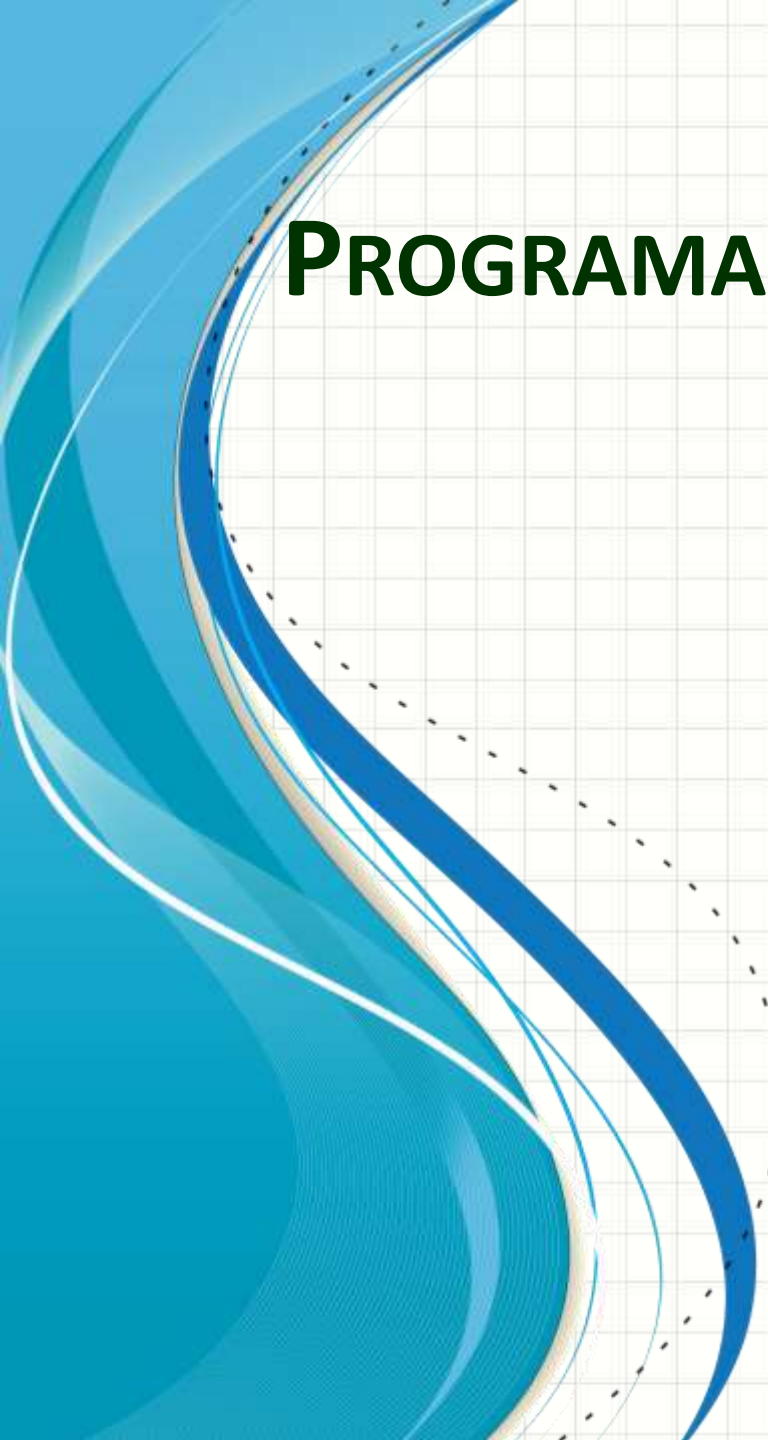




PROGRAMAÇÃO PARA INTERNET RICA

INTRODUÇÃO AO ACTIONSCRIPT 3

Prof. Dr. Daniel Caetano

2012 - 1

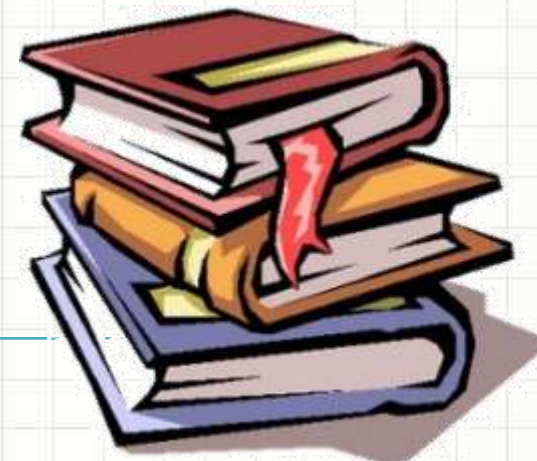
Objetivos

- 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/aulas/pir/>
Adobe Flash CS4

Apresentação

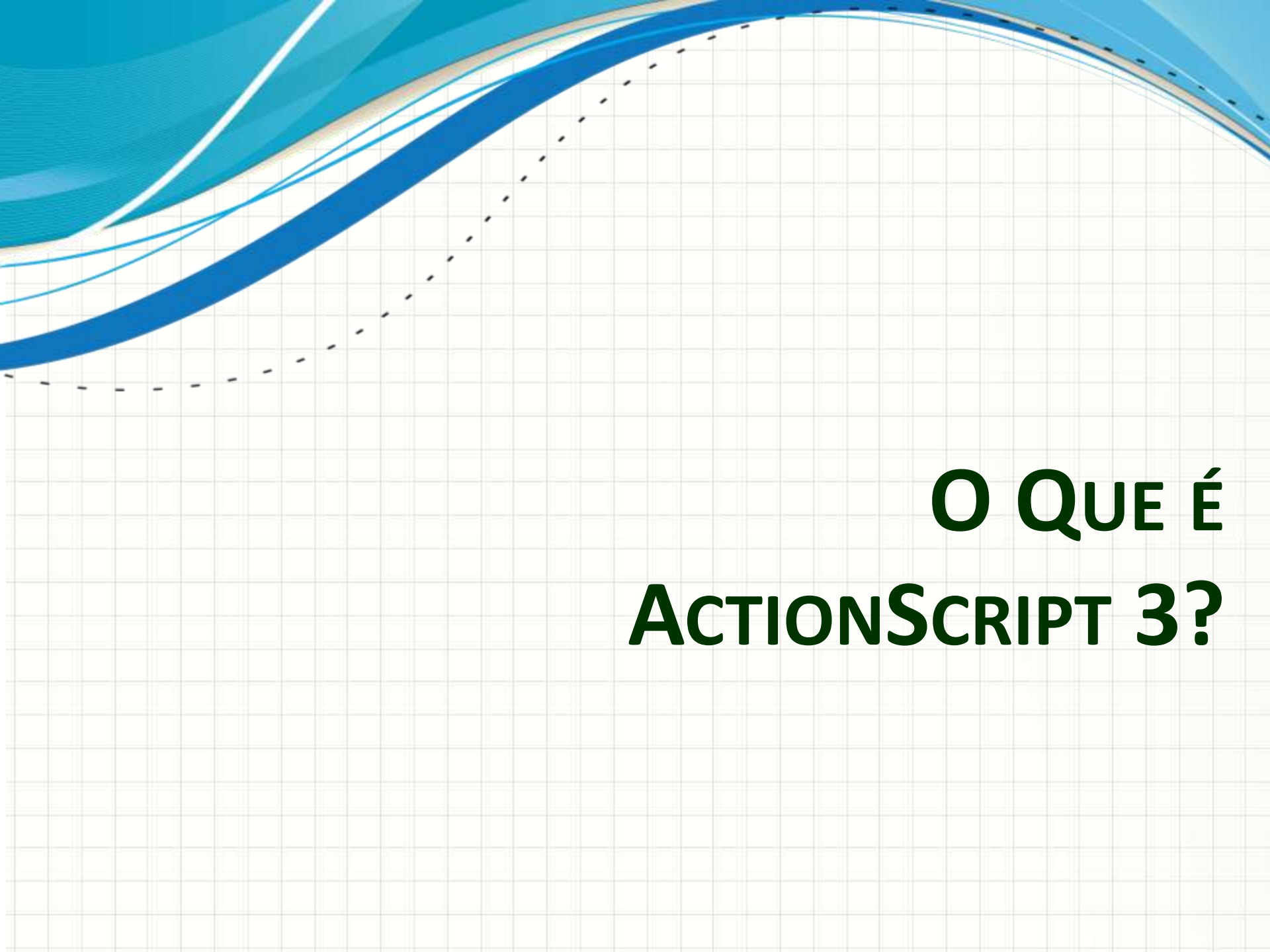
<http://www.caetano.eng.br/aulas/pir/>
(Aula 12)

Material Didático

-

Google

ActionScript 3
AS3



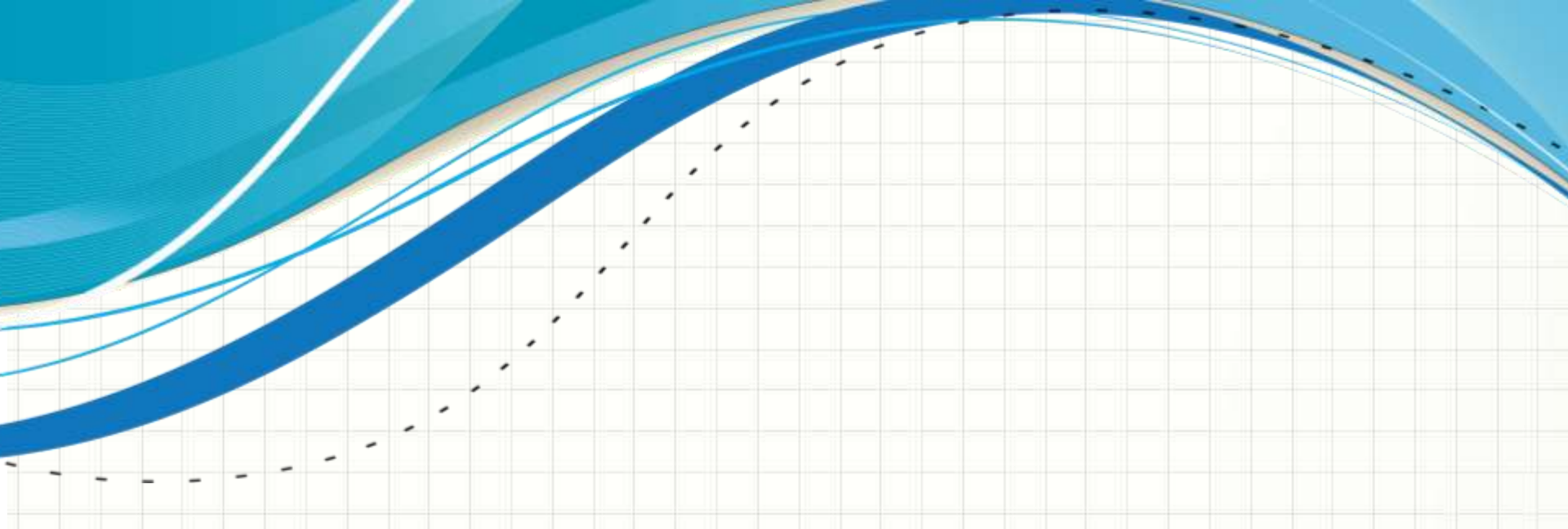
O QUE É ACTIONSCRIPT 3?

Introdução

- 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

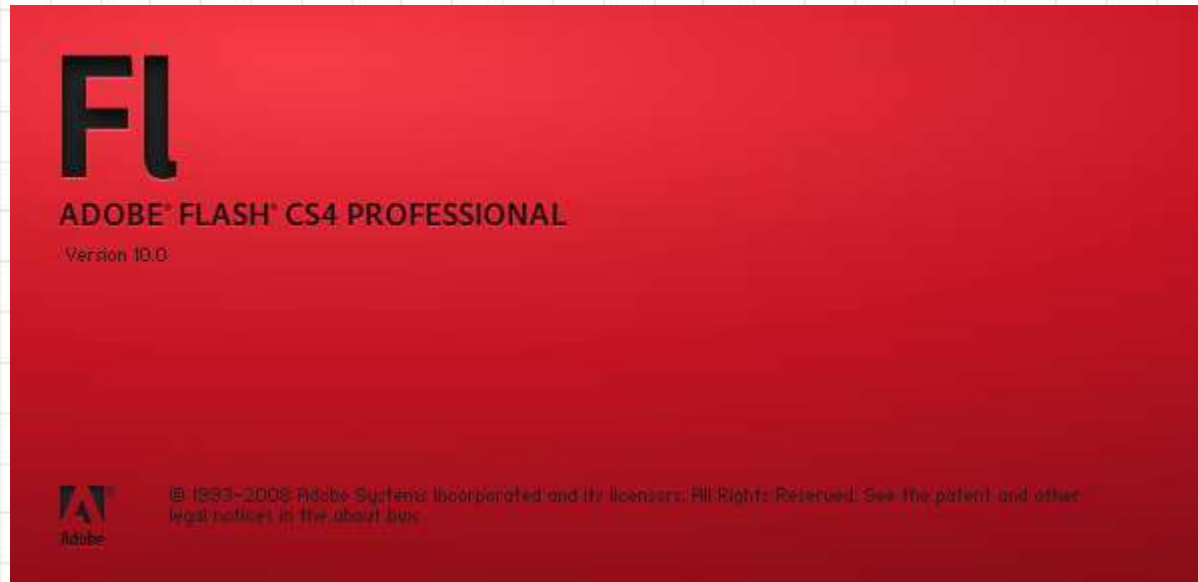
ActionScript 3 na Prática

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



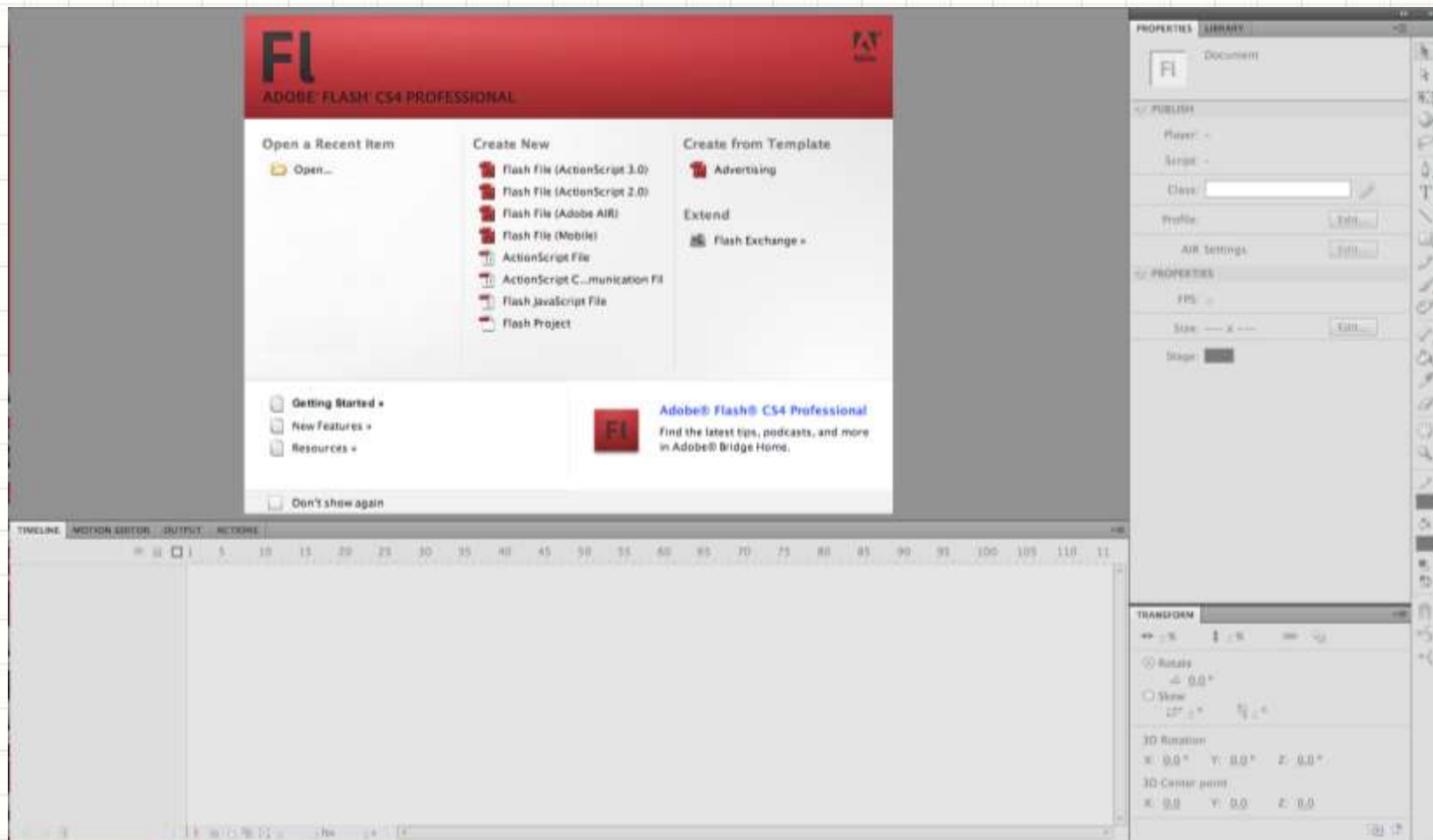
ActionScript 3 na Prática

- Aguarde o carregamento...



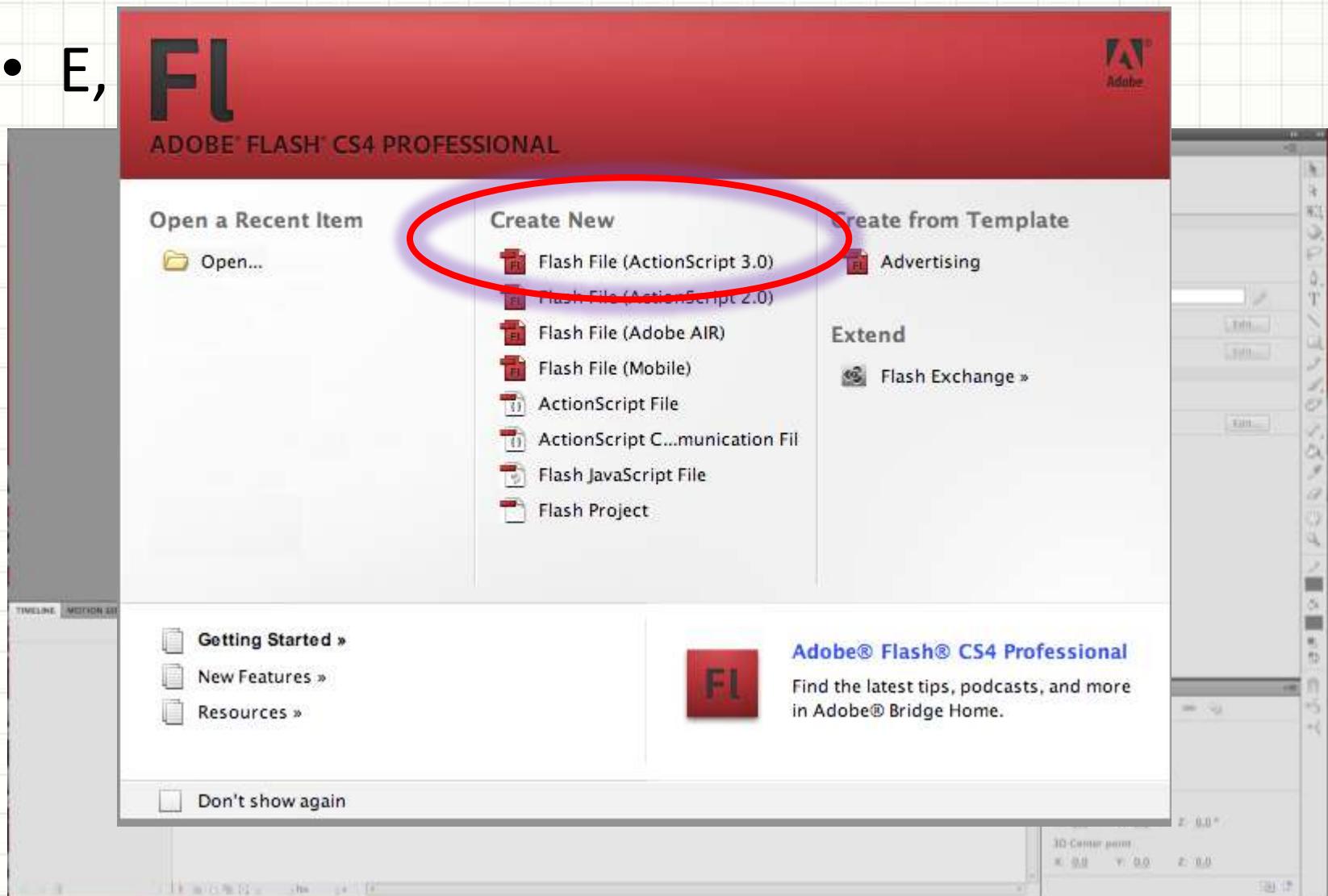
ActionScript 3 na Prática

- E, finalmente, uma janela aparecerá...



