

---

# Introdução a ActionScript 3.0

---

Profa. Flávia Pereira de Carvalho

Fontes:

Adobe Flash CS4 – Design para Web – Processor Alfamídia

Classroom in a Book – Guia de Treinamento Oficial da Adobe

Crie, Anime e Publique seu Site - Fireworks, Flash e Dreamweaver – William Alves

ActionScript 3.0 – Guia de Referência Rápida – Stiller, Shupe, deHaan & Richardson

Programação do ADOBE® ACTIONSCRIPT® 3.0 ([PDF 758pág-10.1MB](#))

<http://www.flexbrasil.com.br/viewtopic.php?f=35&t=980&start=0&sid=55de7d405e35dc4257e3b07832d3d48f>

# Atividade Prática

<http://www.youtube.com/watch?v=B-GVkJJof3o>

- **Tarefa 16: Botões para Controlar Animação**
  - Criar mais 2 camadas e renomeá-las
    - Camada1 = **botoes**
    - Camada2 = **animacoes**
    - Camada3 = **acoes**
  - Selecionar a camada **botoes**: desenhar 3 botões
    - Textos dos botões: **Iniciar, Parar, Voltar**
    - Formatar cores, bordas, fonte

# Atividade Prática

<http://www.youtube.com/watch?v=B-GVkJJof3o>

- **Tarefa 16: Botões para Controlar Animação**
  - Selecionar o botão **Iniciar** (todo botão: **forma e texto**)
  - **Converter em símbolo:**
    - Nome = **btn\_iniciar**
    - Tipo = **button**
    - Fazer nos demais botões:
      - **Parar**: nome = **btn\_parar**
      - **Voltar**: nome = **btn\_voltar**

# Atividade Prática

- **Tarefa 16: Botões para Controlar Animação**
  - **Selecionar o botão Iniciar**
  - **Painel Propriedades > Instance Name**
    - Nomear as instâncias dos 3 botões
      - **Inst\_btn\_iniciar, inst\_btn\_parar, inst\_btn\_voltar**
    - Ctrl + Enter (só pra ver que já são botões)
  - O nome da instância é o que será usado na linguagem ActionScript na hora de se referir ao botão.

---

# Atividade Prática

- **Tarefa 16: Botões para Controlar Animação**
  - Dois cliques no botão **Iniciar** (*ver slide 8*)
    - **Over > F6 >** Mudar cor do botão ou do texto ou da borda
      - 2 cliques fora do botão
    - Fazer os mesmos passos com os outros 2 botões
    - Ctrl + Enter

---

# Atividade Prática

- **Tarefa 16: Botões para Controlar Animação**
  - **F6** no **frame 30** na camada **botoes**: bloquear a camada
  - **Frame 1** da camada **animacoes**: desenhar uma figura
  - **F6** no **frame 30** e mover a figura (na camada **animacoes**)
  - **Create Classic Tween** na camada **animacoes**
    - Mudar trajetória da figura
      - Clicar no frame desejado e mudar a figura de lugar

# Atividade Prática

<http://www.youtube.com/watch?v=ZD-ZF-5agRw>

<http://www.youtube.com/watch?v=VyHYYQXwPYo>

- **Tarefa 16: Botões para Controlar Animação**
  - Selecionar a camada **acoes** (bloquear as outras camadas)
  - **Window > Actions (F9)**
  - Clicar no **frame 1** da camada **acoes**
    - Digitar as ações dos botões: **ActionScript**
  - Aplicar som ao botão (dentro do modo edição do botão):
    - **Window > Common Libraries > Sounds**
      - Clicar com o botão direito = play: para ouvir o som
      - Arrastar o som escolhido para cima do botão (Ctrl+Enter)

# Botões

- **Up:** botão em seu estado normal, sem a proximidade do mouse
- **Over:** estágio em que o usuário passa o mouse sobre o botão
- **Down:** instante em que o usuário está clicando no botão
- **Hit:** área clicável do botão, que não é visualizada pelo usuário

- Voltar ao slide 5



# ActionScript 3.0

- É uma linguagem de programação orientada a objetos (OO) baseada em ECMAScript, utilizada principalmente para construção de aplicações Internet.
- É executada em uma máquina virtual (AVM - ActionScript Virtual Machine), atualmente na versão 3 que está disponível no Adobe Flash Player (plug-in encontrado em navegadores web) e também no ambiente Adobe AIR (Adobe Integrated Runtime).
- Adobe AIR é uma tecnologia que permite a criação de aplicações a partir de tecnologias de desenvolvimento de páginas web, como HTML, Ajax ou Flash.

