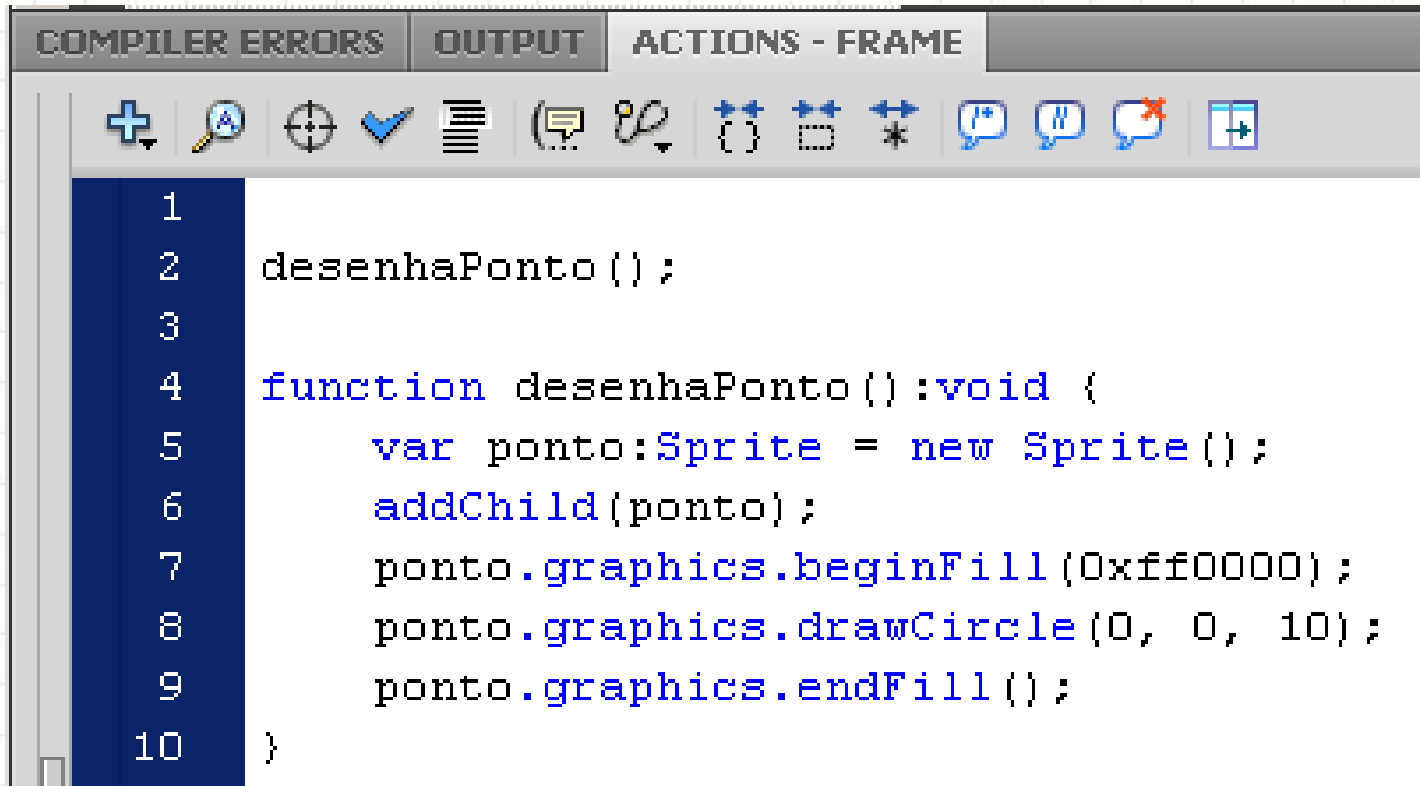
```
1 // Linhas
2 for (var i:int=1; i<=3; i++) {
3     // Colunas
4     for (var j:int=1; j<=3; j++) {
5
6         var ret:Sprite = new Sprite();
7         addChild(ret);
8         ret.graphics.beginFill(0xe06040);
9         ret.graphics.drawRect(0, 0, 50, 20);
10        ret.graphics.endFill();
11        ret.x = j*70;
12        ret.y = i*40;
13        ret.rotation = -10*(j + 3*(i-1));
14
15    }
16 }
```