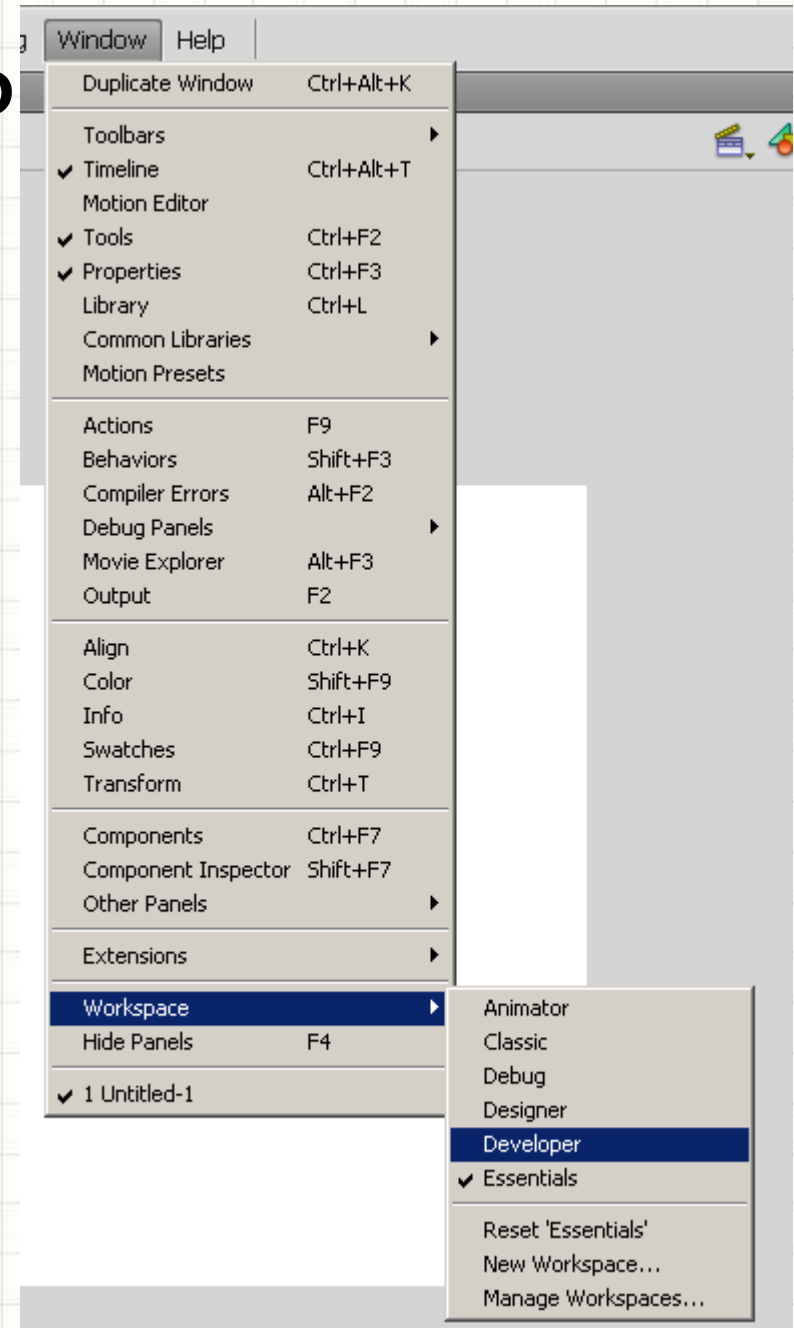
ActionScript 3 na Prática

- E,



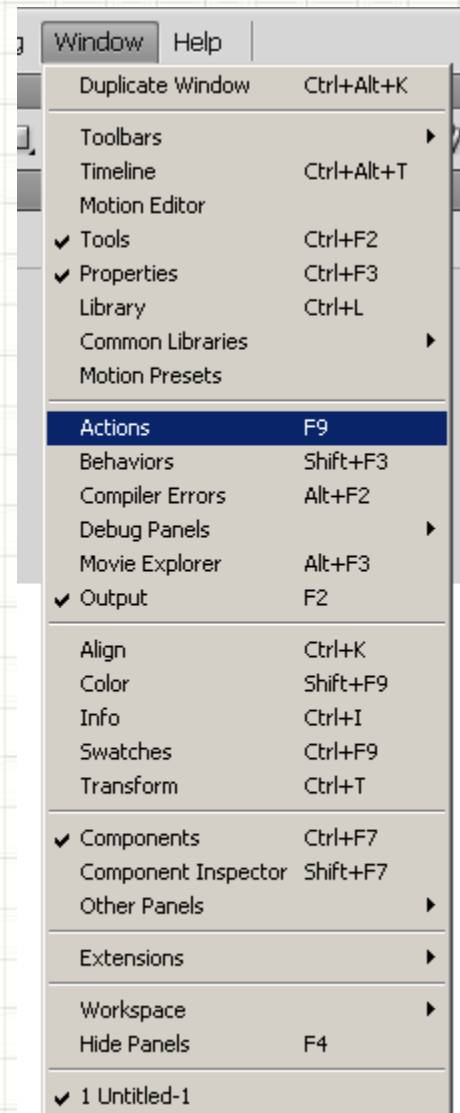
ActionScript 3 na P

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



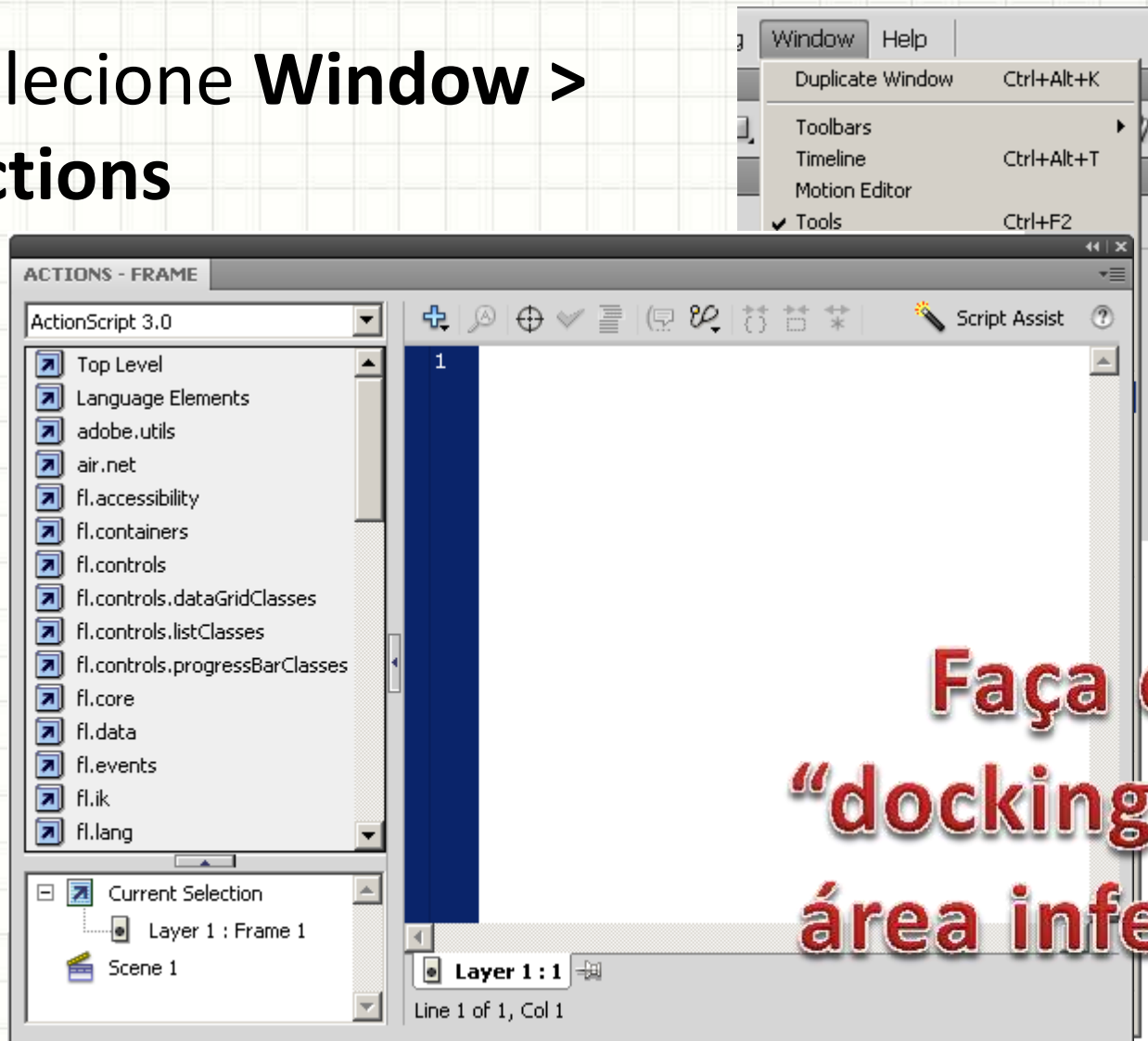
ActionScript 3 na Prática

- Selecione **Window > Actions**



ActionScript 3 na Prática

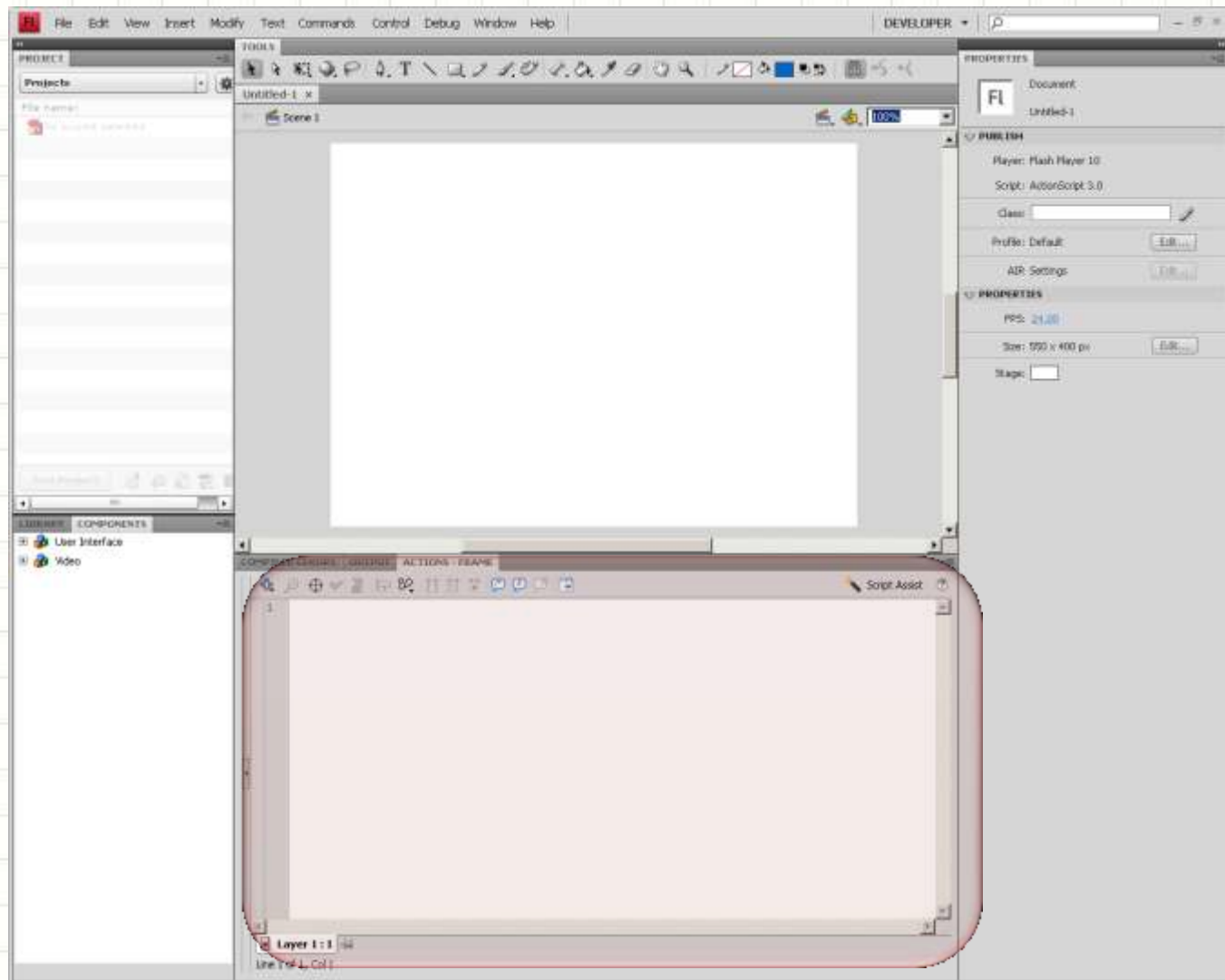
- Selecione **Window > Actions**



Faça o
“docking” na
área inferior

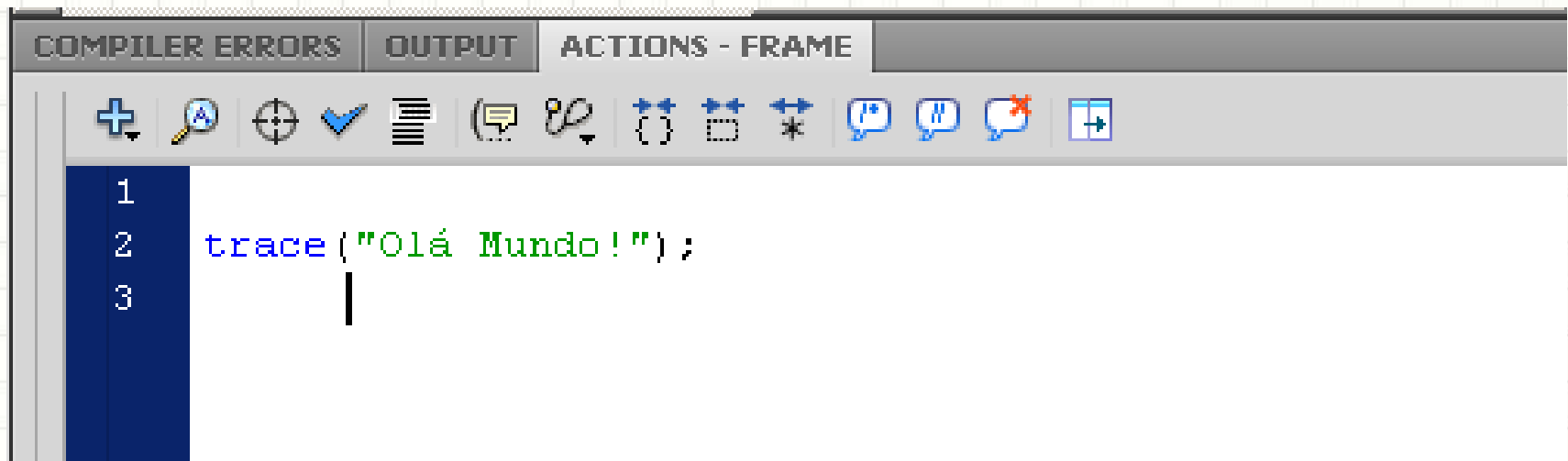
ActionScript 3 na Prática

- Tela de trabalho



ActionScript 3 na Prática

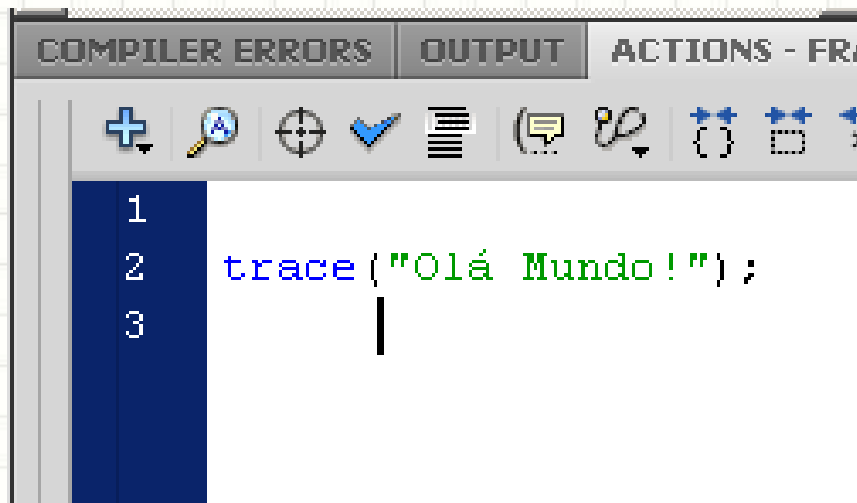
- Primeiro programa...



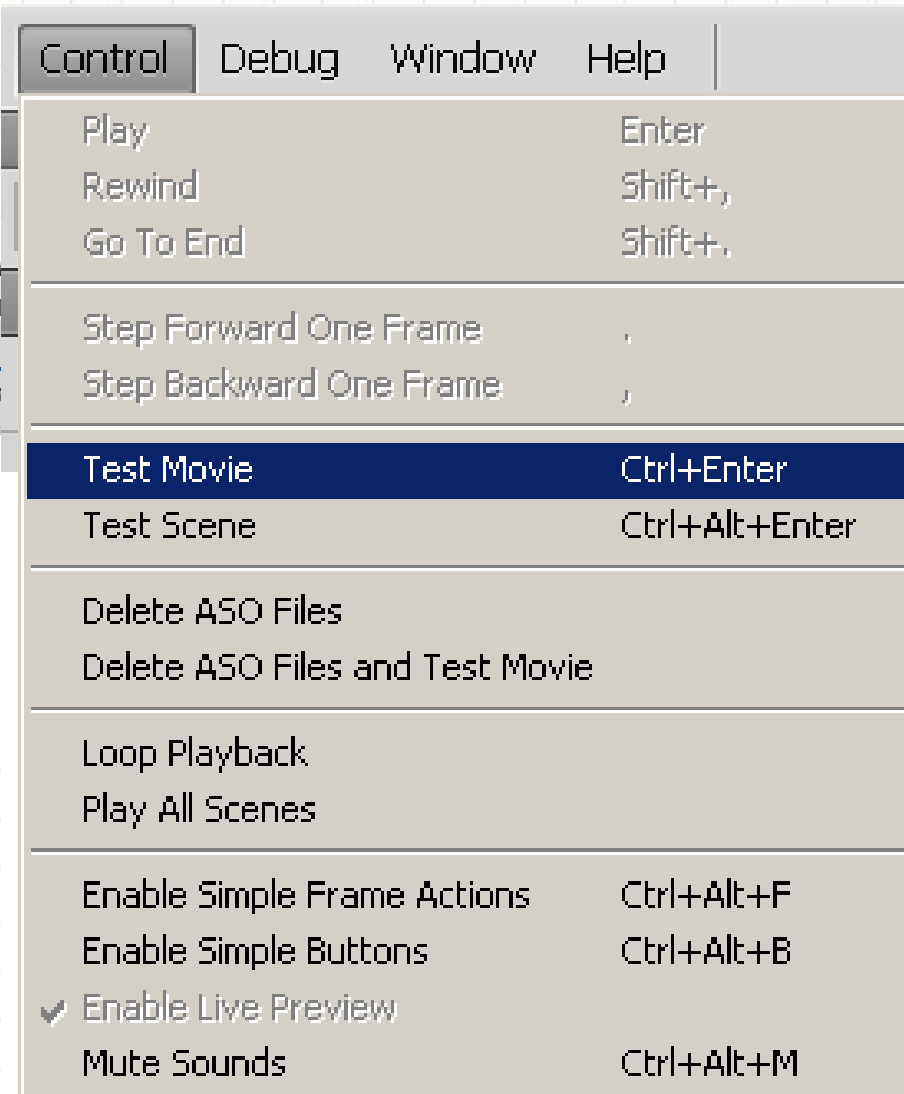
ActionScript 3 na Prática

- Executando...

Control > Test Movie

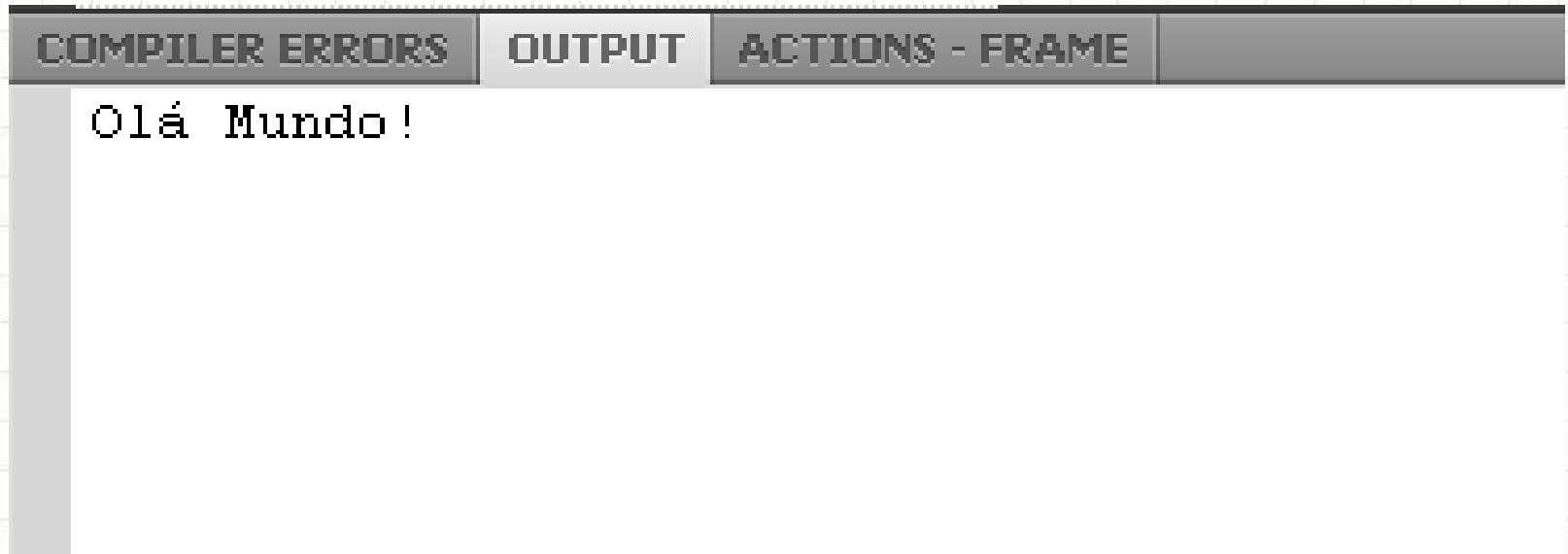


- **Ctrl + Enter** também funciona!



ActionScript 3 na Prática

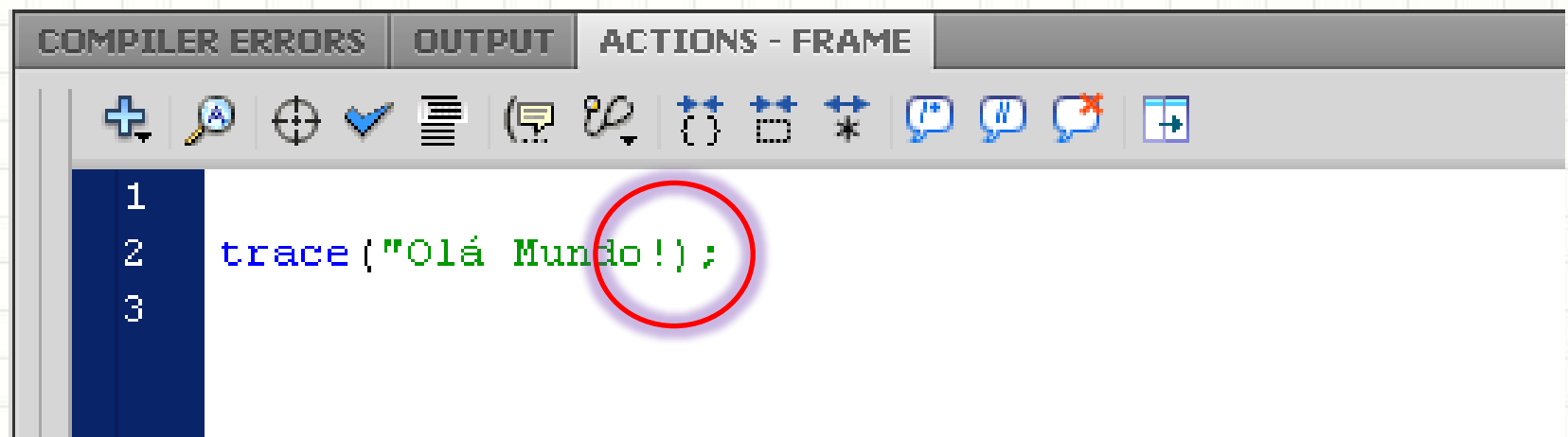
- Resultado...



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

ActionScript 3 na Prática

- E se houver erro...?



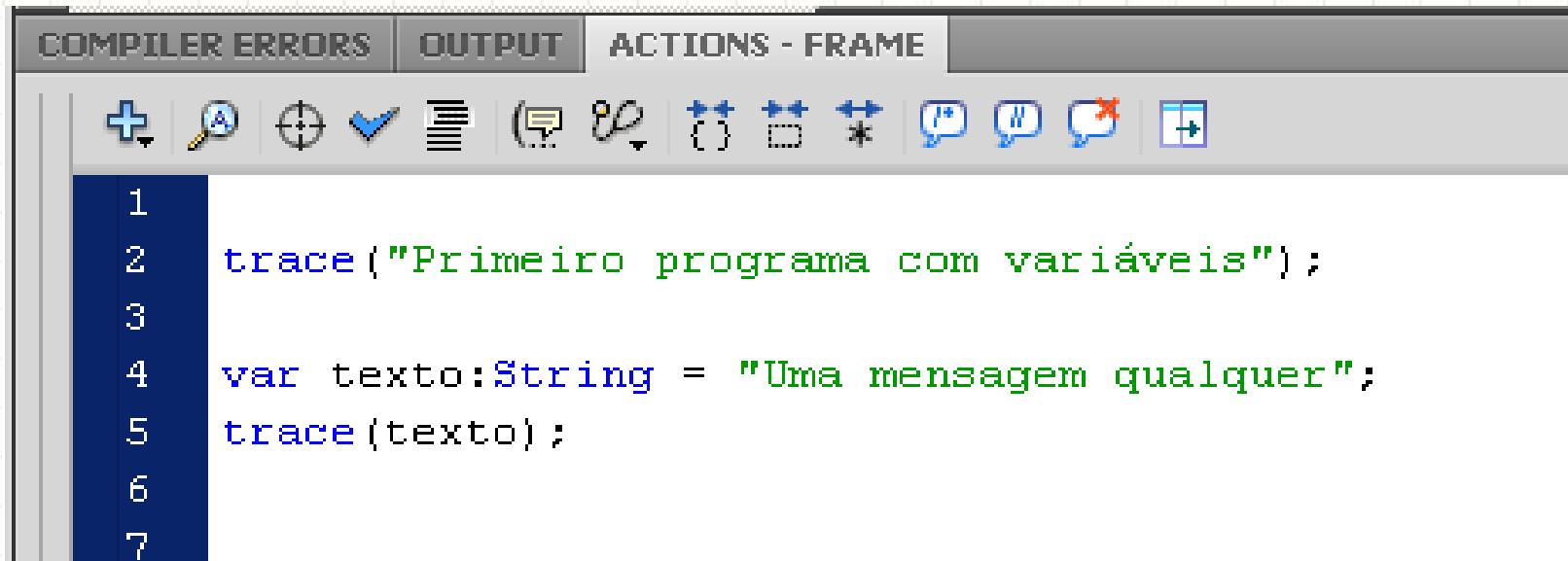
COMPILER ERRORS - 3 REPORTED		OUTPUT	ACTIONS - FRAME
Location	Description	Source	
Scene 1, Layer 'Layer 1', Frame ...	1095: Syntax error: A string literal must be terminated before the line break.	trace("Olá Mundo!");	
Scene 1, Layer 'Layer 1', Frame 1	1084: Syntax error: expecting identifier before 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 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 code editing. The code area contains the following text:

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

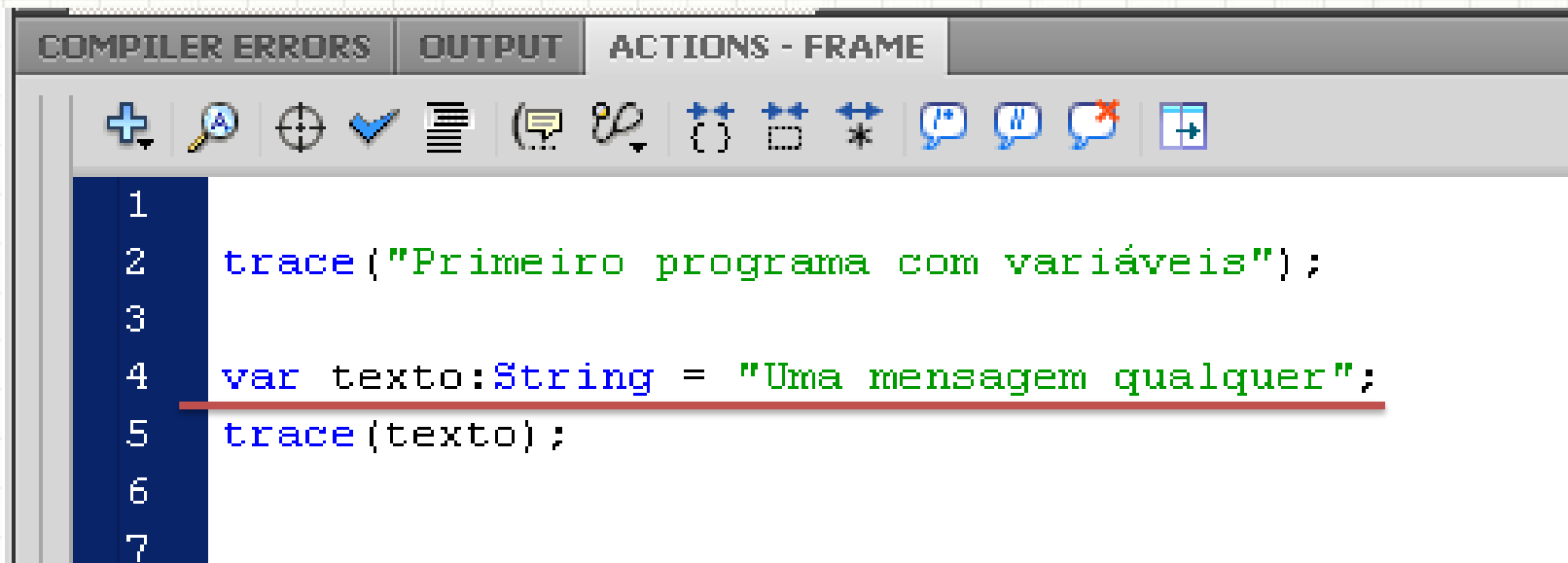
- **texto** é uma variável do tipo **String**

Variáveis em ActionScript 3

- Declaramos variáveis sempre assim:

var *nome:Tipo* = *valor_inicial*;

- Observe!



