



PROGRAMAÇÃO PARA INTERNET RICA

INTRODUÇÃO AO ACTIONSCRIPT 3

Prof. Dr. Daniel Caetano

2013 - 1

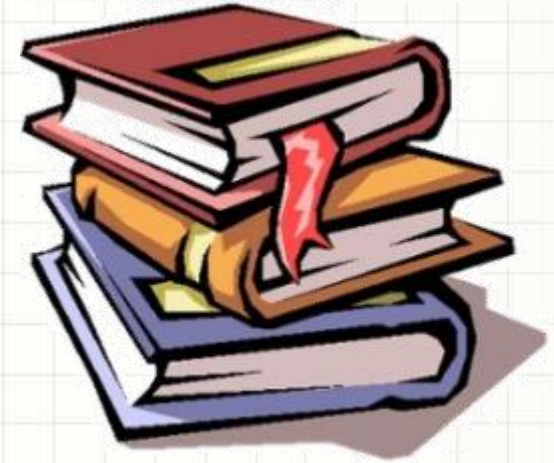
Objetivos

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

- **Trabalho B!**



Material de Estudo



Material

Acesso ao Material

Tutorial Extra

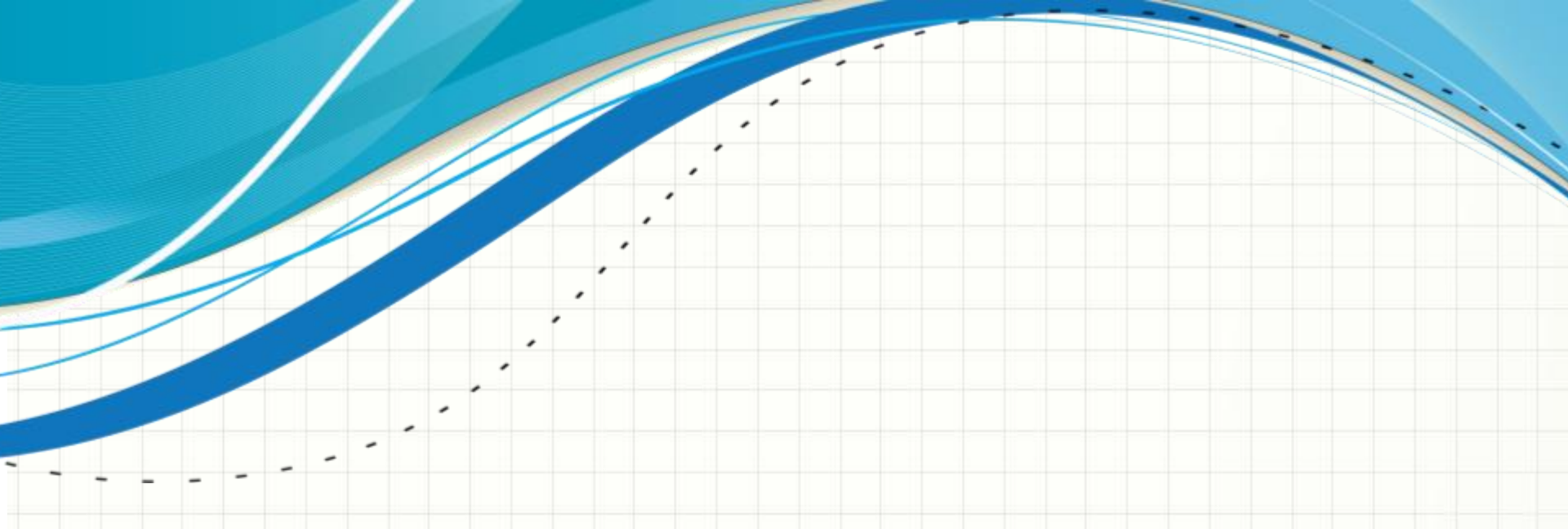
<http://www.caetano.eng.br/>
Adobe Flash CS4 (Material de Apoio)

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



The image shows a screenshot of a web form titled "Contact Information". The form has a yellow header bar with the title. Below the header, there are two input fields for "Name * First *" and "Last *". A horizontal line separates this from the next section, which contains the text "Flash fills in field in automatically. You can replace the text." followed by an "email" input field with the placeholder ".@mm.com". Another horizontal line follows, leading to "Phone" and "Date" input fields. The "Date" field has a small calendar icon on its right side. At the bottom of the form, there are two buttons: "Show Results" and "Reset Fields".

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...



ADOBE® FLASH® CS4 PROFESSIONAL

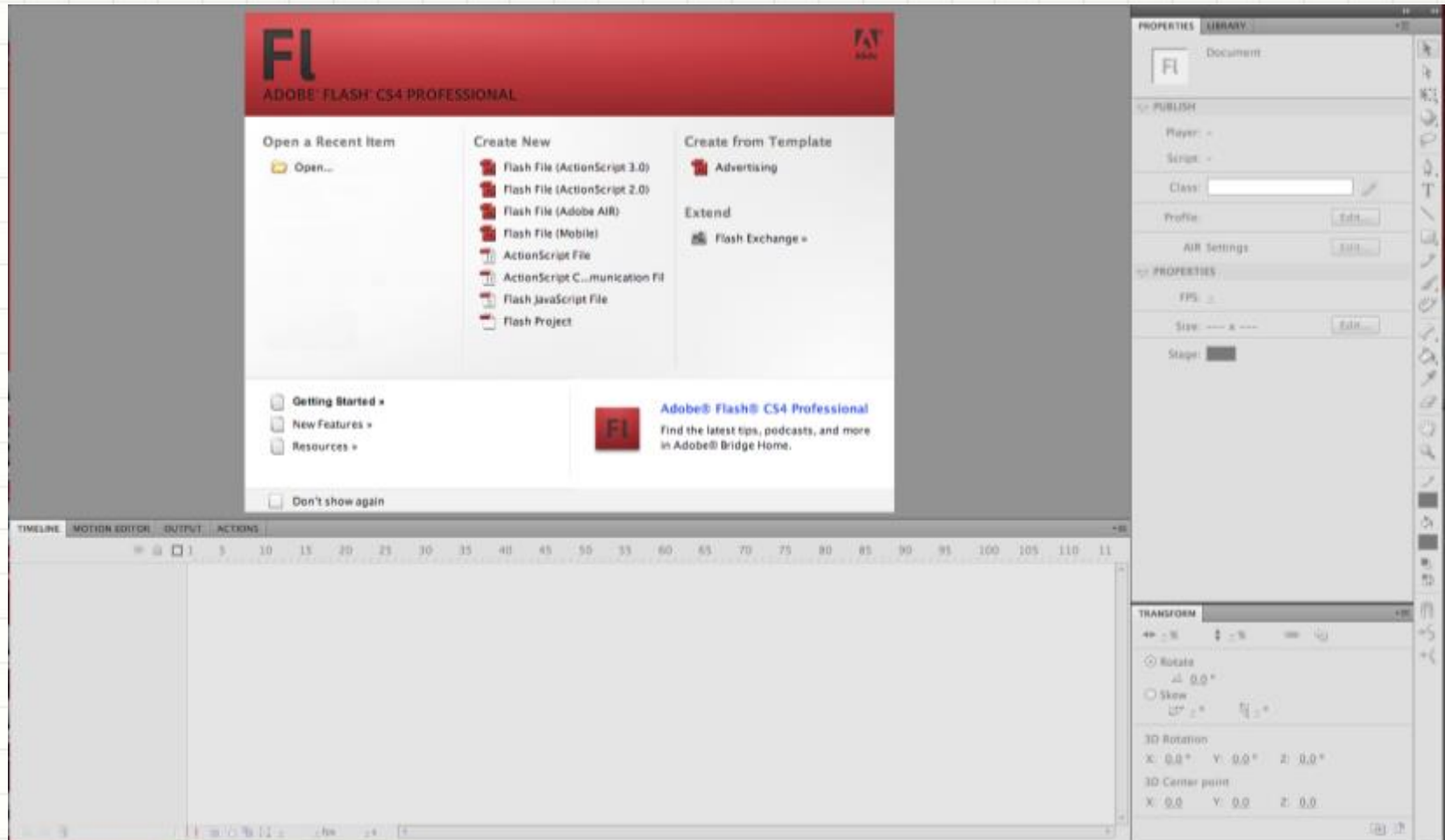
Version 10.0



© 1993–2008 Adobe Systems Incorporated and its licensors. All Rights Reserved. See the patent and other legal notices in the about box.

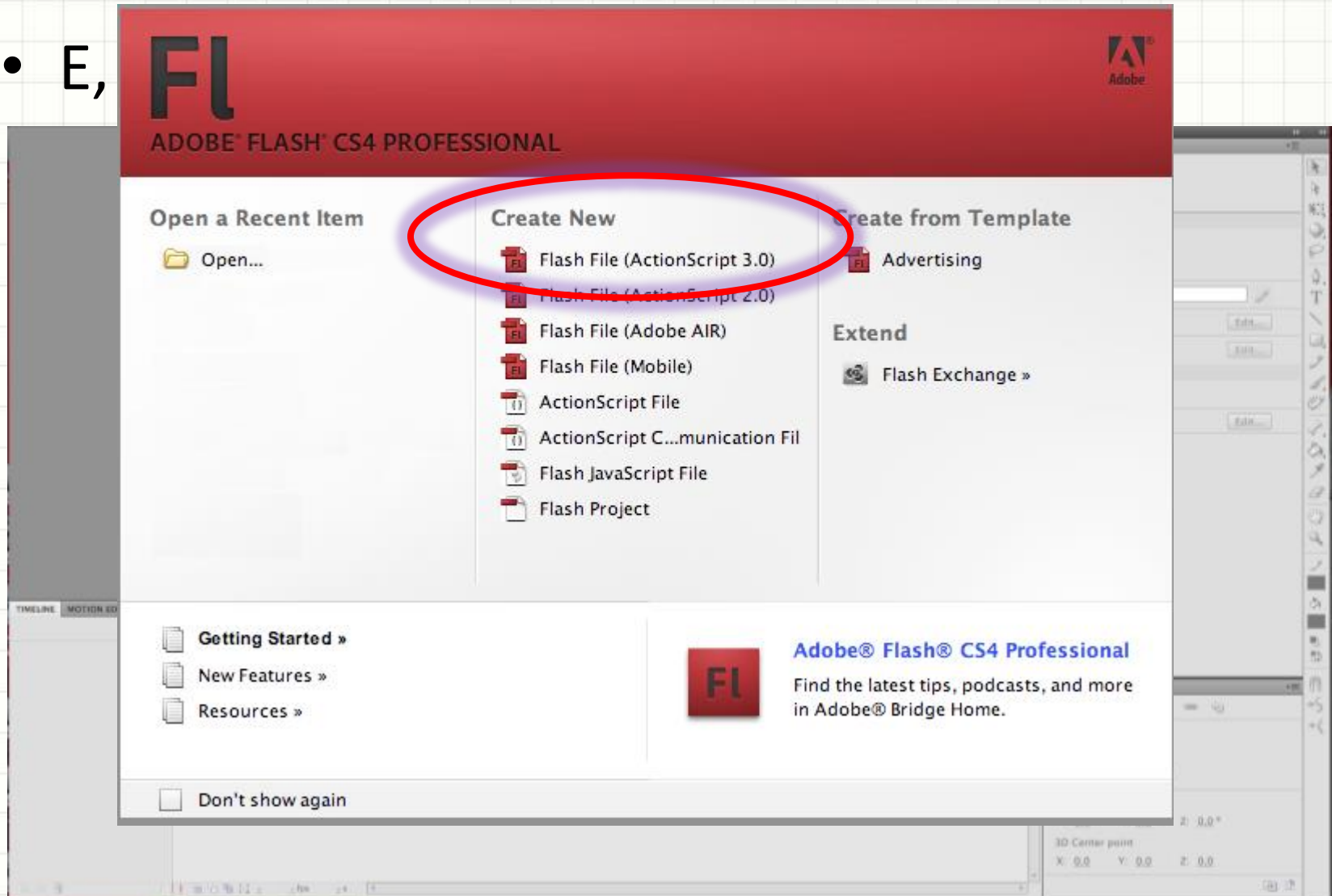
ActionScript 3 na Prática

- E, finalmente, uma janela aparecerá...