# **FUNÇÕES EM ACTIONSCRIPT 3**

# Funções em ActionScript 3

- As funções são parecidas com C



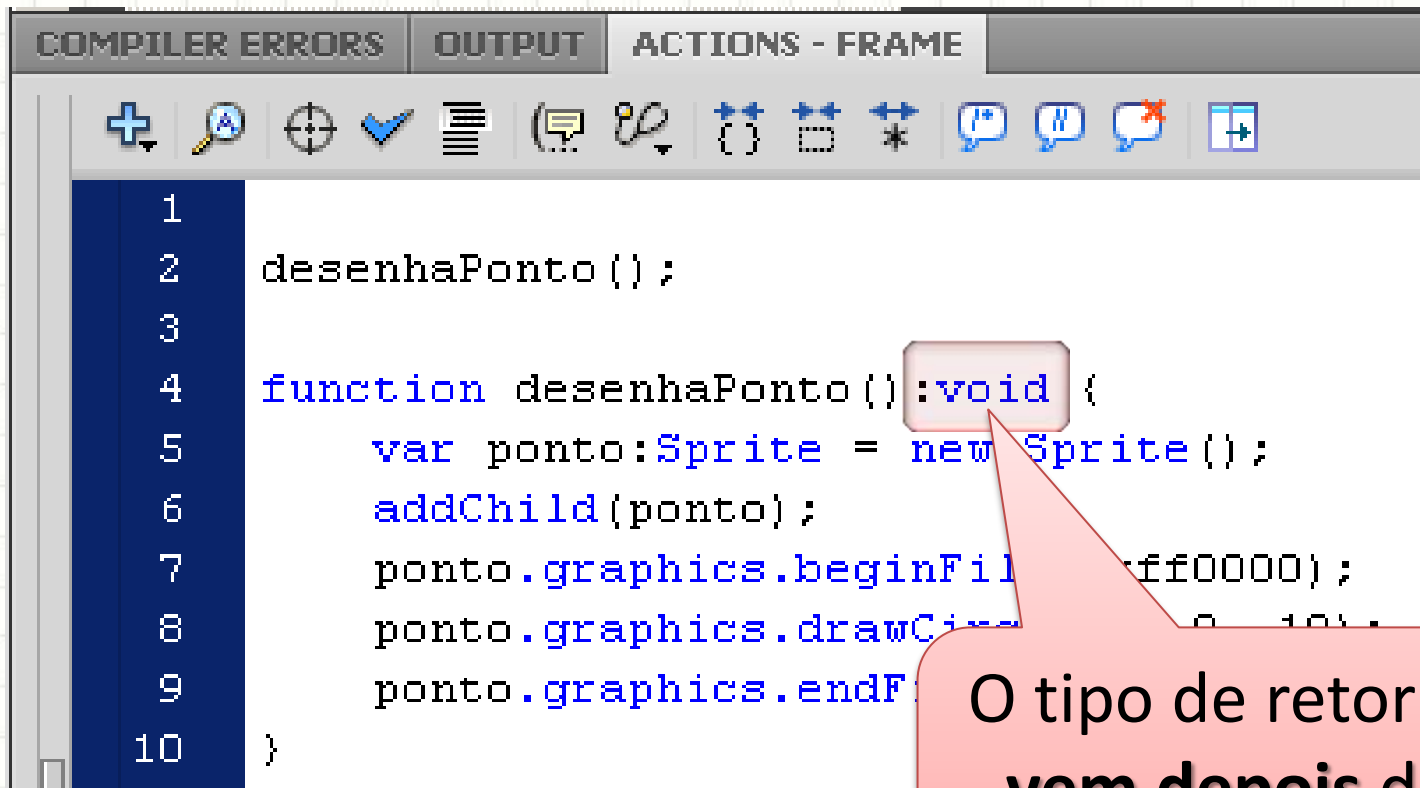
The screenshot shows an IDE window titled 'ACTIONS - FRAME'. The window has three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. Below the tabs is a toolbar with various icons for editing and