

# ANOTAÇÕES DE ESTUDO PESSOAL

Hello World!

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

## Tipos de Dados em Java

Uma **variável** é uma área de memória, associada a um nome, que pode armazenar valores de um determinado tipo.

Estrutura da declaração de variáveis em Java:

```
<tipo> nome;  
  
// ou  
  
<tipo> nome = valorInicial;
```

Tipos e exemplo de declaração:

```
byte < short < int < long  
  
byte b = 8; // 8 bits - 1 byte  
short ss = 16; // 16 bits - 2 bytes  
int x = 10; // 32 bits - 4 bytes  
long l = 20; // 64 bits - 8 bytes  
  
a diferença está na precisão em casas decimais  
float f = 12.9f; // precisão simples (6 casas)  
double d = 12.5; // precisão dupla (10 casas)
```

ambos são para caracteres

```
char c = 'C'; // um caractere

String ola = "Hello World!"; // uma sequência de caracteres

boolean bool = true; // true or false
```

OBS:

1. O tipo int serve para guardar números inteiros, positivos e negativos. Em geral, na dúvida entre os tipos de inteiros use int.
2. O double deve ser utilizado quando o resultado de uma operação em número decimal necessitar ser mais preciso, a exemplo de cálculo em placas de vídeo mais atuais.
3. Float reserva menos memória, então para cálculos simples em casas decimais ele é o tipo recomendado.
4. String utiliza aspas duplas, char utiliza aspas simples.
5. O tipo char é útil em casos como "Pressione alguma tecla para para continuar..." ou "Pressione S para sim ou N para não".
6. Boolean é o tipo de dado mais simples encontrado em Java, retornando apenas True or False. Sendo utilizado em retorno de operações como "a <= b", "x > y", etc.

## Métodos de Exibição (print, printf e println)

### print

print, basicamente, imprime tudo na mesma linha

```
public class ExibicaoDeProcessamento {

    public static void main(String[] args) {

        System.out.print("Oi, meu nome não é Johnny");

        System.out.print("Minha idade é 26");

    }

}

-----

Oi, meu nome não é JohnnyMinha idade é 26
```

## println

println, basicamente, cria uma quebra de linha

```
public class ExibicaoDeProcessamento {  
    public static void main(String[] args) {  
        System.out.println("Oi, meu nome não é Johnny");  
        System.out.print("Minha idade é 26");  
    }  
}
```

```
-----  
Oi, meu nome não é Johnny  
Minha idade é 26
```

## printf

método de impressão com formatação de tipos

```
public class ExibicaoDeProcessamento {  
    public static void main(String[] args) {  
        System.out.printf("%s\n", "Oi, meu nome não é Johnny");  
        System.out.printf("%S\n", "Oi, meu nome não é Johnny");  
  
        // Strings  
  
        Oi, meu nome não é Johnny  
  
        OI, MEU NOME NÃO É JOHNNY
```

```
-----  
  
        System.out.printf("%c\n", 'a');  
  
        System.out.printf("%C\n", 'a');  
  
        // char  
  
        a  
  
        A  
  
-----
```

```

        System.out.printf("%d\n", 26);

        System.out.printf("%f\n", 26.5);

        // int e float

        26

        26,500000

-----

        System.out.printf("%b\n", 1 < 2);

        System.out.printf("%b\n", 1 > 2);

        // %b para Boolean formatado em caixa baixa

        true

        false

        System.out.printf("%B\n", 1 < 2);

        System.out.printf("%B\n", 1 > 2);

        // %B para Boolean formatado para caixa alta

        TRUE

        FALSE

-----

    }
}

```

## Lendo Teclado (Scanner)

```

// É necessário importar a classe Scanner primeiro
import java.util.Scanner;

public class LendoTecladoEmJava {

    public static void main(String[] args) {

```

```
Scanner texto = new Scanner(System.in);

System.out.print("Digite seu nome: ");

String name = texto.nextLine();

System.out.println("Qual a sua idade?");

int age = texto.nextInt();

System.out.printf("%s, você tem %d anos!", name, age);

texto.close();
}
}

-----

Digite seu nome: João da Silva
Qual a sua idade?
29
João da Silva, você tem 29 anos!
```

### exemplo com tipo char

```
Scanner texto = new Scanner(System.in);

System.out.println("Digite uma letra: ");

char c = texto.next().charAt(0);

System.out.printf("Você pressionou a tecla %c!", c);

texto.close();

-----

Digite uma letra:
p
Você pressionou a tecla p!
```

## Condicionais em Java

```
import java.util.Scanner;

public class CondicionaisEmJava {

    public static void main(String[] args) {

        Scanner opcao = new Scanner(System.in);

        System.out.println("Digite S para sim e N para não:");

        char c = opcao.next().charAt(0);

        if (c == 's') {

            System.out.println("Você optou por votar SIM!");

        }

        else if (c == 'n') {

            System.out.println("Você optou por votar NÃO!");

        } else {

            System.out.println("Digite uma opção válida!");

        }

        opcao.close();

    }

}

-----

Digite S para sim e N para não:
s
Você optou por votar SIM!
```