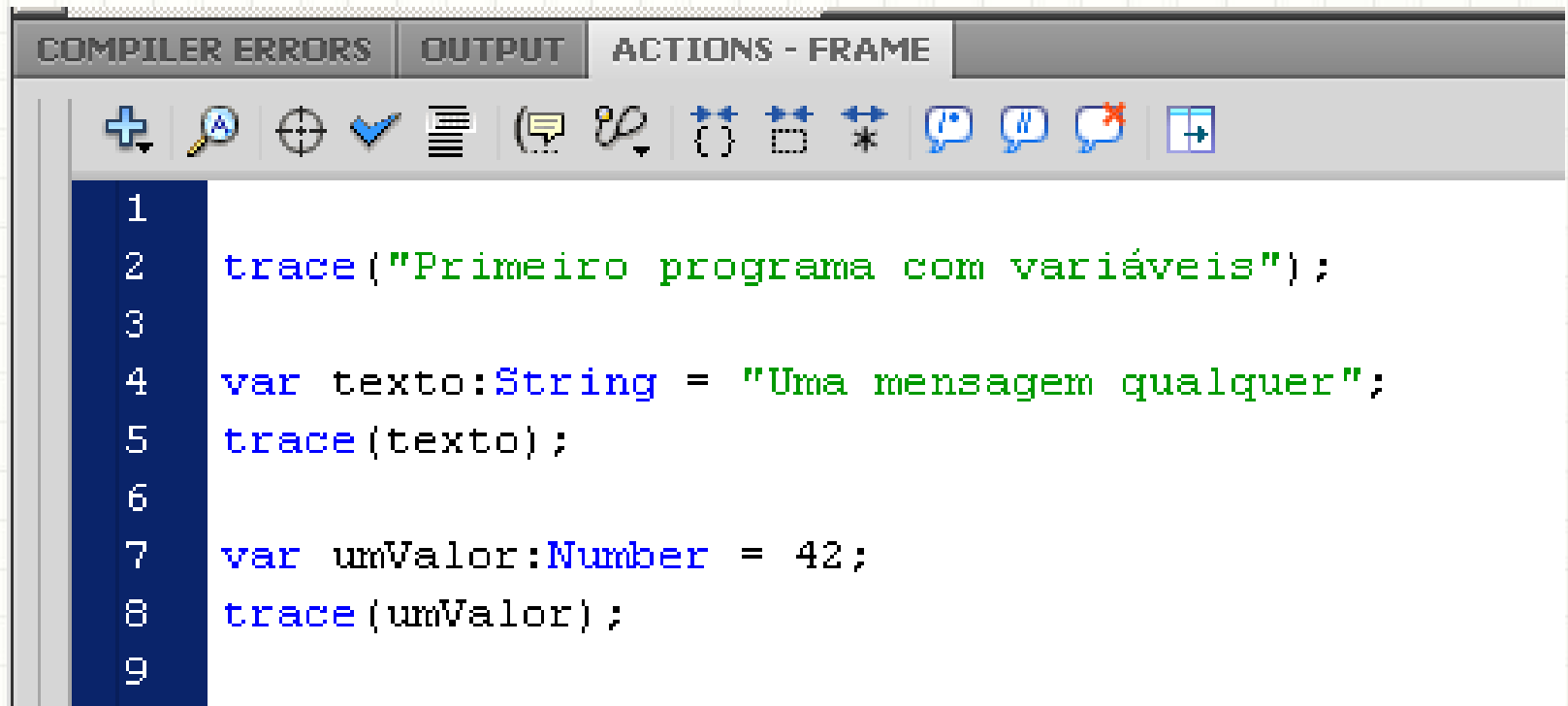
The screenshot shows an IDE window titled 'ACTIONS - FRAME'. The window has a toolbar with various icons for editing and debugging. The code editor displays the following ActionScript 3 code:

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

The line containing the variable declaration (line 4) is highlighted with a red underline.

Variáveis em ActionScript 3

- Incrementando o programa



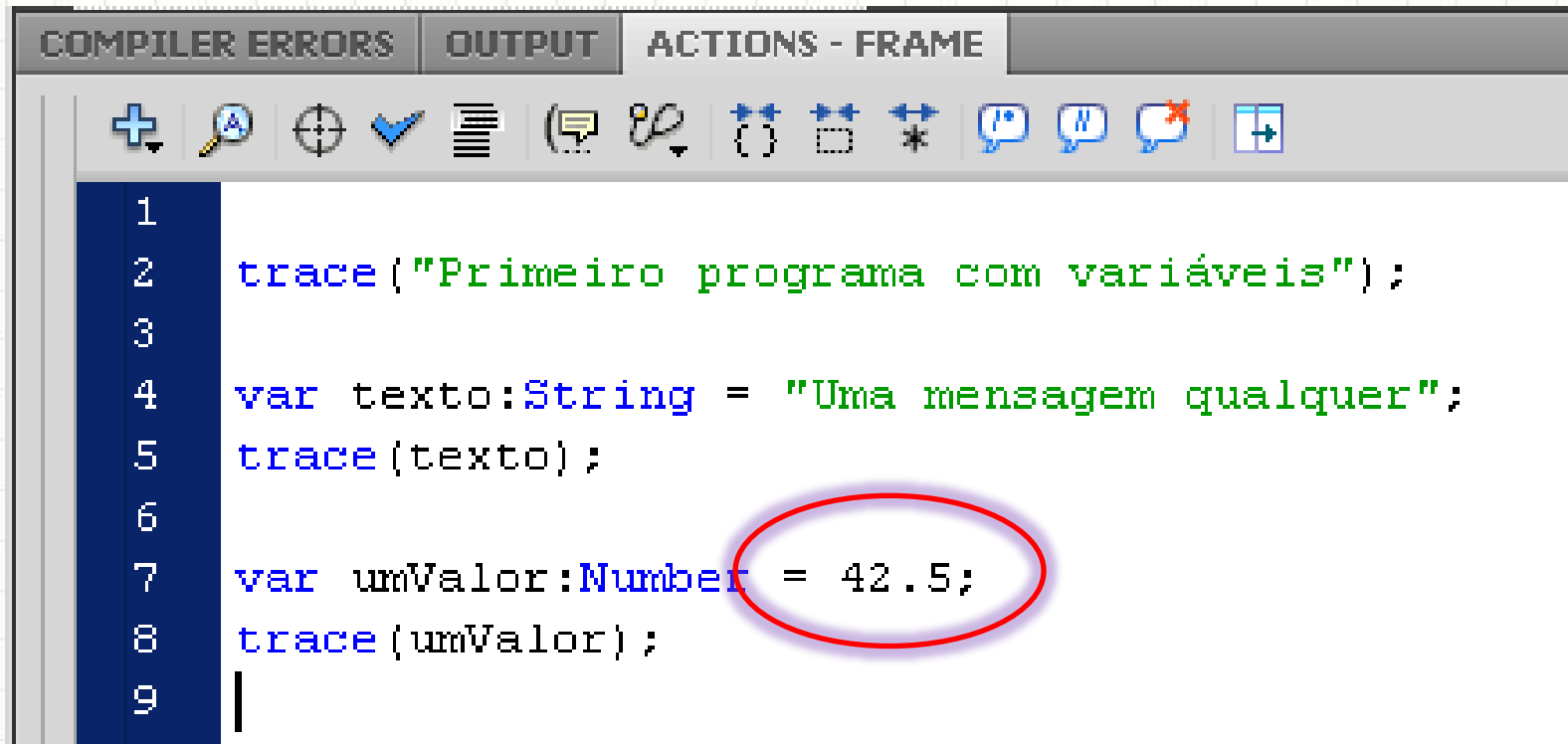
The screenshot shows a code editor window with three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'ACTIONS - FRAME' tab is active. Below the tabs is a toolbar with various icons for editing and debugging. The code area displays the following ActionScript 3 code:

```
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

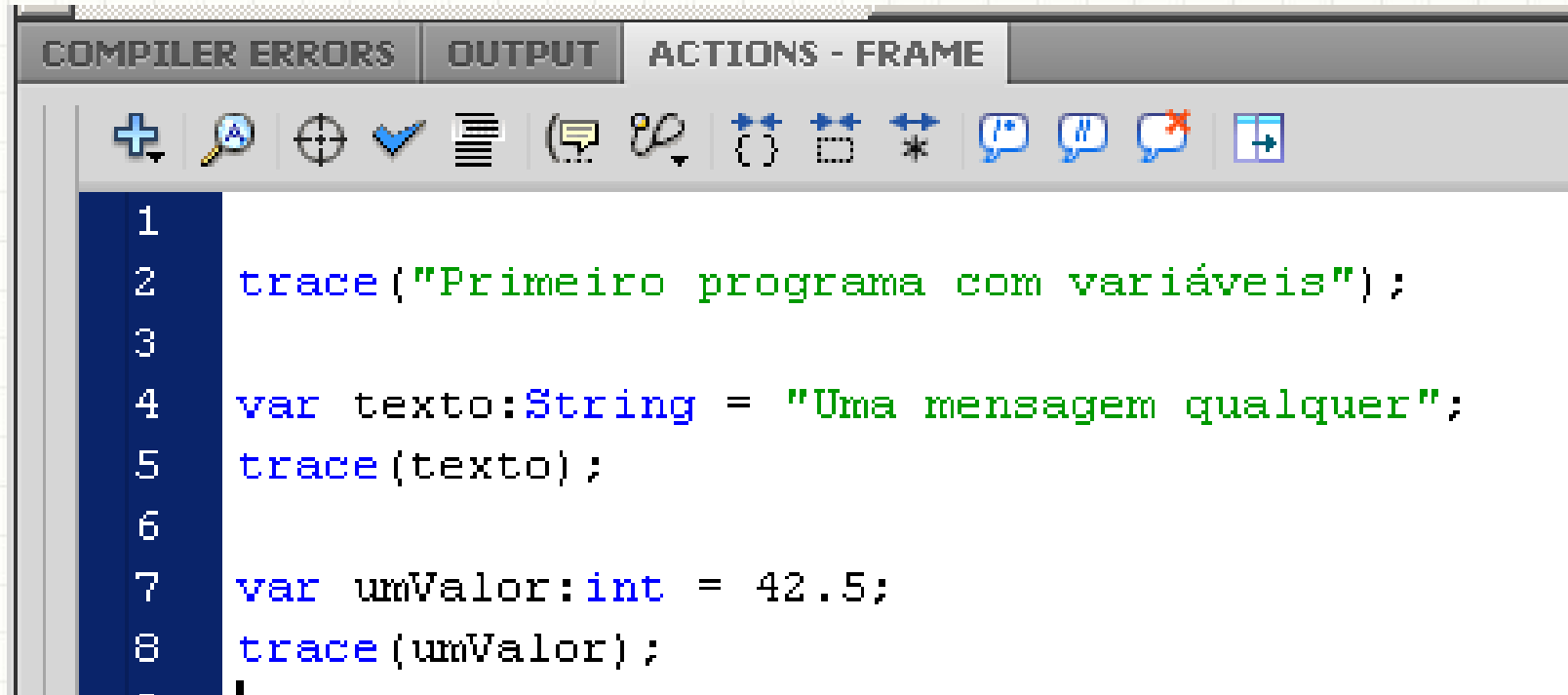
- Number aceita valores fracionários...



```
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...



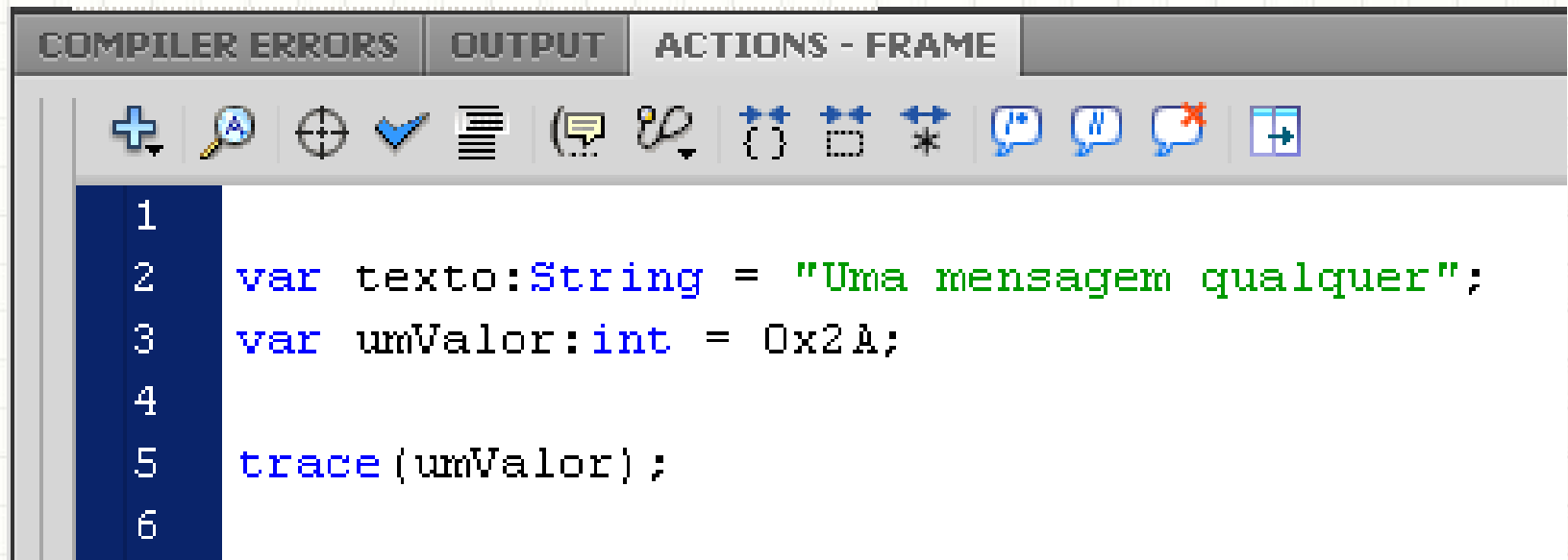
The screenshot shows a code editor window with three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'ACTIONS - FRAME' tab is active. The code is as follows:

```
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);  
9
```

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

Variáveis em ActionScript 3

- Podemos usar valores em hexadecimal...



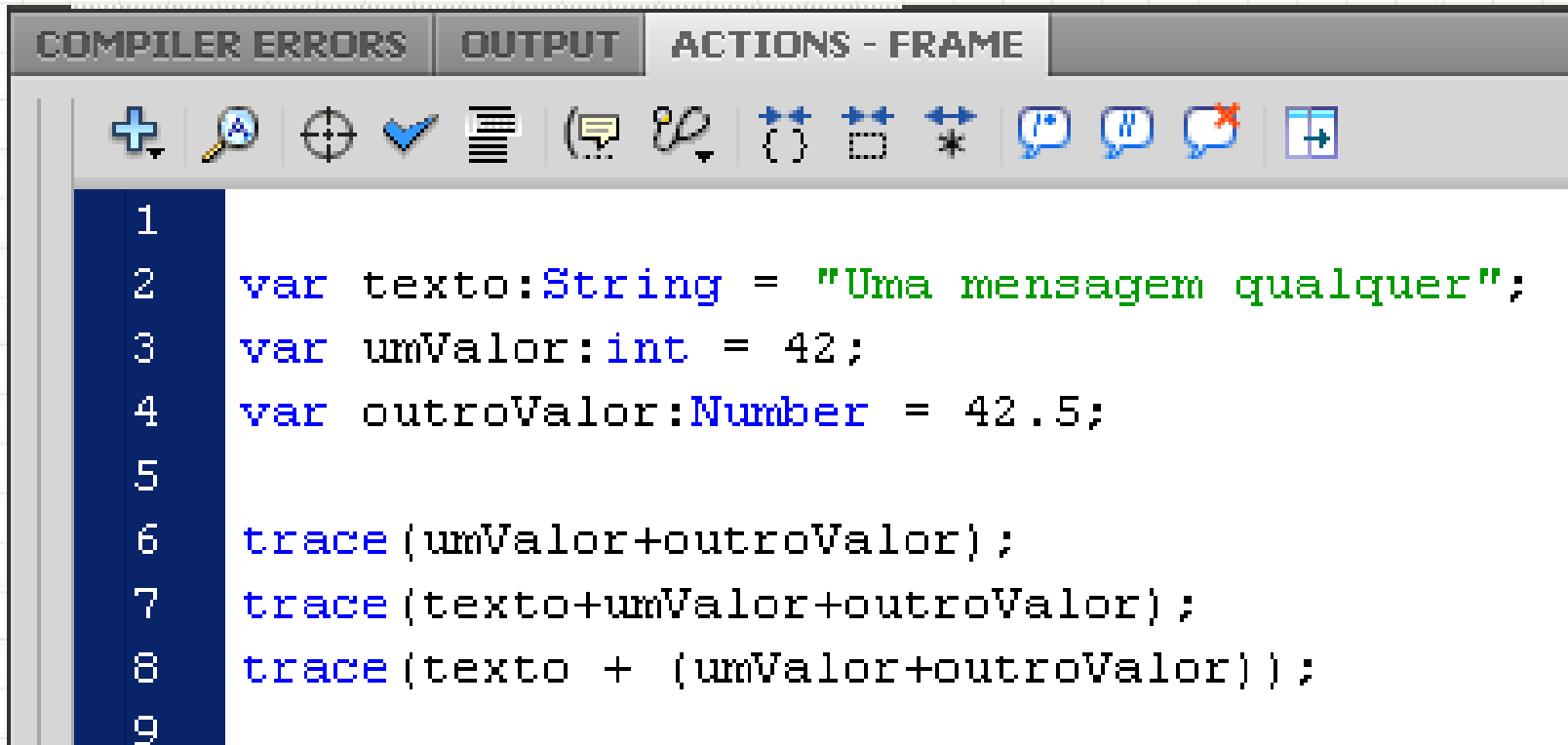
The screenshot shows a code editor window with three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'ACTIONS - FRAME' tab is active. Below the tabs is a toolbar with various icons for editing and debugging. The code area displays the following code:

```
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

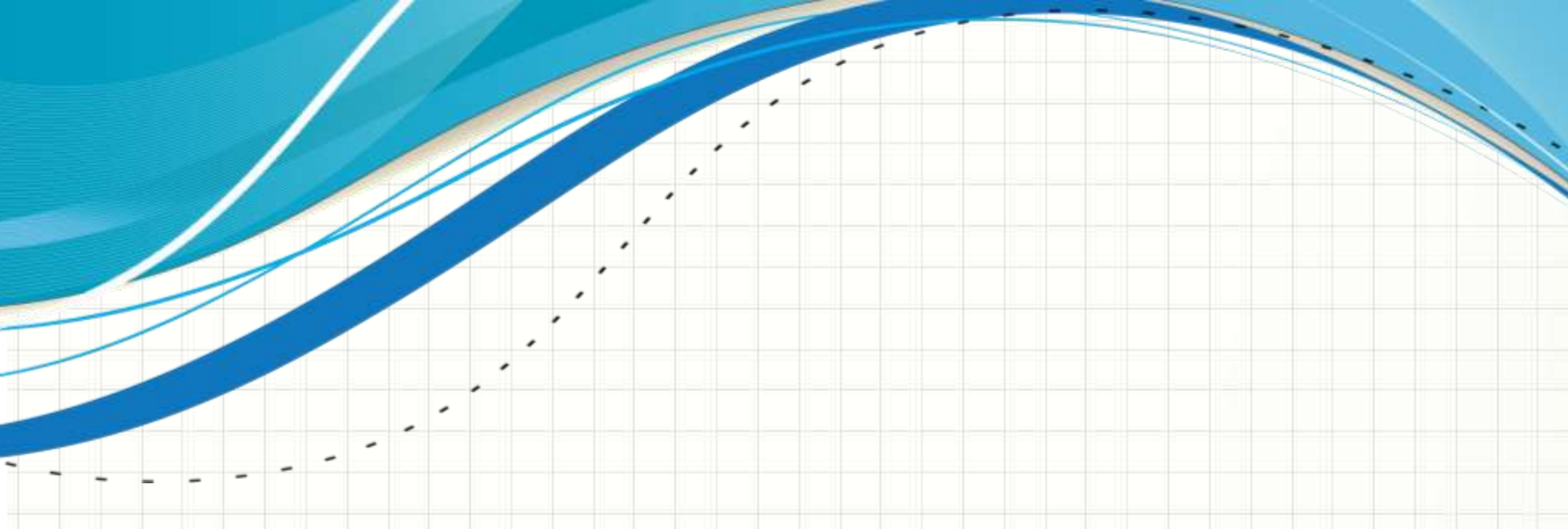
- Podemos misturar variáveis no trace...



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

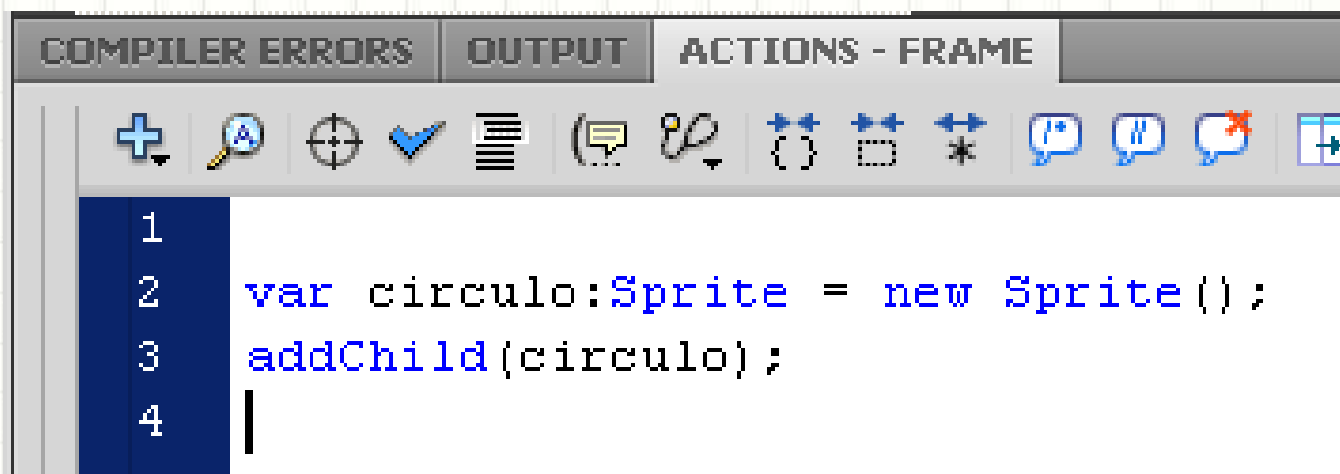
- Execute e observe os resultados...
- Qual a diferença entre eles?



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?

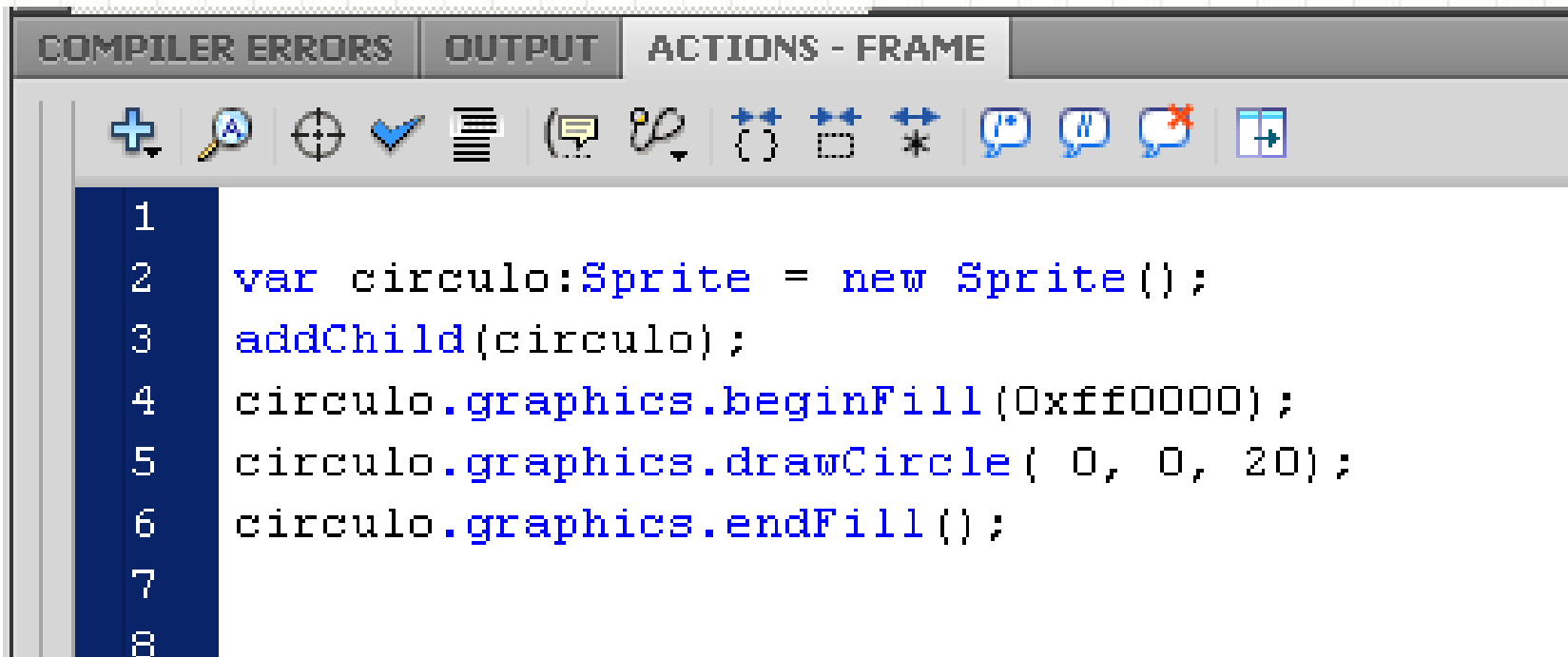


The screenshot shows the 'ACTIONS - FRAME' panel in Adobe Animate. The panel has tabs for 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. Below the tabs is a toolbar with various icons. The main area of the panel displays the following ActionScript 3 code:

```
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 with a tab labeled '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
```

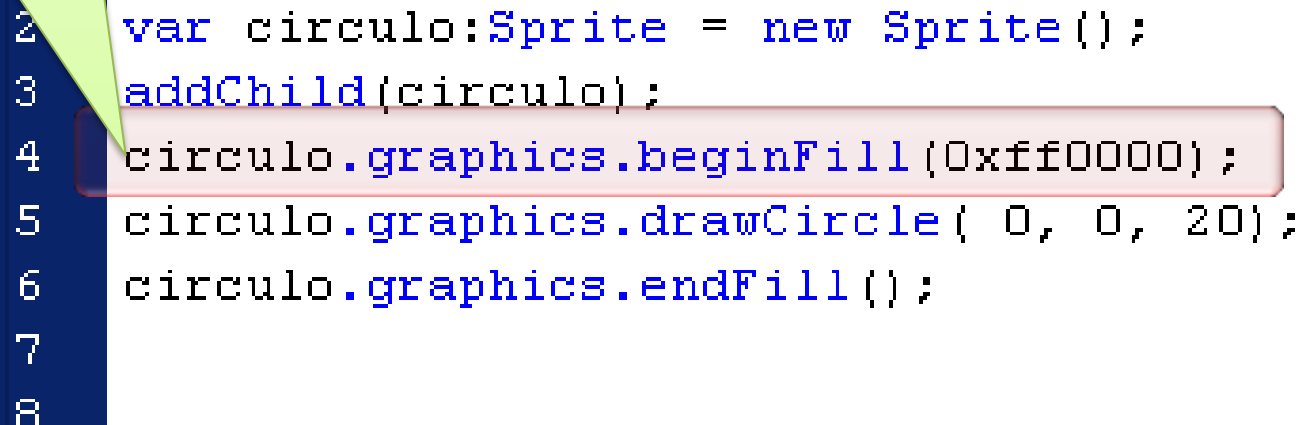
- Experimente!

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

o em ActionScript 3

tem figura ainda...

desenhar dentro do sprite