development. The main area displays the following code:

```
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



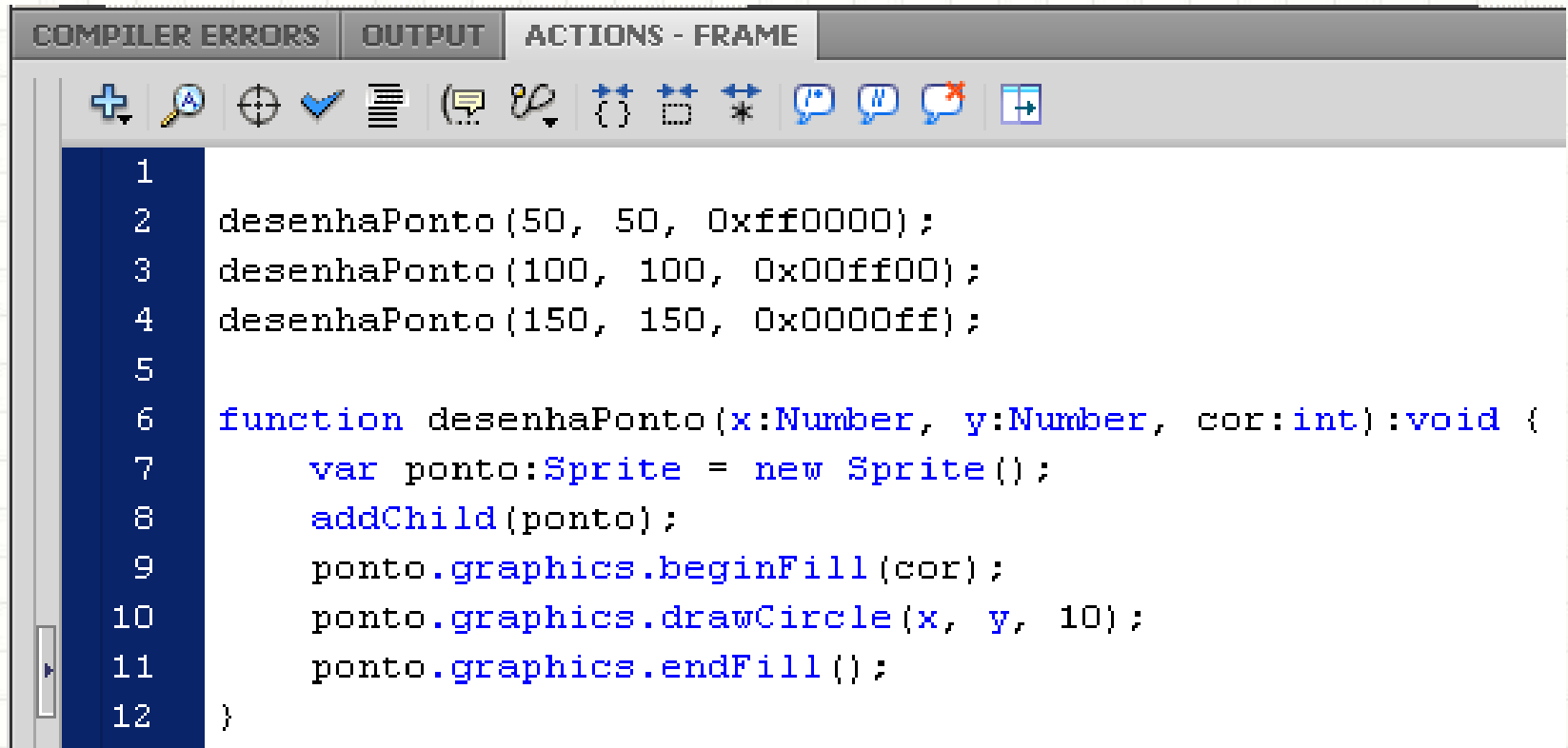
```
1  
2 desenhaPonto();  
3  
4 function desenhaPonto():void {  
5     var ponto:Sprite = new Sprite();  
6     addChild(ponto);  
7     ponto.graphics.beginFill(0xff0000);  
