# JAVA cheat sheet

#### Hello World

```
public class HelloWorld
{
   public static void main(String[] args)
   {
      System.out.println("Hello, World");
   }
}
```

#### Comentários

```
/* Comentário de
  várias
  linhas
*/
// Comentário de linha única
```

## Tipos de variáveis

Tipo	Faixa de valores	Tamanho (bits)
byte	-128 a 127	8
char	0 a 65535	16
sort	-32768 a 32767	16
int	-2147483648 a 2147483647	32
long	-9223372036854775808 a	64
	9223372036854775807	
float	±3.40282347E38	32
double	±1.79769313486231570E308	64
boolean	true ou false	-

Fonte: <a href="mailto:cs-fundamentals.com">cs-fundamentals.com</a>

## Identificadores

Permitidos letras, dígitos, \_ e \$

### CamelCase

Convenção utilizada na criação de identificadores compostos por mais de uma palavra, onde cada palavra é iniciada com letra maiúscula.

Java utiliza 2 variações da CamelCase

- (2) lowerCamelCase: atributos/variáveis, métodos

#### Constantes

final tipoConstante = valor;

## Comandos para impressão

```
System.out.print(...);
System.out.println(...);
System.out.printf(...);
```

## Caracteres de escape

```
\n nova linha
\t tabulação
\\ barra invertida
\" aspas
\b backspace
```

#### Entrada de dados

```
import java.util.Scanner;

public static void main(String[] args) {
   Scanner ent = new Scanner(System.in);

   String palavra = ent.next();
   String frase = ent.nextLine();
   int numInt = ent.nextInt();
   byte numByte = ent.nextByte();
   long numLong = ent.nextLong();
   float numFloat = ent.nextFloat();
   double numDouble = ent.nextDouble();
}
```

## Operadores Básicos

Fonte: <a href="mailto:tutorialspoint.com">tutorialspoint.com</a>

#### **OPERADORES ARITMÉTICOS**

Operador	Operação	Exemplo
+	Adição	a = 2 + 5
-	Subtração	a = x - 4
*	Multiplicação	a = x * y
/	Divisão	a = 9 / 2 (4) b = 9.0 / 2 (4.5) c = 9 / 2.0 (4.5) d = 9.0 / 2.0 (4.5)
%	Resto	a = 9 % 2 (1)
++	Incremento	a++ (pós-incremento) ++a (pré-incremento)
	Decremento	a (pós-decremento) a (pré-decremento)

# JAVA cheat sheet

#### **OPERADORES DE ATRIBUIÇÃO**

Op.	Descrição	Exemplo
=	Atribuição	a = 5
+=	Atrib/Adição	a += 5
		a = a + 5
-=	Atrib/Subtração	a -= 5
		a = a - 5
*=	Atrib/Multiplicação	a *= 3
		a = a * 2
/=	Atrib/Divisão	a /= 2
		a = a / 2
%=	Atrib/Resto	a %= 2
		a = a % 2

#### **OPERADORES RELACIONAIS**

Ор.	Descrição	Exemplo
==	Igualdade	a == b
! =	Desiguladade	a != c
<	Menor que	a < 5
>	Maior que	a > 3
<=	Menor ou igual	a <= 2
>=	Maior ou igual	a >= 0

## Algumas funções matemáticas

Op.	Descrição	Exemplo
Math.abs()	Valor absoluto	<pre>a = Math.abs(- 5)</pre>
Math.pow	Potenciação	a = Math.pow(3,2)
Math.sqrt	Raiz Quadrada	a = Math.sqrt(9)
Math.PI	Valor de pi	C = 2 * Math.PI * R

## Estruturas de Controle

```
if (condição)
   bloco de comandos<sub>1</sub>;
   bloco de comandos<sub>2</sub>;
switch(expressão) {
   case opc1:
      bloco de comandos;
      break;
   case opc2:
      bloco de comandos;
      break;
   default:
      bloco de comandos;
for(inicialização; condição; atualização)
   bloco de comandos;
while (condição)
   bloco de comandos;
do {
  bloco de comandos;
} while (condição);
```

## Arranjos (Vetores e Matrizes)

### **ARRANJOS UNIDIMENSIONAIS (VETORES)**

```
tipo nomeVetor[];
Inicialização
nomeVetor = new tipo[tamanho];
```

Declaração

## Declaração/inicialização

tipo nomeVetor[] = new tipo[tamanho];

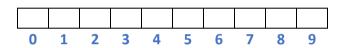
#### Sintaxe alternativa

tipo[] nomeVetor = new tipo[tamanho];

#### Exemplo

```
int[] VET = new int[10];
int VET[] = new int[10];
```

VET.length => retorna a quantidade de colunas



#### **ARRANJOS BIDIMENSIONAIS (MATRIZES)**

#### Declaração

tipo[][]Matriz = new tipo[qtLinhas][qtColunas];

#### Exemplo

```
int[][] MATRIZ = new int[4][5];
int[] MATRIZ[] = new int[4][5];
int MATRIZ[][] = new int[4][5];
```

MATRIZ.Length => retorna a quantidade de Linhas

MATRIZ[0].length => retorna a quantidade
de colunas