---

# ActionScript 3.0

- Lançado em Junho de 2006, em conjunto com o ambiente de desenvolvimento Adobe Flex 2.0 e o Flash Player 9.
- ECMAScript é uma linguagem de programação baseada em scripts, padronizada pela Ecma International na especificação ECMA-262 (European Computer Manufacturers Association). A linguagem é bastante usada em tecnologias para Internet, sendo esta base para a criação do JavaScript/JScript e também do ActionScript.

# ActionScript 3.0

- É obrigatória a declaração de variáveis, funções e seus valores retornados, até mesmo quando a função não retorna valor algum.
- Exemplo de código em AS 3:

```
function helloWorld():void {  
    trace("Olá mundo!");  
}
```

# ActionScript 3.0

- Os programadores JavaScript não terão grandes dificuldades para se adaptar à programação ActionScript. São grandes as semelhanças entre elas devido ao uso do padrão ECMAScript (Ecma-262).
- Em ActionScript, é recomendável separar cada linha do código por um ponto-e-vírgula (;). Não é obrigatório, mas no entanto é extremamente recomendável para separar comandos. Além do mais, diferentemente da versão 2.0, o ActionScript 3.0 não permite tipagem fraca: todas os tipos de variáveis devem ser declaradas (tipagem forte). Exemplo:

```
// Se 'nome' abarcará um tipo String, isso deve ser discriminado
var nome:String = new String();

// Modo errado
var nome = "Fulano";
```

# ActionScript 3.0

- Toda função deve ter um valor com que irá retornar, mesmo que seja um valor vazio (**void**). Por exemplo:

```
// Modo errado  
function Ola(){  
    trace("Olá!");  
}
```

```
// Modo correto, no caso de não retornar nenhum valor, por exemplo, em clique de mouse...  
function Ola(evt:MouseEvent):void {  
    trace("Olá!");  
}
```

```
// Modo correto, quando queremos que a função retorne uma String, por exemplo...  
function exhibeNome(nome:String):String {  
    var frase:String = new String();  
    frase = "Seu nome é " + nome;  
    return frase;  
}
```

# ActionScript 3.0

- Todos os comandos de uma função, cláusula condicional, loops, classes e pacotes devem estar inseridos dentro de chaves ({ }).
- A indentação não é obrigatória, mas é extremamente recomendável, assim como na maioria das linguagens.
- Variáveis e funções não devem ter nomes que iniciam com números, nem com os caracteres "\_" ou "\$".

# ActionScript 3.0

- Os comentários são úteis tanto para inserir lembretes e informações sobre o código como para tirar linhas de código que numa compilação não sejam executadas (útil para testes).
- Há duas maneiras de comentar em ActionScript:

```
// Comentário apenas uma linha
```

```
/* Comentário que  
ocupa várias linhas.  
Aqui é o fim */
```

# ActionScript 3.0

- Além do conhecido Adobe Flash, há outras IDEs (Ambiente de Desenvolvimento Integrado) que permitem a programação de ActionScript. A maioria deles são gratuitas. Dentre as mais conhecidas podemos citar:
- **Flex Builder:** feita para os programadores tradicionais interessados em desenvolver para a Plataforma Flash. Não é apenas um editor de script, pois inclui vários componentes de interface de usuário (UI) e classes indisponíveis para Flash (mas disponíveis no Flex SDK). Feita para a criação de RIAs (*Rich Internet Applications*).
- **FlashDevelop:** IDE gratuita, com vários recursos que agilizam a criação, e código aberto para Windows, construída sobre Microsoft .NET Framework. Para criação de AS em versão 2 e 3, também é possível criar HTML, CSS e outras. Com auto-complete e outros recursos.
- **SE|PY:** editor gratuito para Windows e Mac, desenvolvido em Python por Alessandro Crugnola, originalmente para AS 2.0, mas também permite edição de AS 3.0 e JavaScript.
- **PrimalScript:** criada pela SAPIEN Technologies, suporta mais de 40 linguagens para criação, incluindo HTML/CSS, JavaScript, Java, C#, Flash e Flex, Python, Ruby, Perl, entre muitas outras. Funciona por um período de teste de 45 dias.
- **Notepad++:** mais limitado para uso de AS, é um editor de texto mais avançado que o Bloco de Notas do Windows, por exemplo. Permite coloração de sintaxe AS e de outras linguagens.