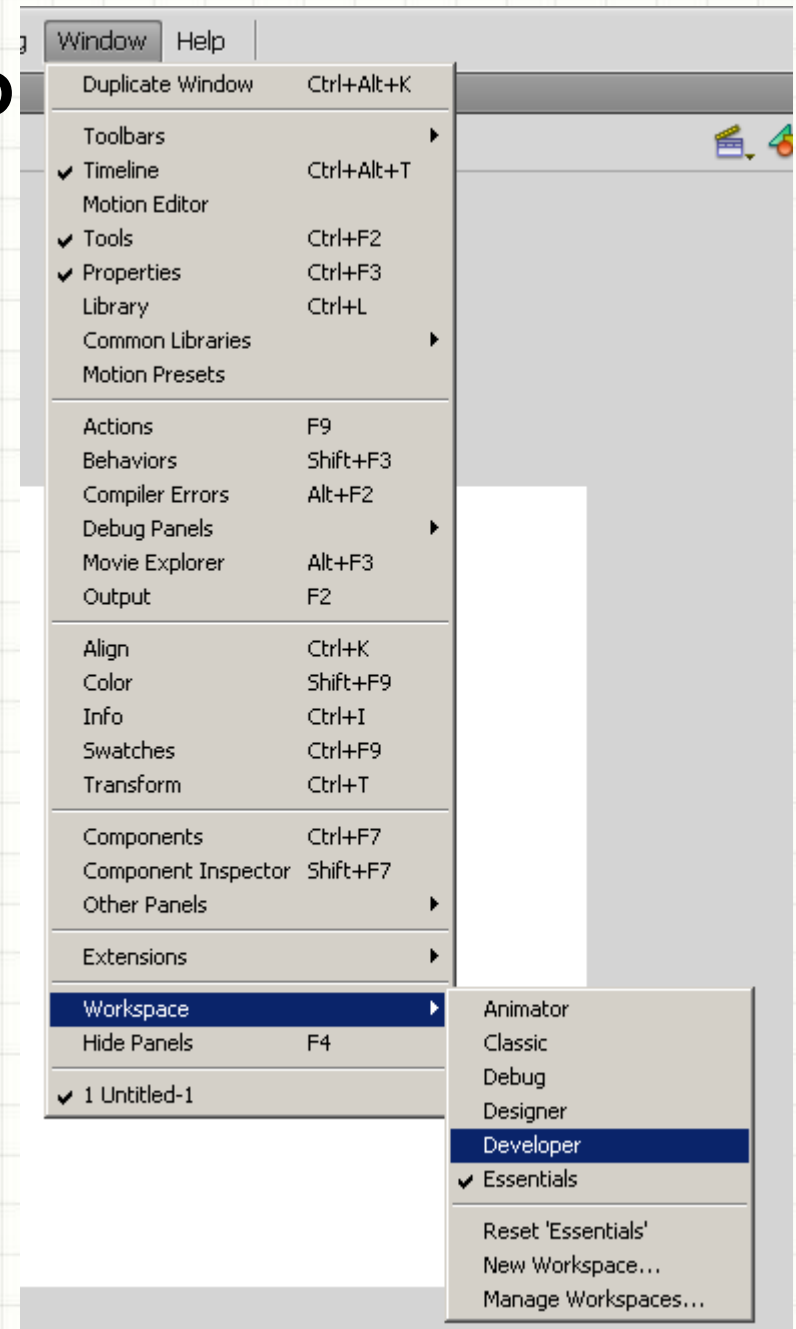
ActionScript 3 na Prática

- E,



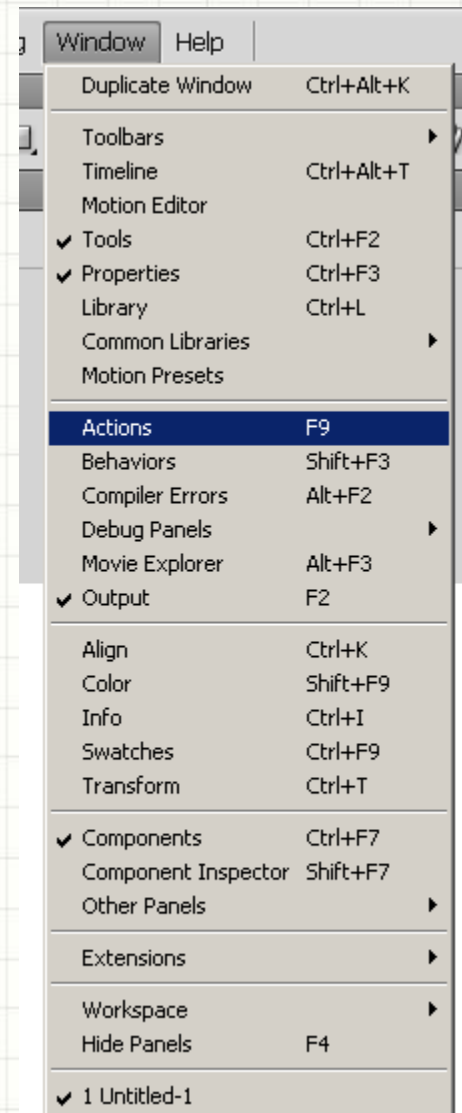
ActionScript 3 na P

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



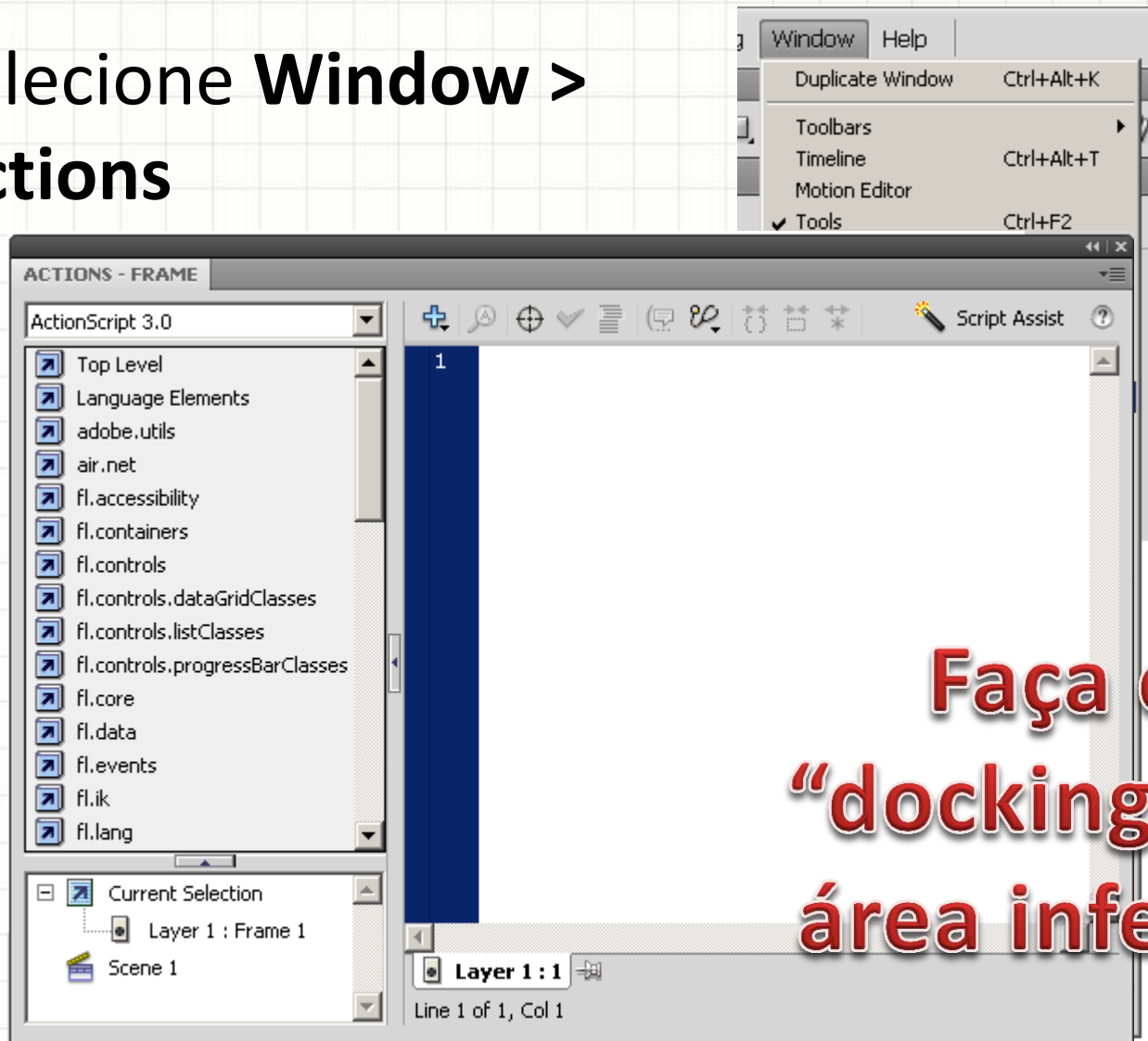
ActionScript 3 na Prática

- Selecione **Window > Actions**



ActionScript 3 na Prática

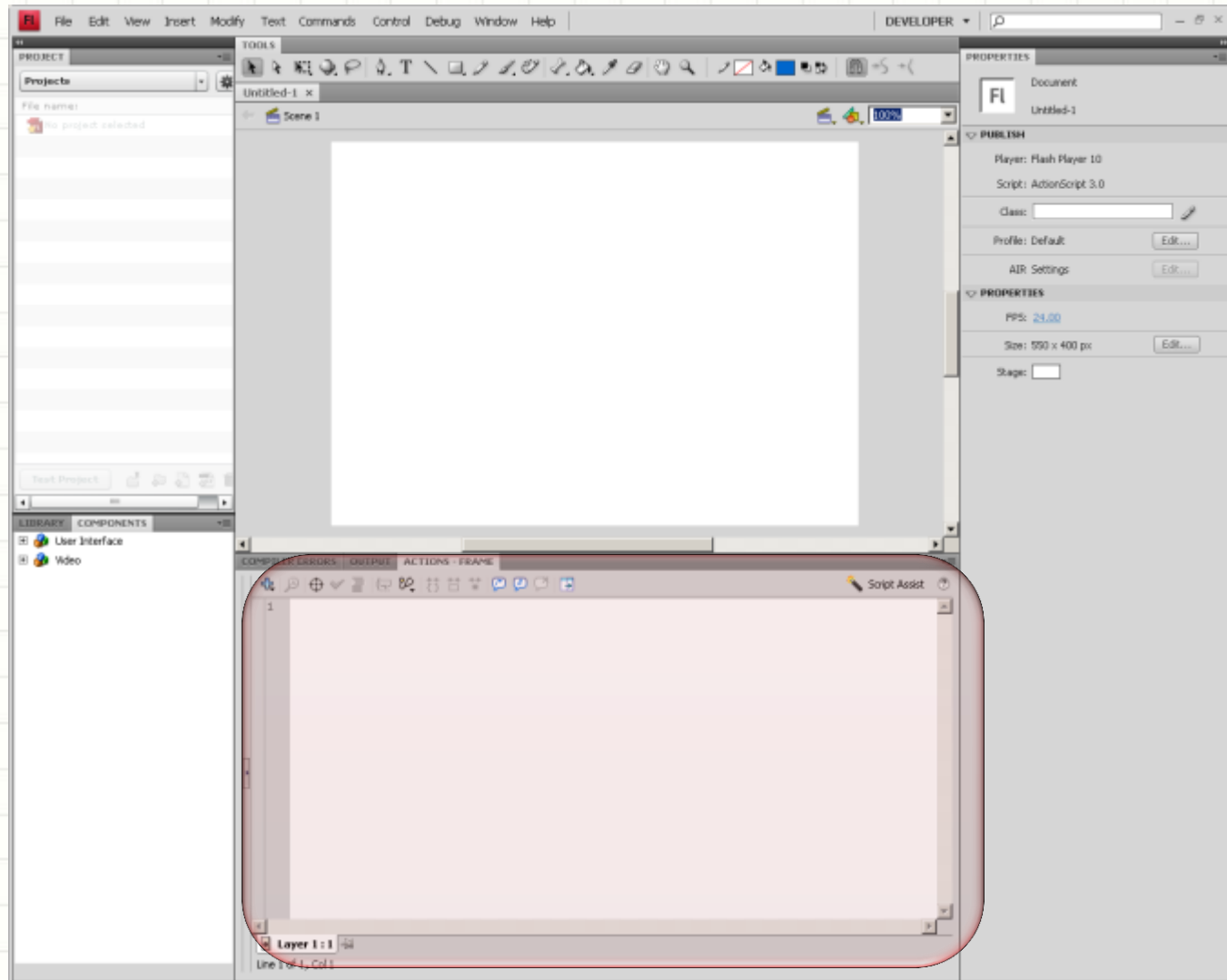
- Selecione **Window > Actions**



Faça o
“docking” na
área inferior

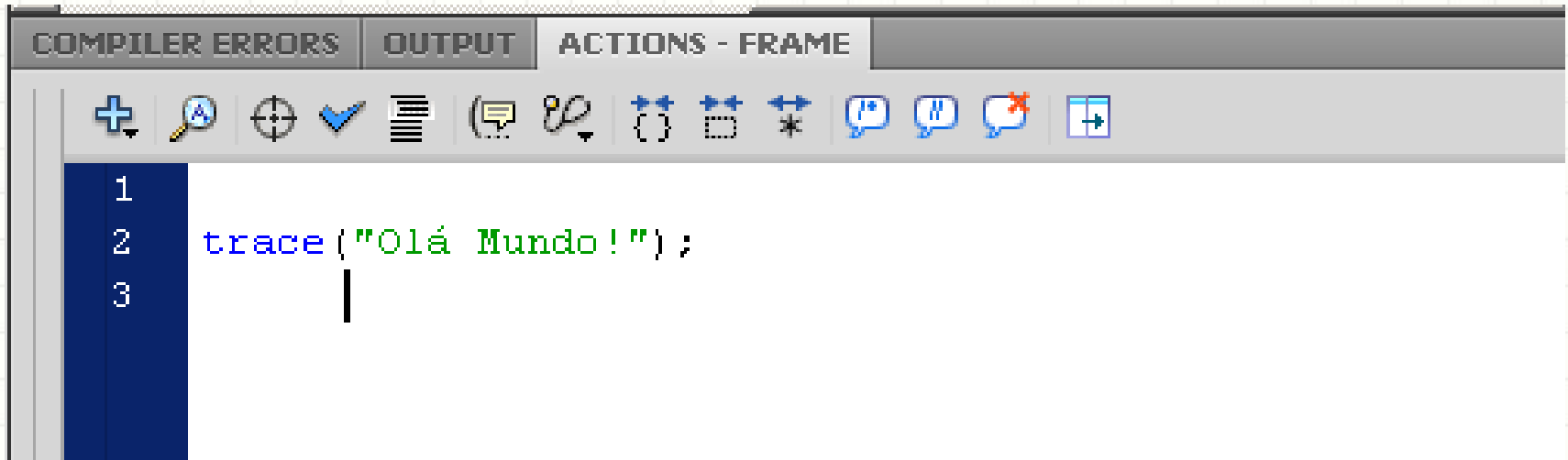
ActionScript 3 na Prática

- Tela de trabalho



ActionScript 3 na Prática

- Primeiro programa...

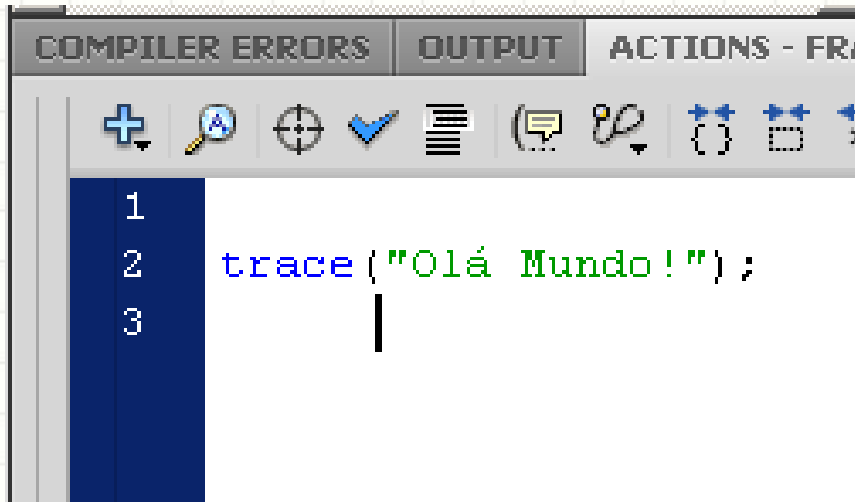


The screenshot shows a software interface with three tabs: "COMPILER ERRORS", "OUTPUT", and "ACTIONS - FRAME". The "ACTIONS - FRAME" tab is active and contains a toolbar with various icons (plus, magnifying glass, crosshair, checkmark, list, speech bubble, eraser, curly braces, square, asterisk, speech bubble with asterisk, speech bubble with double lines, speech bubble with red X, and a window with plus). Below the toolbar, a code editor displays three lines of code:

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


ActionScript 3 na Prática

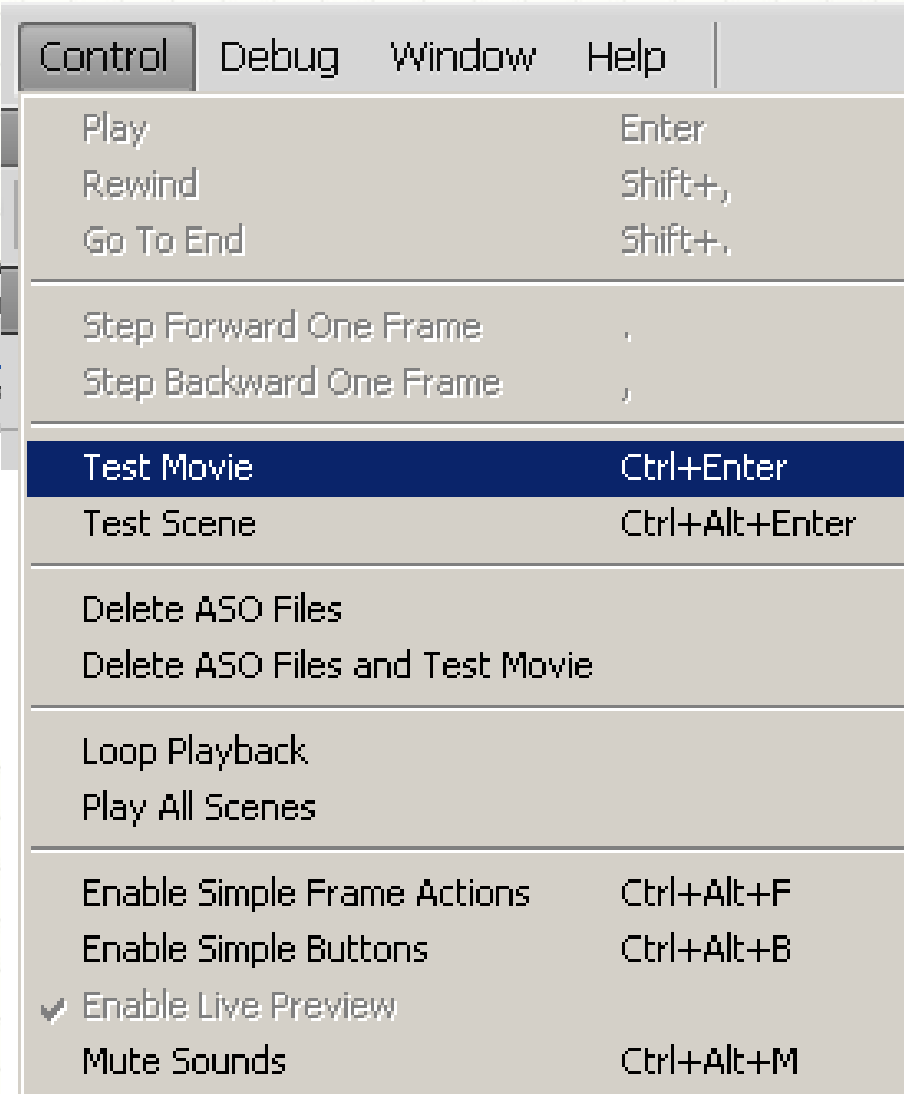
- Executando...
Control > Test Movie



The screenshot shows the ActionScript 3 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!

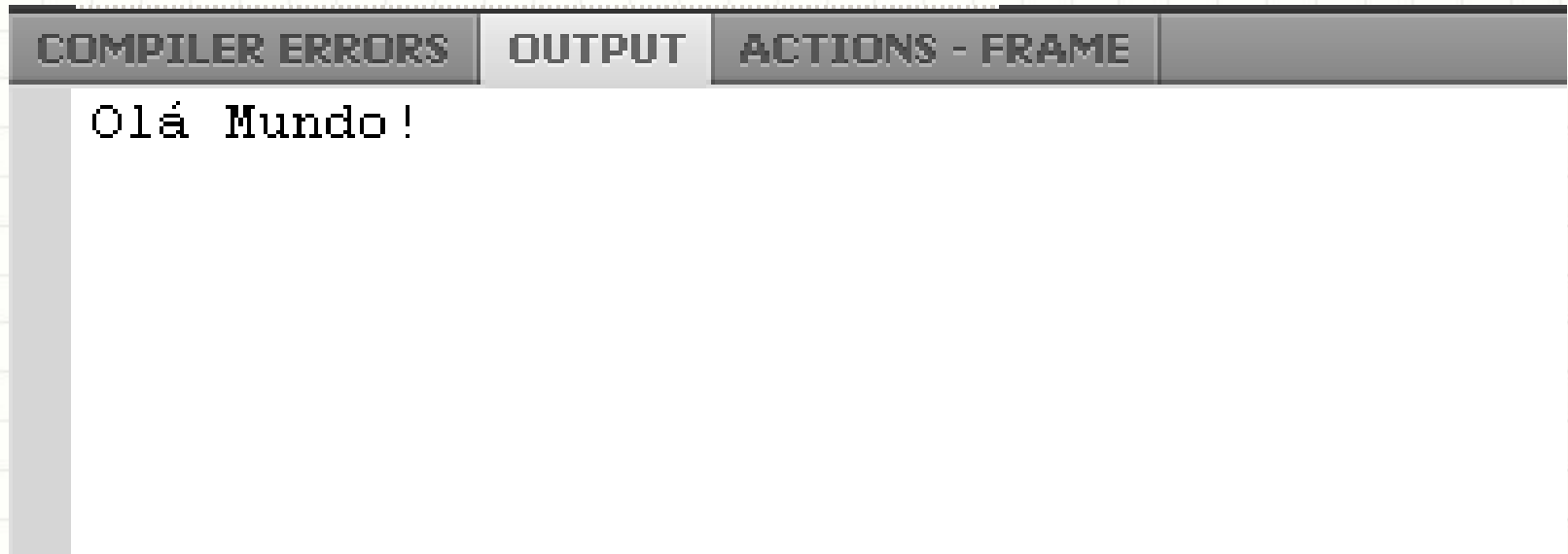


The screenshot shows the 'Control' menu in the ActionScript 3 IDE. The menu items and their keyboard shortcuts are:

Menu Item	Keyboard Shortcut
Play	Enter
Rewind	Shift+,
Go To End	Shift+,
Step Forward One Frame	,
Step Backward One Frame	.
Test Movie	Ctrl+Enter
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...

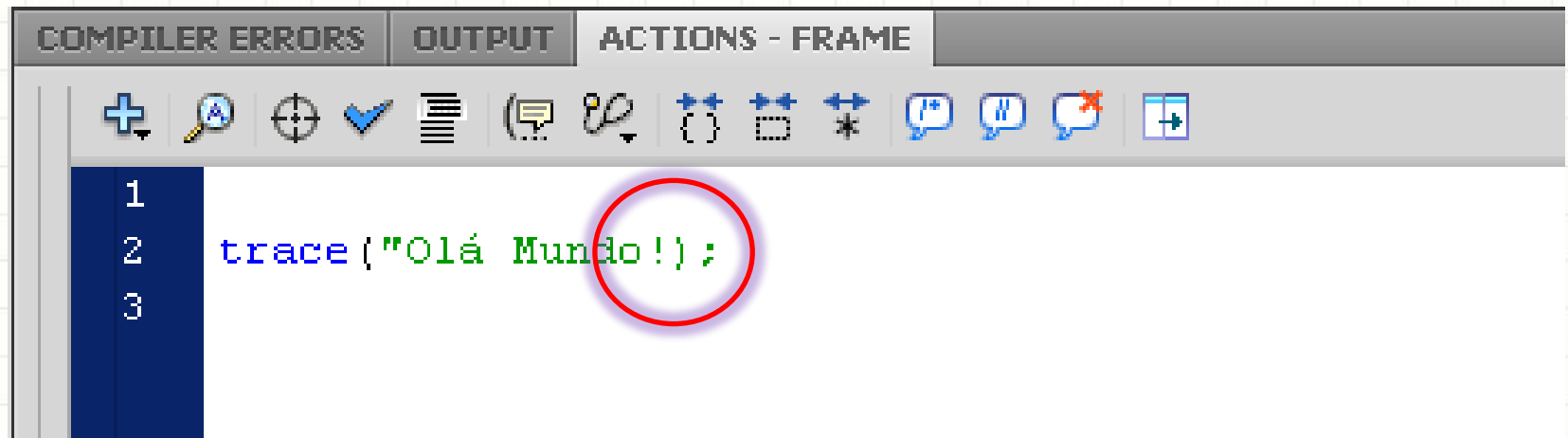
A screenshot of an IDE's output window. The window has a dark grey header with three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'OUTPUT' tab is selected. Below the header, the text 'Olá Mundo!' is displayed in a monospaced font. The background of the IDE window is a light grey grid.

```
COMPILER ERRORS OUTPUT ACTIONS - FRAME
Olá Mundo!
```

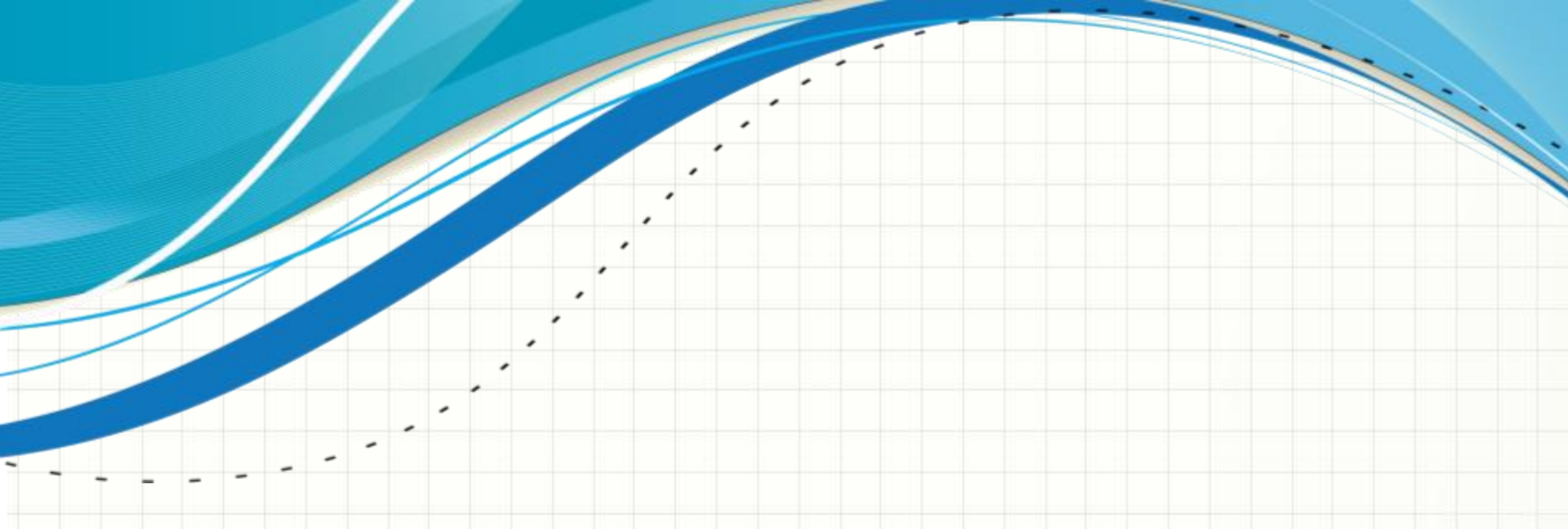
- Não aparece no filme?
 - Não...
 - **trace** é um comando de teste!

ActionScript 3 na Prática

- E se houver erro...?



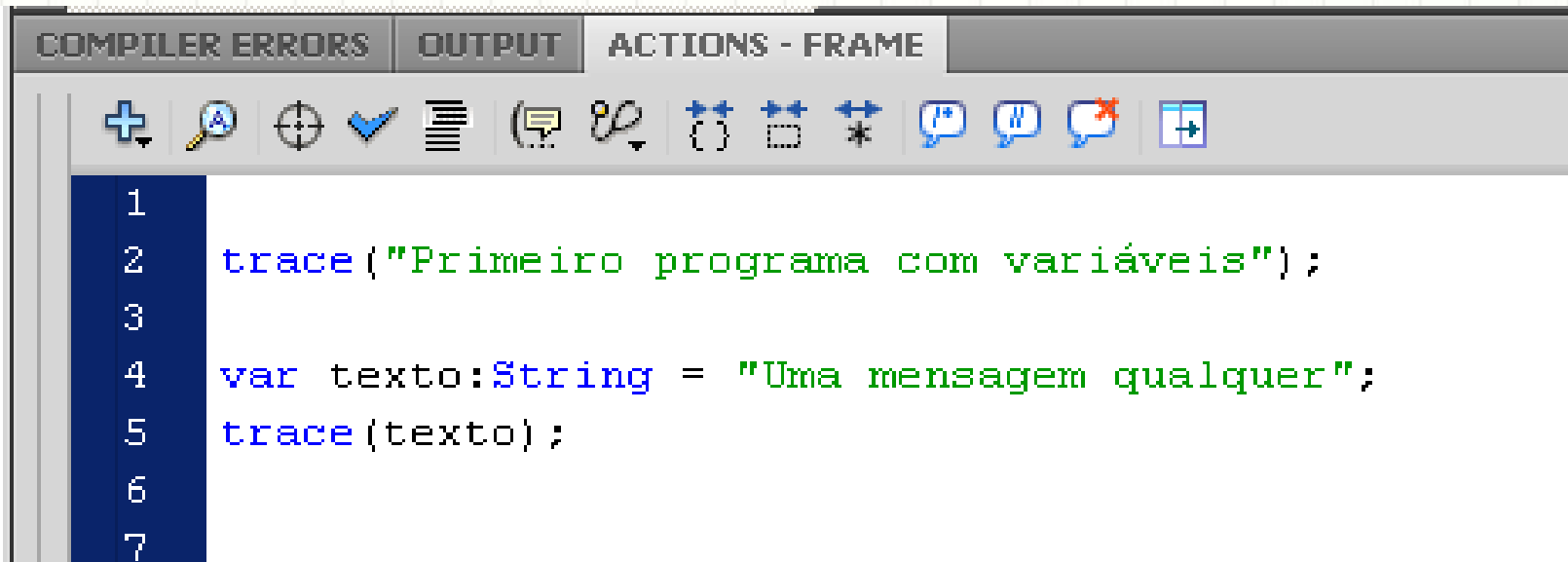
COMPILER ERRORS - 3 REPORTED	OUTPUT	ACTIONS - FRAME
Location	Description	Source
Scene 1, Layer 'Layer 1', Frame ...	1095: Syntax error: A string literal must be terminated before the line break.	trace("Olá Mundo!");
Scene 1, Layer 'Layer 1', Frame 1	1084: Syntax error: expecting identifier before end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



VARIÁVEIS EM ACTIONSCRIPT 3

Variáveis em ActionScript 3

- Teste o programa...