```
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

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!

lo em Action

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

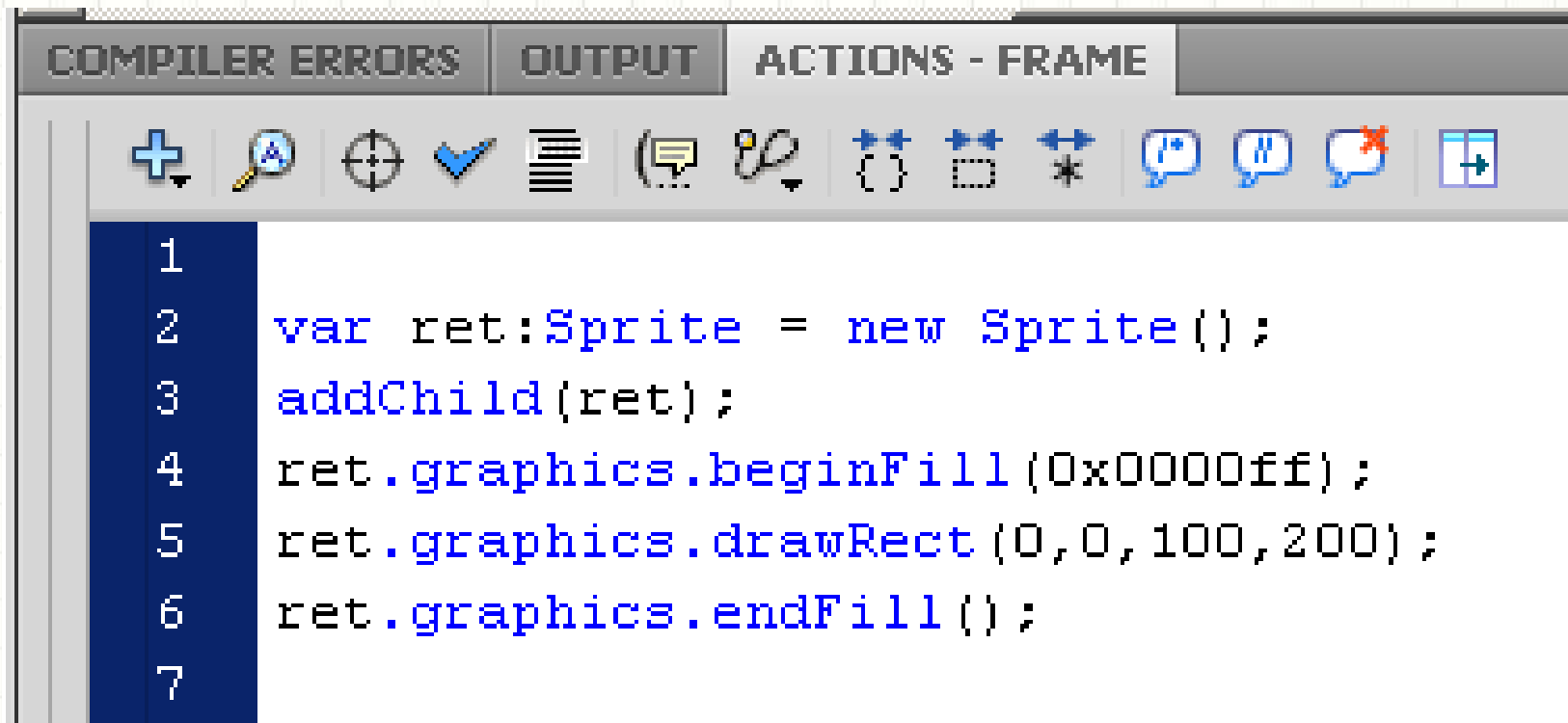
```
2 var circulo:Sprite = new Sprite();  
3 addChild(circulo);  
4 circulo.graphics.beginFill(0xff0000);  
5 circulo.graphics.drawCircle( 0, 0, 20);  
6 circulo.graphics.endFill();  
7  
8
```

Indica que o **desenho do sprite acabou**

- Experimente!

Desenhando em ActionScript 3

- Desenhando um retângulo...



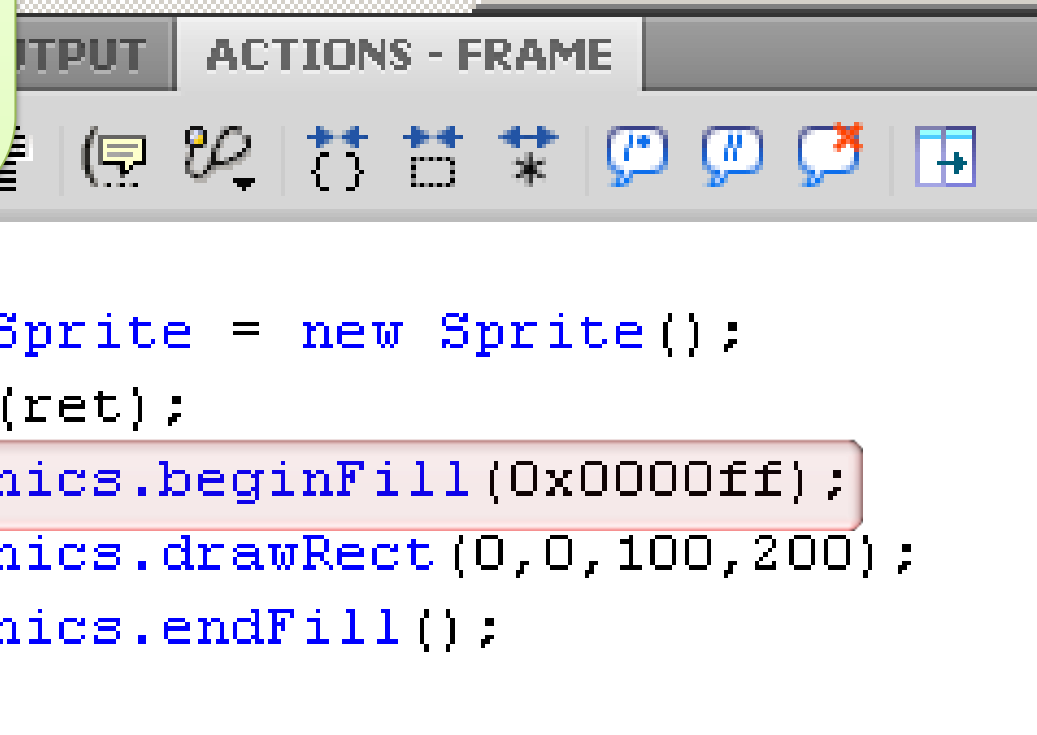
The screenshot shows the Adobe Flash IDE interface. At the top, there are three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'ACTIONS - FRAME' tab is selected. Below the tabs is a toolbar with various icons for actions like adding a new action, finding actions, and inserting code. The main area of the 'ACTIONS' panel displays a list of actions. The first action is selected, and its code is shown in a text area. 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
```

- Experimente!

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

o em ActionScript 3 m retângulo...



```
1 var ret:Sprite = new Sprite();  
2  
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!

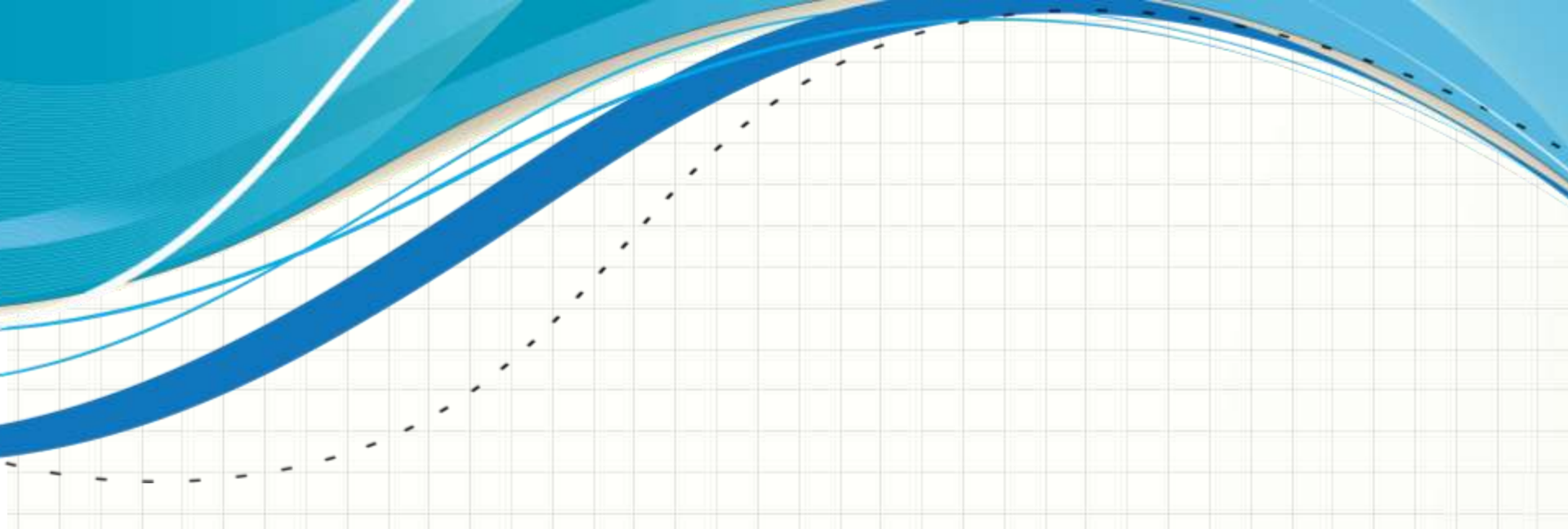
Indica que **iniciaremos o desenho** no sprite usando a cor de preenchimento azul (rrggbb)

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**

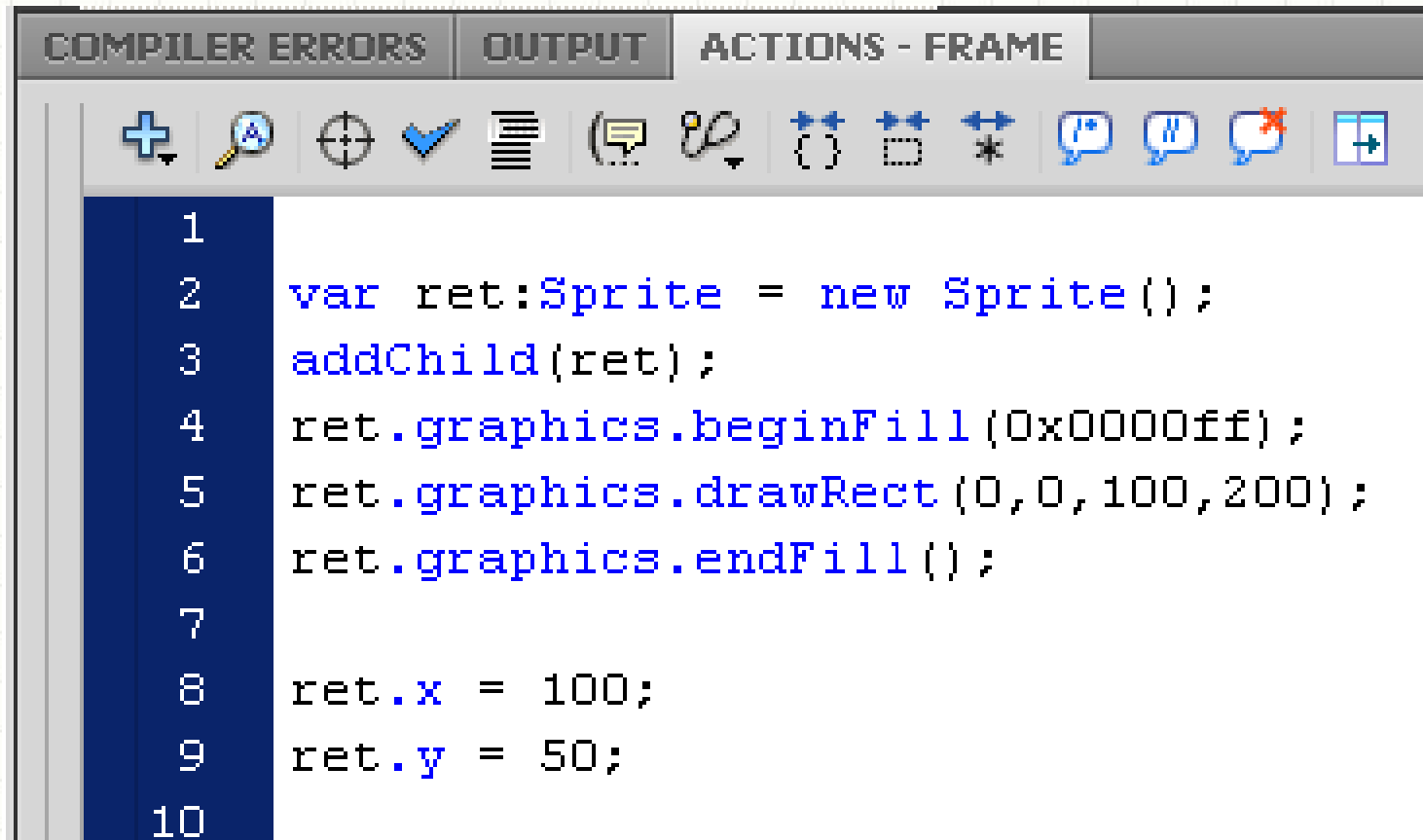
- Experimente!



PROPRIEDADES DOS SPRITES

Desenhando em ActionScript 3

- Podemos mover o nosso retângulo



The screenshot shows the 'ACTIONS - FRAME' panel in the Adobe Flash IDE. The panel has a toolbar with icons for adding, deleting, and moving actions, as well as a list of actions. The code in the panel is as follows:

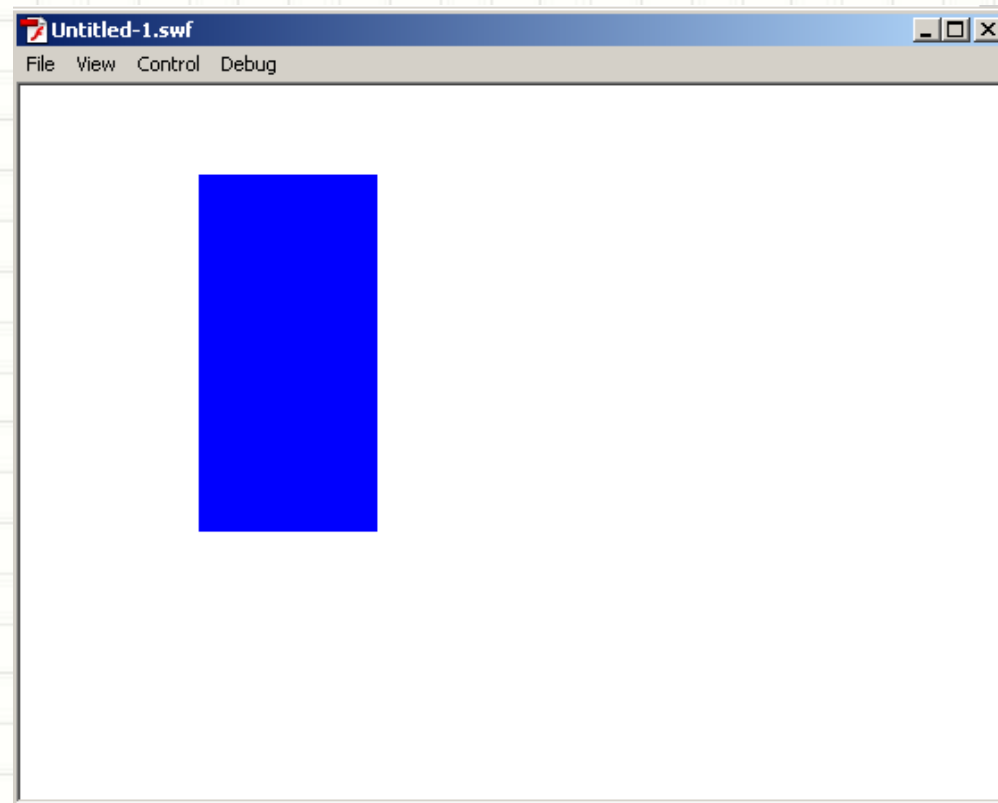
```
1  
2  var ret:Sprite = new Sprite();  
3  addChild(ret);  
4  ret.graphics.beginFill(0x0000ff);  
5  ret.graphics.drawRect(0,0,100,200);  
6  ret.graphics.endFill();  
7  
8  ret.x = 100;  
9  ret.y = 50;  
10
```

- Experimente!

Desenhando em ActionScript 3

- Resultado

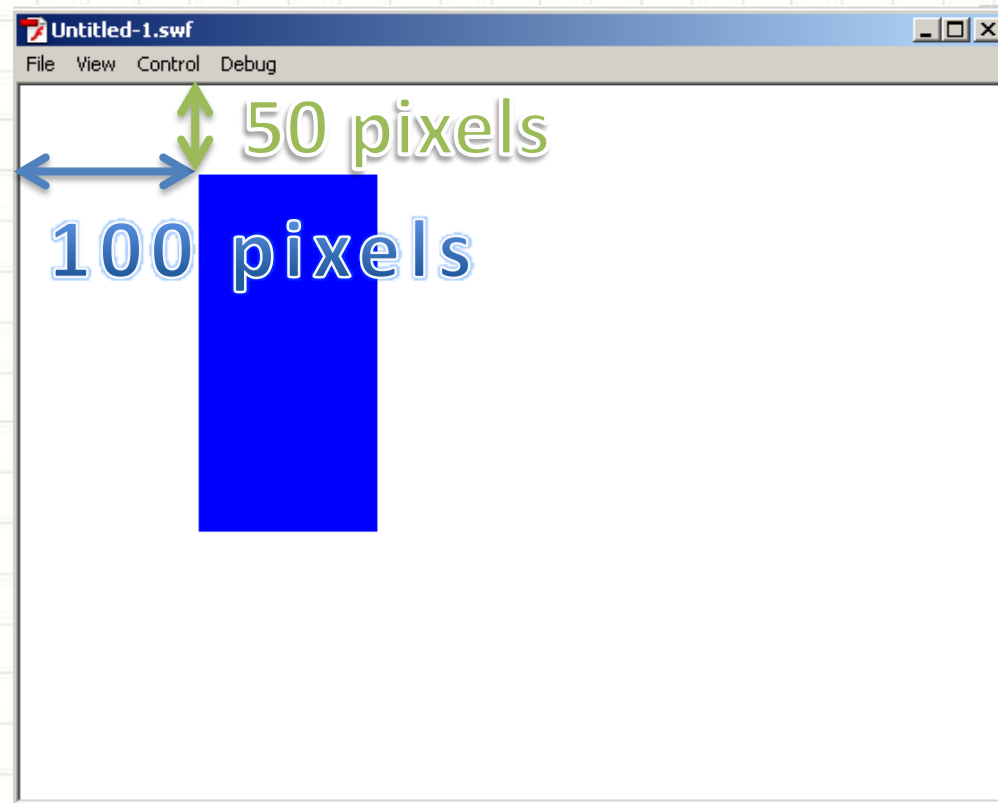
```
8 ret.x = 100;  
9 ret.y = 50;
```



Desenhando em ActionScript 3

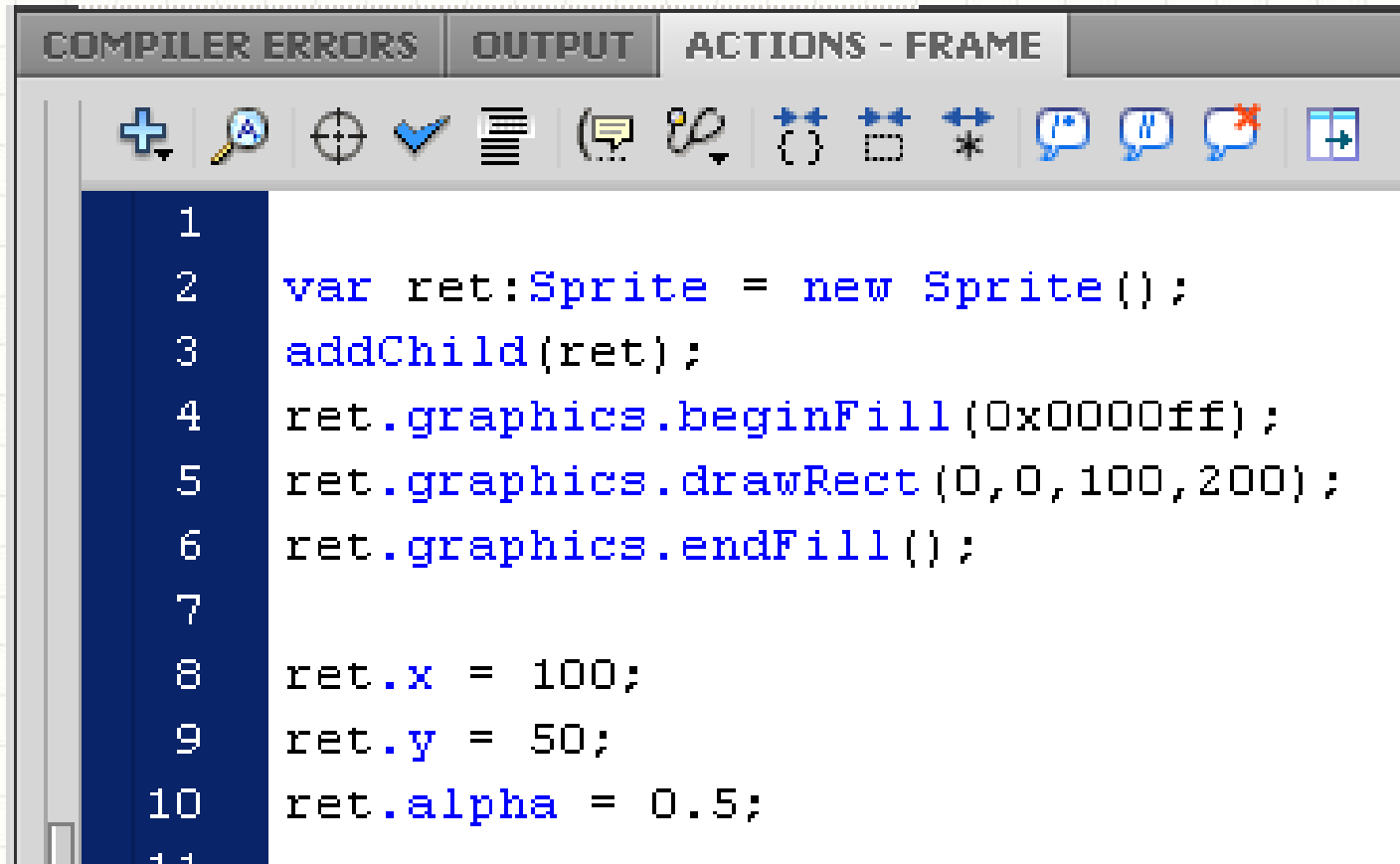
- Resultado

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



Desenhando em ActionScript 3

- Podemos mudar a “opacidade” do sprite



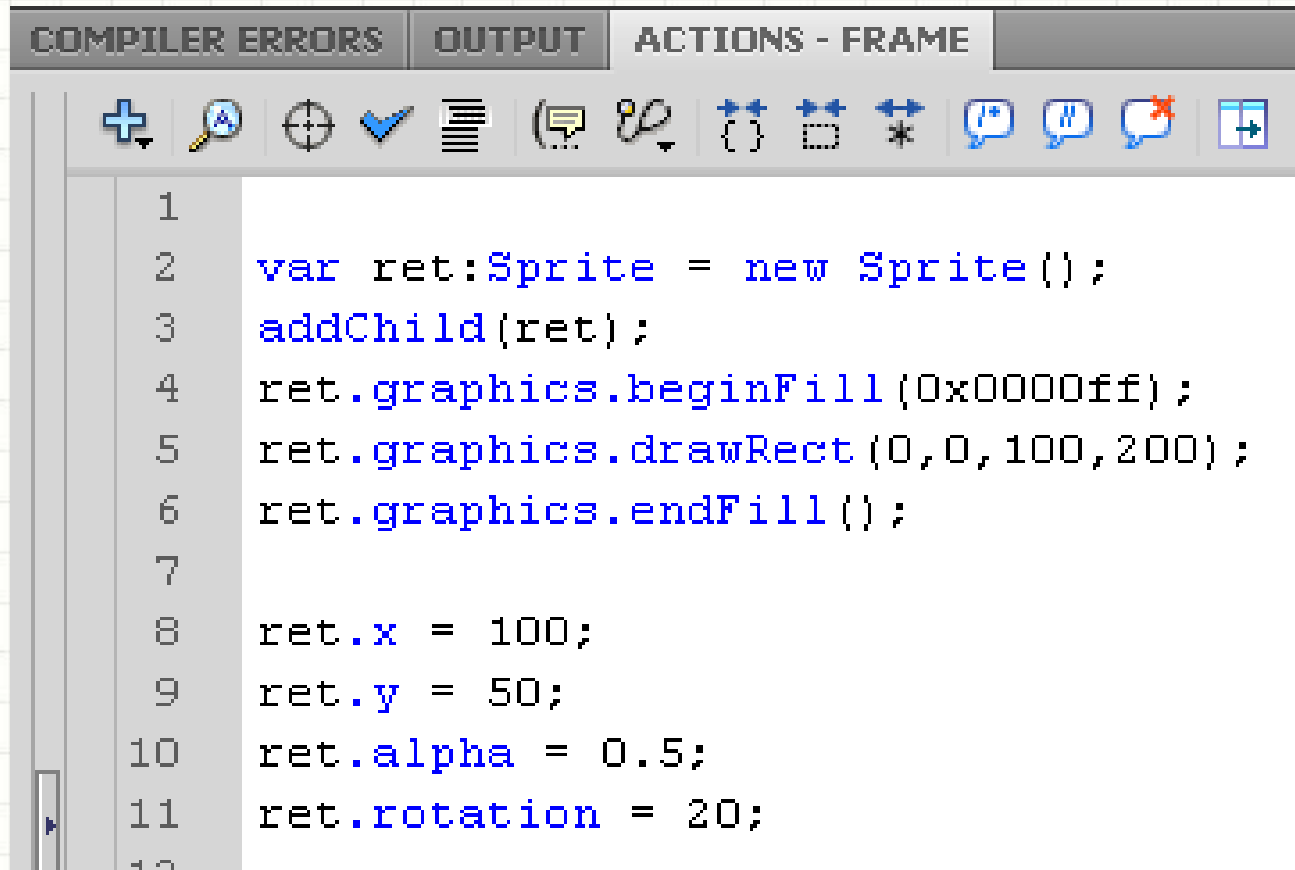
The screenshot shows a code editor window with three tabs: "COMPILER ERRORS", "OUTPUT", and "ACTIONS - FRAME". The "ACTIONS - FRAME" tab is active, displaying a list of actions for a timeline frame. The actions are as follows:

```
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;
```

- 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 a list of ActionScript 3 code lines. The code defines a variable 'ret' of type 'Sprite', adds it to the display, fills it with a light blue color, draws a rectangle, and sets its position, opacity, and rotation.