## Operações aritméticas em Java

```
import java.util.Scanner;

public class OperacoesMatematicasEmJava {
```

```

import java.util.Scanner;

public class OperacoesMatematicasEmJava {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.println("Digite o primeiro número: ");

        int n1 = input.nextInt();

        System.out.println("Digite o segundo número: ");

        int n2 = input.nextInt();

        int soma = n1 + n2;

        int multip = n1 * n2;

        int divisao = n1 / n2;

        int subtracao = n1 - n2;

        int resto = n1 % n2;

        System.out.println("A soma é igual a: " + soma);

        System.out.println("A multiplicação é igual a: " + multip);

        System.out.println("A divisão é igual a: " + divisao);

        System.out.println("A subtração é igual a: " + subtracao);

        System.out.println("O resto da divisão é igual a: " + resto);

        input.close();

    }

}

```

## Incremento e Decremento

1. **variavel++** adiciona 1 ao valor de x, armazena o resultado em x, e retorna o valor original

```
int variavel = 10;
```

```
System.out.println(variavel++); -> 10

System.out.println(variavel); -> 11
```

2. **++variavel** adiciona 1 ao valor de x, armazena o resultado em x, e retorna o novo valor incrementado

```
System.out.println(++variavel); -> 11

System.out.println(variavel); -> 11
```

3. **variavel--** - subtrai 1 do valor de x, armazena o resultado em x, e retorna o valor original

```
System.out.println(variavel--); -> 10

System.out.println(variavel); -> 9
```

4. **--variavel** subtrai 1 do valor de x, armazena o resultado em x, e retorna o novo valor decrementado

```
System.out.println(--variavel); -> 9

System.out.println(variavel); -> 9
```

## Estrutura de Repetição For

```
public class ForEmJava {

    public static void main(String[] args) {

        for (int c = 1; c <= 10; c++) {

            System.out.print(c + " ");

        }

    }

}
```

-----

1 2 3 4 5 6 7 8 9 10



## Estrutura de Repetição While

```
public class WhileEmJava {  
  
    public static void main(String[] args) {  
  
        int i = 0;  
  
        while (i < 10) {  
  
            System.out.print(i + " ");  
  
            i++;  
  
        }  
  
    }  
  
}
```

-----  
0 1 2 3 4 5 6 7 8 9

## Arrays em Java

Array é uma estrutura de dados que permite armazenar uma coleção sequencial de dados do mesmo tipo. O array, diferente de uma lista, não permite aumentar ou diminuir seu tamanho.

Declaração de um Array:

```
tipo De Dado[] = variavelReferencia;  
  
tipoDeDado variavelReferencia = new tipoDeDado[tamanho]
```

```
public class ArraysEmJava {  
  
    public static void main(String[] args) {  
  
        int[] valores = new int[10];  
  
  
        valores[0] = 1;  
  
        valores[1] = 2;
```

```
        valores[2] = 3;

        valores[3] = 4;

        valores[4] = 5;

        valores[5] = 6;

        valores[6] = 7;

        valores[7] = 8;

        valores[8] = 9;

        valores[9] = 10;

        for (int i: valores) {

            System.out.println(i);

        }

    }

}

-----

1
2
3
4
5
6
7
8
9
10
```

imprimindo o valor pela posição

```
System.out.println("O elemento 7 do Array é o número " + valores[6]);

-----

70
```

modificando o valor de uma posição no array

```
valores[6] = 700;

System.out.println("O elemento 7 do Array é o número " + valores[6]);

-----

700
```

imprimindo o valor pela posição

```
System.out.println("O elemento 7 do Array é o número " + valores[6]);

-----

70
```

ArrayList - uma classe útil para manipulação de listas em Java

- Como a classe ArrayList é um agrupamento dinâmico de objetos, isso quer dizer que podemos adicionar ou retirar elementos de, por exemplo, uma lista, sem que seja necessário criar uma nova, mantendo a original com um número diferente dos objetos ali contidos.

#### Array vs ArrayList

```
Array:          int[] valores = new int[10];

ArrayList: ArrayList<String> listaDeFrutas = new ArrayList<>();
```

Como utilizar ArrayList

```
import java.util.ArrayList;

public class ArrayListJava {

    public static void main(String[] args) {

        ArrayList<String> listaDeFrutas = new ArrayList<>();

        listaDeFrutas.add("Maça");

        listaDeFrutas.add("Abacaxi");
```

```
        listaDeFrutas.add("Mamão");

        for (String i : listaDeFrutas) {

            System.out.println(i);

        }

    }

}
```