# ActionScript 3.0

- Tipos de variáveis no ActionScript:
  - **Number**: qualquer número, incluindo os de ponto flutuante (exemplo: 4.5)
  - **int**: qualquer número inteiro, positivo ou não (452, -32...)
  - **uint**: número inteiro e não negativo (2,487,0...)
  - **String**: para textos ("olá", "Fulano Beltrano")
  - **Boolean**: valor booleano (true ou false)
  - **Array**: Registro de um ou mais valores ([25,"Fulano",54.47])
  - **Object**: A estrutura básica de qualquer valor ActionScript, pode ser usado também como alternativa ao tipo Array - ser usado como um dicionário (meuObj.nome = "Fulano").

# ActionScript 3.0

- Para declarar uma variável:

```
var meuNome:String = new String();  
// Ou com um valor definido  
var meuNome:String = "João";
```

- Não é preciso configurar um tipo de codificação (Unicode, ANSI, etc.) em ActionScript: ele trabalha de maneira tranquila com caracteres e acentos como cedilha (ç), til (~), pois, por padrão, utiliza o Unicode, que é uma especificação que reconhece caracteres latinos, japoneses, russos, entre outros. Para alterar a codificação, basta ir em:

**Edit > Preferences > ActionScript** (na IDE Adobe Flash)

# ActionScript 3.0

## ■ Constantes

- No AS 3.0 foi introduzido o conceito de constantes, que nada mais são do que valores com um valor fixo e imutável.

```
const IDIOMA:String = "Português";  
IDIOMA = "Inglês"; // Acusará um erro
```

- Por convenção (e não obrigação) é comum declarar constantes com nome em letras **maiúsculas**.

# ActionScript 3.0

- A declaração **if** é uma das mais comuns. Também podem ser usadas as declarações **else if** e **else**.  
Exemplo:

```
var idade:int = 17;

if (idade < 18){
    trace("Você é menor de idade.");
}

else if (idade > 18 && idade < 65){
    trace("Você é maior de idade e adulto.");
}

else {
    trace("Você tem mais de 65 anos");
}
```

# ActionScript 3.0

- Operadores de comparação:
  - Igualdade: `==`
  - Não-igualdade: `!=`
  - Maior: `>`
  - Menor: `<`
  - Maior ou igual: `>=`
  - Menor ou igual: `<=`
- Operadores lógicos:
  - Operador E: `&&`
  - Operador OU: `||`
  - Operador de negação (false): `!`

# ActionScript 3.0

Exemplo:

```
var nome:String = "Fulano";

if (nome == "Fulano" || nome == "Beltrano" ){
    trace("Aceito.");
}

else if (nome != "Fulano" && nome != "Beltrano"){
    trace("Seu nome não é Fulano nem Beltrano.");
}

else {
    trace("Nome desconhecido");
}
```

Há também o operador `is` que analisa se uma condição é verdadeira ou se uma variável é de algum tipo:

```
var txt:String = new String();

if (txt is String)
{
    txt = "verdadeiro";
}
else
{
    txt = "falso";
}
```

No caso, `txt` será igual a "verdadeiro", pois é uma *String*.

# ActionScript 3.0

## ■ switch

- ❑ Há a opção switch, que é um modo de comparação mais simplificado e leve que o if, porém não pode ser usado para comparações complexas.
- ❑ Em vez de if, esta condicional utiliza a palavra **case** para sinalizar uma comparação.
- ❑ E no fim de cada comparação é necessário inserir a palavra reservada **break**, senão todos os comandos serão lançados até o fim do switch (causando erro de comparação).
- ❑ A palavra **default** executa uma ação caso nenhuma das comparações tenha sido bem-sucedida.