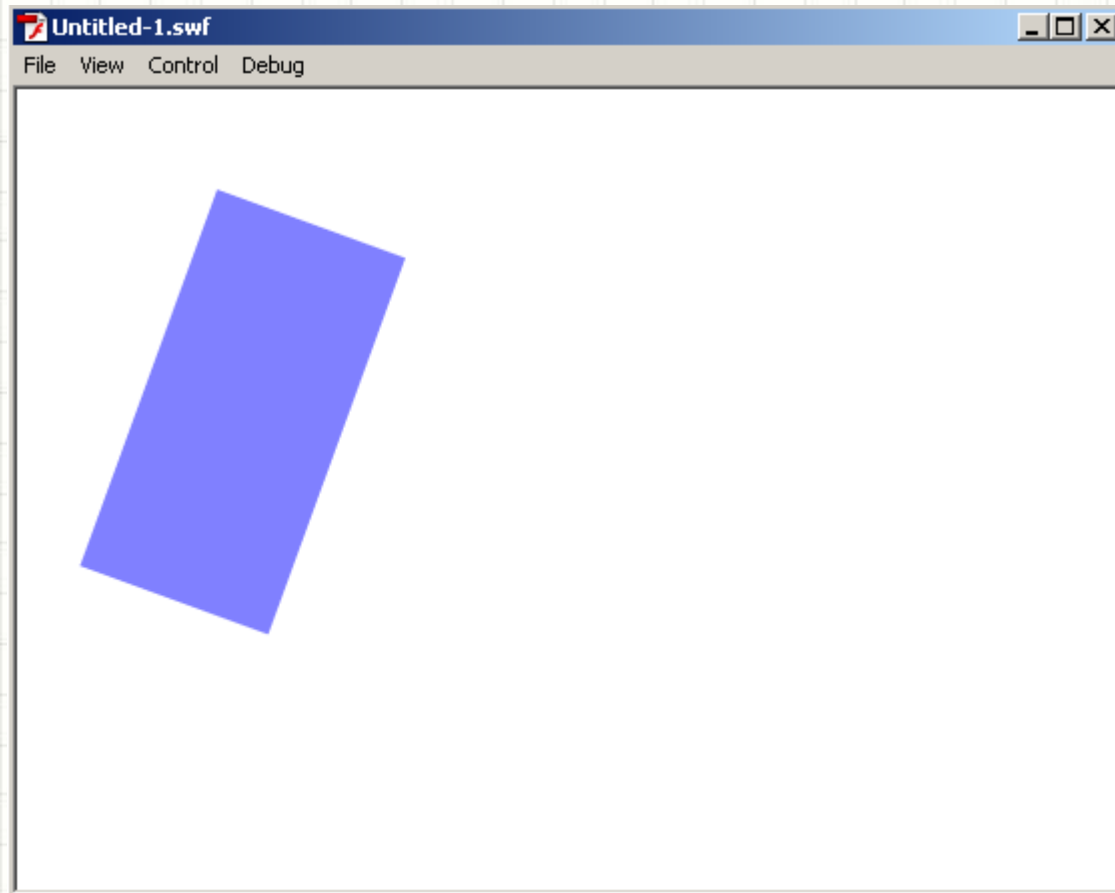
```
1  
2  var ret:Sprite = new Sprite();  
3  addChild(ret);  
4  ret.graphics.beginFill(0x0000ff);  
5  ret.graphics.drawRect(0,0,100,200);  
6  ret.graphics.endFill();  
7  
8  ret.x = 100;  
9  ret.y = 50;  
10 ret.alpha = 0.5;  
11 ret.rotation = 20;  
12
```

- Experimente!

Desenhando em ActionScript 3

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

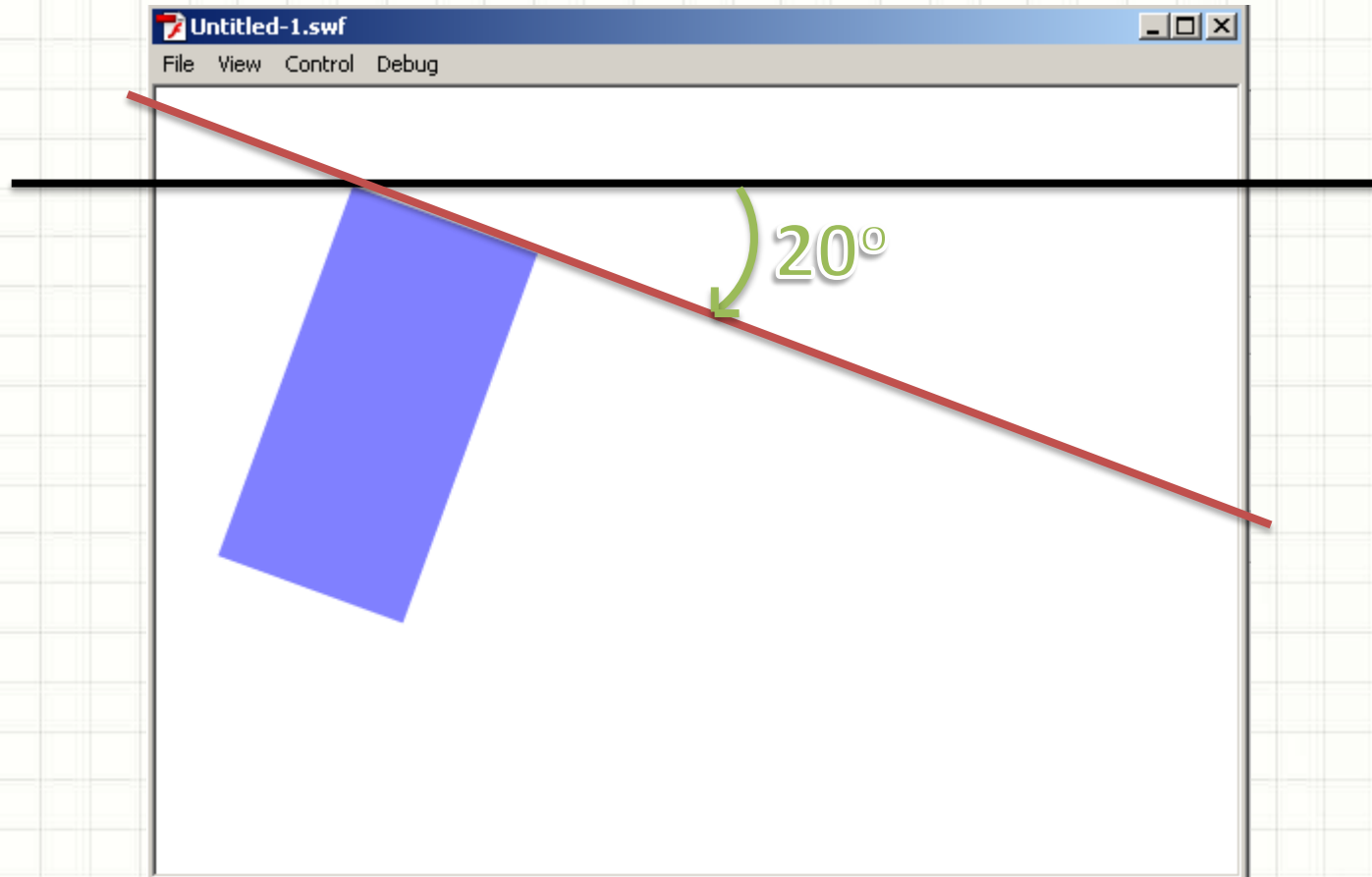
```
11 ret.rotation = 20;
```

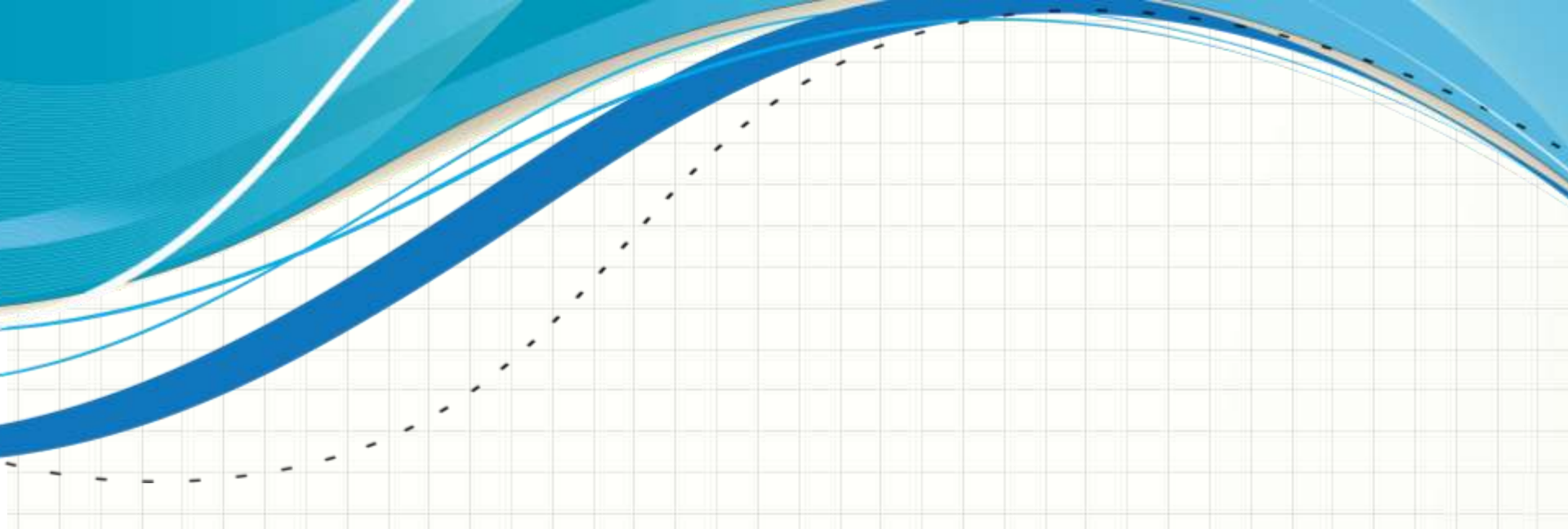


Desenhando em ActionScript 3

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

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





EIXOS DE COORDENADAS

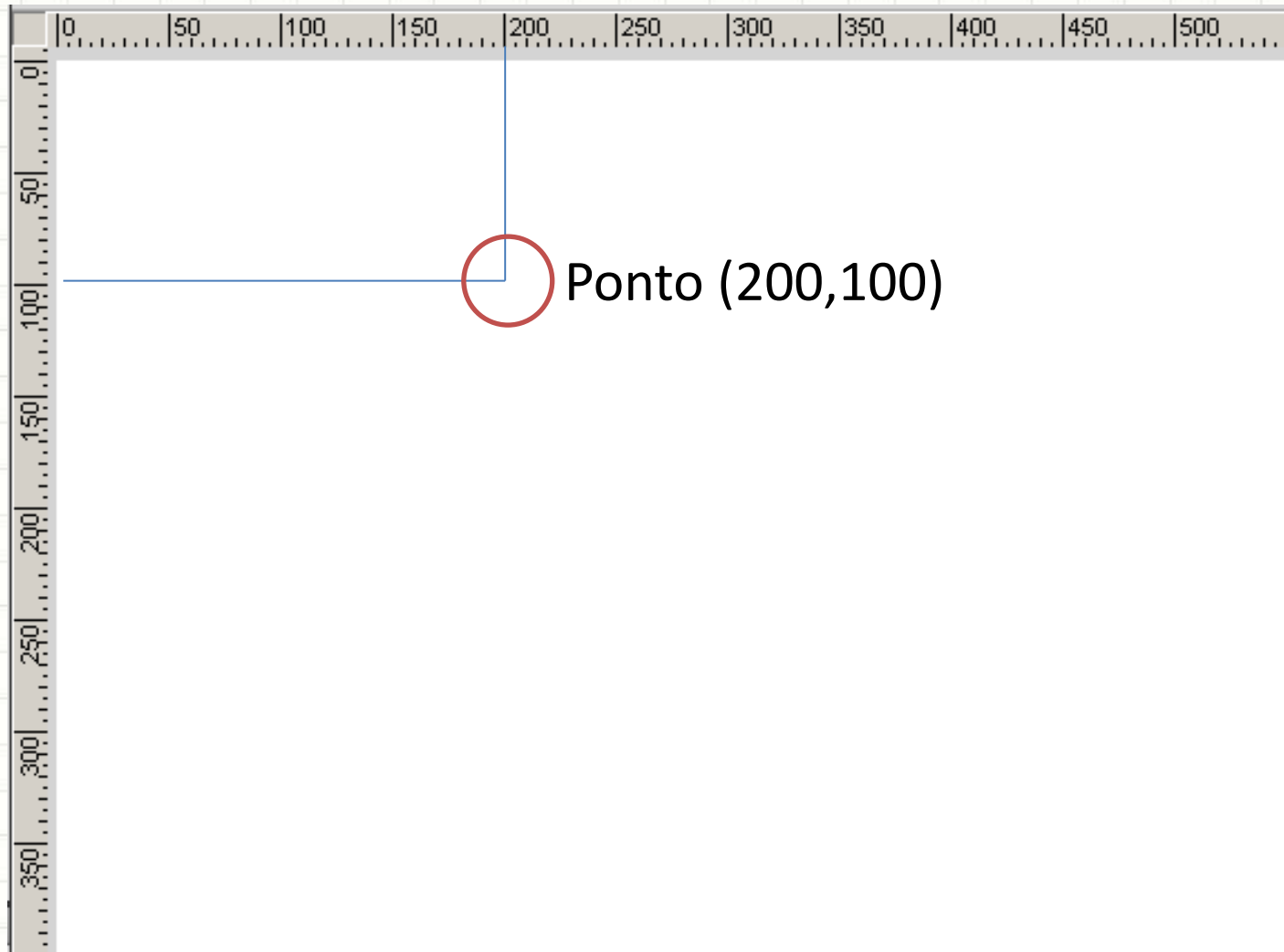
Coordenadas do AS3

- Eixos de coordenadas do cenário



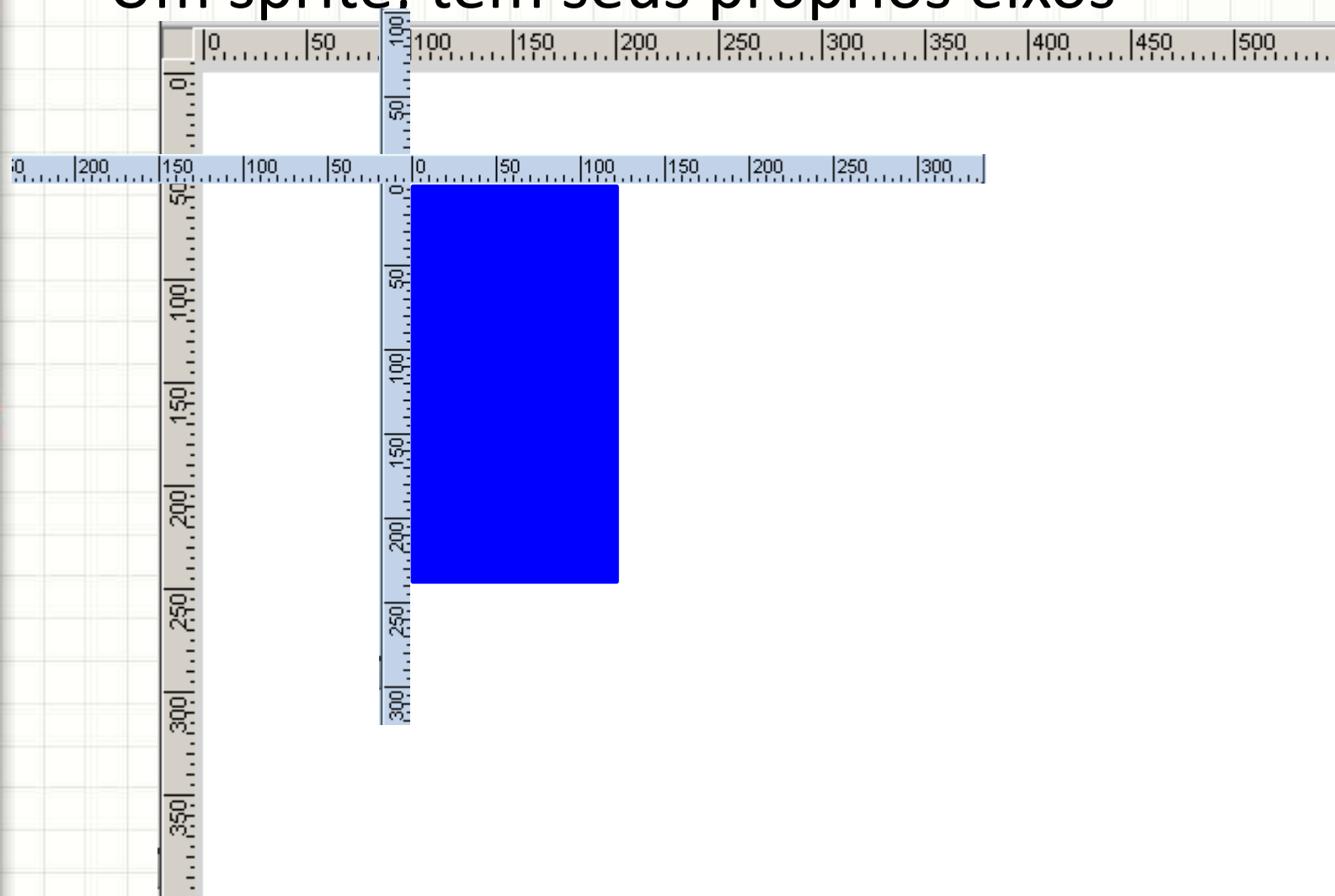
Coordenadas do AS3

- Eixos de coordenadas do cenário



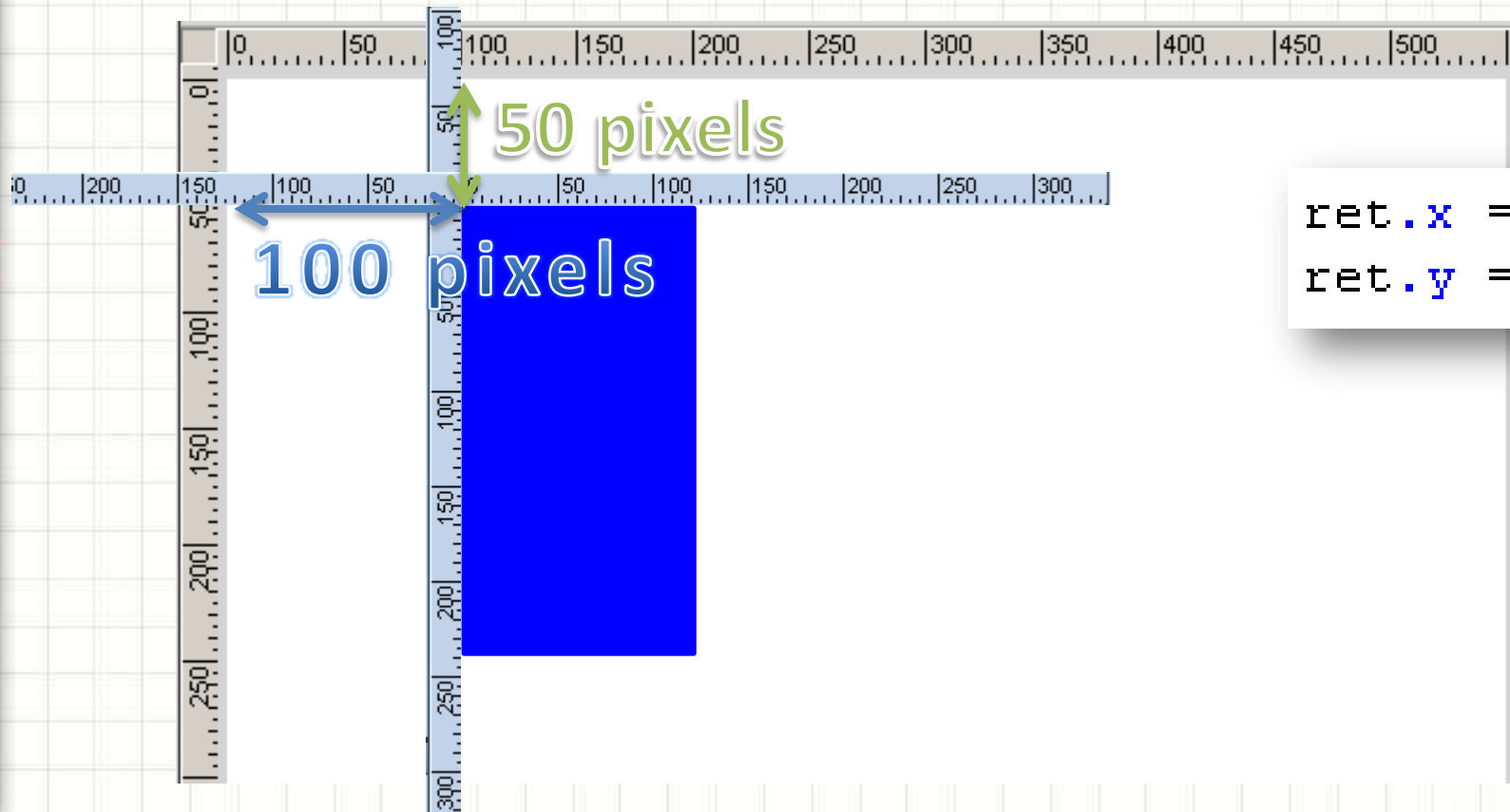
Coordenadas do AS3

- Um sprite, tem seus próprios eixos



Coordenadas do AS3

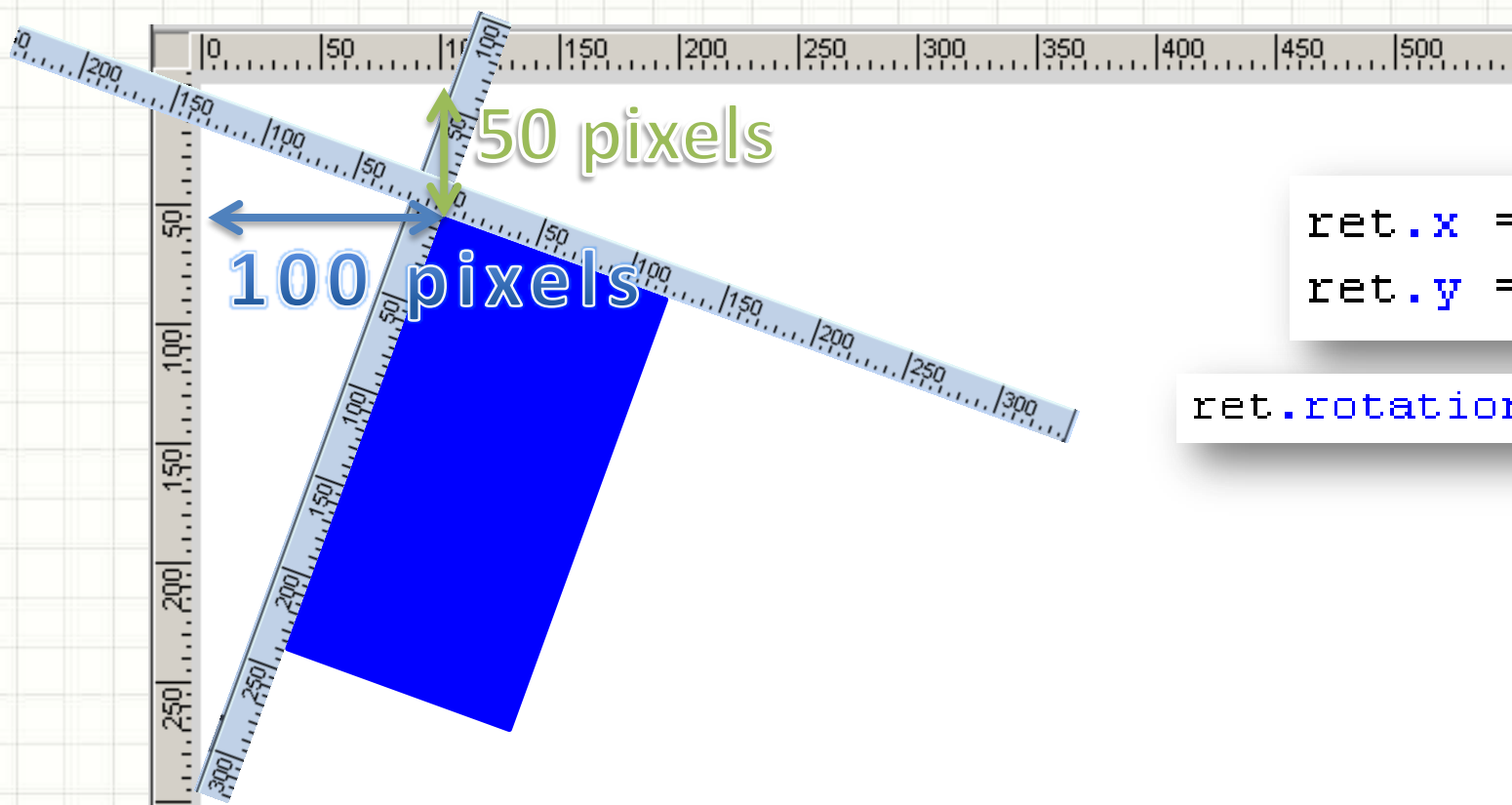
- Quando posicionamos os Sprites, posicionamos o 0,0 do sprite no cenário