The screenshot shows a code editor window titled 'ACTIONS - FRAME'. The 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
```

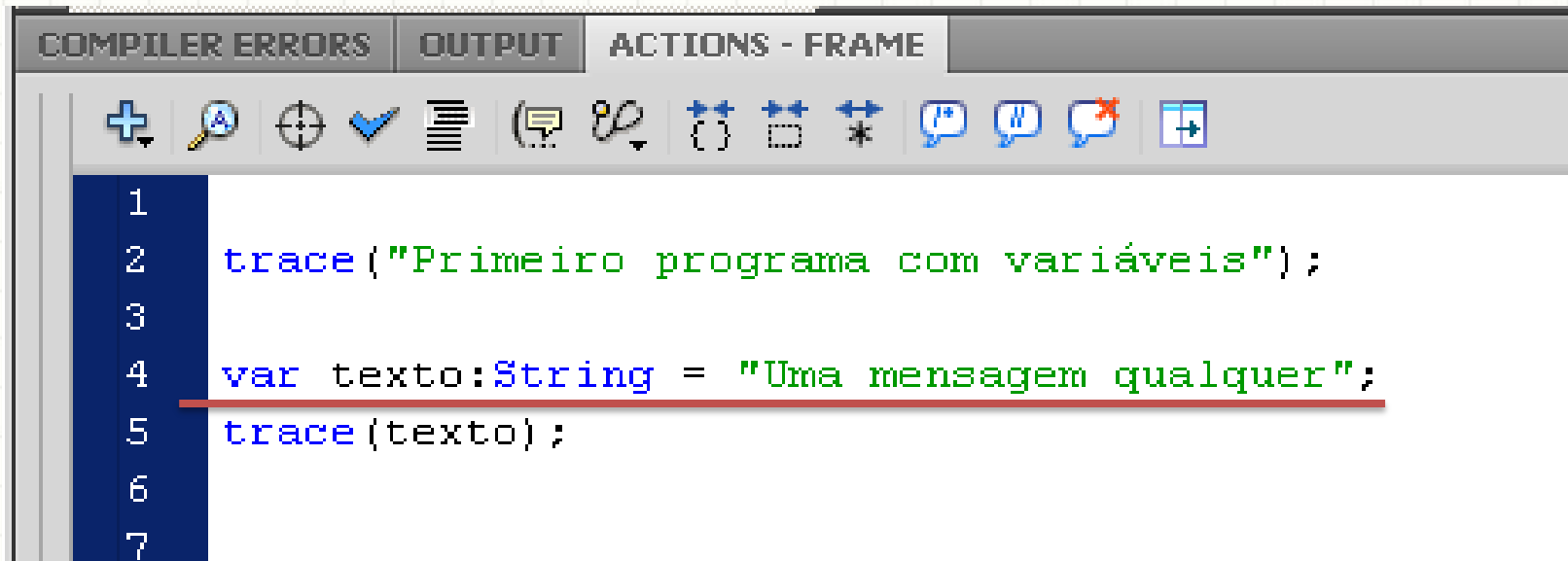
- **texto** é uma variável do tipo **String**

Variáveis em ActionScript 3

- Declaramos variáveis sempre assim:

var *nome:Tipo* = *valor_inicial*;

- Observe!



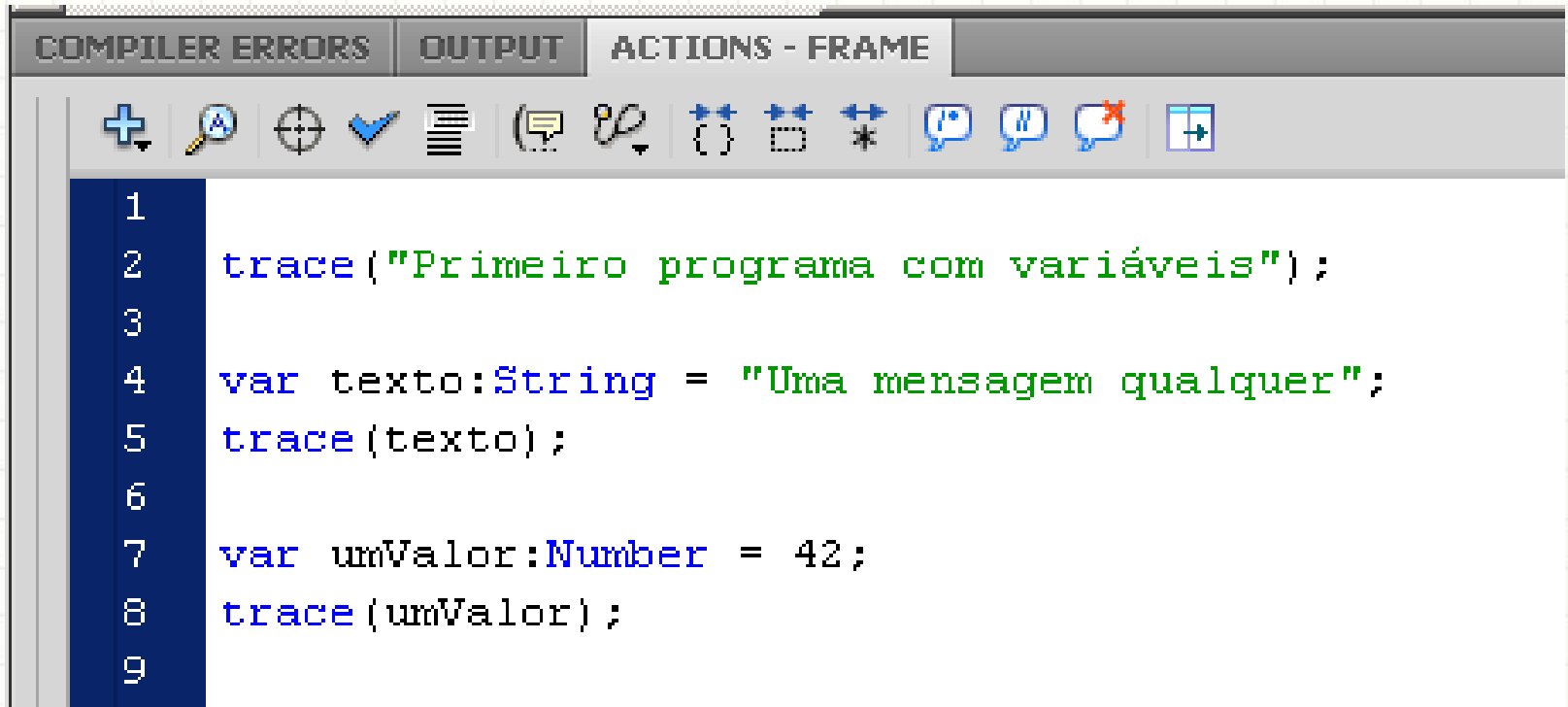
The screenshot shows an IDE window titled "ACTIONS - FRAME" with tabs for "COMPILER ERRORS" and "OUTPUT". The code editor contains the following ActionScript code:

```
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, highlighting the variable declaration syntax.

Variáveis em ActionScript 3

- Incrementando o programa



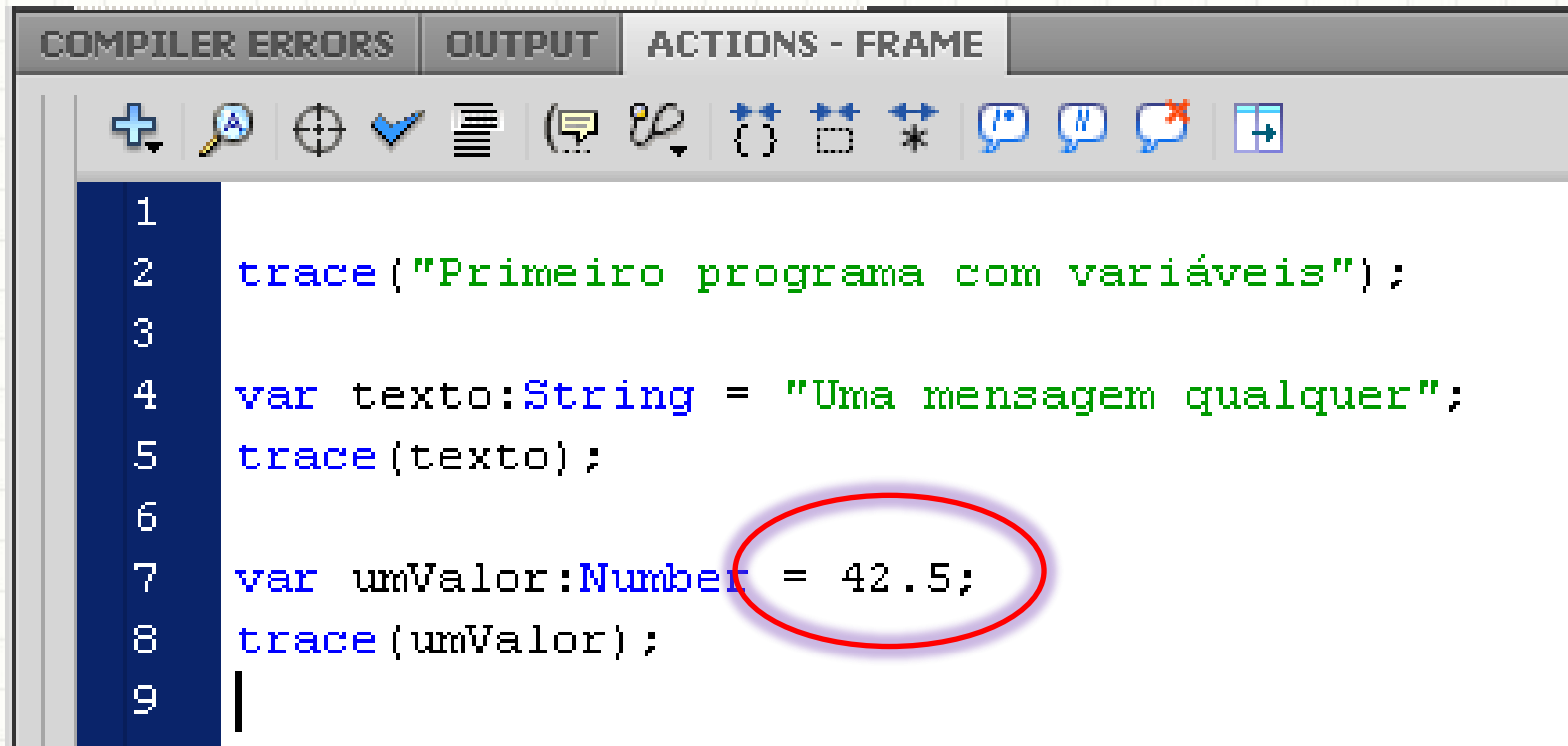
The screenshot shows an IDE window titled 'ACTIONS - FRAME'. The code editor contains the following ActionScript 3 code:

```
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...



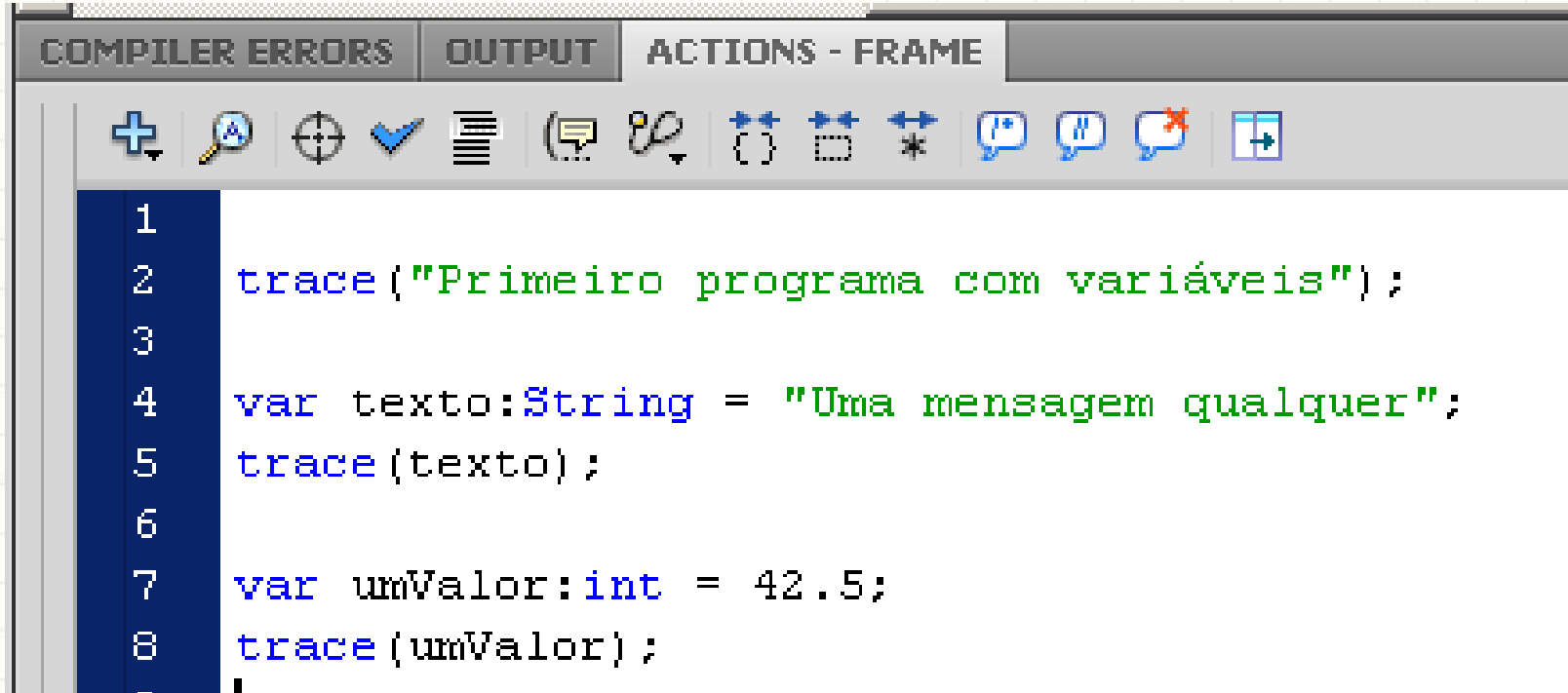
The image shows a screenshot of an IDE window titled "ACTIONS - FRAME". The window has three tabs: "COMPILER ERRORS", "OUTPUT", and "ACTIONS - FRAME". Below the tabs is a toolbar with various icons for editing and development. The main area of the window displays the following ActionScript code:

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

The value `42.5` in line 7 is circled in red, highlighting that the `Number` type accepts fractional values.

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:

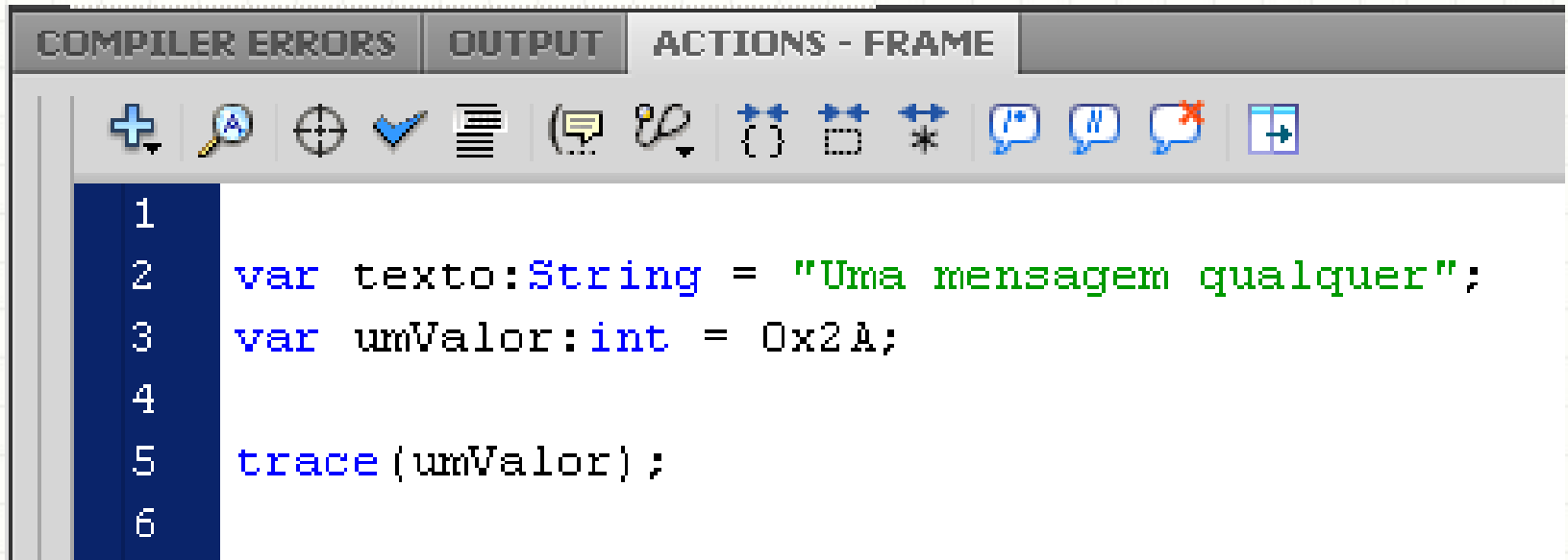
```
1  
2 trace("Primeiro programa com variáveis");  
3  
4 var texto:String = "Uma mensagem qualquer";  
5 trace(texto);  
6  
7 var umValor:int = 42.5;  
8 trace(umValor);
```

The code on line 7, `var umValor:int = 42.5;`, is highlighted in blue, indicating a type mismatch error where a floating-point value is assigned to an integer variable.

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

Variáveis em ActionScript 3

- Podemos usar valores em hexadecimal...



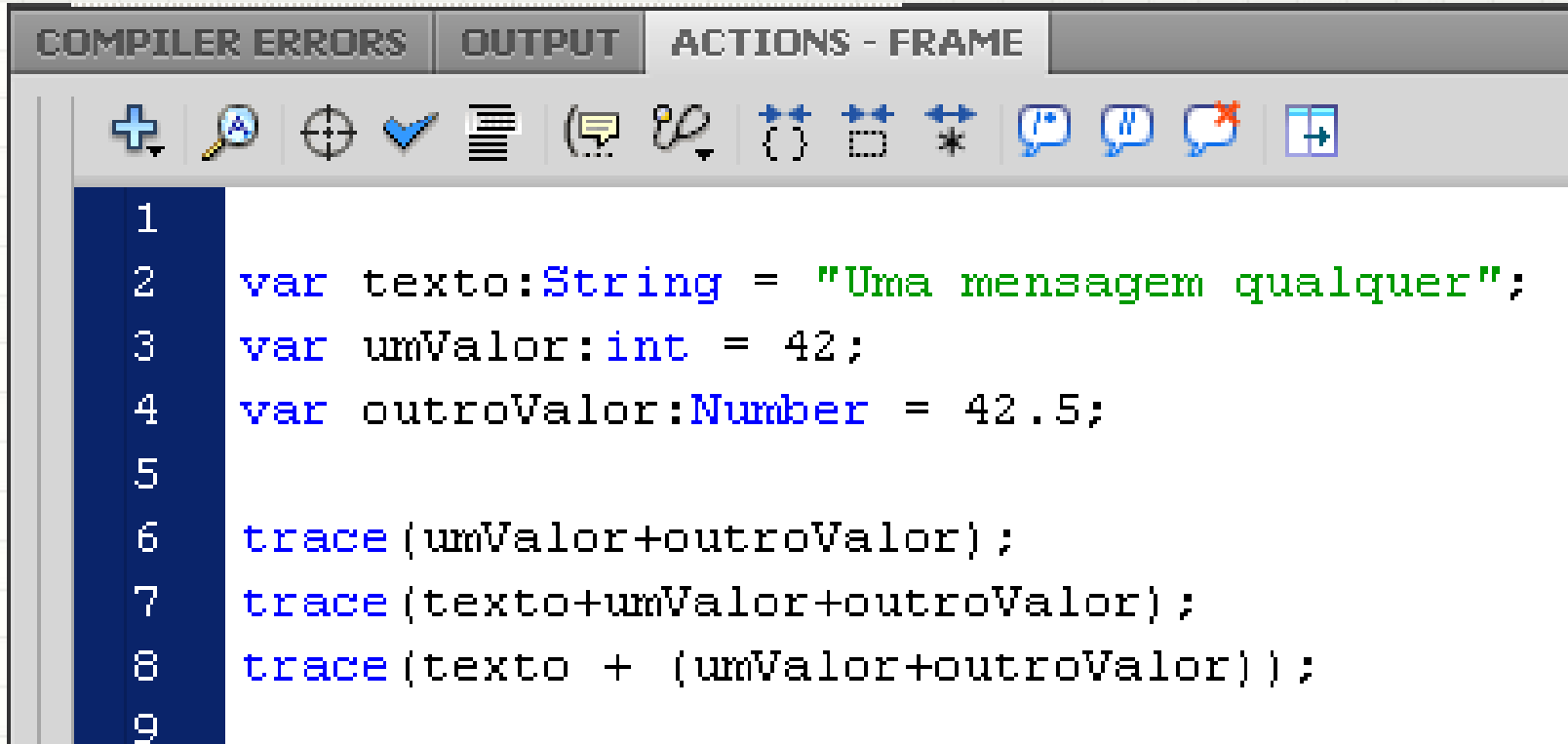
The screenshot shows an IDE window titled 'ACTIONS - FRAME'. The window has three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. Below the tabs is a toolbar with various icons for editing and development. The main area contains the following code:

```
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 and a code editor area. The code editor contains the following ActionScript 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
```

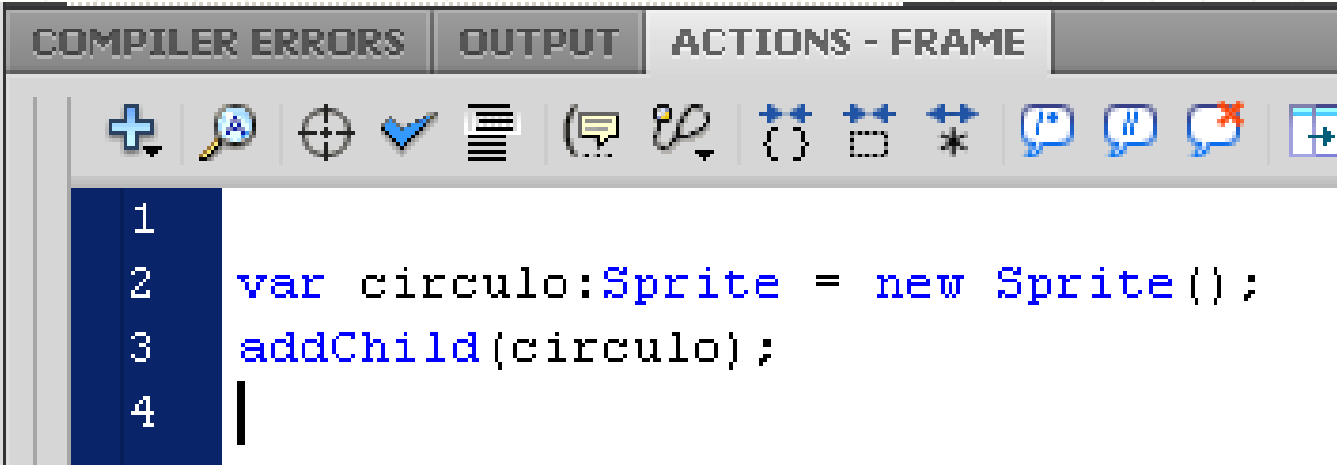
- Execute e observe os resultados...
- Qual a diferença entre eles?



DESENHANDO NA TELA COM ACTIONSCRIPT 3

Desenhando em ActionScript 3

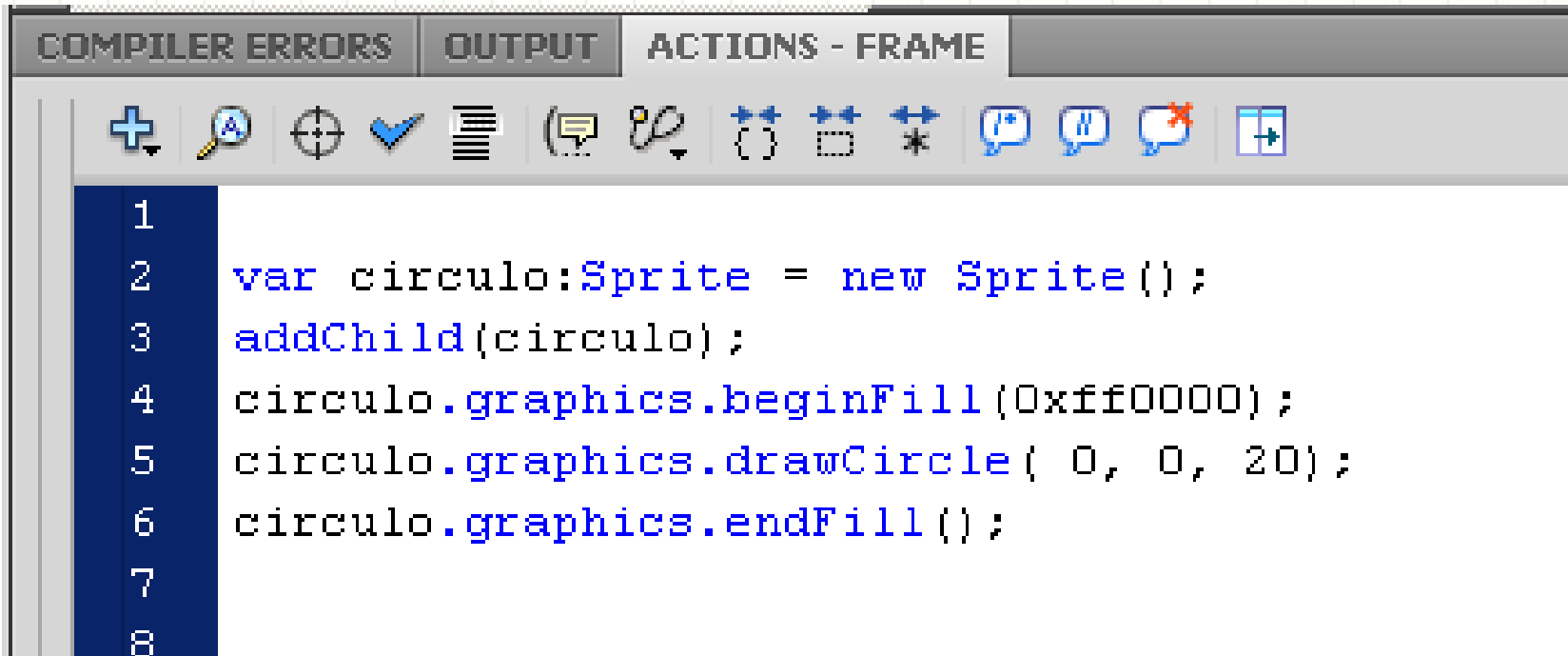
- Para desenhar na tela: criar um **Sprite**
- **Sprite** é um desenho que pode se mover
- Depois, precisamos adicionar o sprite como um filhote da animação corrente (**addChild**)
- Experimente o código abaixo... o que ocorre?



```
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!

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
```

- 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

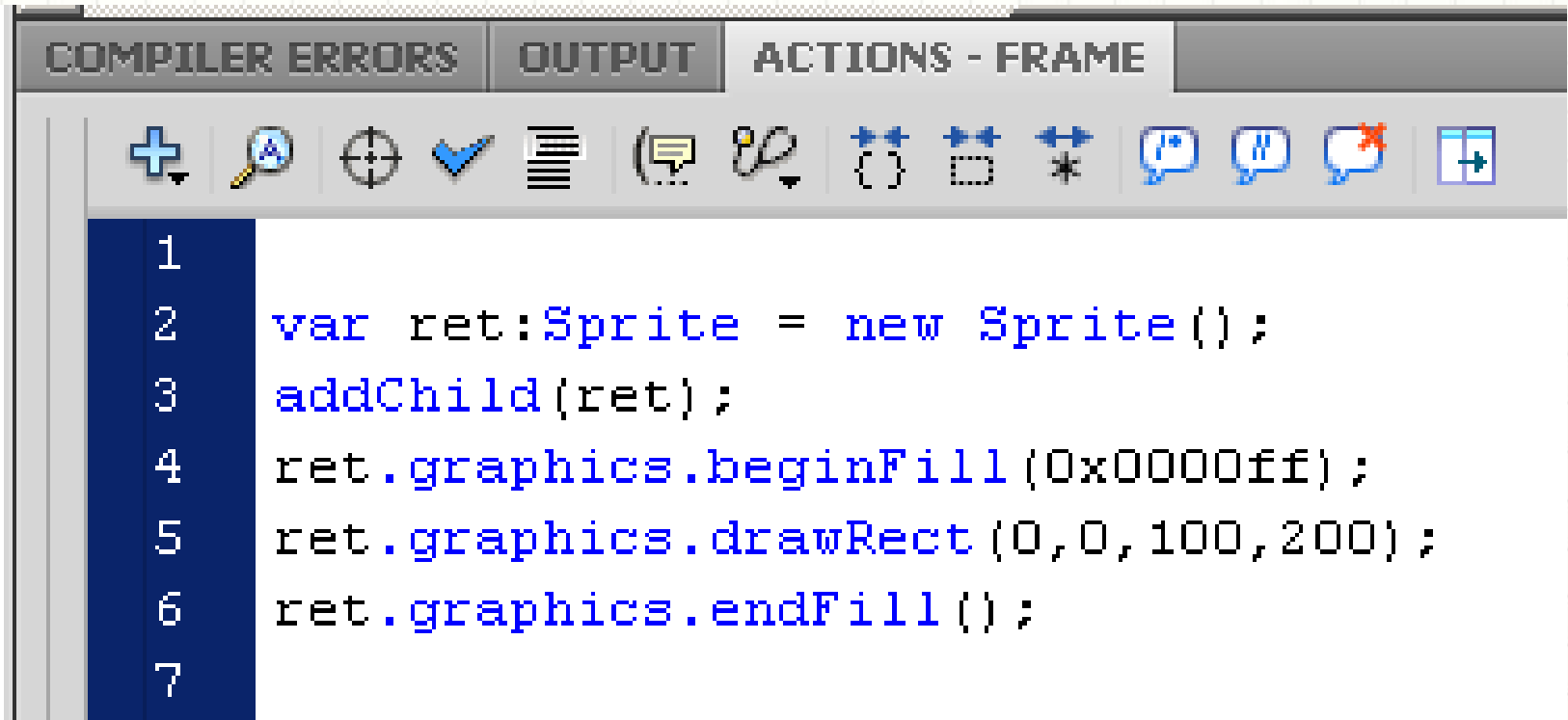
```
2 var circulo:Sprite = new Sprite();
3 addChild(circulo);
4 circulo.graphics.beginFill(0xff0000);
5 circulo.graphics.drawCircle( 0, 0, 20);
6 circulo.graphics.endFill();
7
8
```

Indica que o **desenho do sprite acabou**

- Experimente!

Desenhando em ActionScript 3

- Desenhando um retângulo...



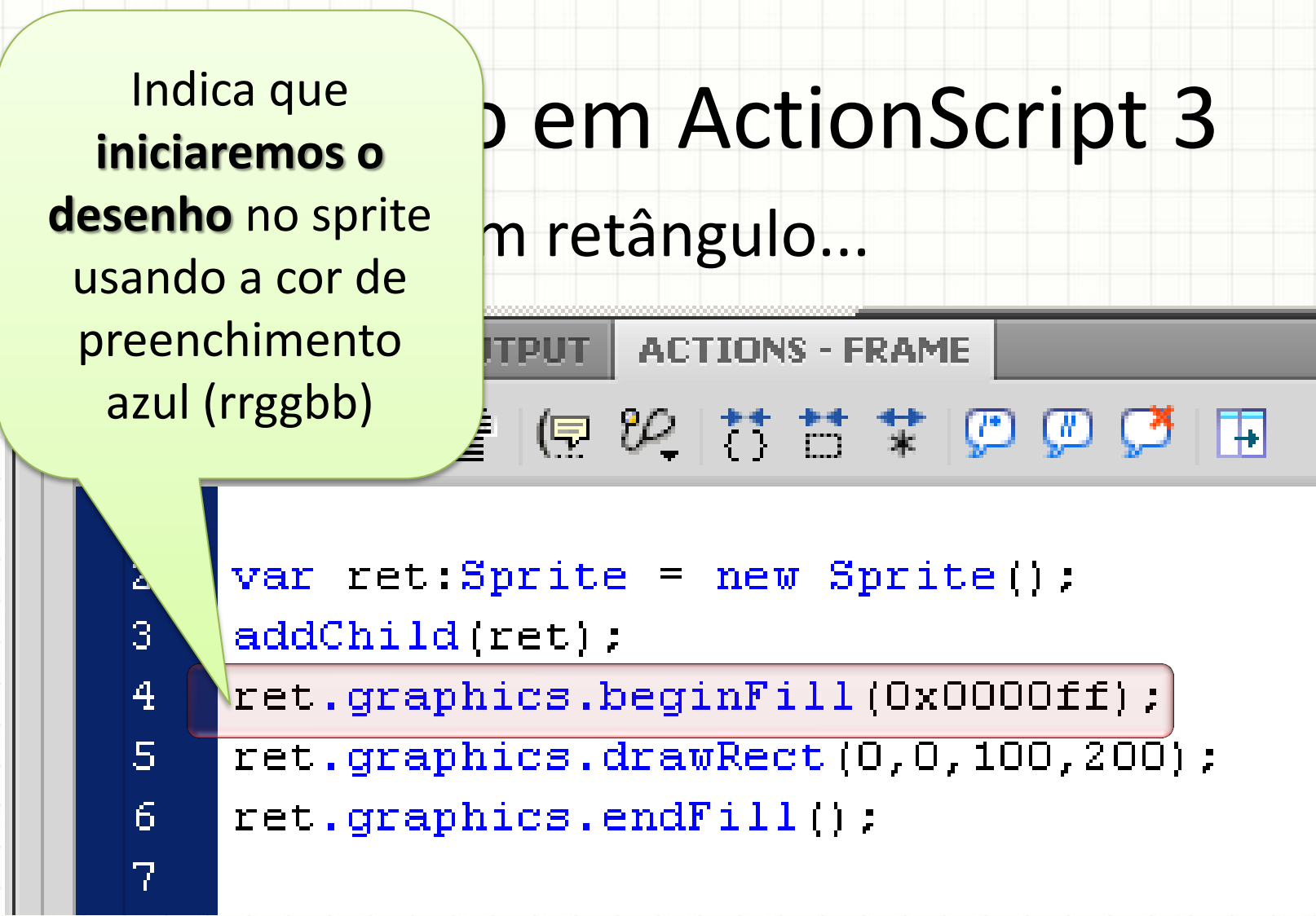
The screenshot shows a software development environment with a toolbar and a code editor. The toolbar includes icons for adding, searching, zooming, undo, redo, and other actions. The code editor displays the following ActionScript 3 code:

```
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
```

- 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**

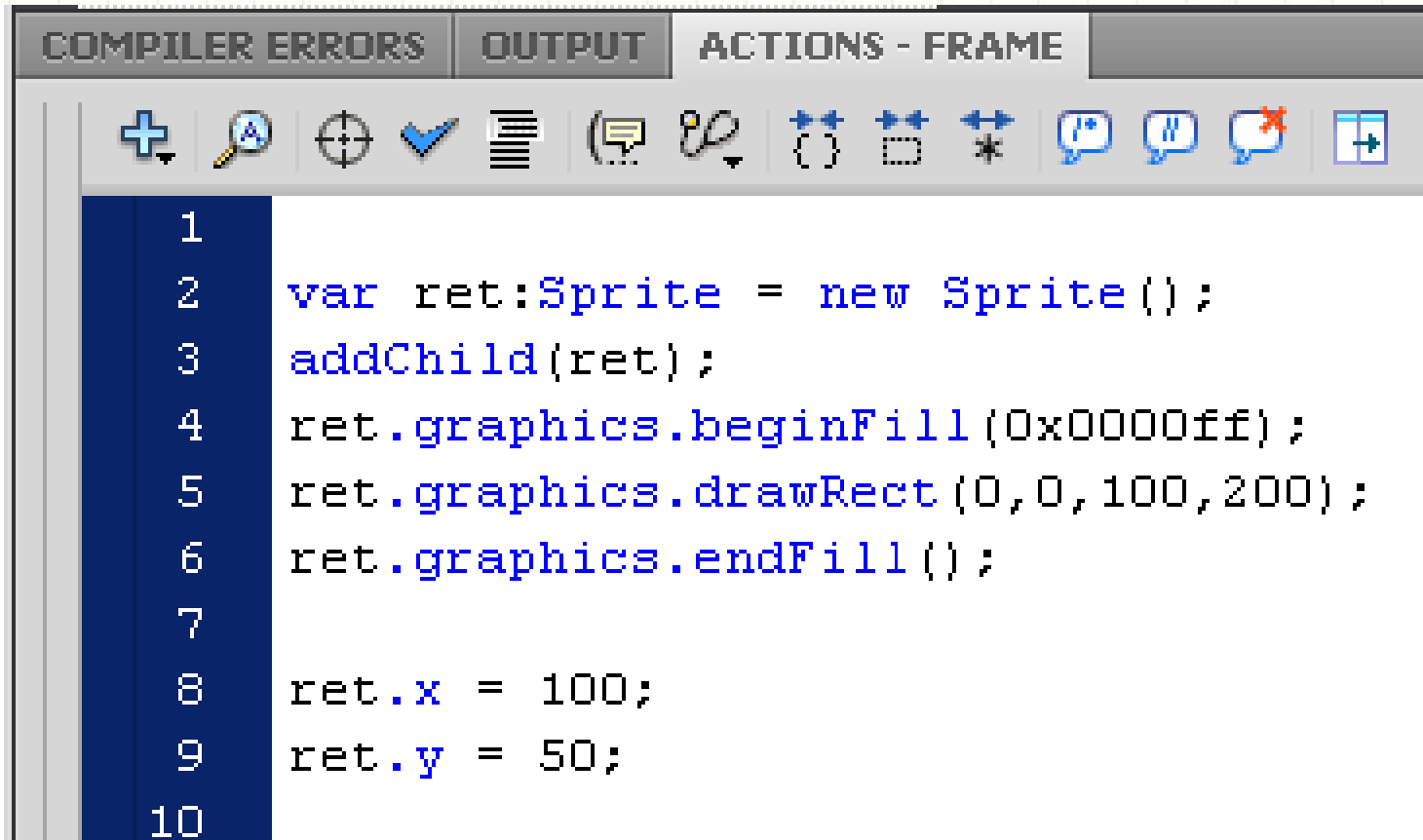
- Experimente!



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:

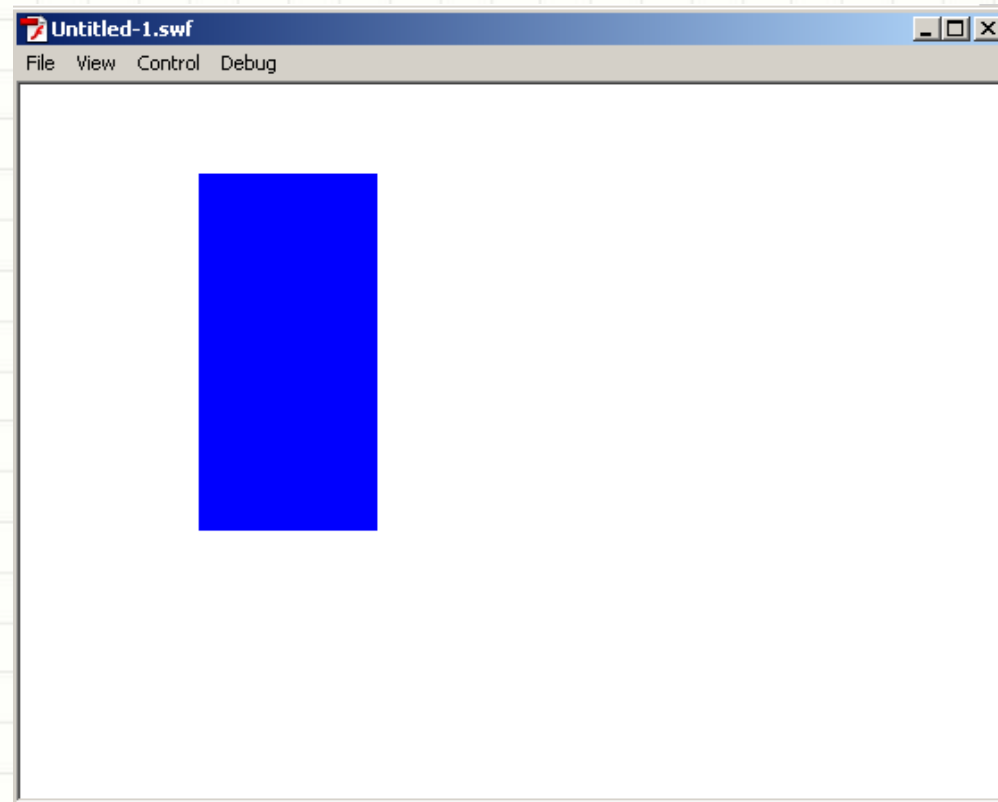
```
1  
2 var ret:Sprite = new Sprite();  
3 addChild(ret);  
4 ret.graphics.beginFill(0x0000ff);  
5 ret.graphics.drawRect(0,0,100,200);  
6 ret.graphics.endFill();  
7  
8 ret.x = 100;  
9 ret.y = 50;  
10
```

- Experimente!

Desenhando em ActionScript 3

- Resultado

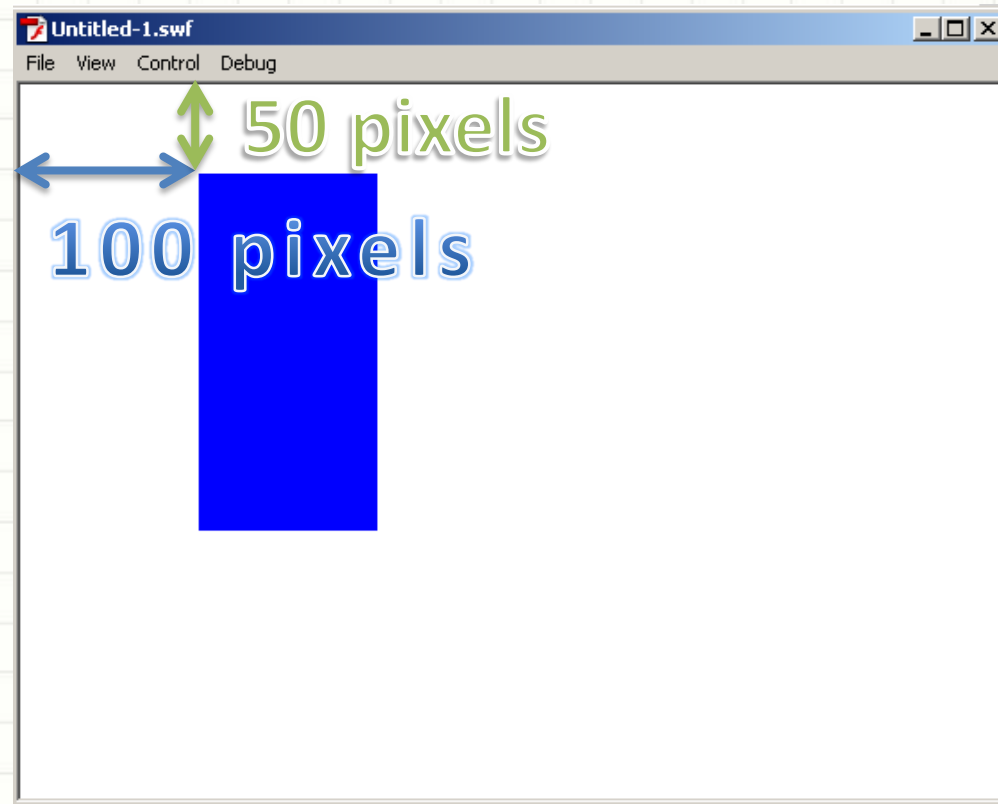
```
8 ret.x = 100;  
9 ret.y = 50;
```



Desenhando em ActionScript 3

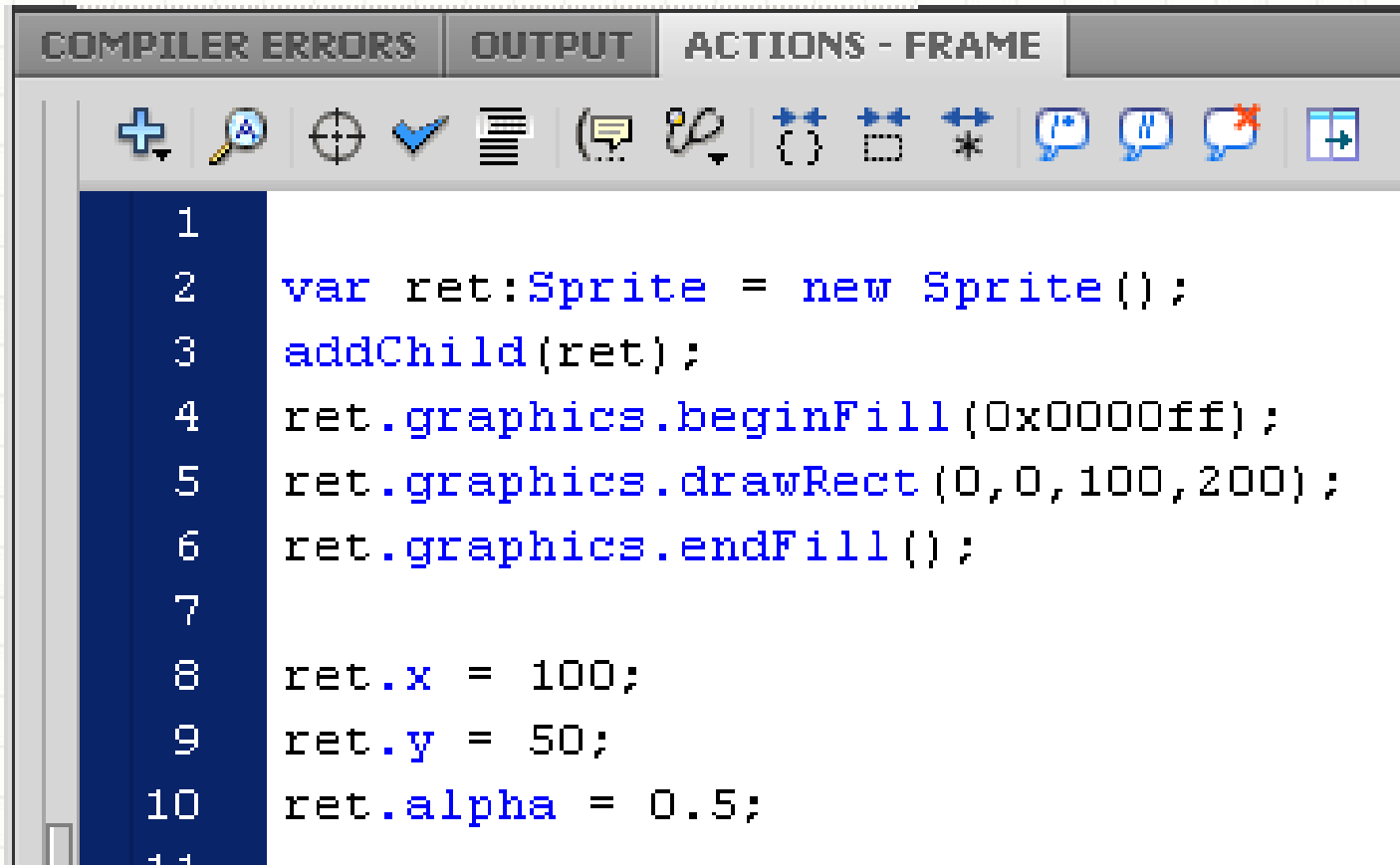
- Resultado

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



Desenhando em ActionScript 3

- Podemos mudar a “opacidade” do sprite



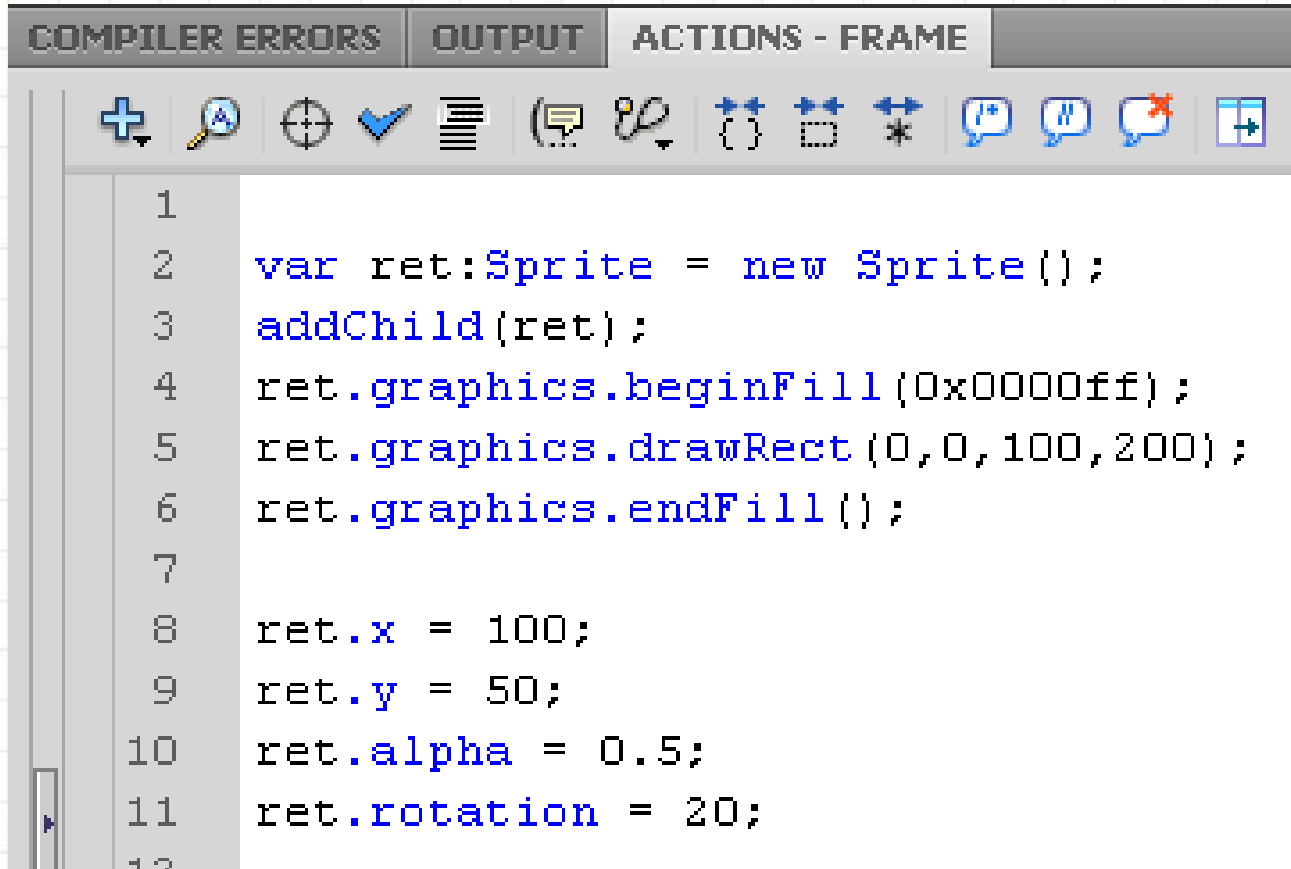
The screenshot shows a code editor window titled "ACTIONS - FRAME". The code is as follows:

```
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 titled "ACTIONS - FRAME" with a toolbar containing various icons for editing and execution. The code is as follows:

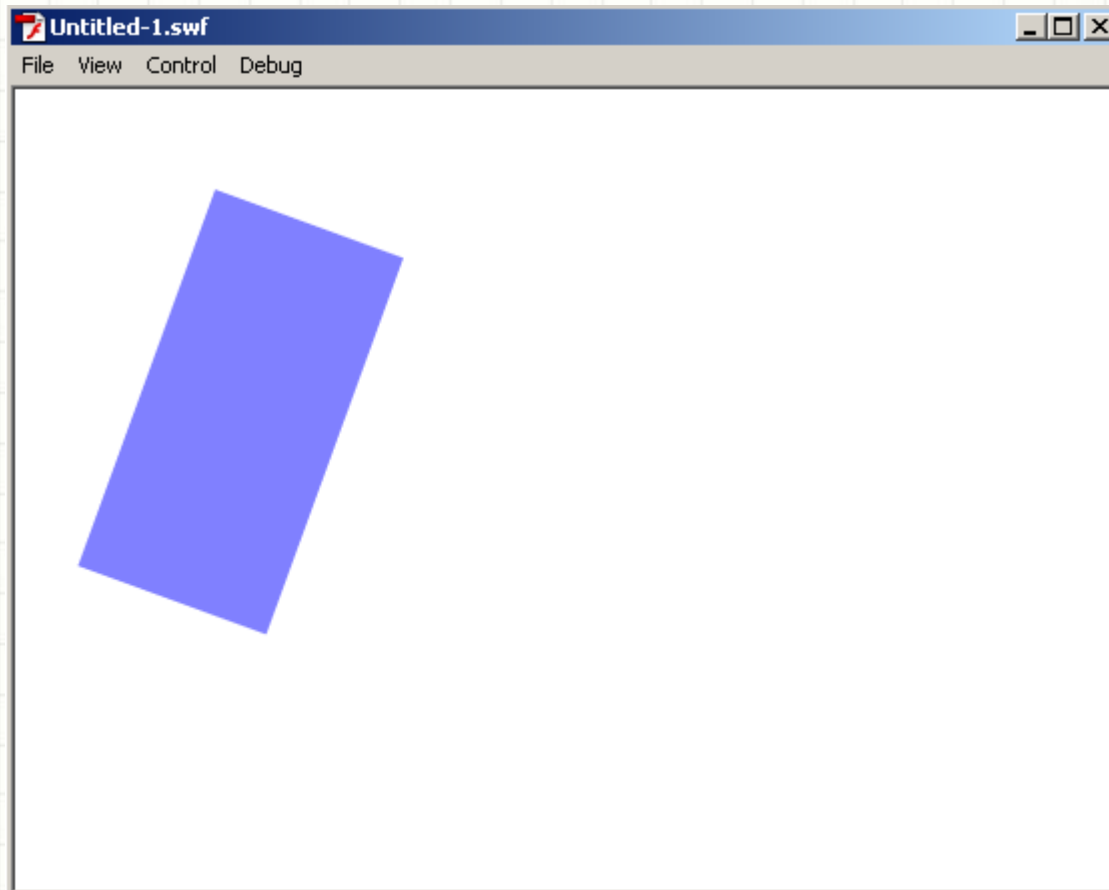
```
1  
2  var ret:Sprite = new Sprite();  
3  addChild(ret);  
4  ret.graphics.beginFill(0x0000ff);  
5  ret.graphics.drawRect(0,0,100,200);  
6  ret.graphics.endFill();  
7  
8  ret.x = 100;  
9  ret.y = 50;  
10 ret.alpha = 0.5;  
11 ret.rotation = 20;  
12
```

- Experimente!