8     ponto.graphics.drawCircle(0, 10);  
9     ponto.graphics.endFill();  
10 }
```

- Experimente!



# Funções em ActionScript 3

- Função com parâmetros são mais úteis:



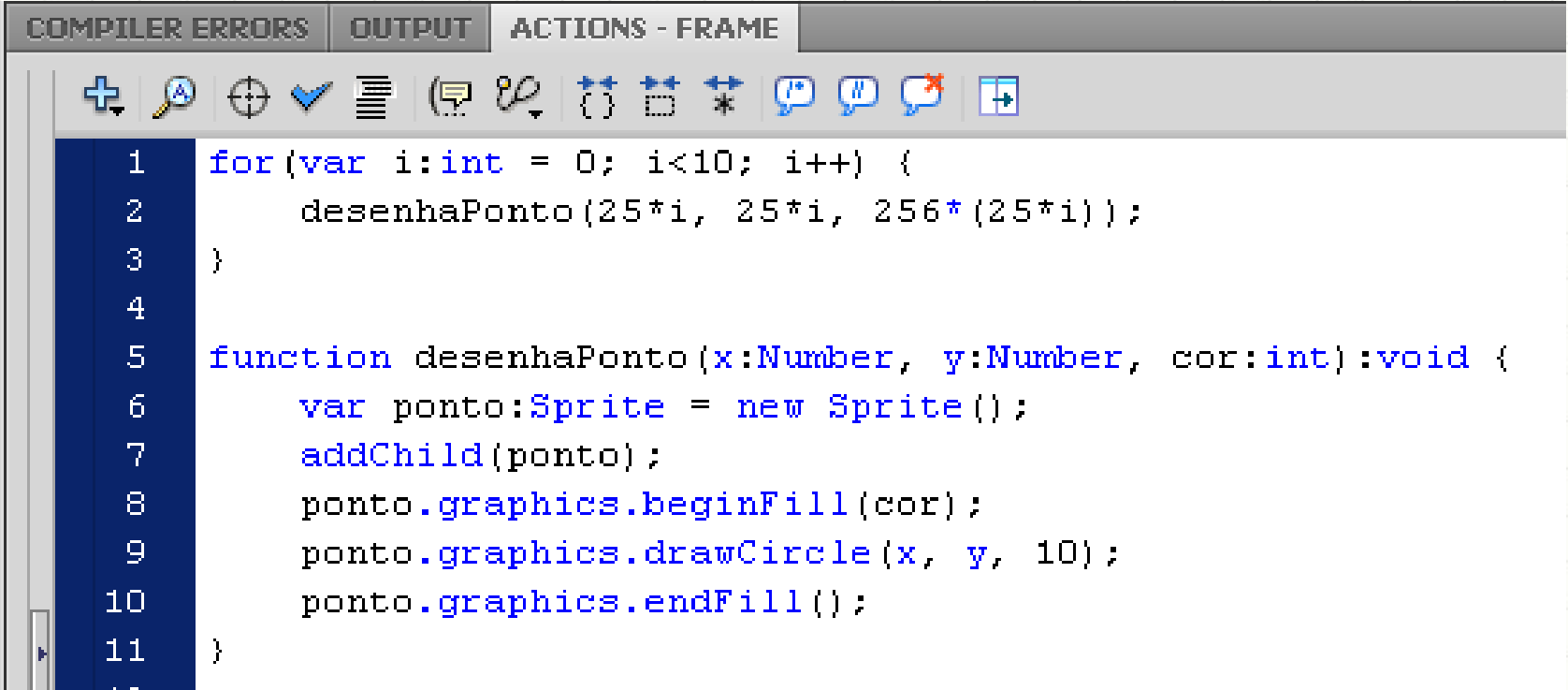
The screenshot shows an IDE window titled 'ACTIONS - FRAME'. The code editor contains the following ActionScript 3 code:

```
1  
2  desenhaPonto(50, 50, 0xff0000);  
3  desenhaPonto(100, 100, 0x00ff00);  
4  desenhaPonto(150, 150, 0x0000ff);  
5  
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 an IDE window titled 'ACTIONS - FRAME'. The code is as follows:

```
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 }
```

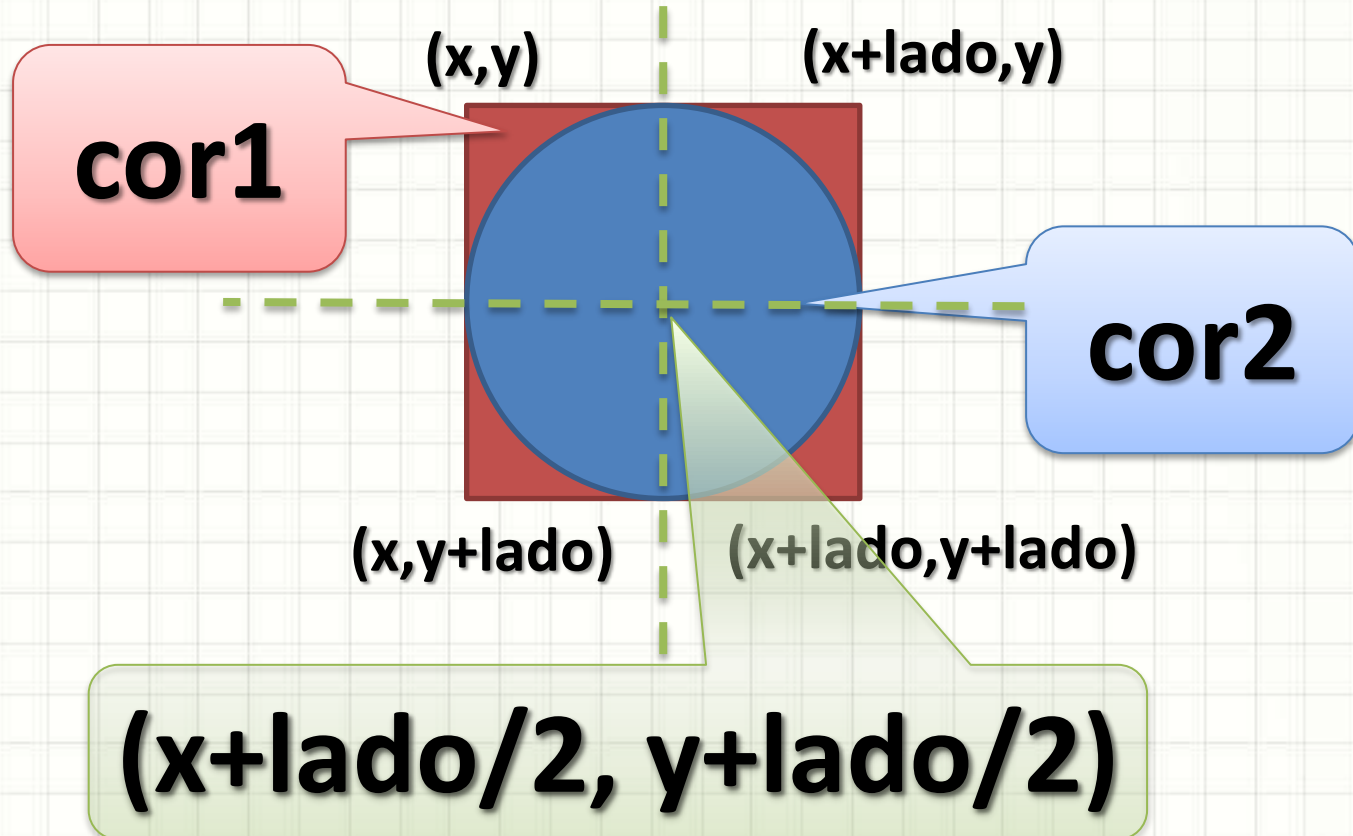
- Experimente!