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

Coordenadas do AS3

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

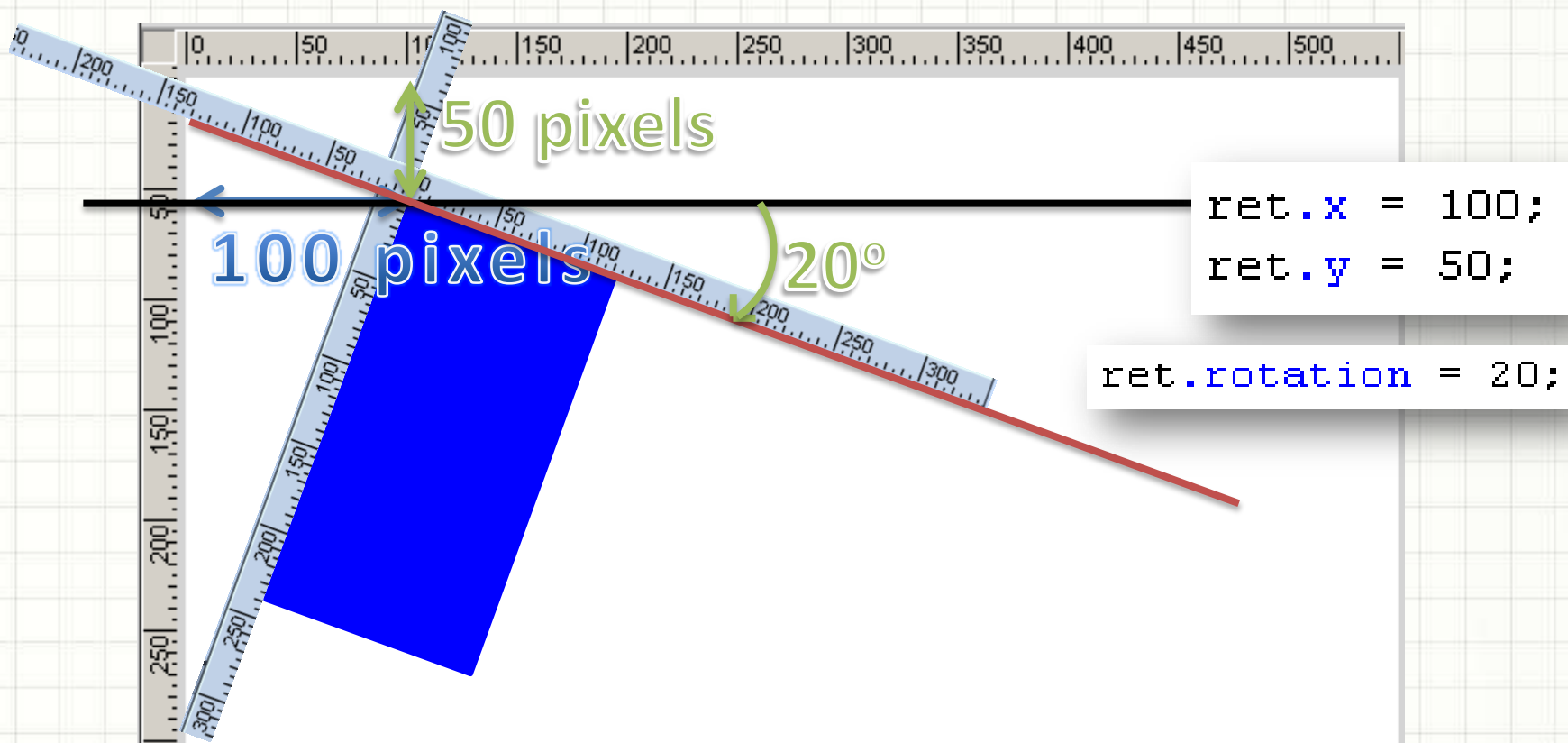


```
ret.x = 100;  
ret.y = 50;
```

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

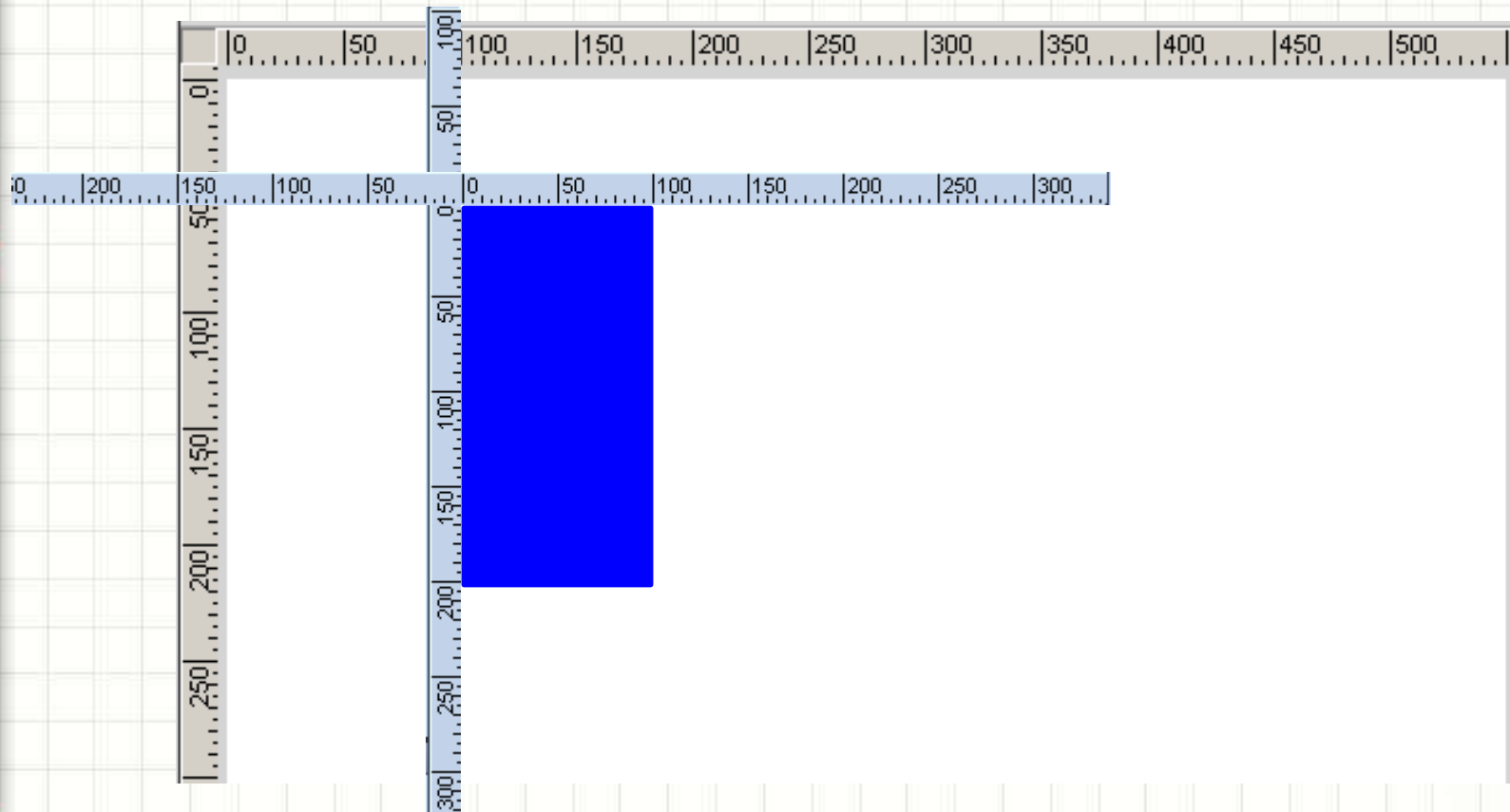
Coordenadas do AS3

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



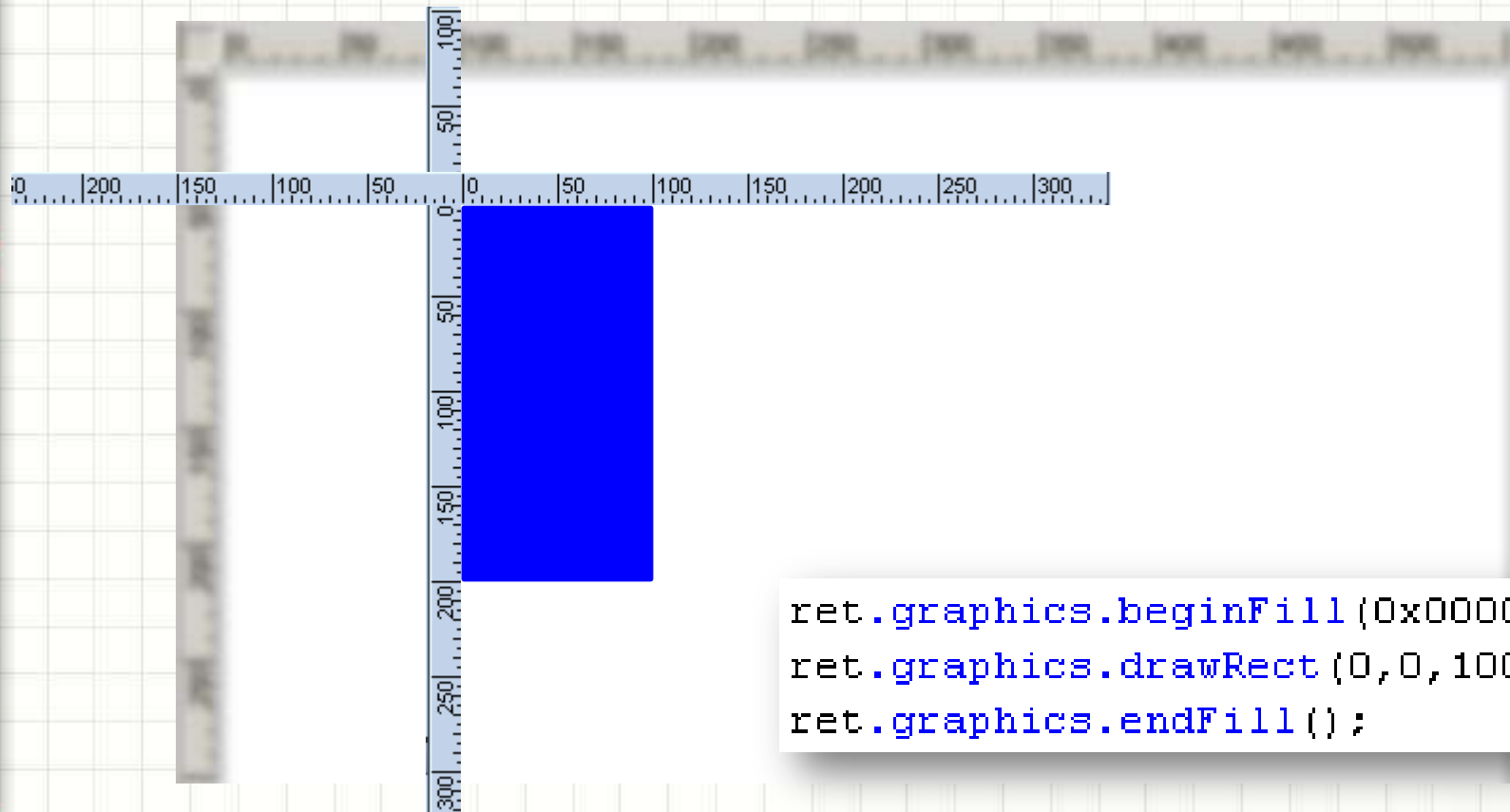
Coordenadas do AS3

- Para que servem os eixos do sprite?



Coordenadas do AS3

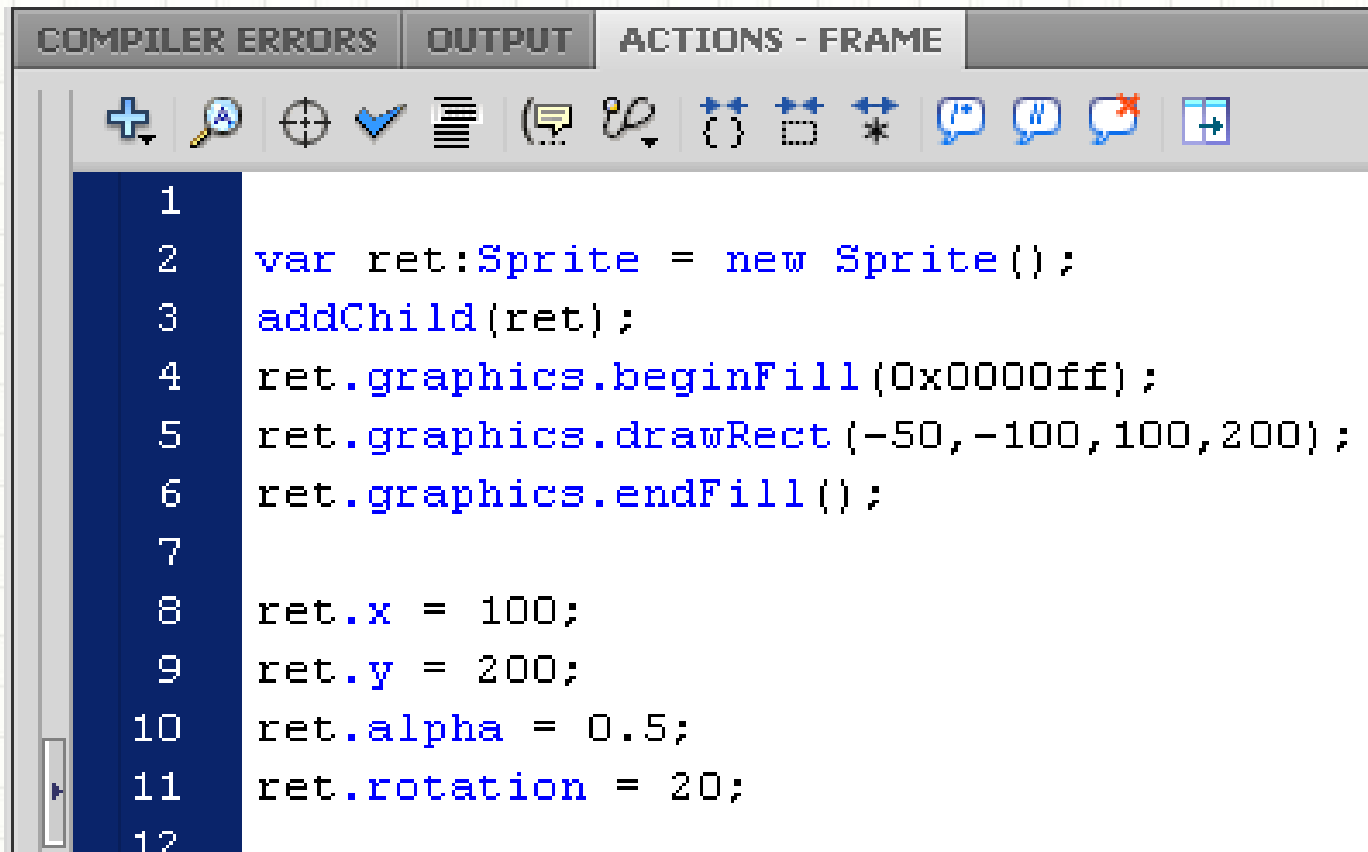
- Para que servem os eixos do sprite?
- Ao **desenhar** o sprite, são eles que valem!



```
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?



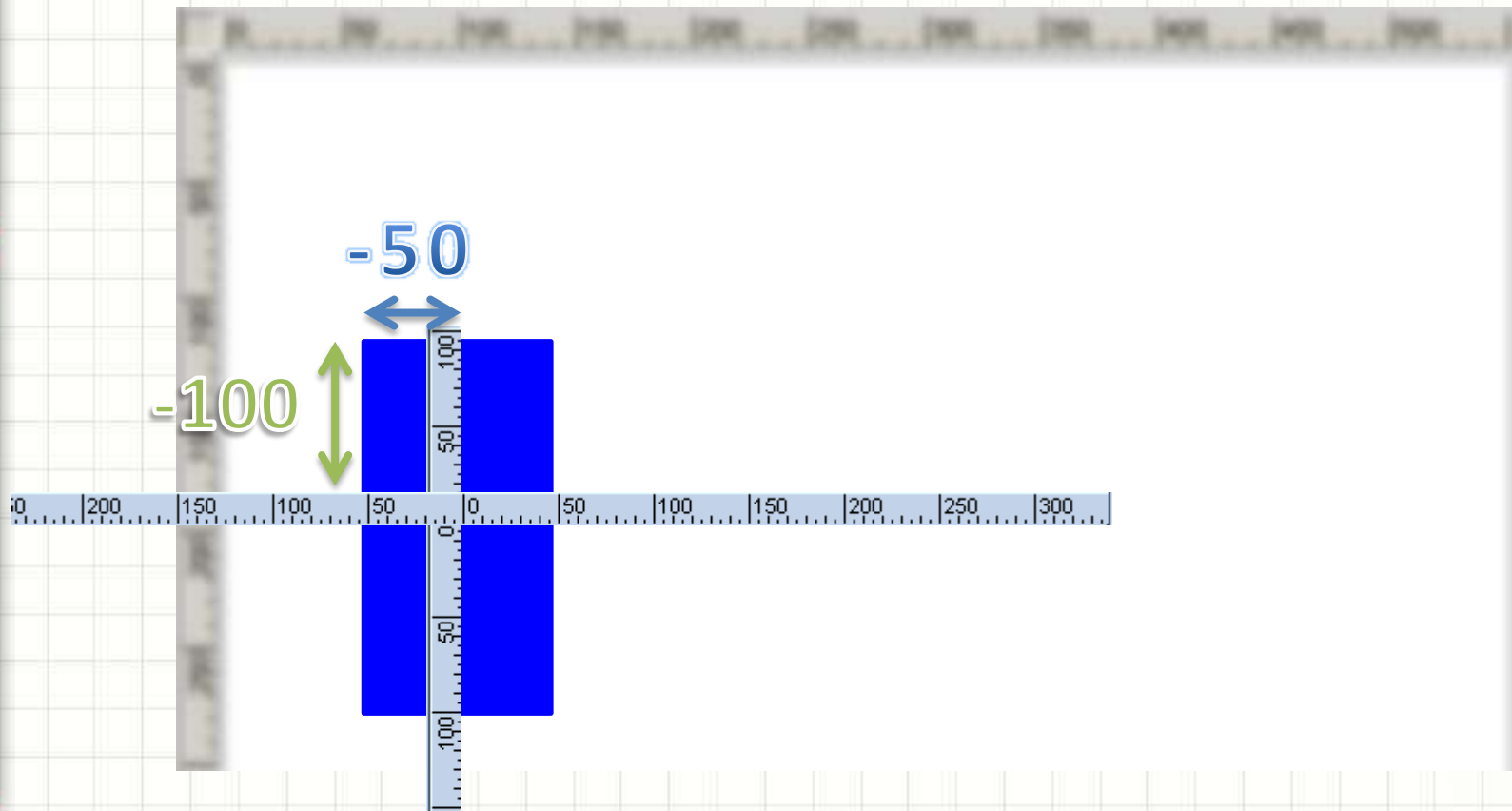
```
1  
2  var ret:Sprite = new Sprite();  
3  addChild(ret);  
4  ret.graphics.beginFill(0x0000ff);  
5  ret.graphics.drawRect(-50,-100,100,200);  
6  ret.graphics.endFill();  
7  
8  ret.x = 100;  
9  ret.y = 200;  
10 ret.alpha = 0.5;  
11 ret.rotation = 20;  
12
```

- Experimente!

Desenhando em ActionScript 3

- Observe como desenhamos...

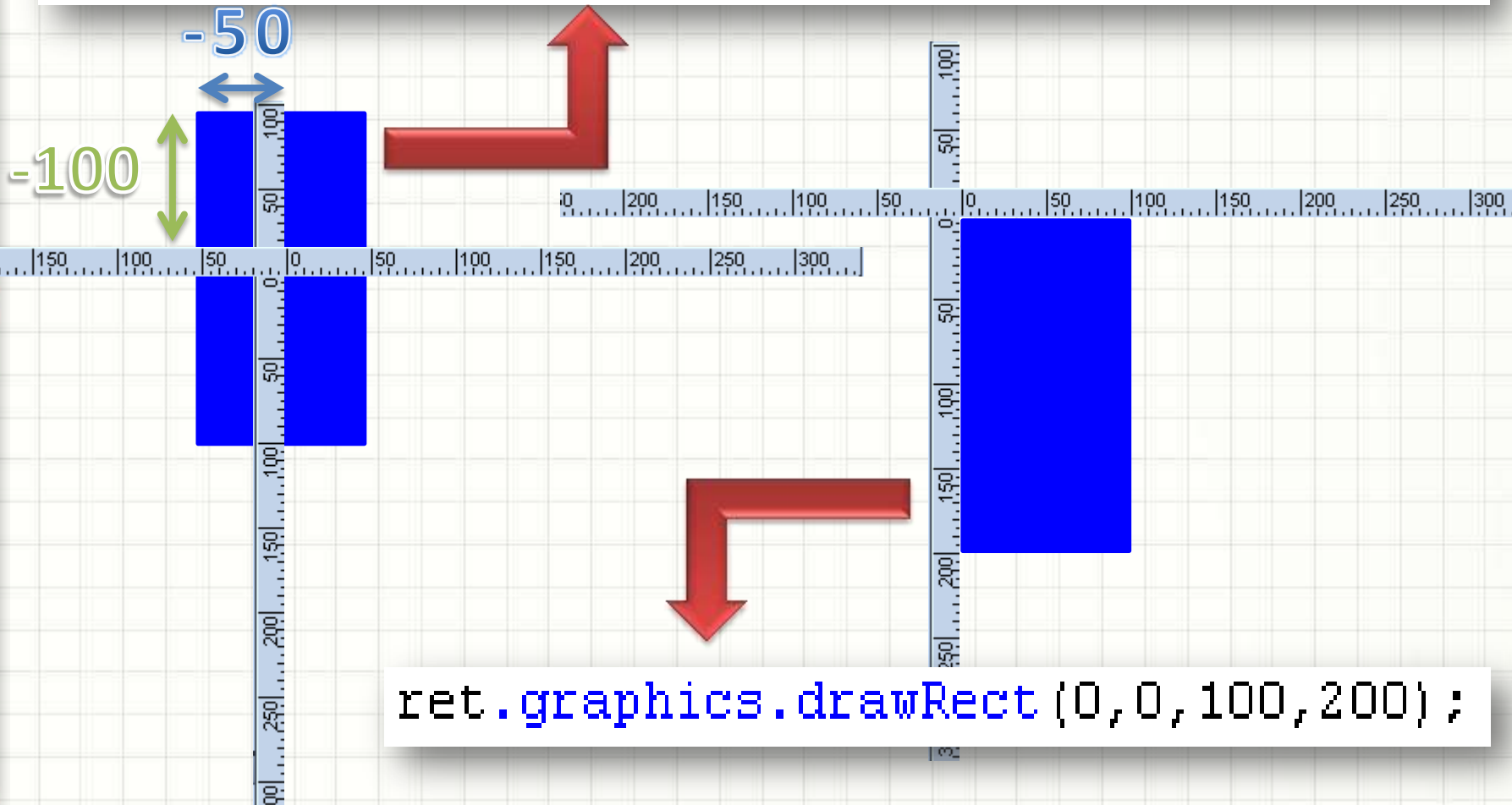
```
ret.graphics.drawRect (-50, -100, 100, 200) ;
```



Desenhando em ActionScript 3

- Compare

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



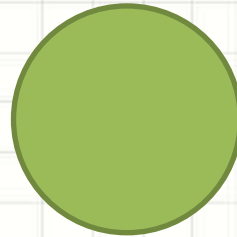
```
ret.graphics.drawRect (0,0,100,200) ;
```



ATIVIDADE

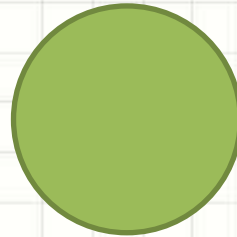
Atividade

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



Atividade

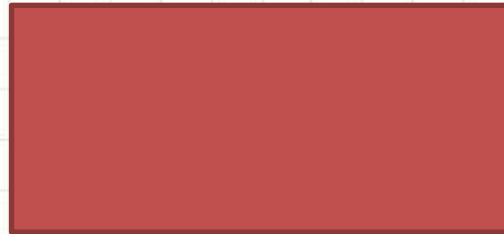
- Faça um programa que mostre 1 círculo verde de raio 20 pixels



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

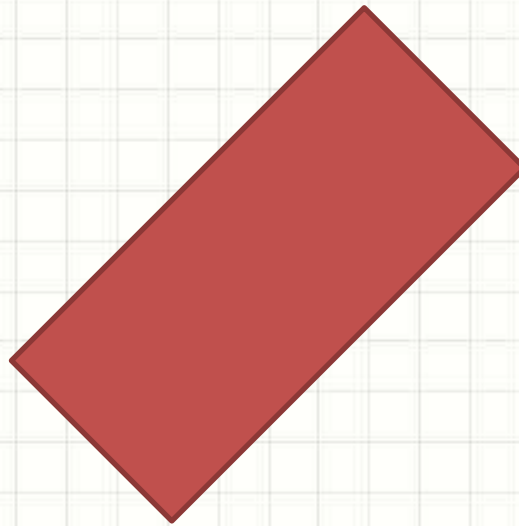
Atividade

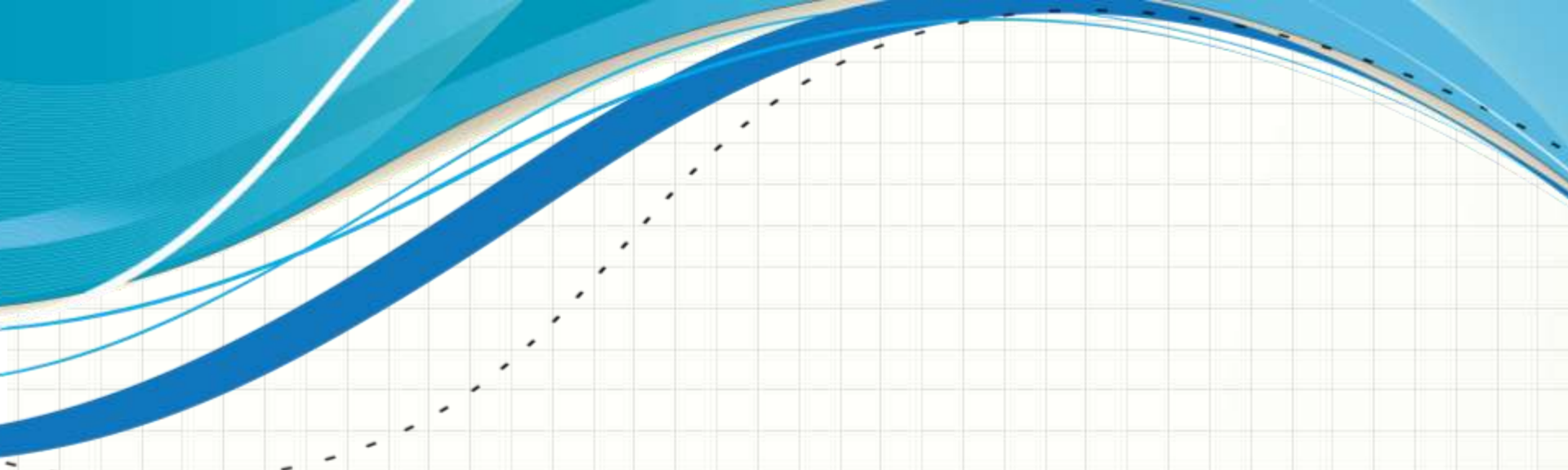
- Faça um programa que mostre 1 retângulo vermelho com 50 x 20 pixels



Atividade

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

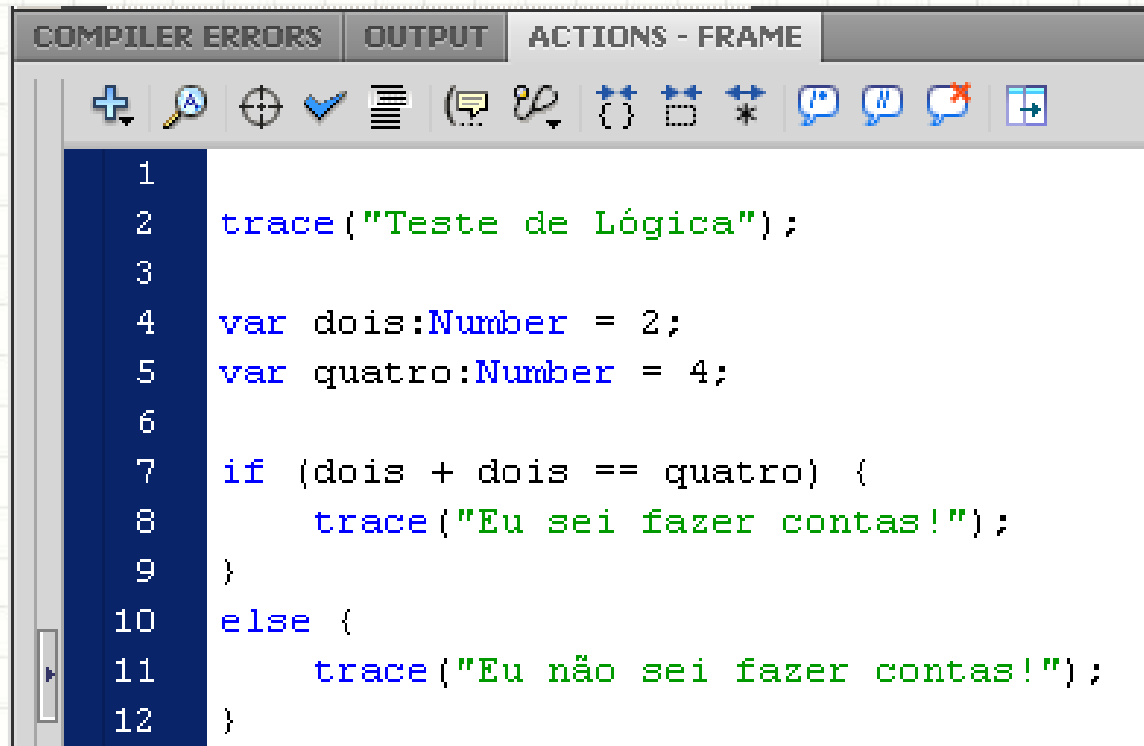




LÓGICA E LAÇOS

Logica e Laços em ActionScript 3

- As estruturas de decisão são idênticas ao C



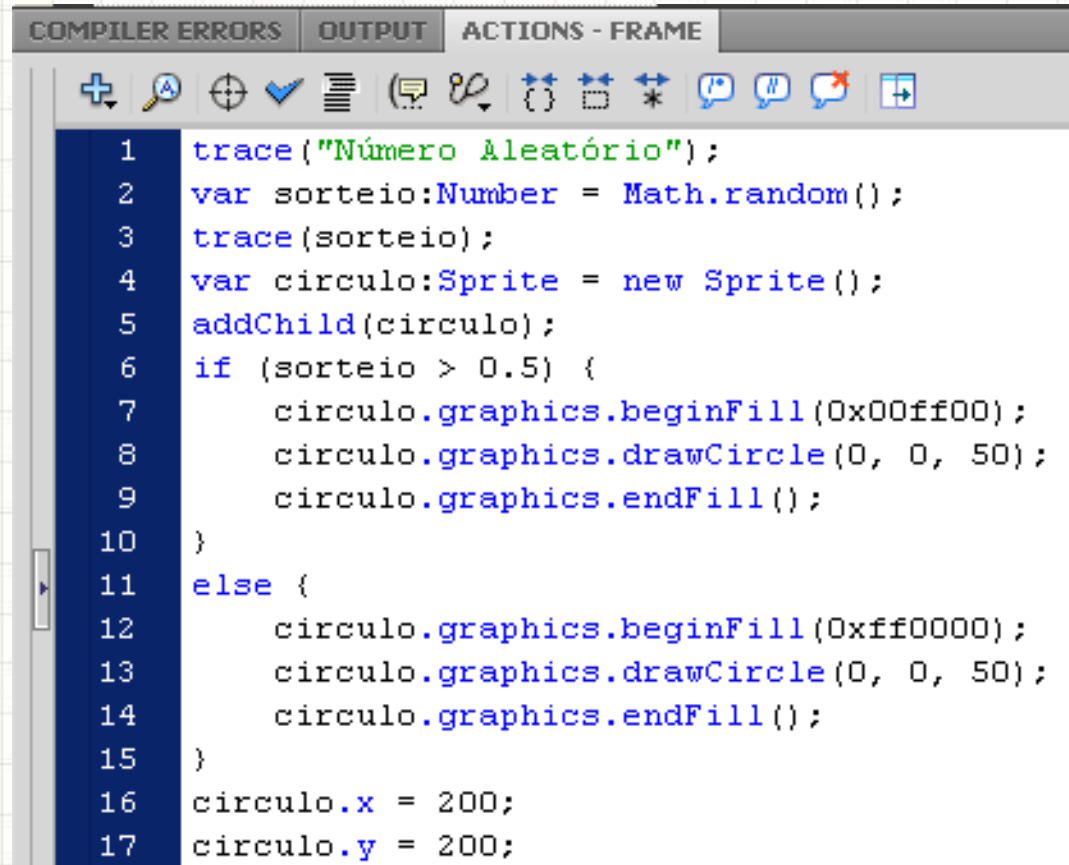
The screenshot shows the 'ACTIONS - FRAME' panel of an IDE. The code editor contains the following ActionScript 3 code:

