

Exibindo texto com printf

O método **System.out.printf** (f significa “formato”) exibe os dados *formatados*. A figura abaixo utiliza esse método para gerar a saída em duas linhas das strings “Welcome to” e “Java Programming!”.

As linhas 7 e 8:

```
System.out.printf("%s\n%s\n",  
    "Welcome to", "Java Programming!");
```

chamam o método `System.out.printf` para exibir a saída do programa. A chamada de método especifica três argumentos. Quando um método exige vários argumentos, estes são colocados em uma *lista separada por vírgulas*.

As linhas 7 e 8 representam somente *uma* instrução. O Java permite que instruções grandes sejam divididas em muitas linhas.

A linha 8 foi indentada para indicar que ela é uma *continuação* da linha 7.

Arquivo fonte

```
1 // Arquivo: Welcome4.java  
2 // Exibindo várias linhas com o método System.out.printf.  
3  
4 public class Welcome4 {  
5     // método main inicia a execução do aplicativo Java  
6     public static void main(String[] args) {  
7         System.out.printf("%s\n%s\n",  
8             "Welcome to", "Java Programming!");  
9     } // fim do método main  
10 } // fim da classe Welcome4
```

Saída do programa

```
Welcome to  
Java Programming!
```

As linhas 7 e 8 também poderiam ser escritas da seguinte forma:

```
7     System.out.printf("%s\n%s\n",  
8         "Welcome to", "Java Programming!");
```

Com `printf`, no lugar de `\n`, podemos usar o `%n` e teremos o mesmo efeito.

Especificadores de formato para uso com `printf`:

%d	int
%f	float / double
%b	boolean
%c	char

Assim como em Python, podemos usar `%.Nf`, onde N indica o número de casas decimais que serão apresentadas. Por exemplo, para apresentar um número com 2 casas decimais, devemos escrever `%.2f`.

```
double x = 58.3367;  
System.out.printf("%.2f", x); //Será apresentado o valor 58.34
```

Operadores Aritméticos

Os **operadores aritméticos** da linguagem Java estão resumidos na tabela abaixo. Esses operadores são chamados *binários*, porque cada um funciona em *dois* operandos. Por exemplo, a expressão $f + 7$ contém o operador binário $+$ e os dois operandos f e 7 .

A **divisão de inteiros** produz um quociente inteiro. Por exemplo, a expressão $7 / 4$ é avaliada como 1 e a expressão $17 / 5$, como 3 . Qualquer parte fracionária na divisão de inteiros é simplesmente *truncada* (isto é, *descartada*) — não são feitos *arredondamentos*. O Java fornece o operador de resto, $\%$, que oferece o resto depois da divisão. A expressão $x \% y$ produz o que sobra depois que x é dividido por y . Portanto, $7 \% 4$ produz 3 , e $17 \% 5$ produz 2 . Esse operador é normalmente utilizado com operandos inteiros, mas também pode ser usado com outros tipos de dados. Vamos examinar vários aplicativos interessantes do operador de resto, como determinar se um número é um múltiplo de outro.

Operação Java	Operador	Expressão Java
Adição	$+$	$f + 7$
Subtração	$-$	$p - c$
Multiplificação	$*$	$b * m$
Divisão	$/$	x / y
Resto	$\%$	$r \% s$

Tabela com operadores aritméticos.

Operadores de igualdade e operadores relacionais

Uma **condição** é uma expressão que pode ser **true** ou **false**. Este tópico apresenta a **instrução de seleção if** do Java que permite a um programa tomar uma **decisão** com base no valor de uma condição. Por exemplo, a condição “nota é maior ou igual a 60” determina se um aluno passou em um teste. Se a condição em uma estrutura if for *verdadeira*, o corpo da estrutura if é executado. Se a condição for *falsa*, o corpo não é executado.

As condições nas instruções if podem ser formadas utilizando os **operadores de igualdade** ($==$ e $!=$) e os **operadores relacionais** ($>$, $<$, $>=$ e $<=$), resumidos na tabela abaixo. Ambos os operadores de igualdade têm o mesmo nível de precedência, que é *mais baixo* do que o dos relacionais.

Os operadores de igualdade são associados *da esquerda para a direita*. Todos os operadores relacionais têm o mesmo nível de precedência e são associados *da esquerda para a direita*.

Operador algébrico	Operador de igualdade ou relacional Java	Exemplo de condição em Java	Significado da condição em Java
Operadores de igualdade			
	$==$	$x == y$	x é igual a y
	$!=$	$x != y$	x é não igual a y
Operadores relacionais			
	$>$	$x > y$	x é maior que y
	$<$	$x < y$	x é menor que y
	$>=$	$x >= y$	x é maior que ou igual a y
	$<=$	$x <= y$	x é menor que ou igual a y