# ActionScript 3.0

Exemplo:

```
var nota:int = 7;

switch (nota)
{
    // Notas para ser aprovado...
    case 7:
        trace("Aprovado no limite!");
        break;

    case 8:
        trace("Boa nota, aprovado.");
        break;

    case 9:
        trace("Ótimo, aprovado, parabéns.");
        break;

    case 10:
        trace("NOTA DEZ! Excelente!!!");
        break;

    // Qualquer nota abaixo de 7
    default:
        trace("Você não foi aprovado...");
        break;
}
```



# ActionScript 3.0

**for:** loop para executar ações de acordo com uma condição.

```
for (var i:Number = 0; i < 10; i++) {  
    trace(i.toString());  
}  
  
// Imprimirá: 0,1,2,3,4,5,6,7,8,9
```

Na primeira linha do loop criamos uma variável especificamente para o loop (no caso, **i**).

Logo após, fazemos uma comparação: se **i** for menor que 10, iremos executar `trace(i.toString())` e também incrementaremos o valor de **i** (`i++`).

O `i++` traz o mesmo resultado que `i = i+1`. Ou seja, ele incrementa 1 a cada iteração. Há também a possibilidade de subtrair: através de `i--`.

# ActionScript 3.0

## while

---

O `while`, diferentemente do loop `for`, realiza a iteração se uma expressão for verdadeira. Exemplo:

```
var num:Number = 0;
```

```
// Fará trace("Tá no looping...") infinitamente porque num é menor que 0.5
```

```
while (num < .5) {  
    trace("Tá no looping...");  
}
```

# ActionScript 3.0

É preciso ter cuidado: é muito fácil errar com este tipo de loop, porque se você não fizer nada para tornar a expressão **false** ele irá repetir a operação infinitamente, o que poderá fazer o seu programa travar (caso do exemplo anterior). Para fazer o loop parar, é só fazer a expressão tornar-se falsa. Exemplo:

```
var num:Number = 0;

// O loop parará quando num for maior que 0.5, ou seja, na próxima iteração

while (num < .5) {
    trace("Tá no looping...");
    num++;
}
```

# ActionScript 3.0

Funções podem conter parâmetros ou não. E podem retornar em determinados tipos (String, Number...) ou não (retornando void).

```
function digaOla (seuNome:String) :void  
{  
    trace("Olá " + seuNome + "!");  
}
```

```
trace("Fulano");  
// Exibirá "Olá Fulano!"
```

# ActionScript 3.0

Se você quer que uma função retorne um valor em um determinado tipo, esse tipo de dado deverá ser discriminado depois do sinal de dois pontos (:). Exemplo:

```
// Deverá retornar um tipo Number
function Dobro(numero:Number) :Number
{
    return numero * numero;
}

trace (Dobro (2) ) ;
// Exibirá 4
```

# Atividade Prática

Apostila Adobe: Programação do ActionScript 3.0 – Pág. 23

- **Tarefa 17: Botões para Controlar Animação e Abrir Site**
  - Criar mais 2 camadas e renomeá-las
    - Camada1 = **botoes**
    - Camada2 = **animacoes**
    - Camada3 = **acoes**
  - Selecionar a camada **botoes**: desenhar 2 botões
    - Textos dos botões: **Animar, Google**
    - Formatar cores, bordas, fonte

# Atividade Prática

Apostila Adobe: Programação do ActionScript 3.0 – Pág. 23

- **Tarefa 17: Botões para Controlar Animação e Abrir Site**
  - Selecionar o botão **Animar** (todo botão: **forma e texto**)
  - **Converter em símbolo:**
    - Nome = **btn\_animam**
    - Tipo = **button**
    - Fazer o mesmo no botão Home:
      - **Google**: nome = **btn\_google**

# Atividade Prática

Apostila Adobe: Programação do ActionScript 3.0 – Pág. 23

- **Tarefa 17: Botões para Controlar Animação e Abrir Site**
  - Selecionar o botão **Animar**
  - **Painel Propriedades > Instance Name**
    - Nomear as instâncias dos 2 botões
      - **inst\_btn\_animar, inst\_btn\_google**
    - **Ctrl + Enter** (só pra ver que já são botões)
  - O nome da instância (ocorrência) é o que será usado na linguagem ActionScript na hora de se referir ao botão. Instância é uma cópia do objeto.