```
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 brincar com “if”, vamos gerar números aleatórios?

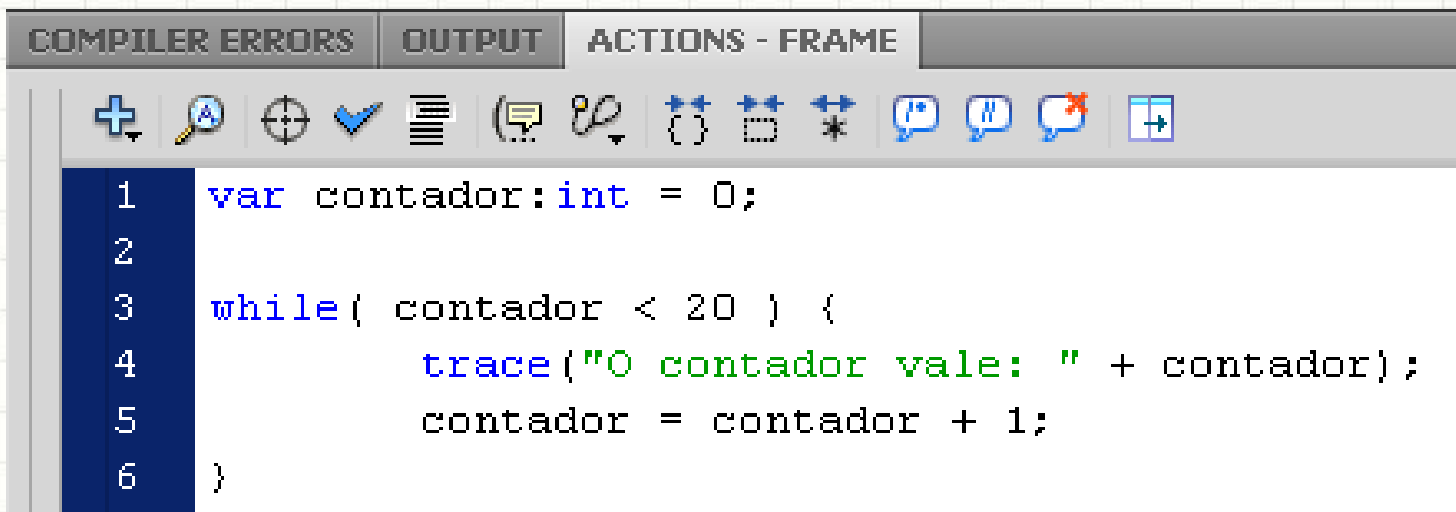


```
COMPILER ERRORS OUTPUT ACTIONS - FRAME  
1 trace("Número Aleatório");  
2 var sorteio:Number = Math.random();  
3 trace(sorteio);  
4 var circulo:Sprite = new Sprite();  
5 addChild(circulo);  
6 if (sorteio > 0.5) {  
7     circulo.graphics.beginFill(0x00ff00);  
8     circulo.graphics.drawCircle(0, 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?

Logica e Laços em ActionScript 3

- Estruturas de repetição são iguais às em C



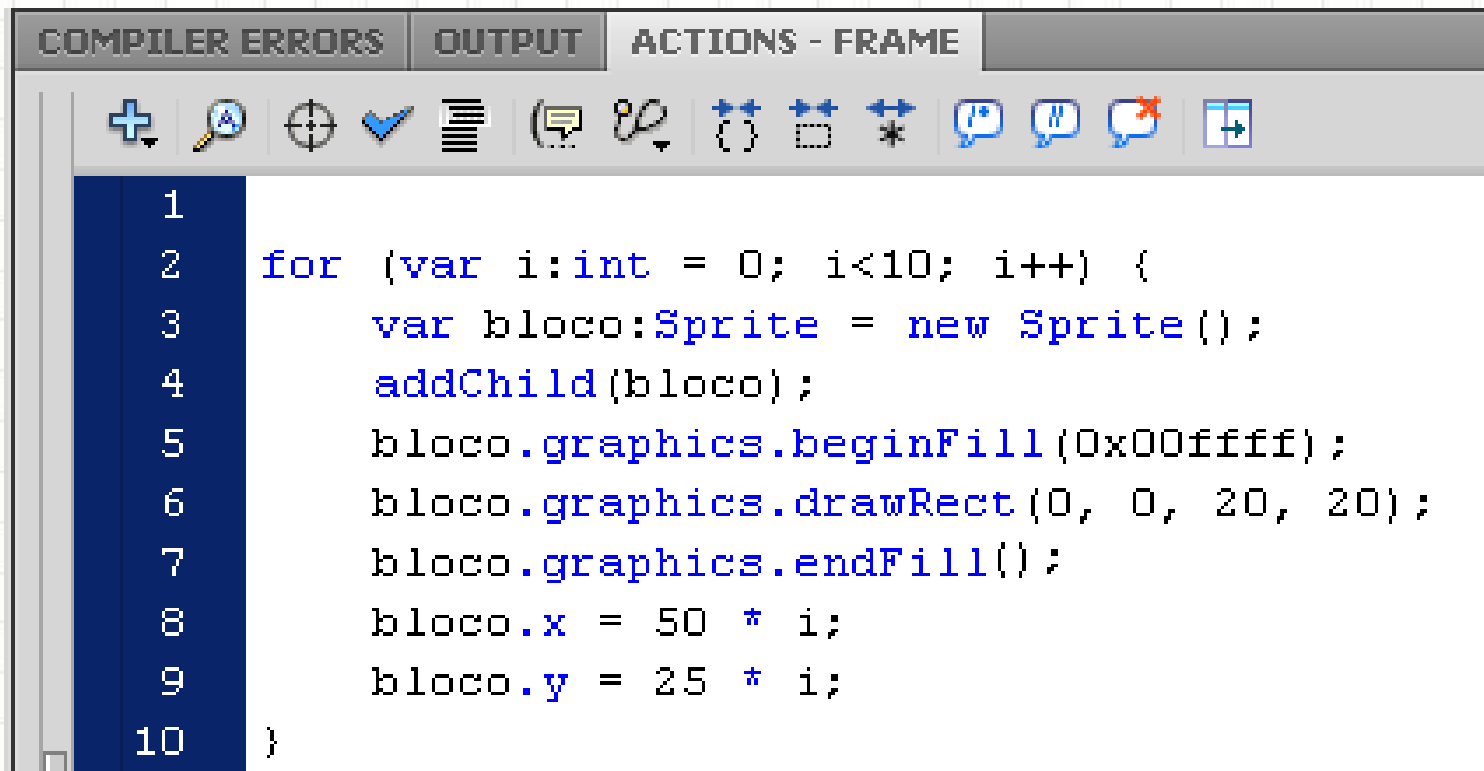
The screenshot shows a code editor window with three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'ACTIONS - FRAME' tab is active. Below the tabs is a toolbar with various icons for code editing. The code area displays the following ActionScript 3 code:

```
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...



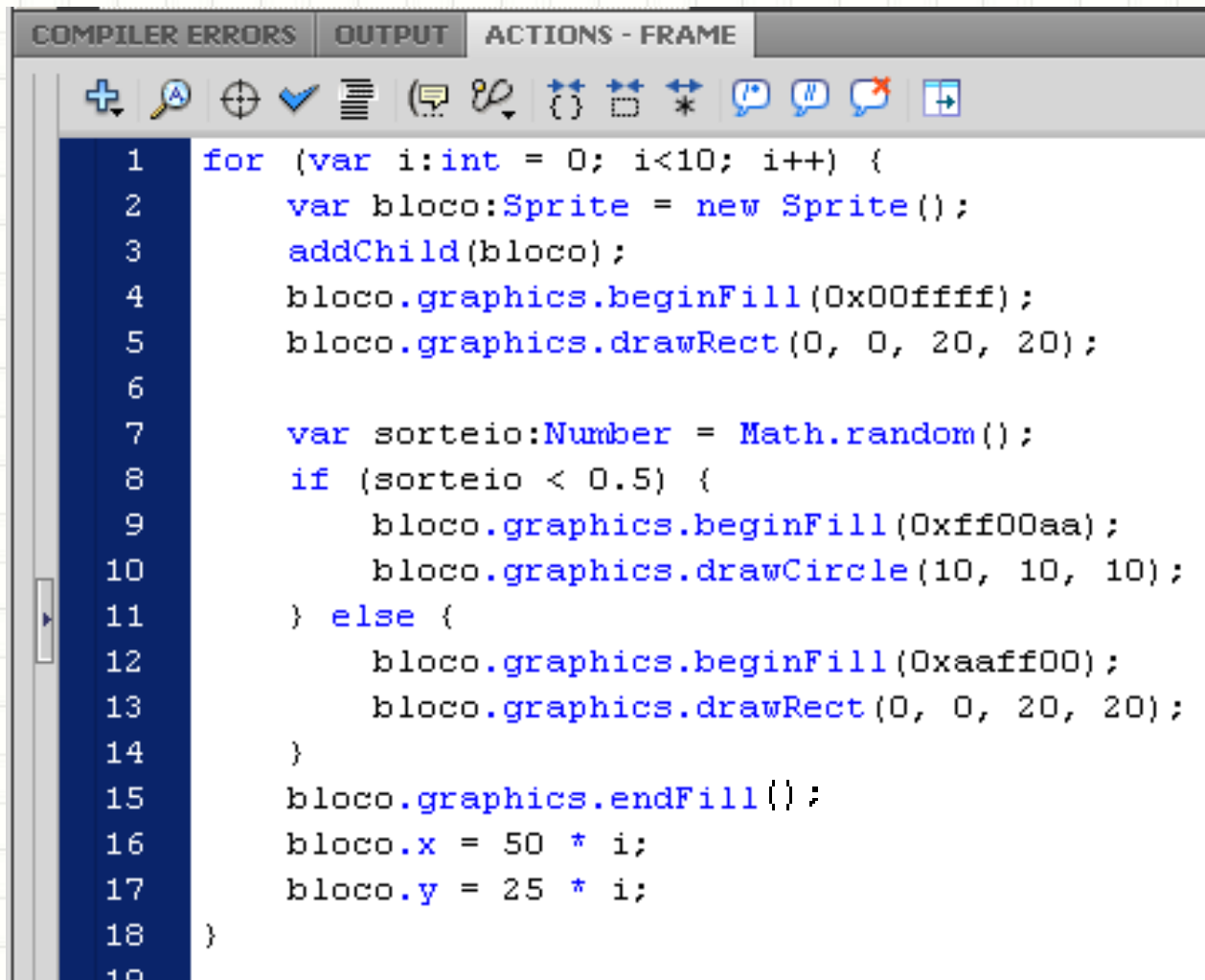
The screenshot shows the 'ACTIONS - FRAME' panel in Adobe Animate. The panel has a toolbar with icons for adding, deleting, and organizing actions. Below the toolbar, a list of 10 actions is displayed, each preceded by a line number. The actions form a loop that creates and draws 10 squares.

```
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...



```
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

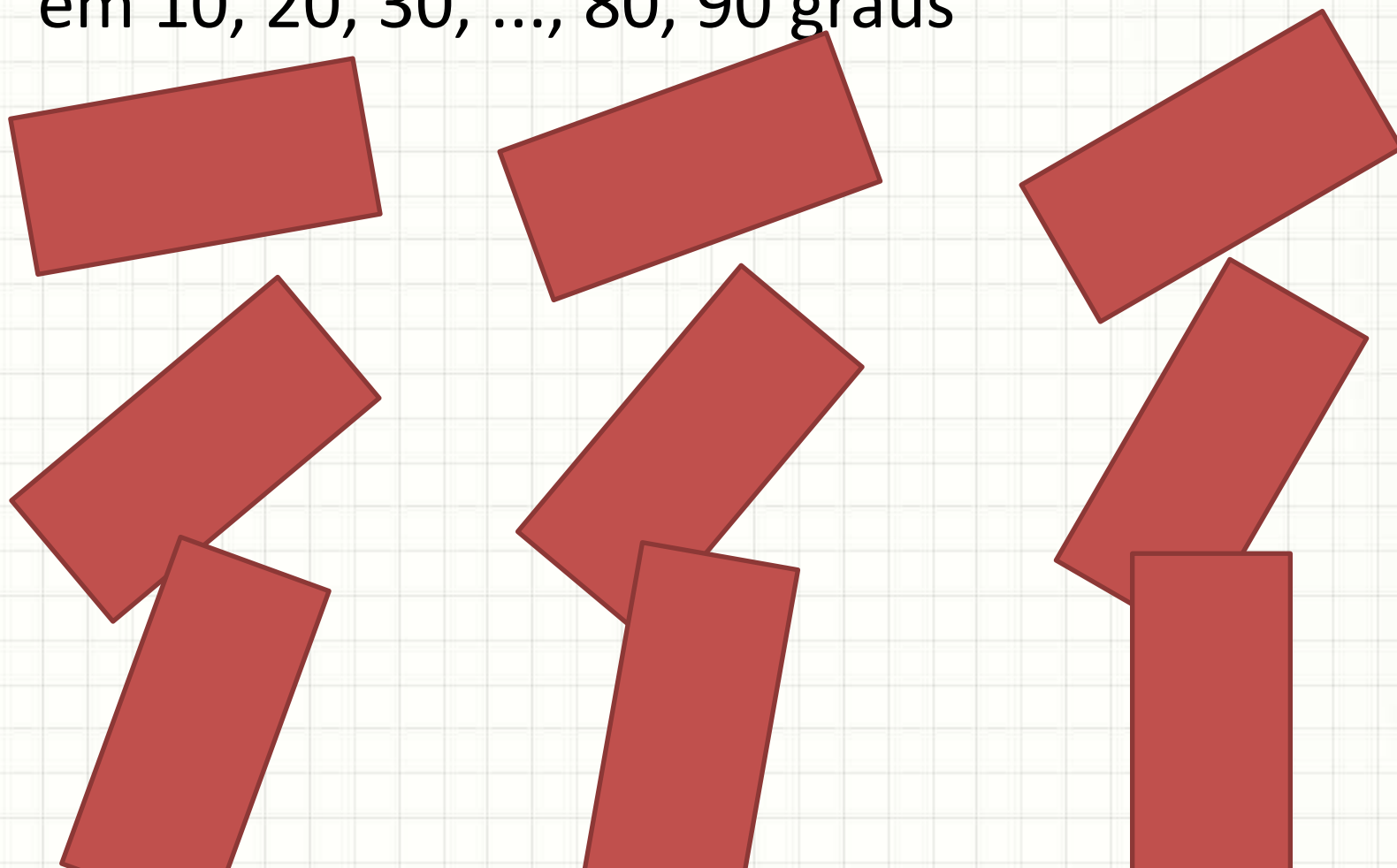
Atividade

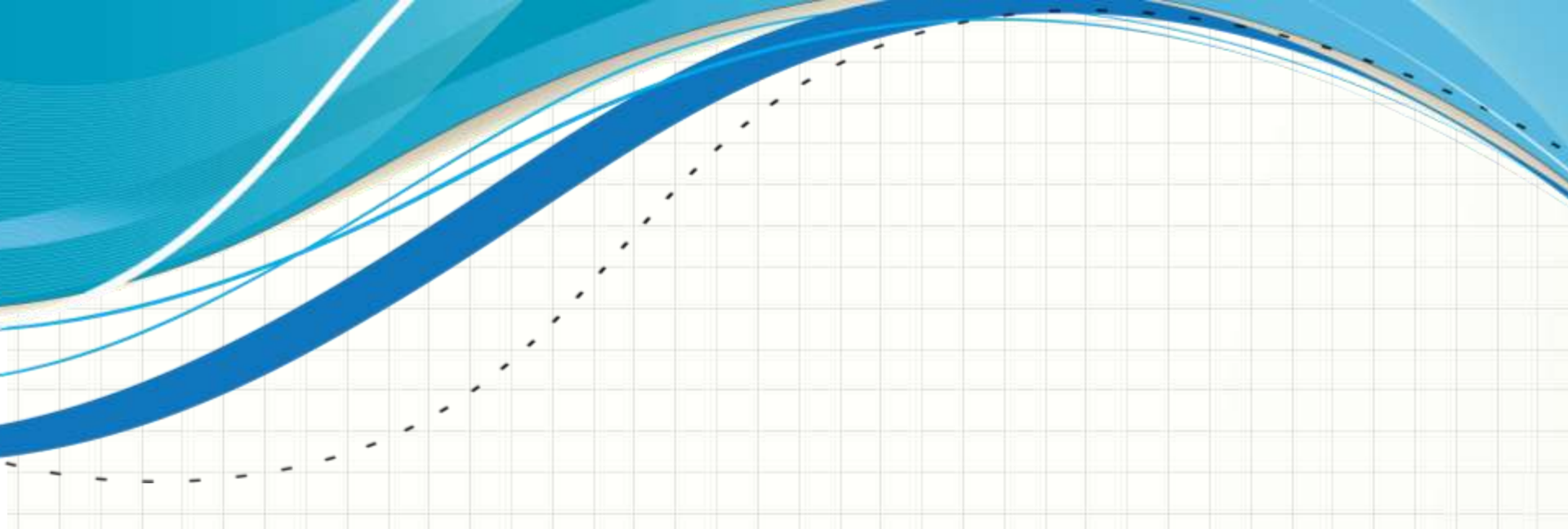
- Faça um programa que mostre 9 retângulos vermelhos 50 x 20 pixels da seguinte forma