Desenhando em ActionScript 3

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

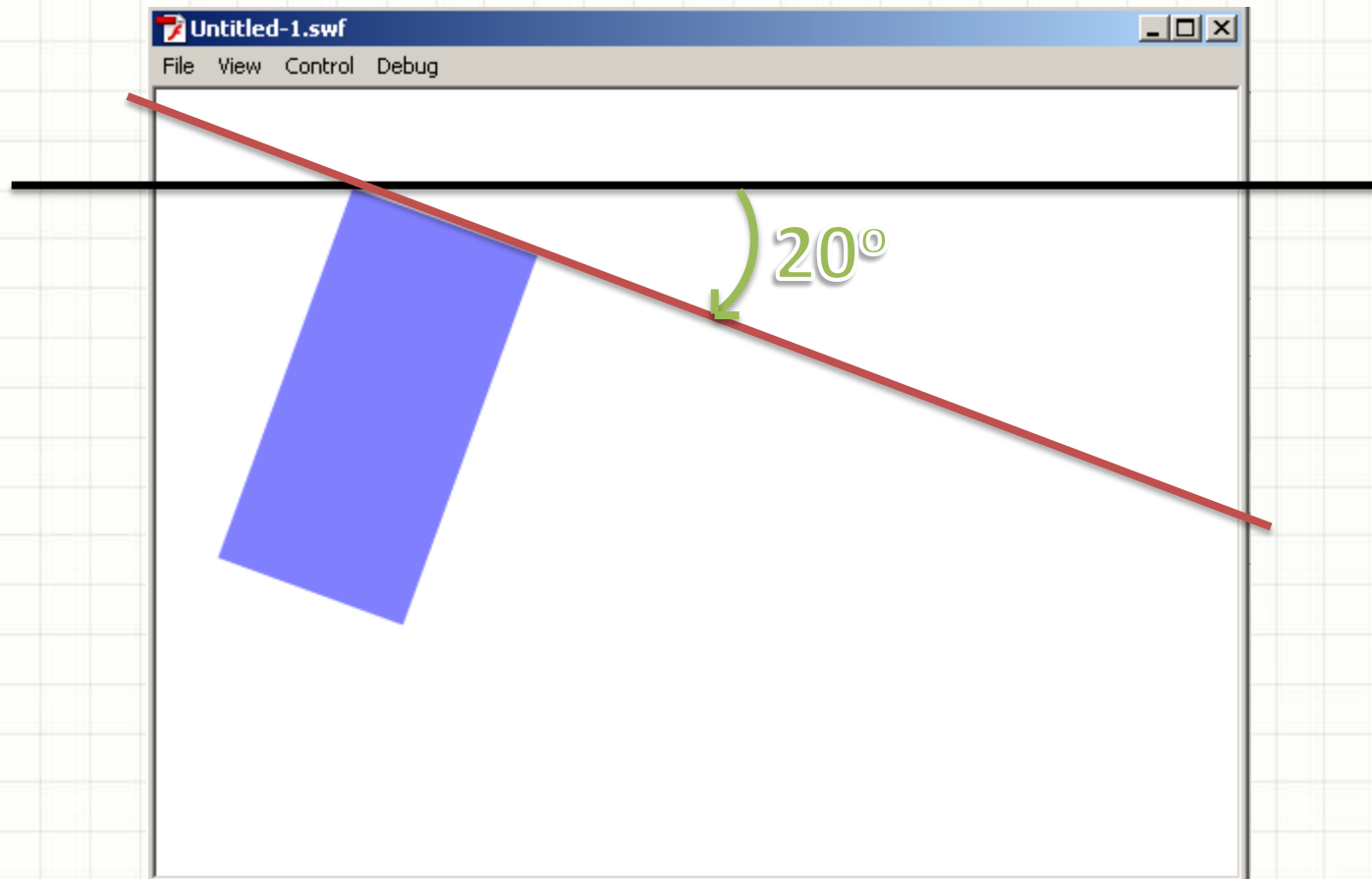
```
11 ret.rotation = 20;
```



Desenhando em ActionScript 3

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

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





EIXOS DE COORDENADAS

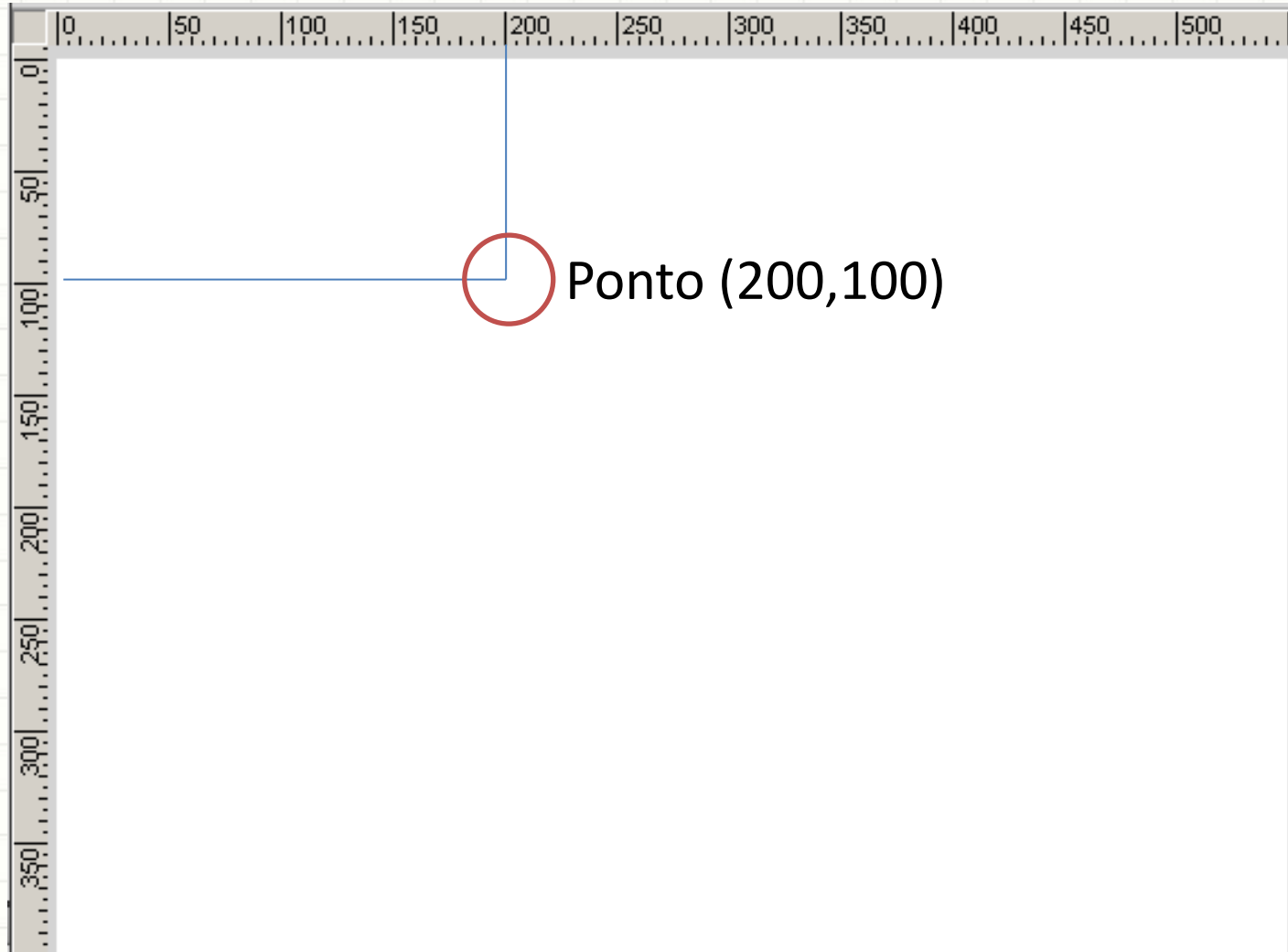
Coordenadas do AS3

- Eixos de coordenadas do cenário



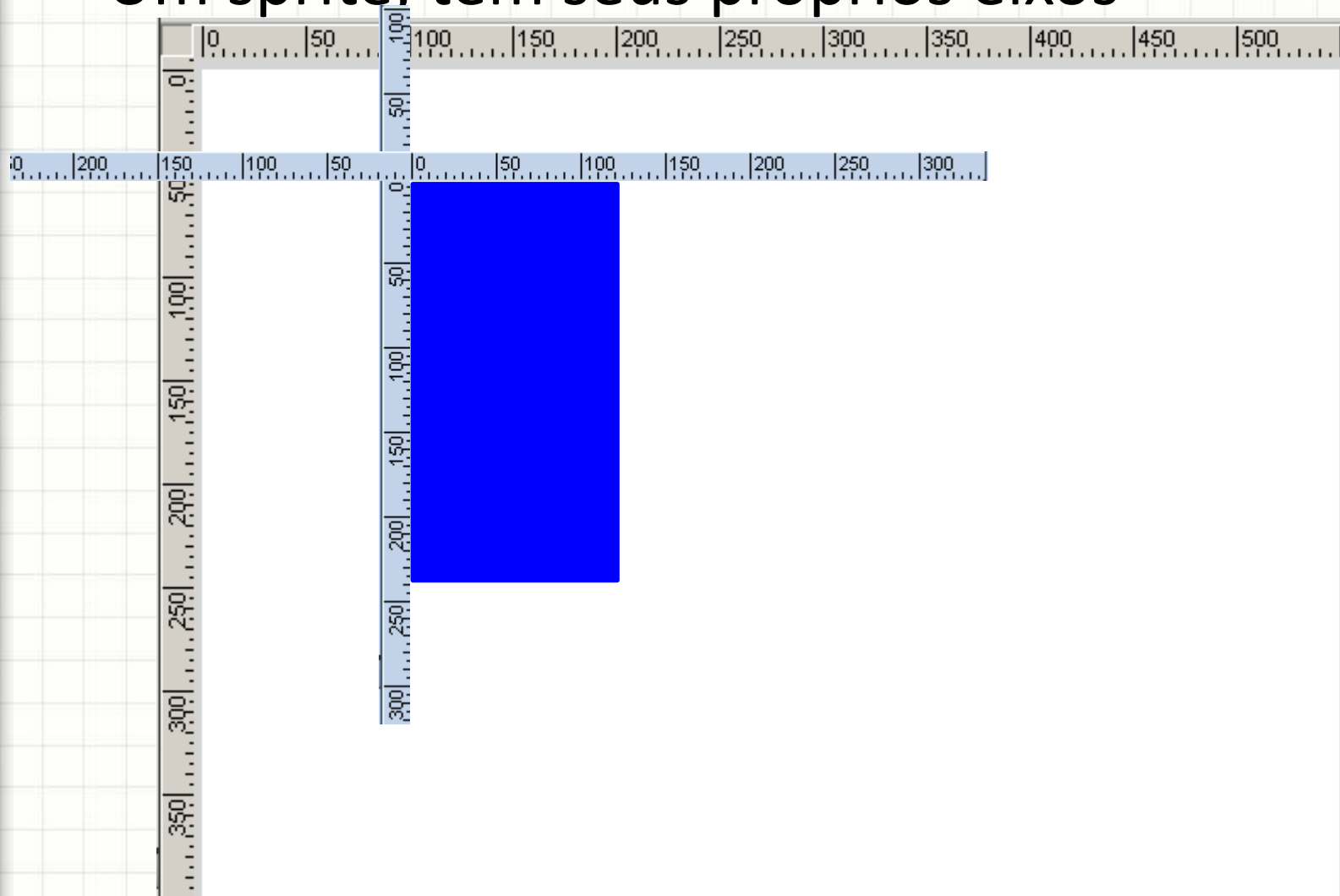
Coordenadas do AS3

- Eixos de coordenadas do cenário



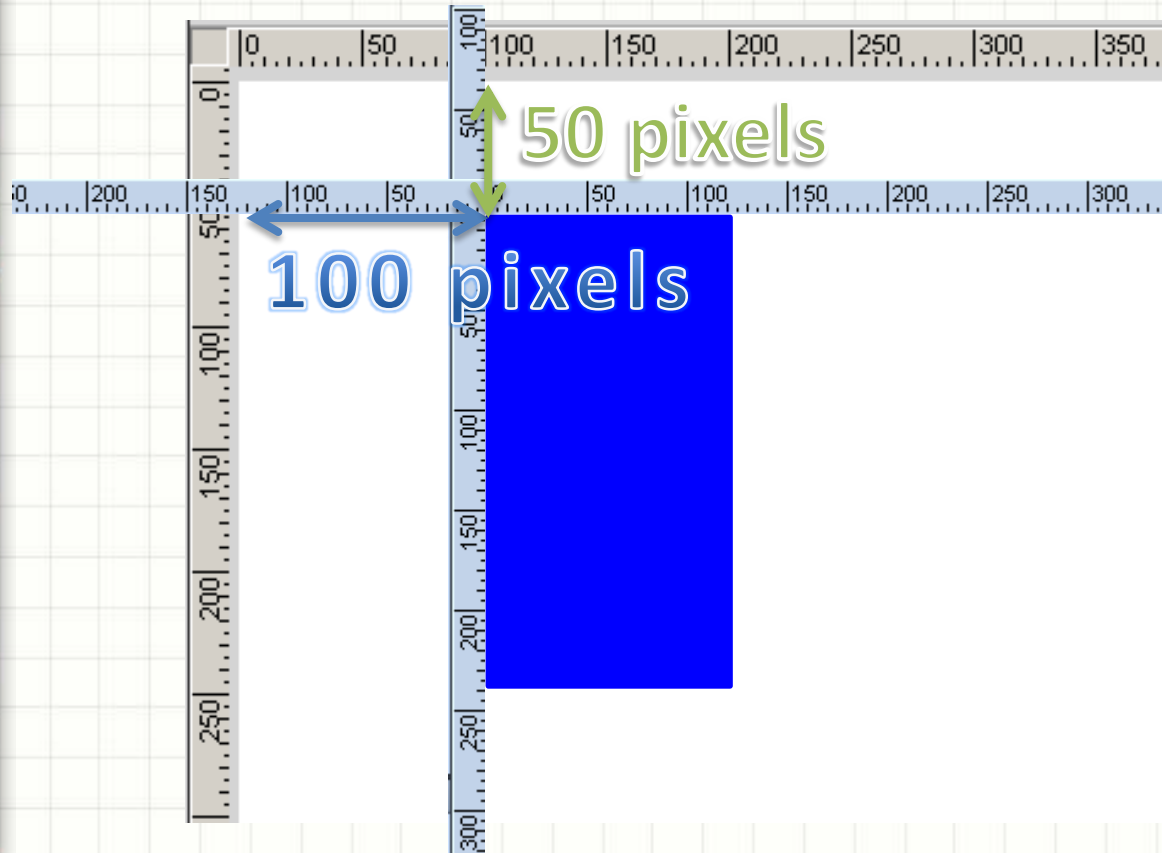
Coordenadas do AS3

- Um sprite, tem seus próprios eixos



Coordenadas do AS3

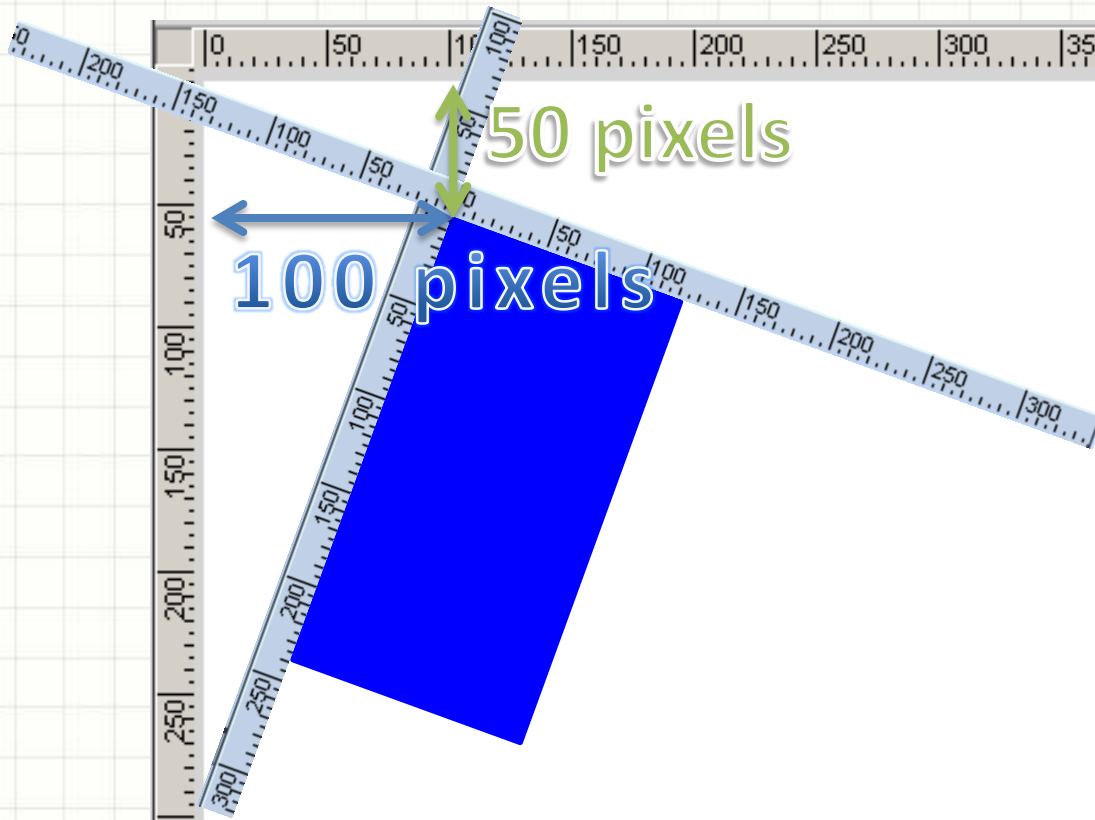
- Quando posicionamos os Sprites, posicionamos o 0,0 do sprite no cenário



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

Coordenadas do AS3

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

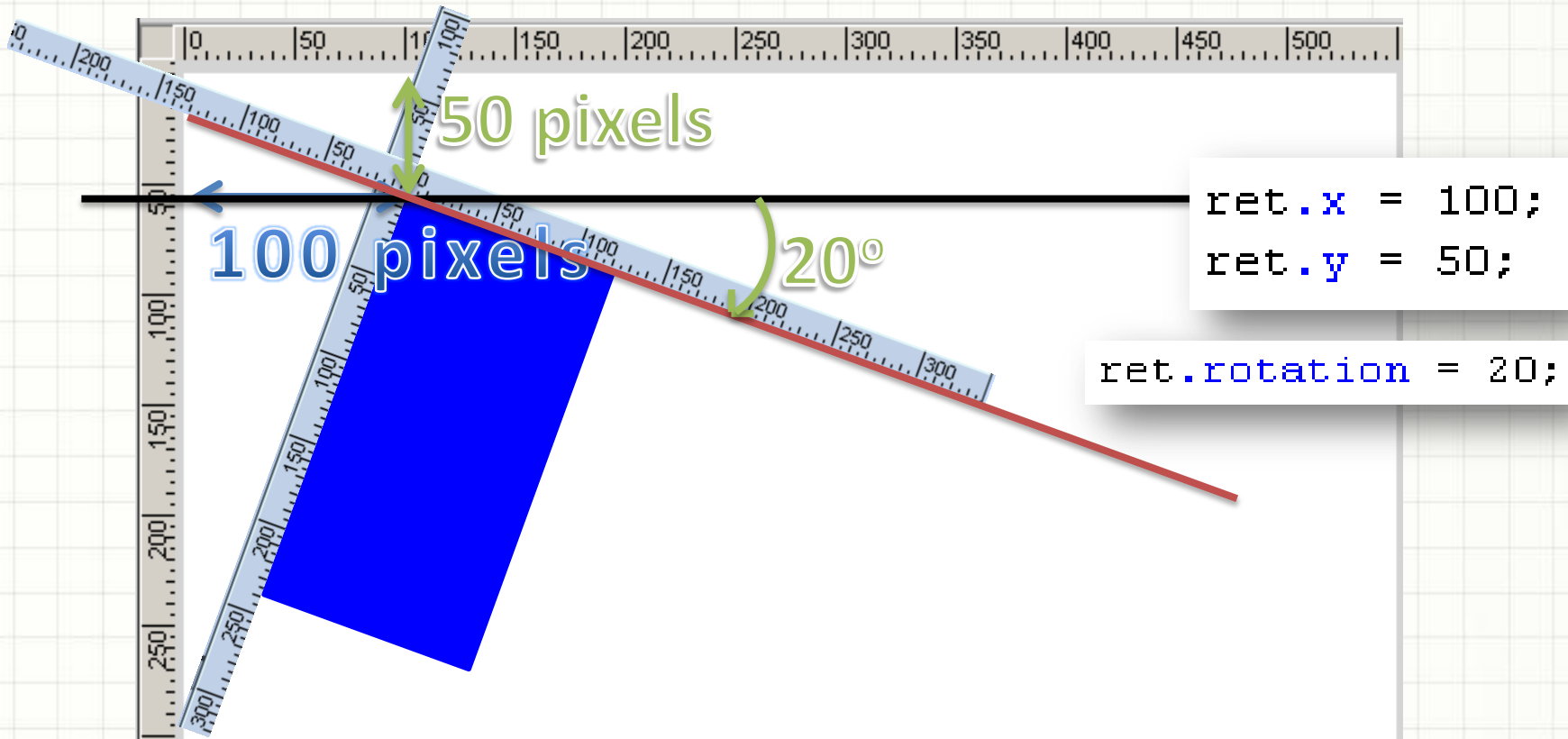


```
ret.x = 100;  
ret.y = 50;
```

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

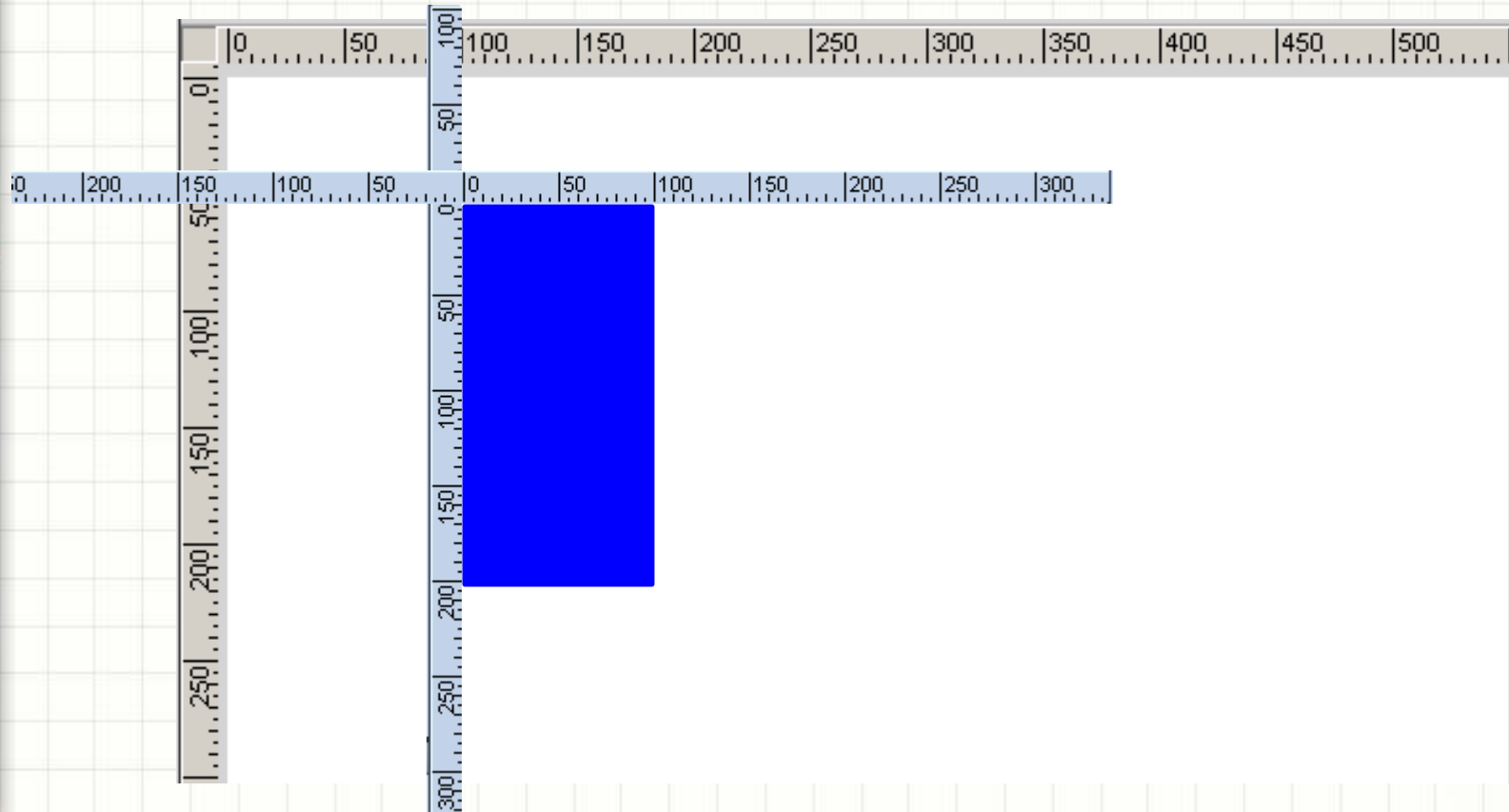
Coordenadas do AS3

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



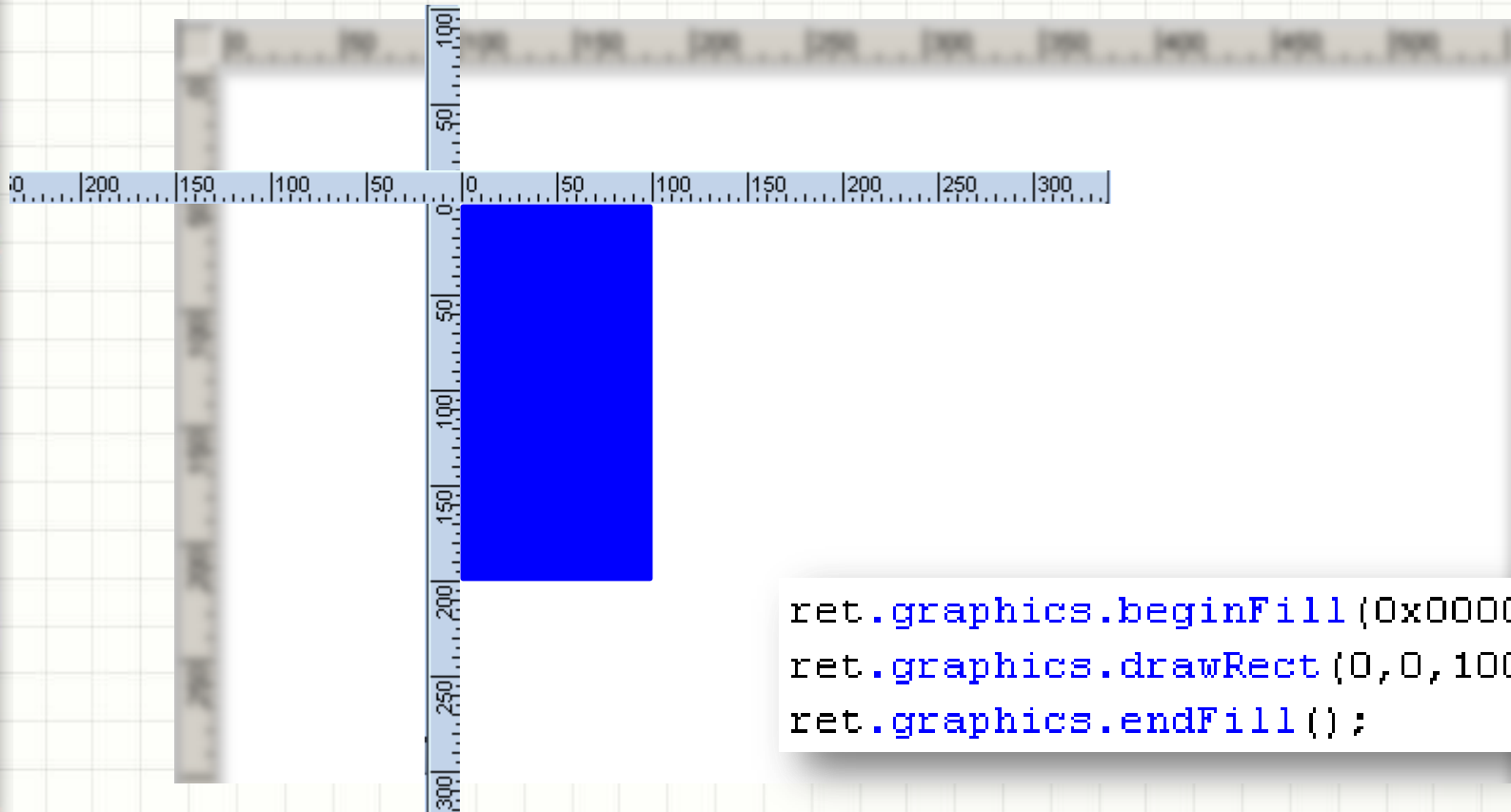
Coordenadas do AS3

- Para que servem os eixos do sprite?