# Atividade Prática

Apostila Adobe: Programação do ActionScript 3.0 – Pág. 23

- **Tarefa 17: Botões para Controlar Animação e Abrir Site**
  - Na camada **animacoes**
    - Fazer as animações desejadas
  - Clicar no **frame 1** da camada **acoes** = **F9**
    - Digitar as ações em **ActionScript**

# Atividade Prática

Apostila Adobe: Programação do ActionScript 3.0 – Pág. 31

- **Tarefa 18: Criar Aplicativo do ActionScript**
  - **File > New = ActionScript File**
    - Digite as ações
  
  - **File > Save = selecione o local e o nome do arquivo**
    - Nome = **greeter.as**

# Atividade Prática

Apostila Adobe: Programação do ActionScript 3.0 – Pág. 31

- **Tarefa 18: Criar Aplicativo do ActionScript**
  - **File > New = Flash File (ActionScript 3.0)**
  - Ferramenta **Texto**: faça um campo de texto
  - Ferramenta **Free Transform**: 300 px X 100 px
  - Ferramenta **Texto**: tamanho, cor, tipo de letra
    - Painel Propriedades
      - Instance Name = **mainText**
      - **Dynamic Text**

# Atividade Prática

Apostila Adobe: Programação do ActionScript 3.0 – Pág. 31

- **Tarefa 18: Criar Aplicativo do ActionScript**
  - **No frame 1 = F9**
    - Digitar o código script
  - **Salvar o arquivo = helloworld fla**
    - Salvar no mesmo local do arquivo **greeter.as**
    - **Ctrl + Enter**

# Atividade Prática

Apostila Adobe: Programação do ActionScript 3.0 – Pág. 31

- **Tarefa 18: Criar Aplicativo do ActionScript**
  - Abrir o arquivo **Greeter.as**
    - Digitar o código script
  
  - Abrir o arquivo **helloworld fla**
    - Clicar no **frame 1 = F9**
    - Modificar o script
    - Ferramenta **Texto**
      - Desenhar 2 campos abaixo do mainText existente
      - No primeiro campo = **User Name:**

# Atividade Prática

Apostila Adobe: Programação do ActionScript 3.0 – Pág. 31

- **Tarefa 18: Criar Aplicativo do ActionScript**
  - Selecionar o segundo campo: Painel Propriedades
    - **Input Text**
    - Instance Name = **textIn**
    - Behavior = **Single Line**
  - Clicar no frame 1 e digitar as ações
  - Salvar e testar

# Atividade Prática

<http://www.youtube.com/watch?v=RyjPwI89An0&feature=related>

## ■ Tarefa 19: Menu Animado

- Renomear a camada para home
  - Desenhar um retângulo: ver posição e medidas
  - Selecionar o retângulo: converter em símbolo
    - Movie Clip – Nome = home
  - Entrar no modo edição do MovieClip: dois cliques
    - Camada1 = retângulo1
    - Camada2 = retângulo2
    - Na camada retangulo2: desenhar um retângulo da mesma altura do outro, mas mais escuro e na mesma posição

# Atividade Prática

<http://www.youtube.com/watch?v=RyjPwI89An0&feature=related>

## ■ Tarefa 19: Menu Animado

- Camada retangulo1 = frame 20 = F6
- Camada retangulo2 = frame 10 = F6 e frame 20 = F6
- Camada retangulo2 = frame 10
  - ❑ Ferramenta Free Transform: aumentar retângulo escuro
  - ❑ Clicar no frame 1: botão direito = copiar frame
  - ❑ Clicar no frame 20: botão direito = colar frame
  - ❑ Botão direito entre os frames: Create Shape Tween
    - Entre o frame 1 e 10 e também entre o 10 e o 20



# Atividade Prática

## ■ Tarefa 19: Menu Animado

- Camada **retangulo2**: clicar no **frame 1 = F9**
  - Digitar: **stop()**;
  - Fazer o mesmo no **frame 10**
- Inserir nova camada = **texto**
  - **Ferramenta Texto**: digitar Home (em cima do retângulo)
  - Frame 20 = F6

---

# Atividade Prática

## ■ Tarefa 19: Menu Animado

- Voltar para cena 1
- Selecionar o objeto todo: F9 (digitar o script)

```
on (rollOver) {  
    gotoAndPlay(2);  
}  
on (rollOut) {  
    gotoAndPlay(11);  
}
```

---

# Atividade Prática

## ■ Tarefa 19: Menu Animado

```
on (release) {  
    getURL( "http://fit.faccat.br" )  
}
```