Atividade

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

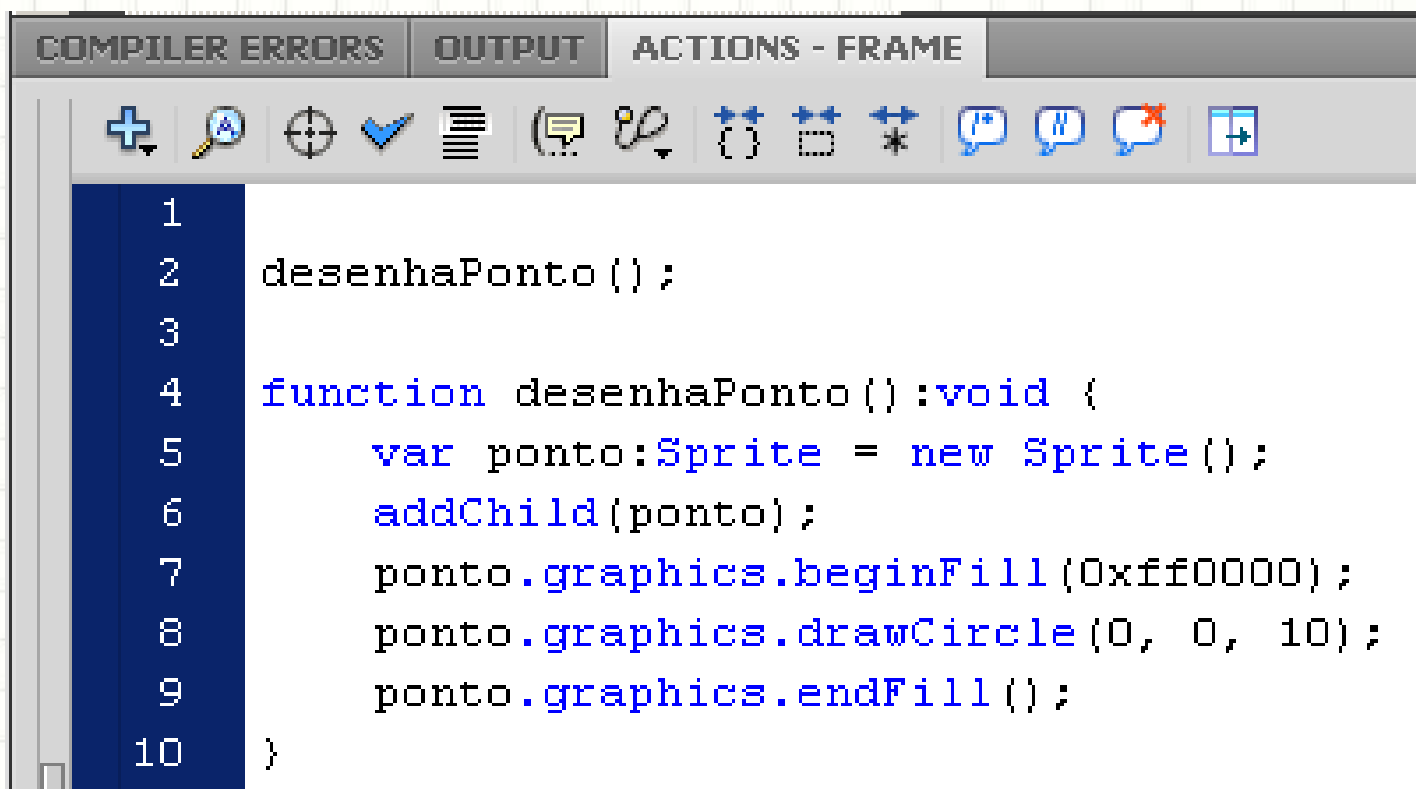




FUNÇÕES EM ACTIONSCRIPT 3

Funções em ActionScript 3

- As funções são parecidas com C



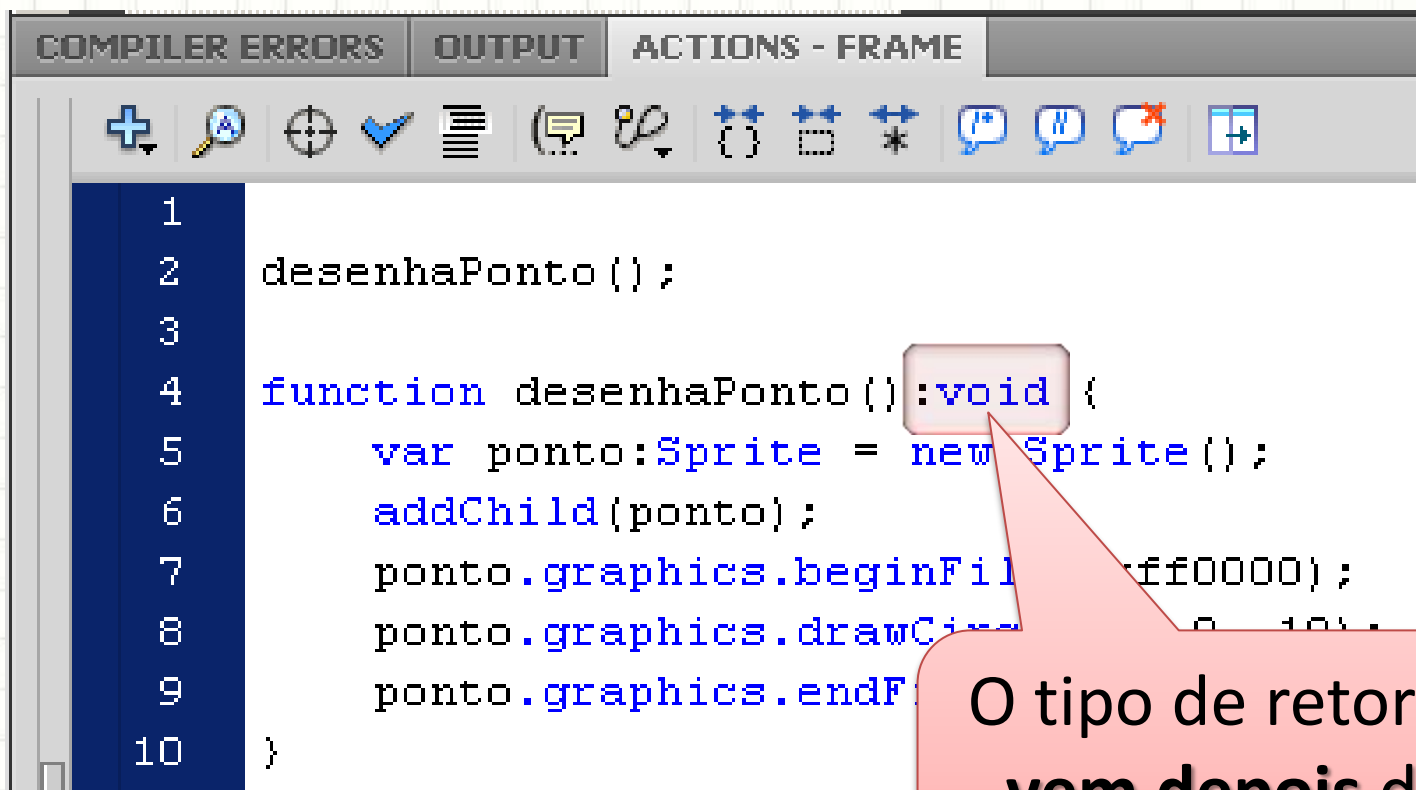
The screenshot shows the 'ACTIONS - FRAME' panel of an IDE. It contains a code editor with the following ActionScript 3.0 code:

```
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 'ACTIONS - FRAME' panel of an IDE. It contains a code editor with the following text:

```
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 }
```

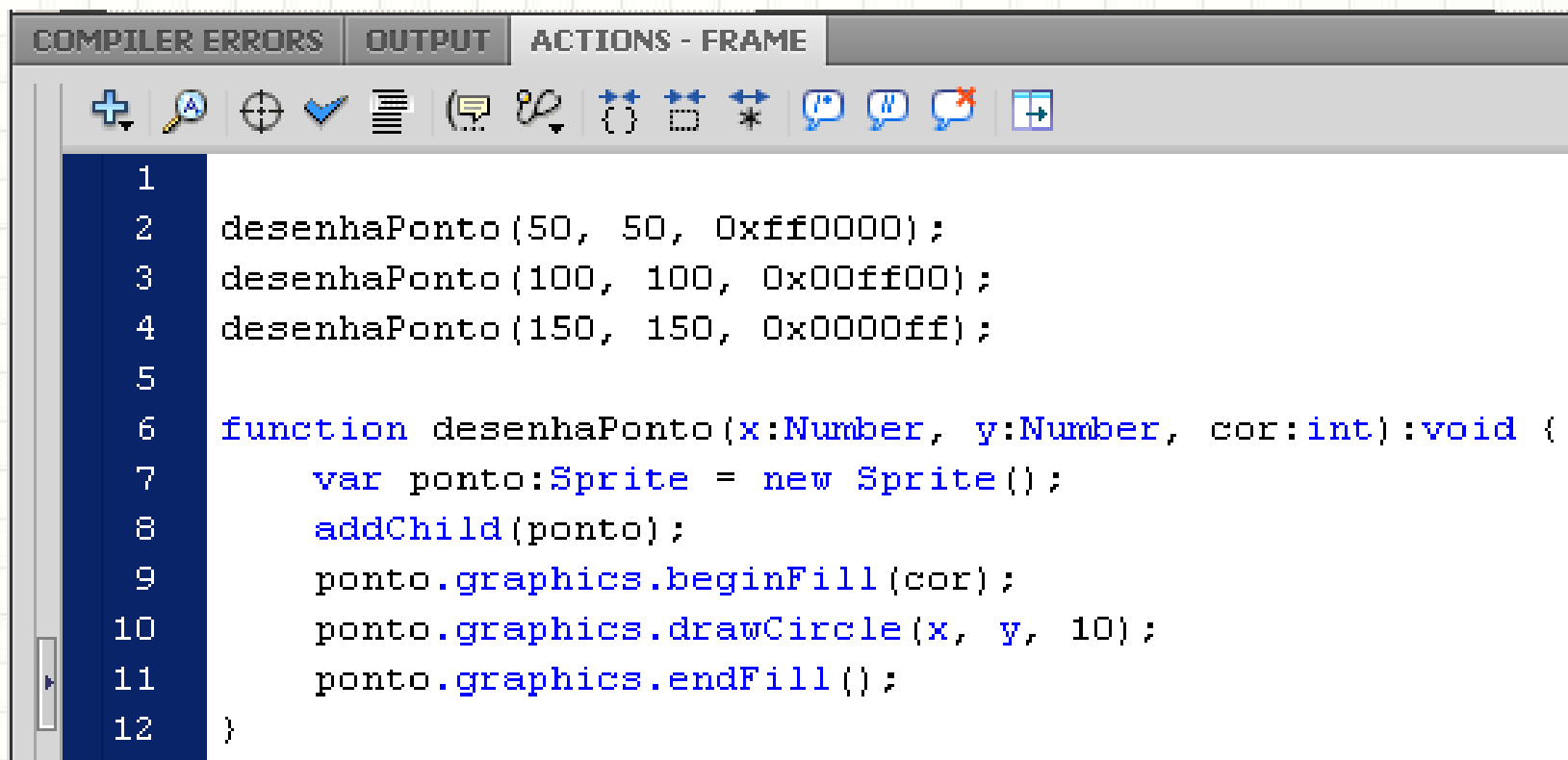
A red callout bubble points to the `:void` return type in line 4.

O tipo de retorno
vem depois do
nome da função

- Experimente!

Funções em ActionScript 3

- Função com parâmetros são mais úteis:



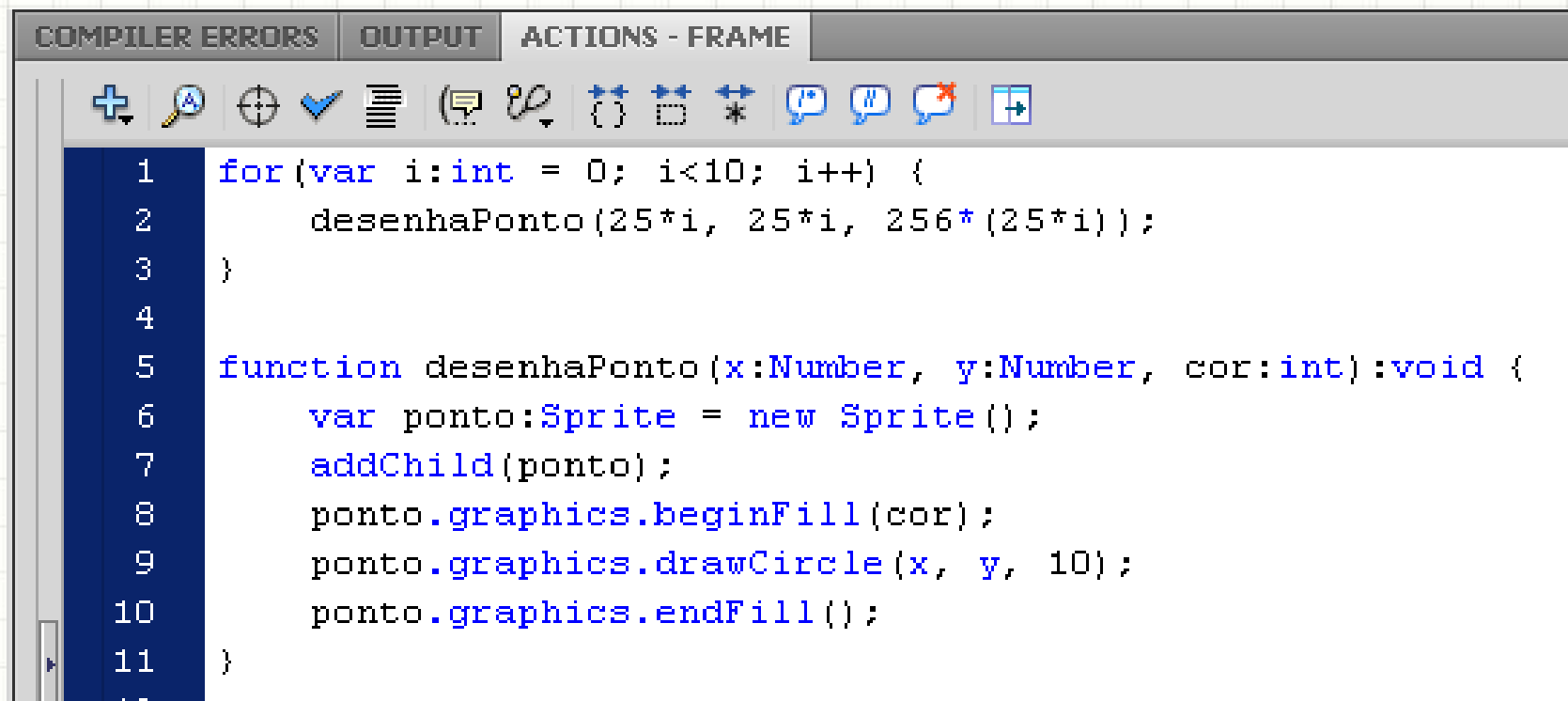
The screenshot shows an IDE window with three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'ACTIONS - FRAME' tab is active, displaying a list of icons for actions like adding, deleting, and commenting. Below the icons, the code is as follows:

```
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 a code editor window with three tabs: 'COMPILER ERRORS', 'OUTPUT', and 'ACTIONS - FRAME'. The 'ACTIONS - FRAME' tab is active. The code is written in ActionScript 3 and consists of a for loop and a function definition. The for loop iterates from 0 to 9, calling a function named 'desenhaPonto' with arguments (25*i, 25*i, 256*(25*i)). The function 'desenhaPonto' takes three parameters: 'x:Number', 'y:Number', and 'cor:int', and returns 'void'. Inside the function, a new 'Sprite' object is created and added to the stage. Then, a circle is drawn at the specified coordinates with a radius of 10, and the fill color is set to the provided color. 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 }  
12
```

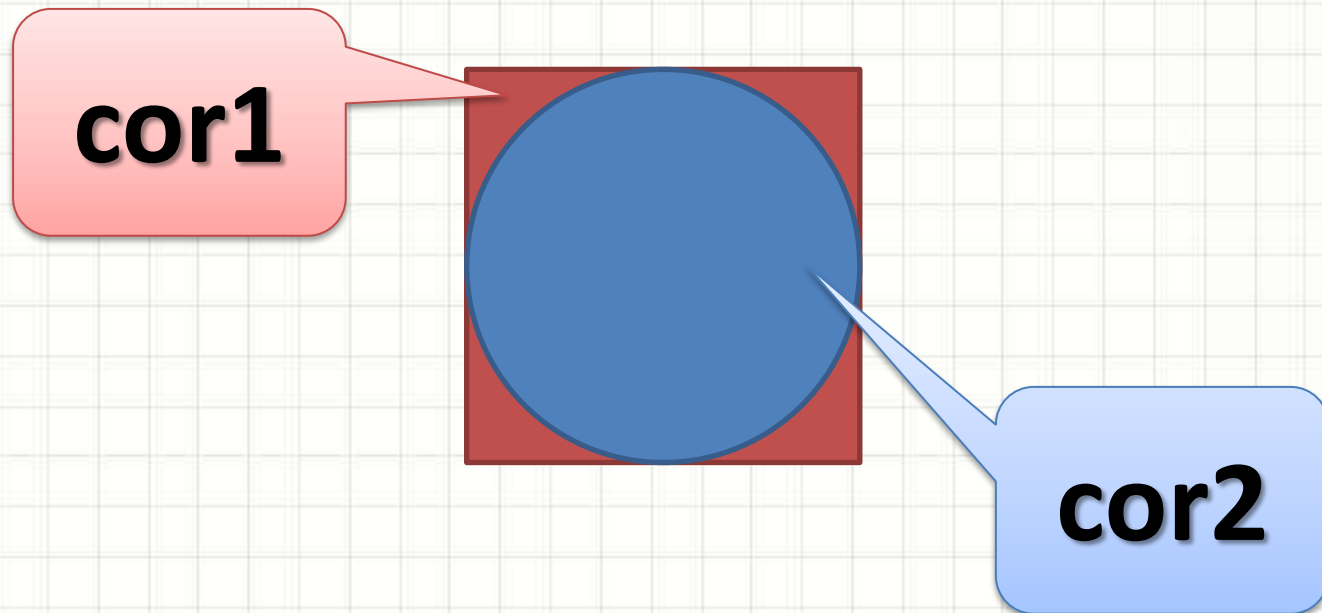
- Experimente!



ATIVIDADE

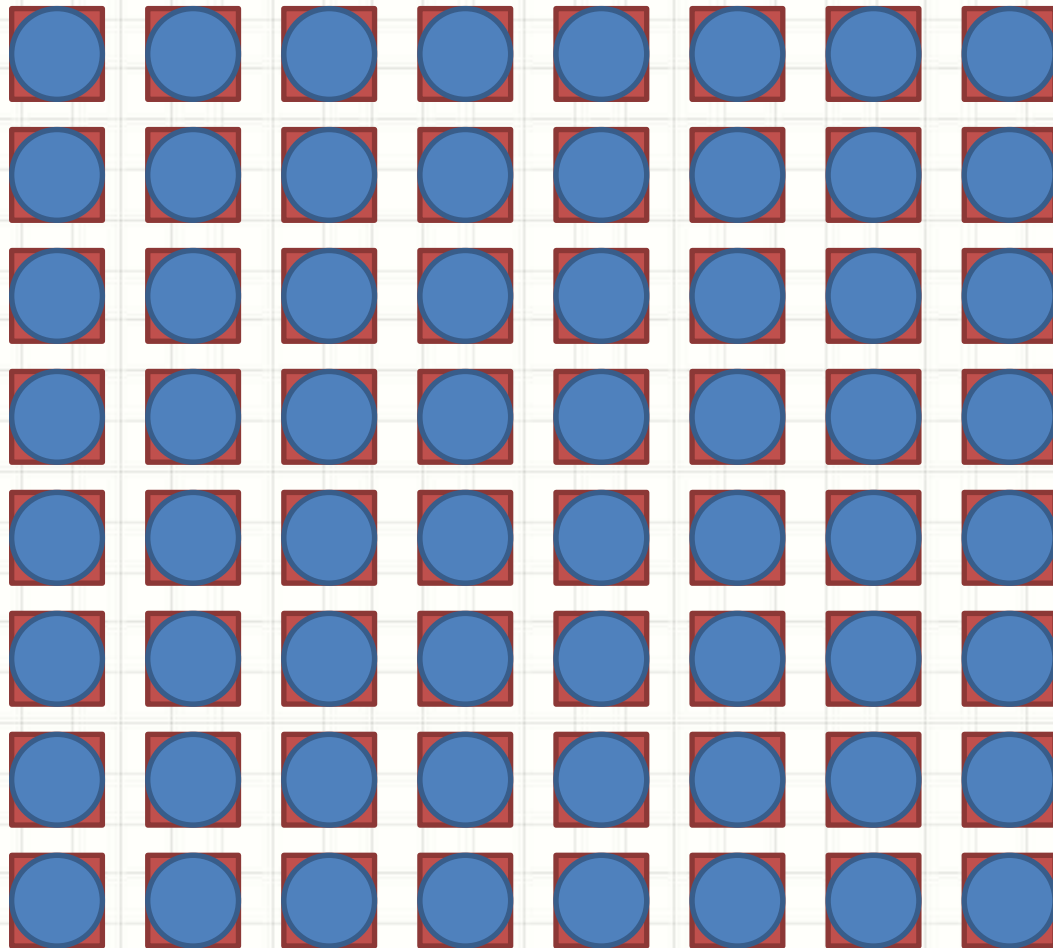
Atividade

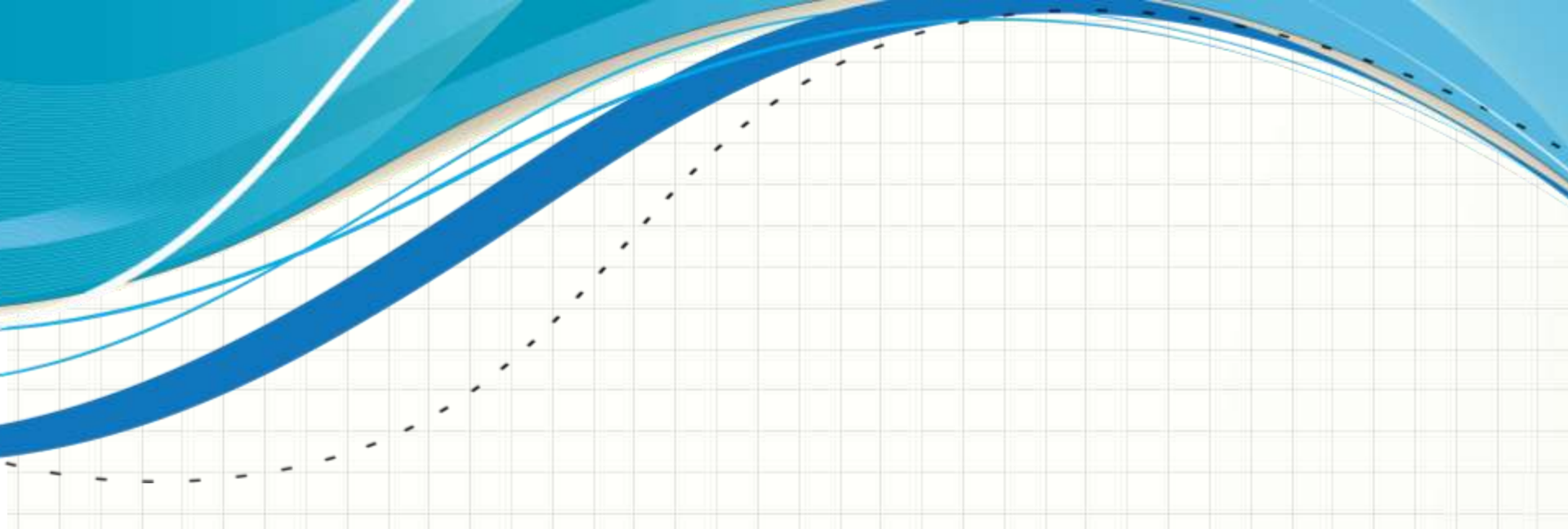
- Faça uma função que receba x , y , lado, cor1 e cor2 e desenhe:



Atividade

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

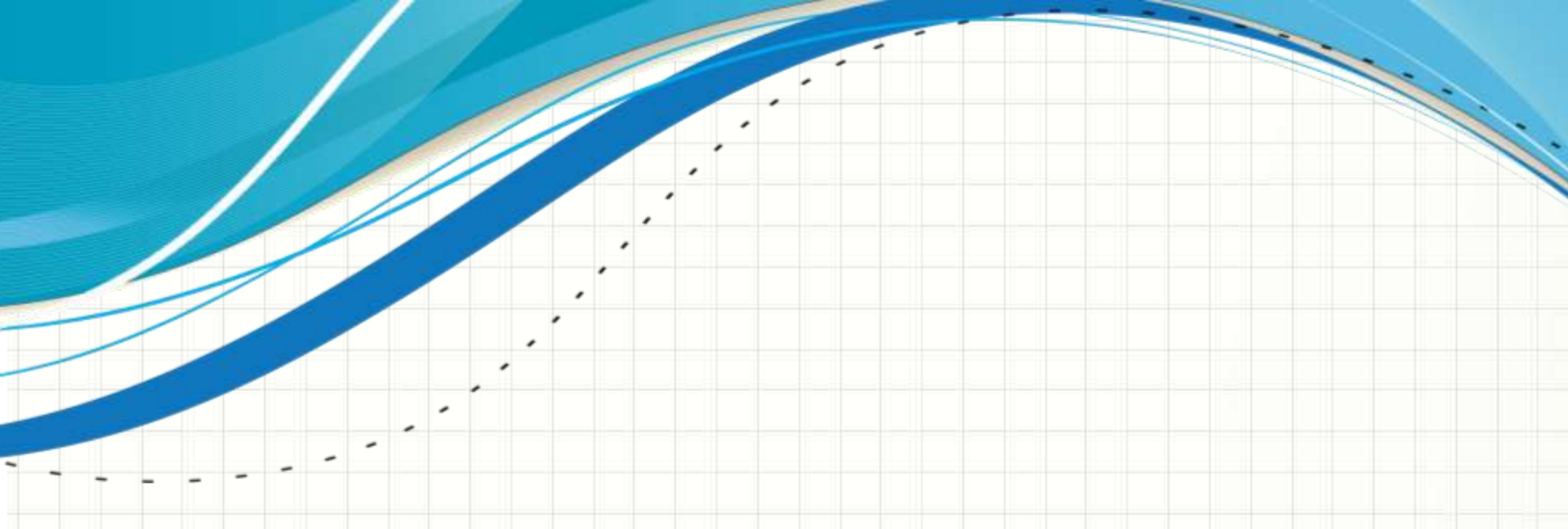




CONCLUSÕES

Resumo

- O ActionScript 3 é uma linguagem similar ao JavaScript/C
- Há uma grande facilidade em trabalhar com formas geométricas
- A interface do Flash é prática para o desenvolvimento de aplicações AS3
- **TAREFA**
 - Trabalho B!

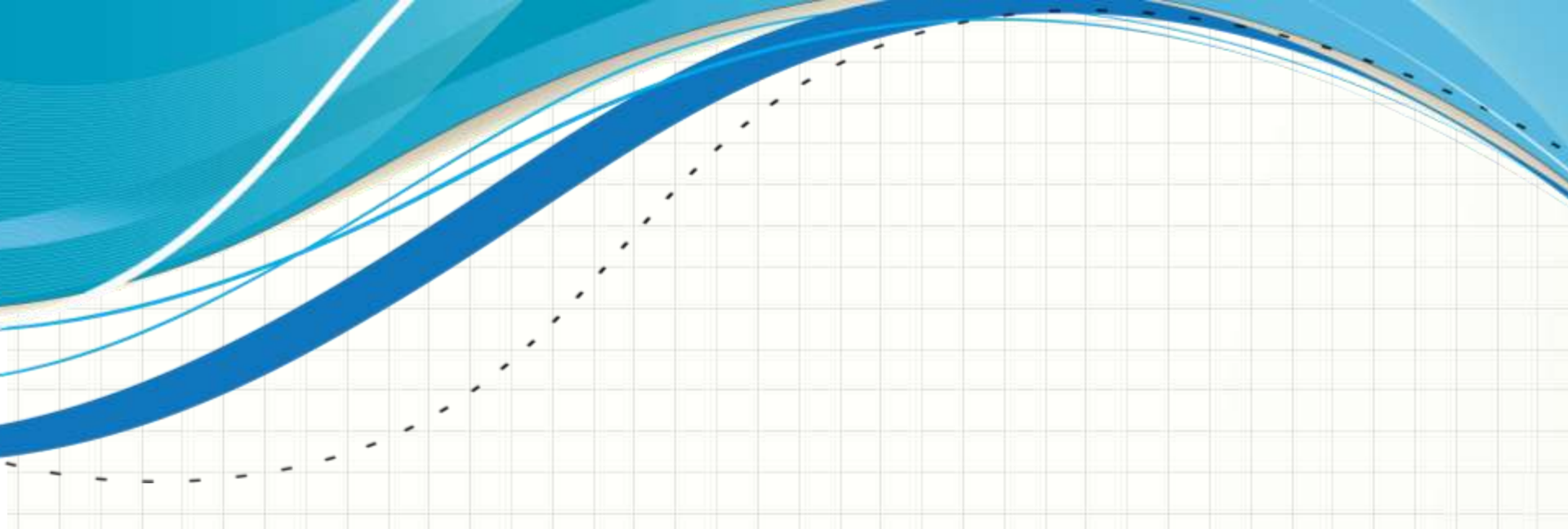


PERGUNTAS?

Próxima Aula



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



**BOM DESCANSO
A TODOS!**