Coordenadas do AS3

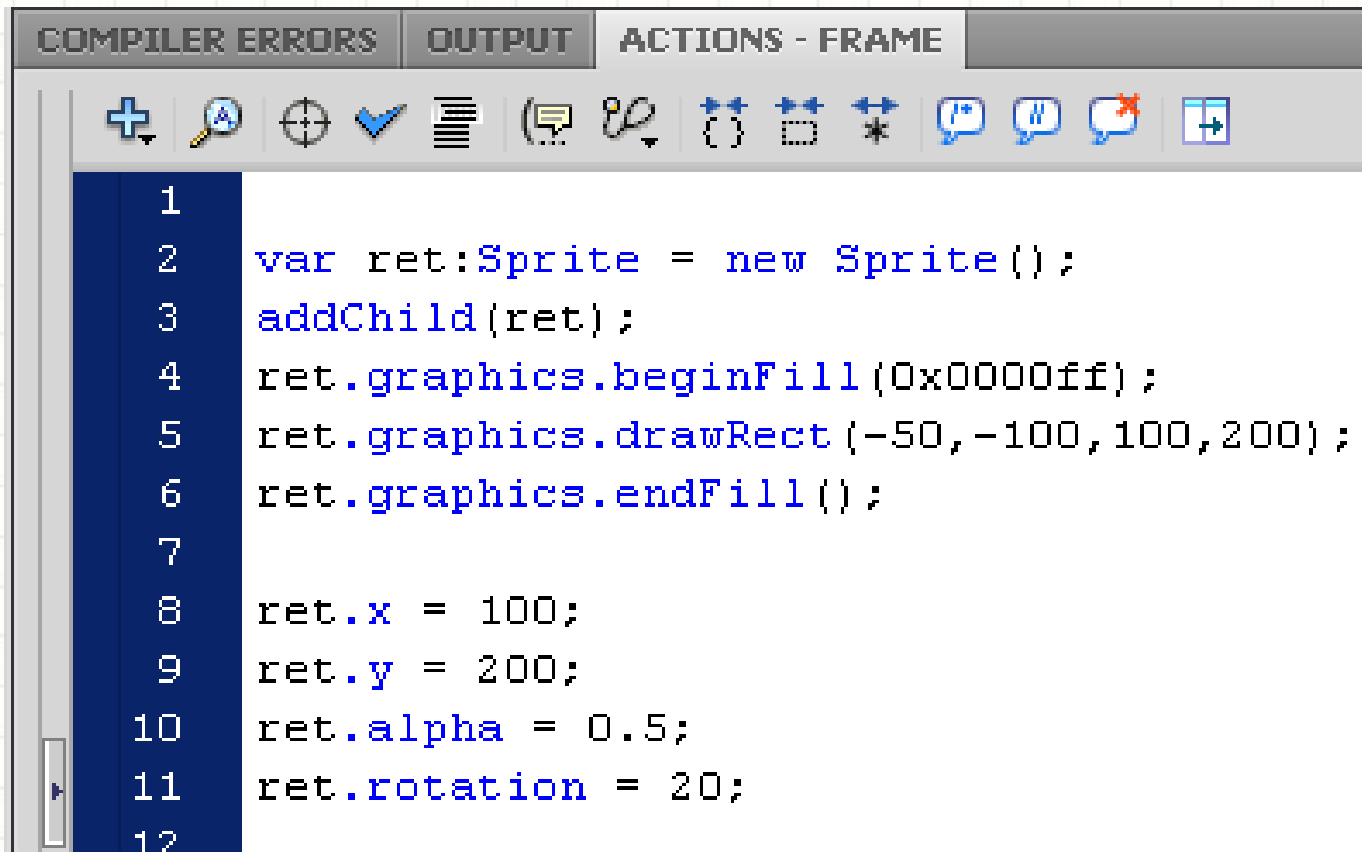
- Para que servem os eixos do sprite?
- Ao **desenhar** o sprite, são eles que valem!



```
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?



The screenshot shows a code editor window titled 'ACTIONS - FRAME'. The code is as follows:

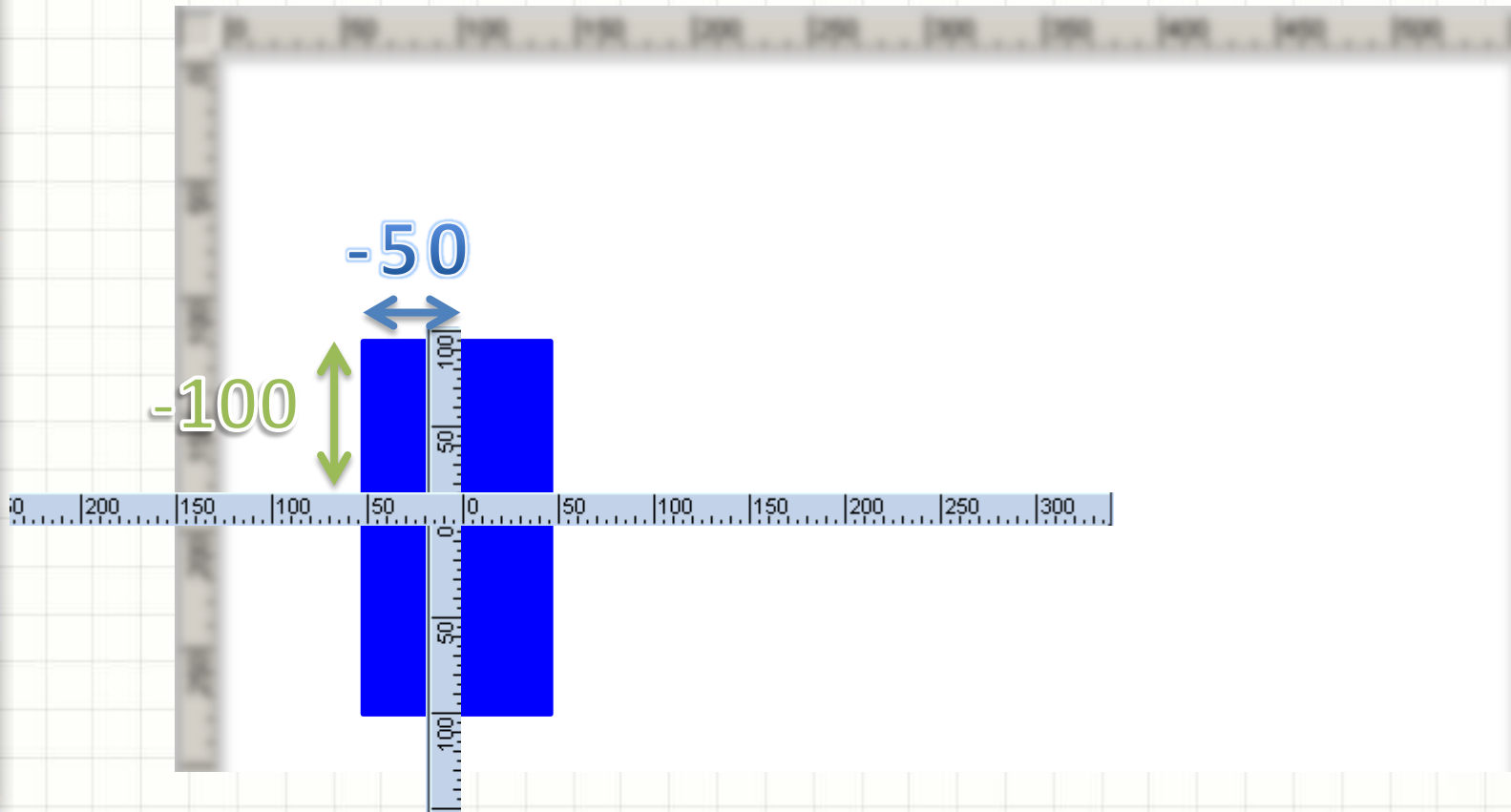
```
1  
2 var ret:Sprite = new Sprite();  
3 addChild(ret);  
4 ret.graphics.beginFill(0x0000ff);  
5 ret.graphics.drawRect(-50,-100,100,200);  
6 ret.graphics.endFill();  
7  
8 ret.x = 100;  
9 ret.y = 200;  
10 ret.alpha = 0.5;  
11 ret.rotation = 20;  
12
```

- Experimente!

Desenhando em ActionScript 3

- Observe como desenhamos...

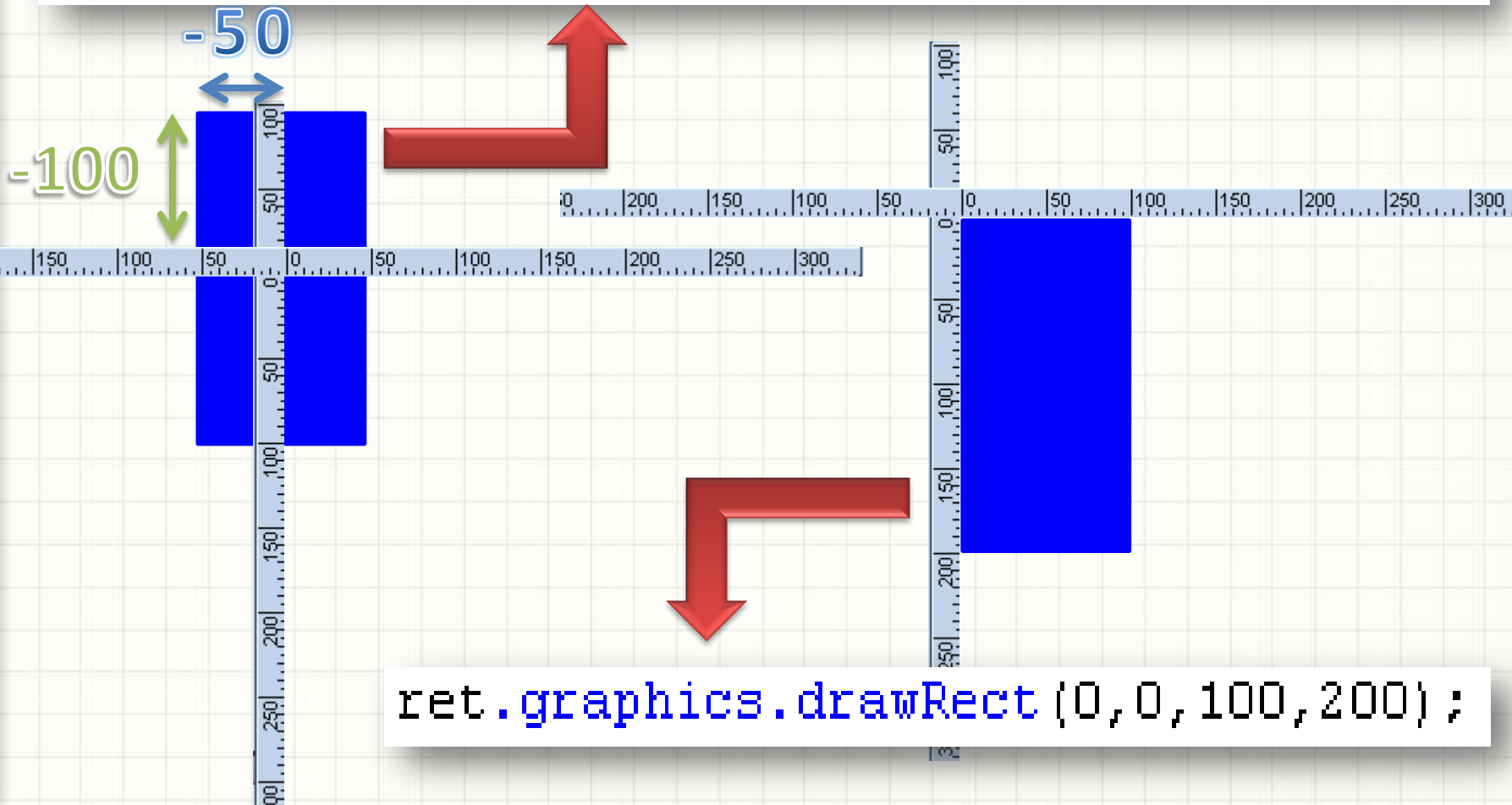
```
ret.graphics.drawRect (-50, -100, 100, 200) ;
```



Desenhando em ActionScript 3

- Compare

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



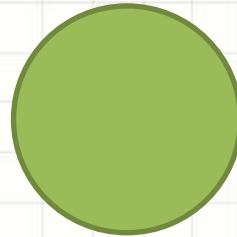
```
ret.graphics.drawRect (0,0,100,200) ;
```



ATIVIDADE

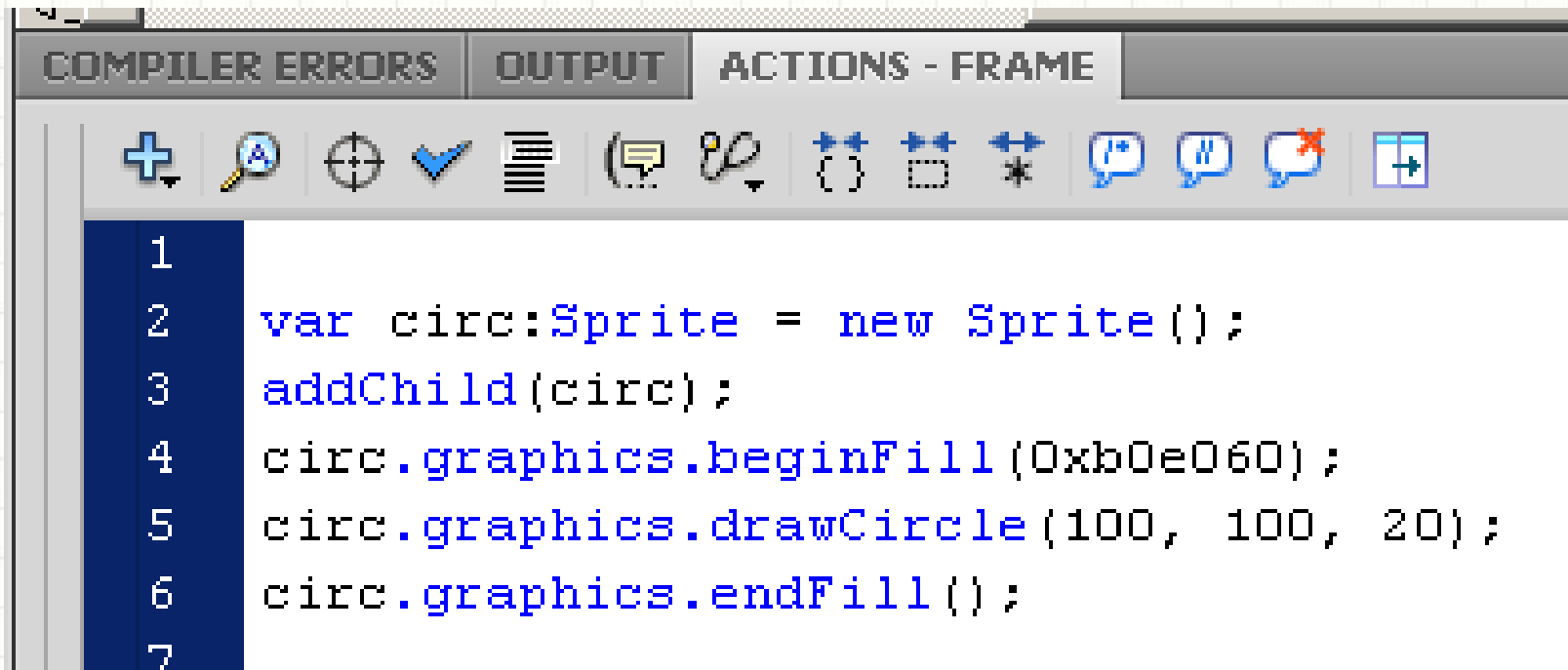
Atividade

- 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

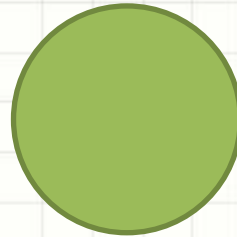


The image shows a screenshot of an IDE window with three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'ACTIONS - FRAME' tab is active. Below the tabs is a toolbar with various icons including a plus sign, a magnifying glass, a crosshair, a checkmark, a list, a speech bubble, a pencil, a pair of brackets, a square, an asterisk, a speech bubble with a plus sign, a speech bubble with a 'W', a speech bubble with a red 'X', and a square with a plus sign. The main area of the IDE contains 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

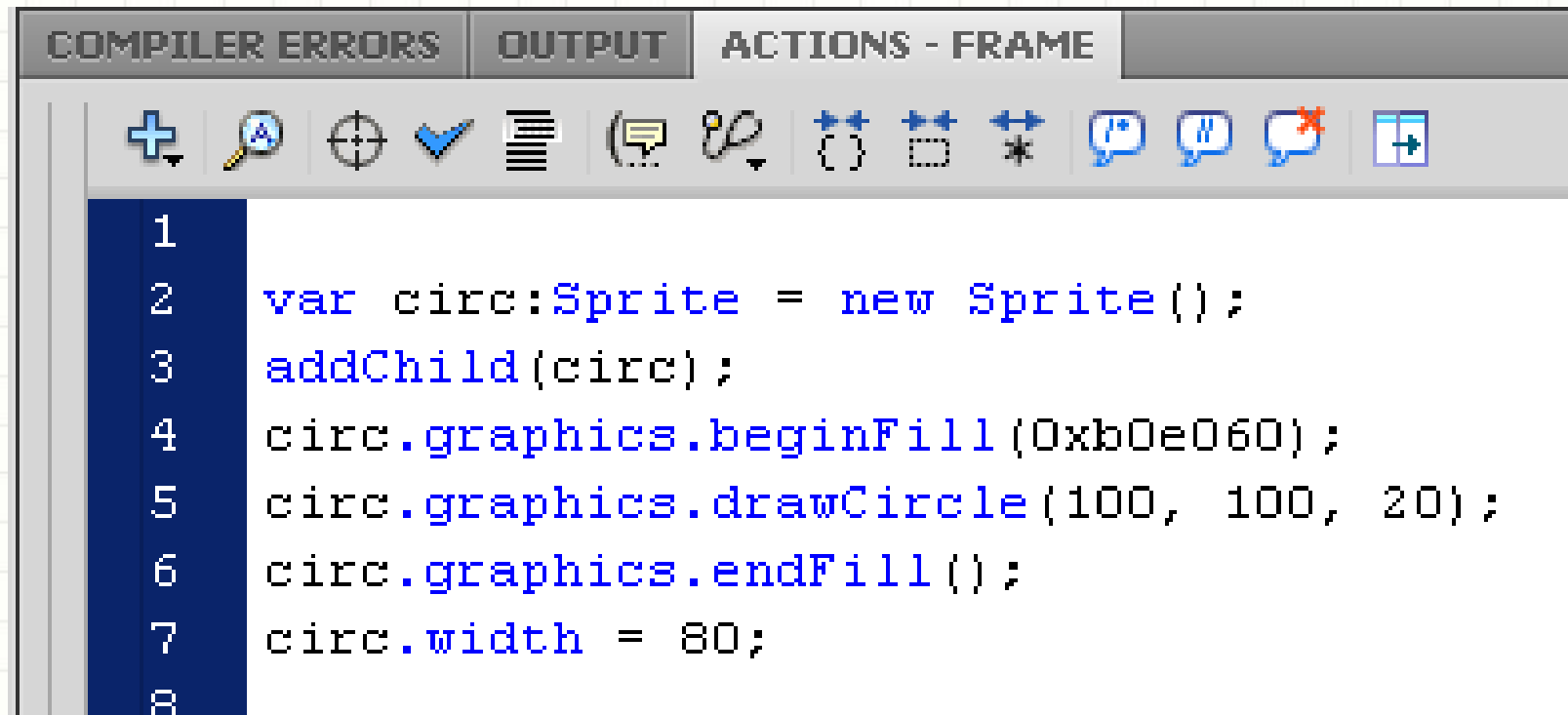
- Faça um programa que mostre 1 círculo verde de raio 20 pixels



- 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!

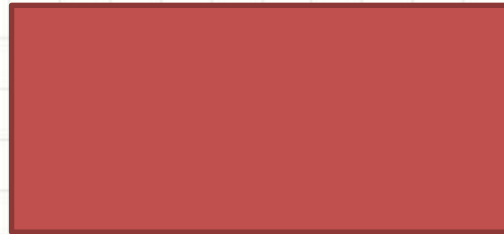


The image shows a screenshot of an IDE window titled "ACTIONS - FRAME". The window has a toolbar with various icons for editing and debugging. The code editor displays the following code:

```
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
```

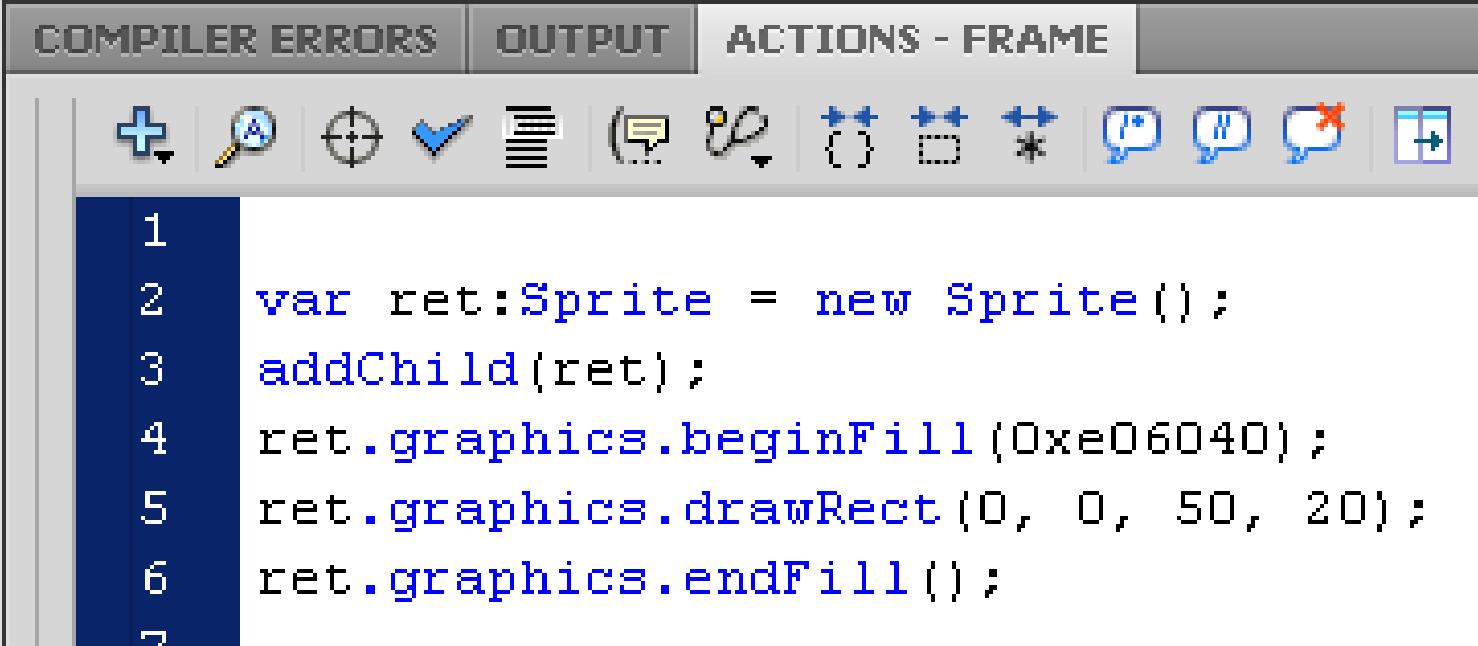
Atividade

- 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

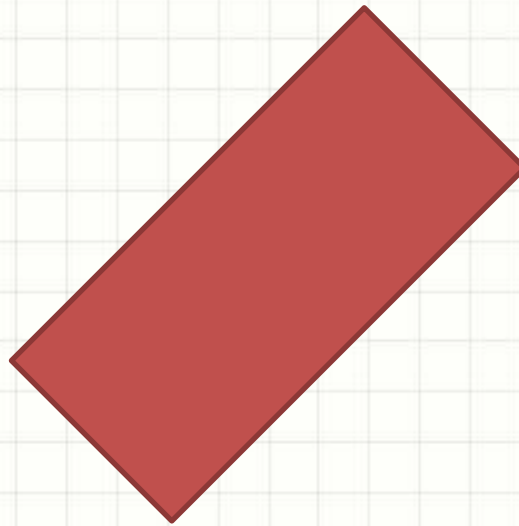


The image shows a screenshot of an IDE interface. At the top, there are three tabs: "COMPILER ERRORS", "OUTPUT", and "ACTIONS - FRAME". Below the tabs is a toolbar with various icons for editing and development. The main area is a code editor with a dark blue background. The code is written in a light blue font and consists of six lines of code. The first line is a blank line. The second line is `var ret:Sprite = new Sprite();`. The third line is `addChild(ret);`. The fourth line is `ret.graphics.beginFill(0xe06040);`. The fifth line is `ret.graphics.drawRect(0, 0, 50, 20);`. The sixth line is `ret.graphics.endFill();`. The line numbers 1 through 7 are visible on the left side of the editor.

```
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
```

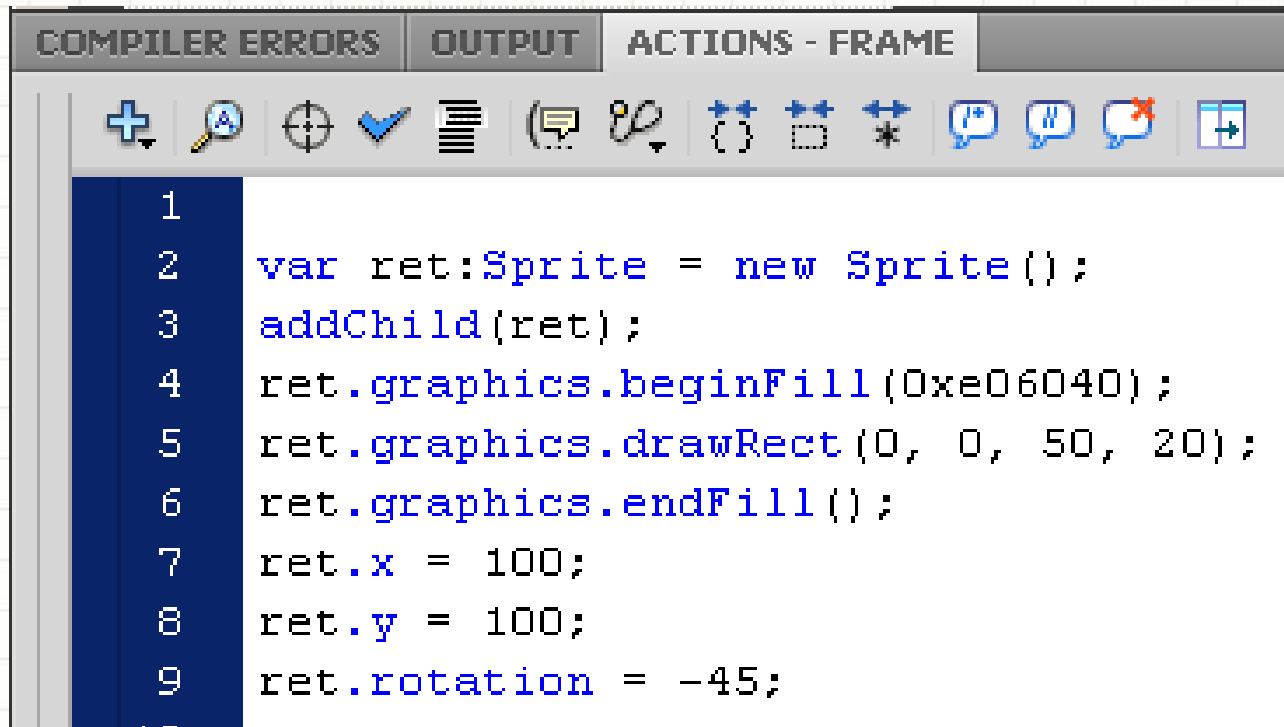

Atividade

- 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



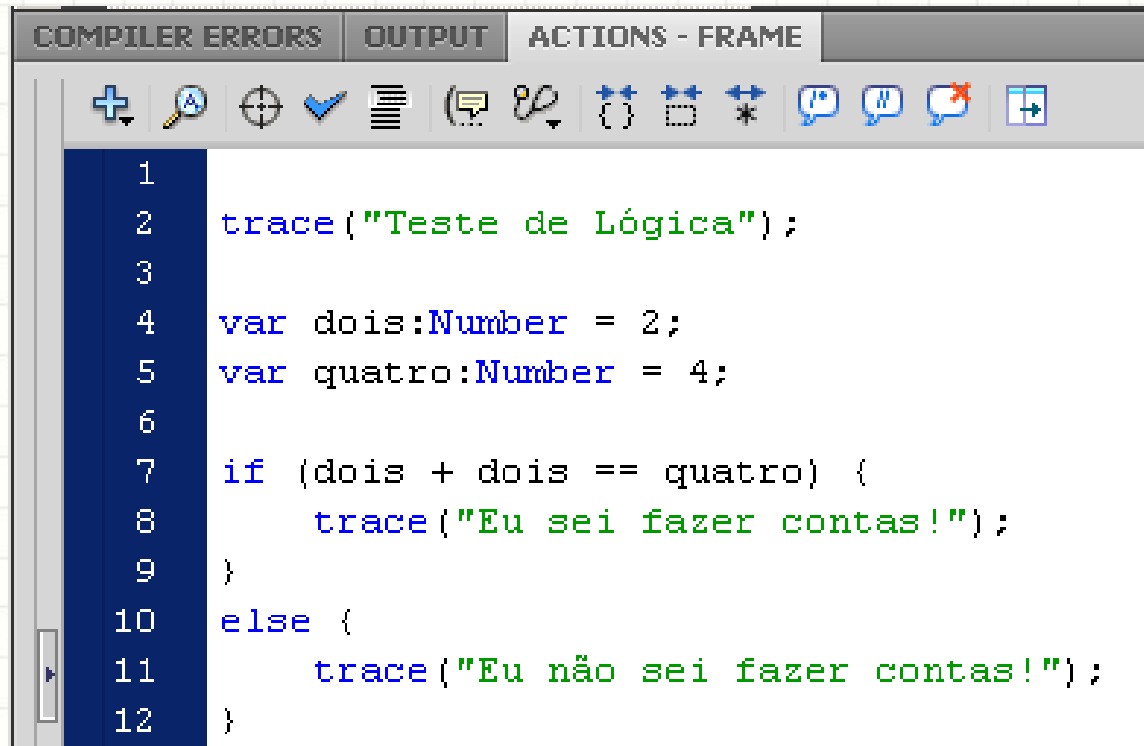
```
COMPILER ERRORS | OUTPUT | ACTIONS - FRAME  
+ 🔍 📏 ✓ ☰ (💬) ✍️ {} ☐ * 💬 💬 ✖️ 📄  
1  
2 var ret:Sprite = new Sprite();  
3 addChild(ret);  
4 ret.graphics.beginFill(0xe06040);  
5 ret.graphics.drawRect(0, 0, 50, 20);  
6 ret.graphics.endFill();  
7 ret.x = 100;  
8 ret.y = 100;  
9 ret.rotation = -45;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