The image features a background with a light gray grid. In the upper left corner, there are several overlapping, wavy lines in shades of blue and white, creating a sense of motion and depth. A prominent, thick blue curve arches across the top of the page. Below this, a dashed black line follows a similar path, curving upwards and then downwards. The word "ATIVIDADE" is positioned in the lower right quadrant of the page.

**ATIVIDADE**

# Atividade

- Faça uma função que receba  $x$ ,  $y$ ,  $lado$ ,  $cor1$  e  $cor2$  e desenhe:



# Atividade - Solução

- Faça uma função que receba x, y, lado, cor1 e cor2 e desenhe...

```
COMPILER ERRORS OUTPUT ACTIONS - FRAME
+ 🔍 📏 ✓ 📄 🗨️ 📌 📁 📌 📌 🗨️ 🗨️ 🗨️ 🗨️
1
2 desenha(100, 100, 50, 0xe04000, 0x0080e0);
3
4 function desenha(x:Number, y:Number, l:Number, c1:int, c2:int):void {
5     var fig:Sprite = new Sprite();
6     addChild(fig);
7     fig.graphics.beginFill(c1);
8     fig.graphics.drawRect(0, 0, l, l);
9     fig.graphics.beginFill(c2);
10    fig.graphics.drawCircle(l/2, l/2, l/2);
11    fig.graphics.endFill();
12    fig.x = x;
13    fig.y = y;
14 }
15
```

# Atividade - Solução

ERRORS OUTPUT ACTIONS - FRAME

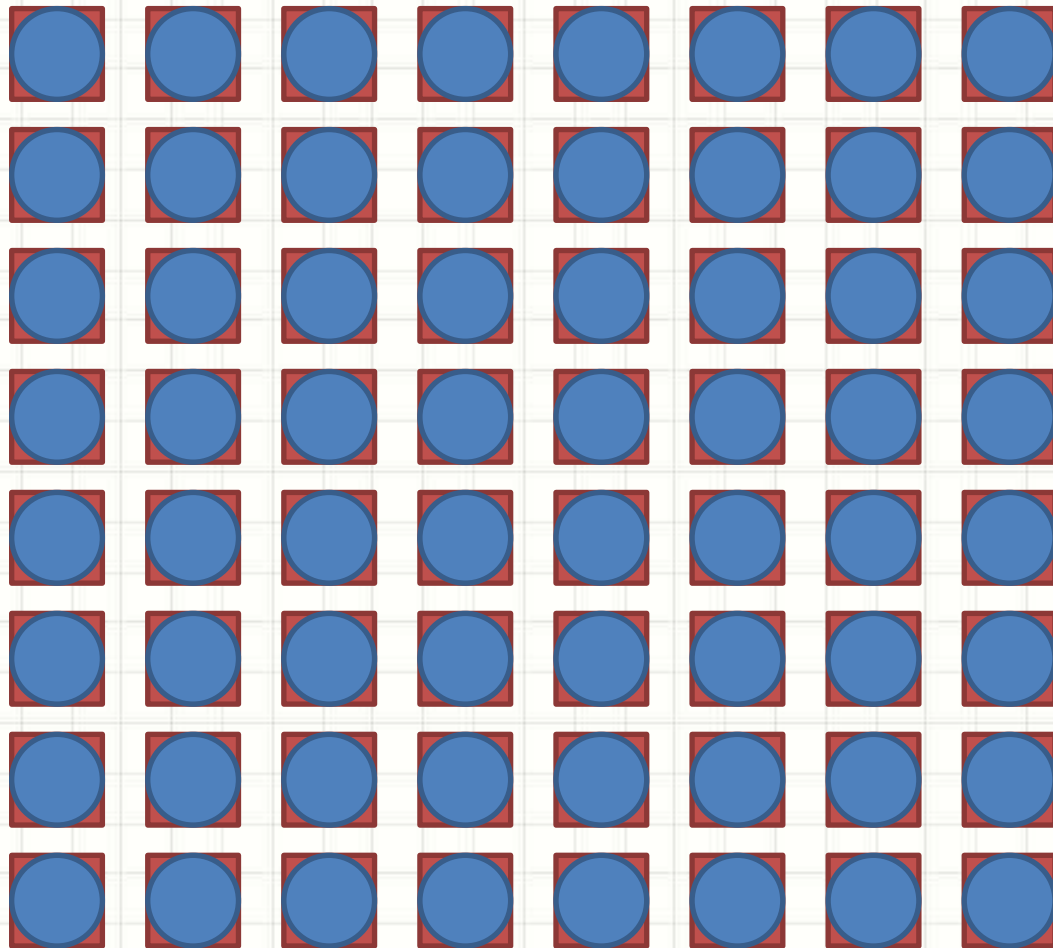


```
desenha(100,100,50,0xe04000,0x0080e0);
```

```
function desenha(x:Number, y:Number, l:Number, c1:int, c2:int):void {  
    var fig:Sprite = new Sprite();  
    addChild(fig);  
    fig.graphics.beginFill(c1);  
    fig.graphics.drawRect(0, 0, 1, 1);  
    fig.graphics.beginFill(c2);  
    fig.graphics.drawCircle(1/2, 1/2, 1/2);  
    fig.graphics.endFill();  
    fig.x = x;  
    fig.y = y;  
}
```

