

Web Games

Prof. Luciano Alves

Variáveis e Tipos de dados

O que são variáveis

Variáveis, na programação, nada mais são do que endereços de memória onde cada posição de memória “assume” um determinado valor durante a execução de um programa (no nosso caso, um jogo).

Os valores armazenados na memória podem, durante o jogo, serem modificados. Por isso chama-se “variáveis”.

Como utilizar uma variável ?

Toda variável, na programação, precisa ser “declarada” antes de ser utilizada no programa (jogo). Para isso, precisamos conhecer a “sintaxe” da linguagem para podermos fazer a devida declaração. Eis abaixo a sintaxe de declaração de uma variável.

Sintaxe :

var <nome da variável> : <tipo> ;

Exemplo :

var nome : **String** ;

Tipos de dados

Na programação toda variável assume um valor , e esse valor precisa corresponder a um determinado tipo de dados. Veja abaixo os tipos de dados mais comuns na linguagem ActionScript:

int : Armazena números inteiros de 32 bits (positivos e negativos), cujo intervalo é 2.147.483.648 a 2.147.483.647.

uint : Armazena números inteiros de 32 bits (somente), cujo intervalo é 0 a 4.294.967.295.

Number : Armazena números inteiros e de ponto flutuante (positivos e negativos) de 64 bits.

Tipos de dados

String : Armazena valores alfanuméricos (incluindo números, letras e símbolos) no formato Unicode (caracteres de 16 bits cada). Seu tamanho é variável.

Boolean : Armazena valores lógicos como **true** (Verdadeiro) e **false** (Falso).

Variáveis

Exemplos de declarações de variáveis :

```
var nome_jogador : String ;
```

```
var total_de_pontos : int;
```

```
var inimigo_attingido : Boolean ;
```

Variáveis

Observações sobre declarações de variáveis :

Quando temos mais de uma variável de um mesmo tipo, podemos declará-las separando-as por vírgula na mesma linha:

```
var nome_jogador, nome_jogador2, nome_jogador3 : String ;
```

Variáveis

Comando de atribuição :

O comando de atribuição é uma expressão que utilizamos para atribuir um determinado valor para uma variável. Veja a sintaxe abaixo:

Sintaxe :

<variável> = <valor>;

Variáveis

Comando de atribuição :

Exemplos :

```
var nome_jogador: String;
```

```
nome_jogador = "Luciano";
```

```
var preco: Number;
```

```
idade= 12.5;
```

```
var idade: int;
```

```
idade= 23;
```

```
var pode_atacar: Boolean;
```

```
pode_atacar= true;
```

Variáveis

Operadores aritméticos:

A linguagem **ActionScript** oferece um conjunto de operadores aritméticos que podemos utilizar em nossas expressões em uma atribuição por exemplo. Veja os operadores abaixo:

Operador	Simbolo	Com quais tipos de dados funcionam ?
Adição	+	(unit, int, String, Number)
Subtração	-	(unit, int, Number)
Multiplicação	*	(unit, int, Number)
Divisão	/	(unit, int, Number)
Módulo	%	(unit e int)

Variáveis

Operadores aritméticos (Exemplos):

```
soma = numero1 + numero2;
```

```
media = (numero1 + numero2) / 2;
```

```
distancia = velocidade * tempo;
```

```
resto_divisao = 5 % 2;
```

Variáveis

Operadores aritméticos (Exemplos):

```
soma = numero1 + numero2;
```

```
media = (numero1 + numero2) / 2;
```

```
distancia = velocidade * tempo;
```

```
resto_divisao = 5 % 2;
```

Variáveis

Atribuição Composta

É muito comum acumularmos um valor dentro de uma variável (tanto com soma ou multiplicação). Veja um exemplo abaixo:

```
numero = numero + 5;
```

A instrução acima poderia ser simplificada para :

```
numero+=5;
```

Nos utilizamos acima o que chamamos de **atribuição composta**.

Variáveis

Atribuição Composta

Veja abaixo algumas das atribuições compostas que podemos utilizar :

Atribuição composta	Equivale à :
<variável> += <valor>	<variável> = <variável> + <valor>
<variável> -= <valor>	<variável> = <variável> - <valor>
<variável> *= <valor>	<variável> = <variável> * <valor>
<variável> /= <valor>	<variável> = <variável> / <valor>
<variável> %= <valor>	<variável> = <variável> % <valor>

Variáveis

Atribuição Composta (Exemplos)

numero += 5;

fatorial*=incremento;

valor/=3;

Variáveis

Operadores de sufixo/préfixo

Aprendemos que podemos simplificar algumas expressões utilizando a **atribuição composta** . Veja um exemplo:

```
numero = numero + 1;
```

Que pode ser simplificada por

```
numero+=1;
```


Variáveis

Operadores de sufixo/prefixo

Nós podemos utilizar o que chamados de operadores de sufixo ou prefixo. Para demonstrar o uso , poderíamos simplificar mais ainda o exemplo que demonstramos utilizando por exemplo o **operador de sufixo**, veja:

numero ++; (Operador de sufixo)

Isso equivale a :

numero+=1;

Que também equivale à:

numero = numero + 1;

Variáveis

Operadores de sufixo/prefixo

Nós podemos utilizar o que chamados de operadores de sufixo ou prefixo. Para demonstrar o uso , poderíamos simplificar mais ainda o exemplo que demonstramos para:

numero ++; (Operador de sufixo)

Isso equivale a :

numero+=1;

Que também equivale à:

numero = numero + 1;

Variáveis

Operadores de sufixo/prefixo

Agora vamos utilizar o operador de prefixo para o mesmo exemplo:

```
++numero; (Operador de prefixo)
```

Isso equivale a :

```
numero+=1;
```

Que também equivale à:

```
numero = numero + 1;
```

Variáveis

Operadores de sufixo/prefixo (Tabela)

Op. de sufixo	Op. prefixo	Equivale à	Que também equivale à
<variável>++;	++<variável>;	<variável>+=1;	<variável> = <variável> + 1;
<variável>--;	--<variável>;	<variável>-=1;	<variável> = <variável> - 1;

Variáveis

Operadores de sufixo/prefixo (Diferenças)

Ambos os operadores fazem a mesma coisa mas, qual a diferença entre eles ? Irei demonstrar isso agora. Podemos realizar uma operação de sufixo/prefixo durante uma atribuição a uma variável. Veja:

```
var valor: int = 12;
```

```
soma = valor++;
```

O comando acima equivale à :

```
soma = valor;
```

```
valor+=1;
```

No final soma = 12 e valor = 13;

Variáveis

Operadores de sufixo/prefixo (Diferenças)

Agora vamos demonstrar o uso do operador de prefixo. Veja

```
var valor: int = 12;
```

```
soma = ++valor;
```

O comando acima equivale à :

```
valor+=1;
```

```
soma = valor;
```

No final soma = 13 e valor = 13;