LÓGICA E LAÇOS

Logica e Laços em ActionScript 3

- As estruturas de decisão são idênticas ao C

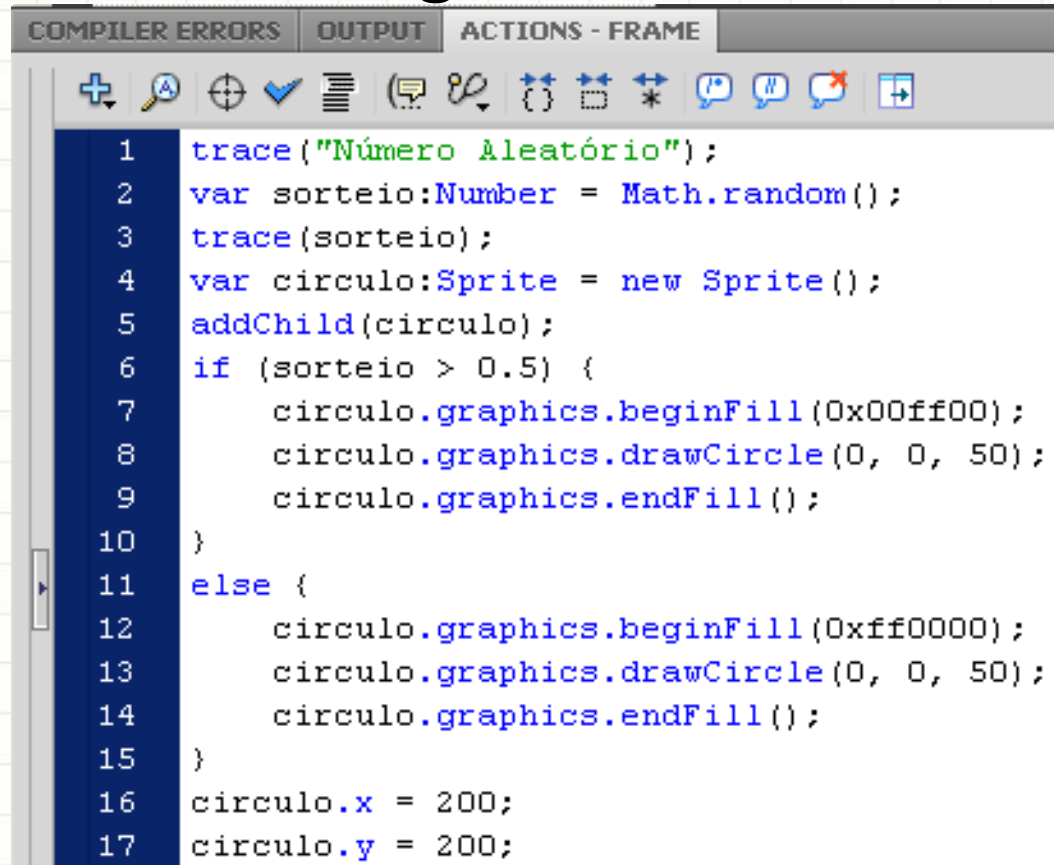


```
COMPILER ERRORS OUTPUT ACTIONS - FRAME
+ 🔍 📍 ✓ ☰ (💬) 🔄 {} 📅 * 💬 💬 💬 📄
1
2 trace("Teste de Lógica");
3
4 var dois:Number = 2;
5 var quatro:Number = 4;
6
7 if (dois + dois == quatro) {
8     trace("Eu sei fazer contas!");
9 }
10 else {
11     trace("Eu não sei fazer contas!");
12 }
```

- 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

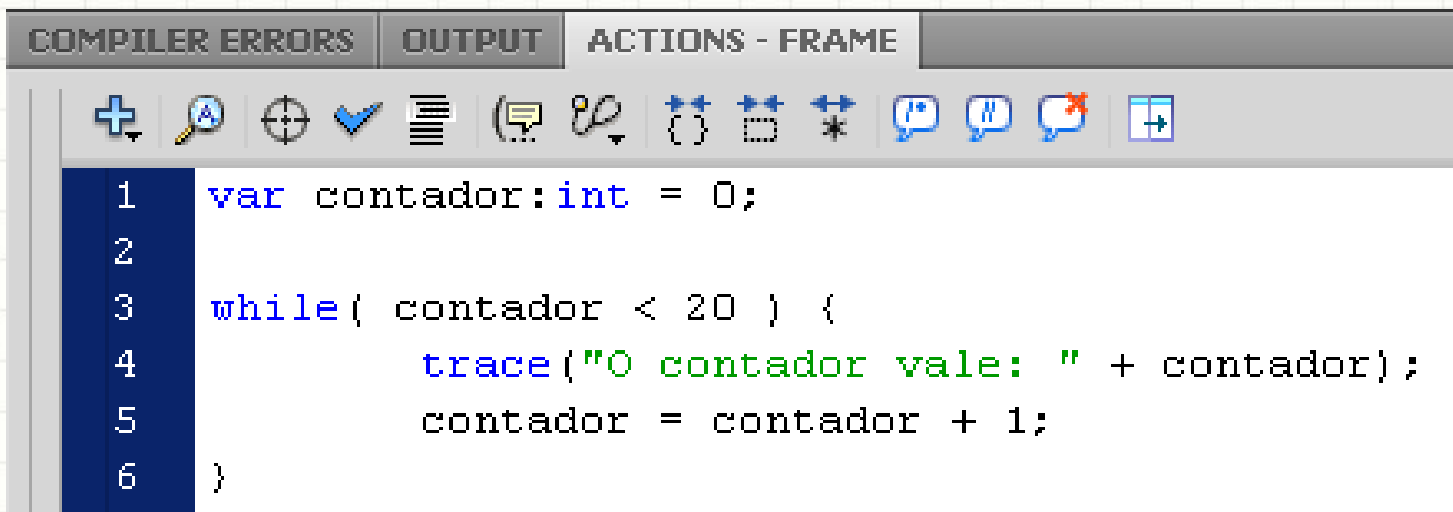
ACTIONS - FRAME



```
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



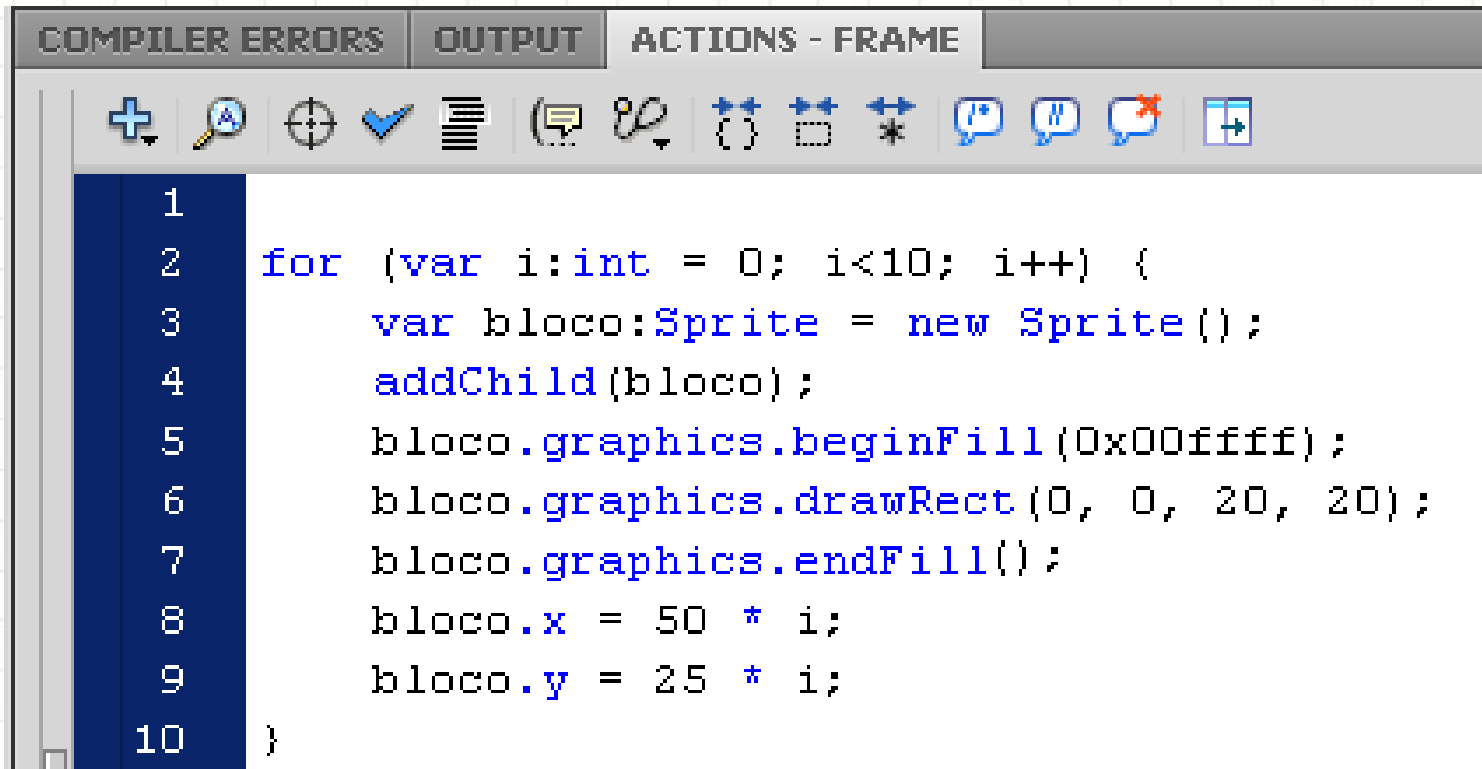
The screenshot shows a code editor window titled 'ACTIONS - FRAME'. The editor contains the following ActionScript 3 code:

```
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...



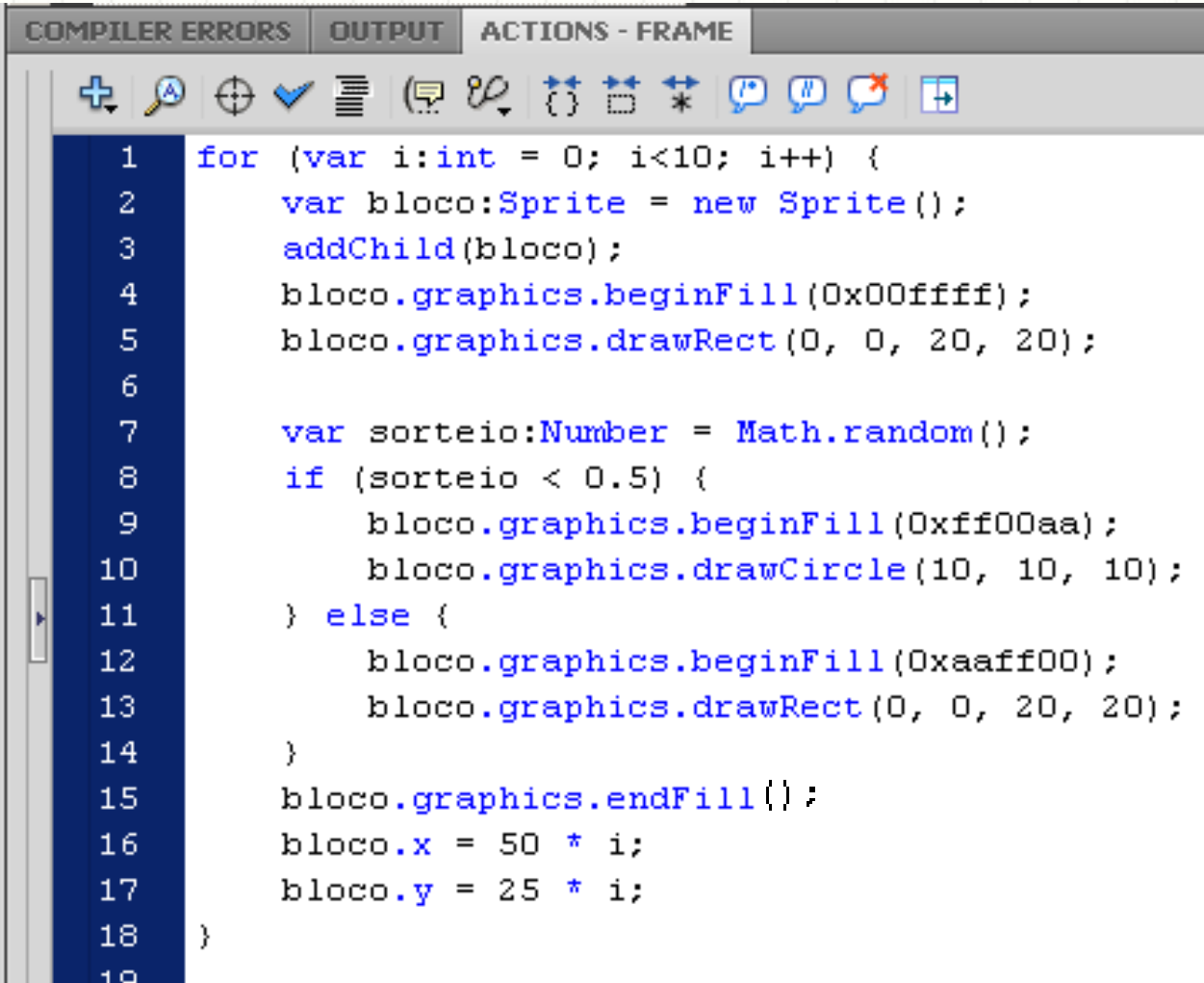
The screenshot shows an IDE window titled "ACTIONS - FRAME" with a toolbar and a code editor. The code editor contains the following ActionScript 3 code:

```
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(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
```



```
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(0xaa00ff);
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
```



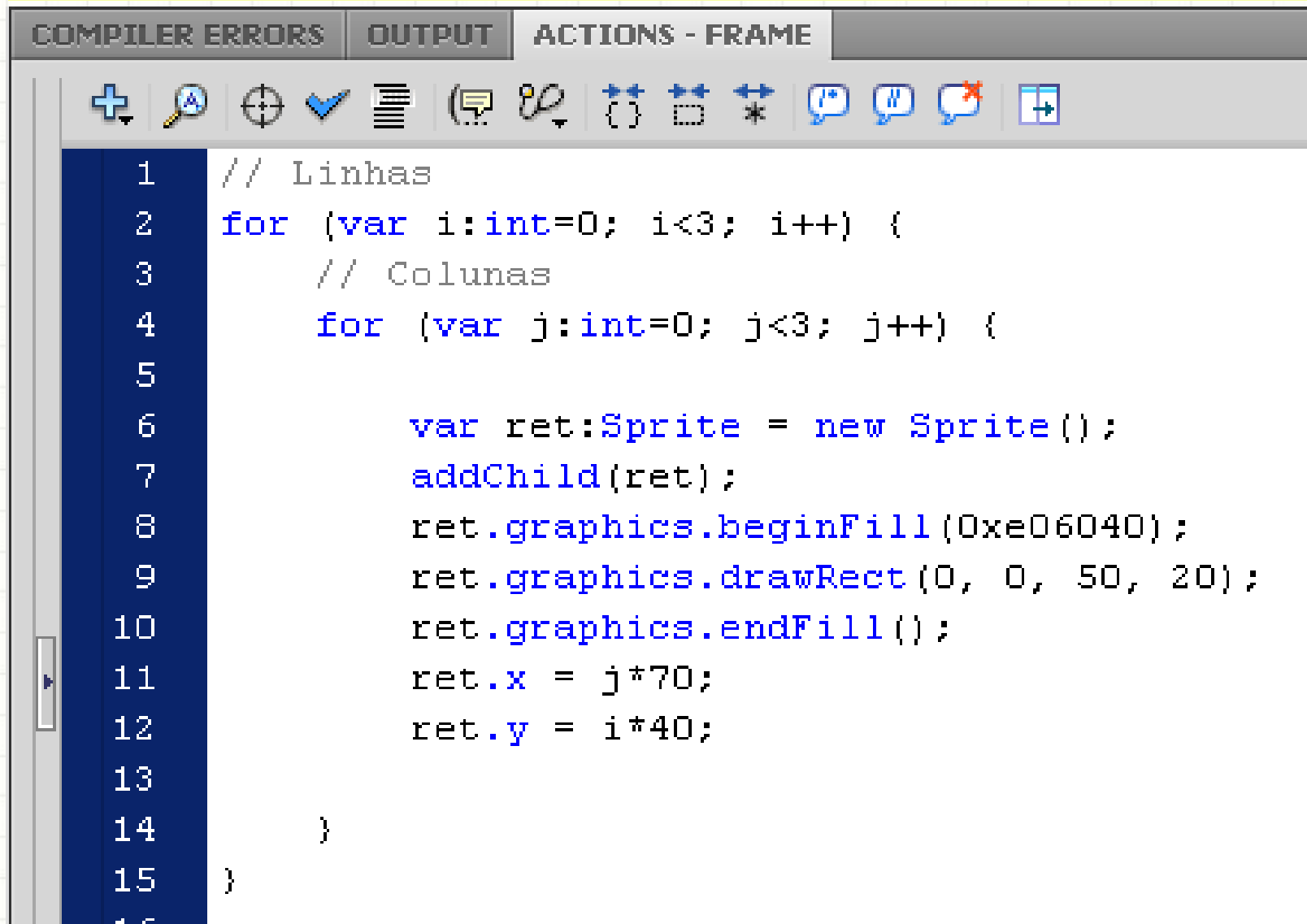
ATIVIDADE

Atividade

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



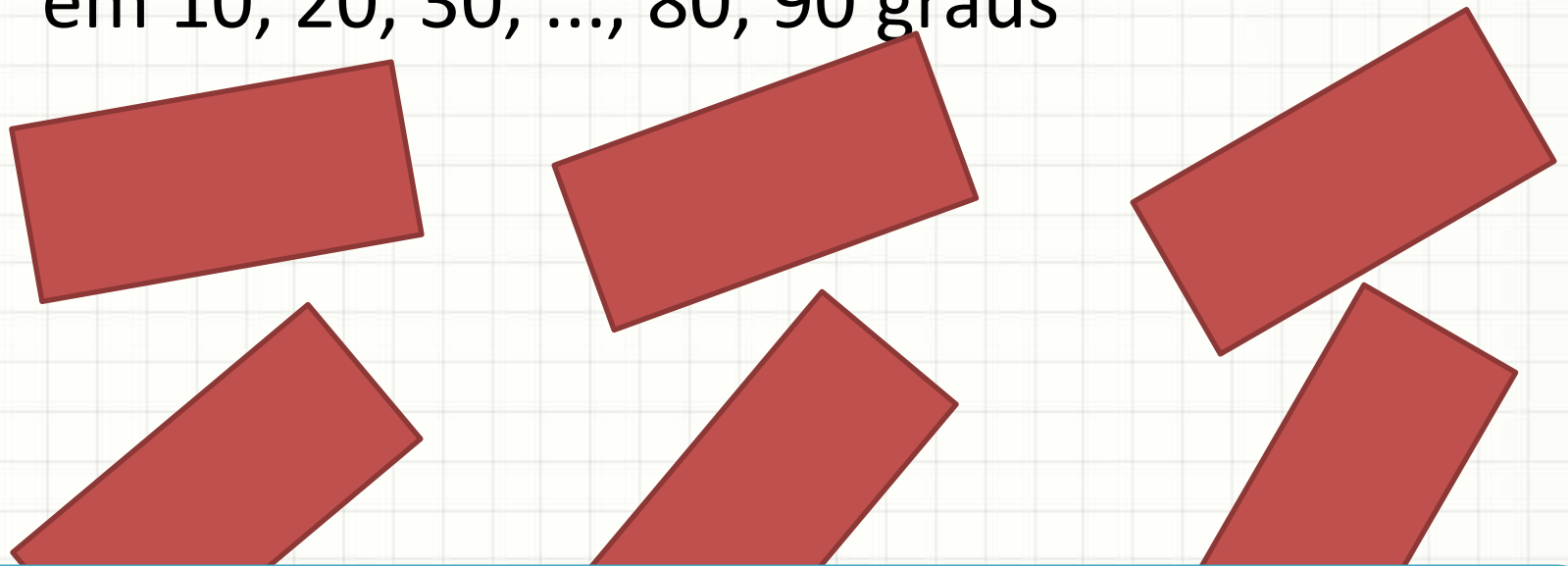
Atividade - Solução



```
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 }
```



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!**