# Atividade

- Use a função, com dois “for”, para desenhar:



# Atividade - Solução

- Use a função, com dois “for”, para desenhar...

```
COMPILER ERRORS OUTPUT ACTIONS - FRAME
+ 🔍 📏 ✓ ☰ (💬) 🔄 {} 🌐 * 💬 💬 ✖️ 📄
1 for (var i:int=1; i<=8; i++) {
2     for (var j:int=1; j<=8; j++) {
3         desenha(j*30,i*30,20,0xe04000,0x0080e0);
4     }
5 }
6
7 function desenha(x:Number, y:Number, l:Number, c1:int, c2:int):void {
8     var fig:Sprite = new Sprite();
9     addChild(fig);
10    fig.graphics.beginFill(c1);
11    fig.graphics.drawRect(0, 0, 1, 1);
12    fig.graphics.beginFill(c2);
13    fig.graphics.drawCircle(1/2, 1/2, 1/2);
14    fig.graphics.endFill();
15    fig.x = x;
16    fig.y = y;
17 }
18
```



# Atividade - Solução

ERRORS OUTPUT ACTIONS - FRAME



```
for (var i:int=1; i<=8; i++) {  
    for (var j:int=1; j<=8; j++) {  
        desenha(j*30,i*30,20,0xe04000,0x0080e0);  
    }  
}  
  
function desenha(x:Number, y:Number, l:Number, c1:int, c2:int):void {  
    var fig:Sprite = new Sprite();  
    addChild(fig);  
    fig.graphics.beginFill(c1);  
    fig.graphics.drawRect(0, 0, 1, 1);  
    fig.graphics.beginFill(c2);  
    fig.graphics.drawCircle(1/2, 1/2, 1/2);  
    fig.graphics.endFill();  
    fig.x = x;  
    fig.y = y;  
}
```



**CONCLUSÕES**

# Resumo

- ActionScript 3: linguagem similar ao JScript/C
- Grande facilidade: criar formas geométricas
- Interface do Flash: prática
  - Desenvolvimento de aplicações AS3
  
- **TAREFA**
  - Trabalho B!



**PERGUNTAS?**

# Próxima Aula



- AS3 parece legal...
  - Mas como interagir com o programa?
  - Como fazer animações?



**BOM DESCANSO  
A TODOS!**