

# Introdução ao Java

//Sintaxe, variáveis, tipos de dados e operadores

IT BOARDING

**BOOTCAMP**





# Índice



**01**

O que é Java? Sintaxe da linguagem

**03**

Variáveis + Tipos de dados

**02**

Sintaxe da linguagem

**04**

Operadores

IT BOARDING

**BOOTCAMP**

# O que é Java?

IT BOARDING

**BOOTCAMP**



# Java



**Java é uma linguagem de programação** e tecnologia pensada para o desenvolvimento de programas de grande porte, que são altamente escaláveis, que tem grande integração com outras tecnologias e sobretudo muito robustas. Se destaca por ser uma linguagem fortemente tipada.



Nasceu com o objetivo de ser uma linguagem de programação de estrutura simples que pode ser executada em diversos sistemas operacionais.





# Versões do Java

Java tem duas edições principais:

- **Java Standard Edition (Java SE):** Era inicialmente conhecido como J2SE, mas a partir da versão 5.0 foi padronizado apenas como Java SE. É o ambiente Java básico que permite criar programas padrão com as ferramentas e coleções básicas já incorporadas à linguagem.
- **Java Enterprise Edition (Java EE):** É usado principalmente para o desenvolvimento de aplicações de grande porte ou de negócios. Considera a web como o ambiente principal, utilizando diferentes APIs como JDBC, JPA, servlets, JSP, EJB (Enterprise Java Beans), etc.



# O que é necessário para programar em Java?



Para poder programar em Java, o dispositivo com o qual estamos trabalhando deve ter dois acessórios básicos:

- **JRE (Java Runtime Environment):** É um conjunto de utilitários que o Java disponibiliza para a execução de aplicações desenvolvidas nesta linguagem. O JRE é geralmente incluído na grande maioria dos sistemas operacionais e até mesmo como um complemento para alguns ambientes de desenvolvimento (IDEs).
- **JDK (Java Development KIT):** É um software que fornece as ferramentas necessárias para desenvolver aplicativos ou elementos de software usando Java. Possui versões diferentes, porém as duas consideradas mais estáveis são a 8 e a 11.



# Sintaxe

IT BOARDING

**BOOTCAMP**





# Sintaxe

**Java** se destaca por ser uma linguagem **fortemente tipada**. O que significa isto? O que é **muito exigente** tanto com sua **sintaxe** quanto com seus **tipos de dados**.

Entre alguns desses requisitos, temos:

- É **Case sensitive** (diferencia maiúsculas de minúsculas)
- Cada linha de código deve terminar com **;** (**ponto e vírgula**)
- O início e o fim de cada estrutura, bem como os blocos de código são delimitados por **{}** (**chaves**)
- Os parâmetros são passados entre **()** (**parênteses**)



# Sintaxe de exemplo



```
package com.bootcamp.Exercicio1;
```

```
public class Main {
```

Chave  
de  
início

```
    public static void main(String[] args) {
```

```
        System.out.println("Olá Mundo");
```

Linha que  
termina  
com “,”

```
    }
```

Chave  
de fim

```
}
```

# Variáveis & Tipos de Dados

IT BOARDING

**BOOTCAMP**





# Variáveis



Uma **variável** é uma posição na memória que é alocada ou reservada para poder armazenar dados dentro dela.



Como na **matemática**, é um valor que pode **mudar** em um determinado período de tempo.



Em **Java**, para usar uma variável ou atribuir determinados valores a ela, ela deve ter sido **declarada** de antemão. Para declarar variáveis, usamos **Tipos de dados**.



# Tipos de Dados

Os tipos de dados definem o que pode ser armazenado em uma variável, bem como os limites de tamanho a serem considerados para isso. Em Java, existem **tipos de dados primitivos**, bem como classes que agem como se fossem tipos de dados.

## Numéricos

- **short** (inteiro, 2 bytes)
- **int** (inteiro, 4 bytes)
- **long** (inteiro, 8 bytes)
- **float** (flutuante, 4 bytes)
- **double** (flutuante, 8 bytes)

## Caracteres

- **char** (caracter, 2 bytes)
- **String** (cadeia de caracteres, não é primitiva, é **uma classe**)

## Otros

- **boolean** (lógico, 2 bytes)
- **void** (tipo de dados "nulo", para funções que não retornam valor)



# Como declarar uma variável?

## Nome da variável

Representa o nome que terá a variável em questão. Sugere-se que seja um nome representativo do que armazenará.

Java

```
tipo de dado nome_da_variavel;
```

## Tipo de Dado

Indica o tipo de dado que conterá a variável a ser declarada.



## Exemplo de declaração

```
{  
    int numero;  
    boolean concluido;  
    double salario;  
    String nome;  
    char letra;  
}
```



## Regras para nomes de variáveis



Não pode começar com um número. Deve começar com uma carta. Como boa prática, sugere-se uma letra minúscula.



Você não pode usar caracteres especiais (% , \$ , \* , etc) ou caracteres reservados para outras operações.



Você não pode incluir espaços em branco.



Não pode ser chamado da mesma forma que palavras reservadas no idioma (como: int, for, if, else, private, entre outros).

# Operadores

IT BOARDING

**BOOTCAMP**







# Operadores aritméticos, lógicos e relacionais

Aritméticos	Relacionais	Lógicos
<b>+</b> Soma	<b>==</b> Igual	<b>&amp;&amp;</b> (AND / E)
<b>-</b> Subtração	<b>&gt;,&lt;</b> Maior, Menor	<b>  </b> (OR / OU)
<b>*</b> Multiplicação	<b>&gt;=,&lt;=</b> Maior = , Menor =	<b>!</b> (NOT / Negação)
<b>/</b> Divisão	<b>!=</b> Diferente	
	<b>=</b> Atribuição	



# Exemplos

## //Aritméticos

```
int soma = 3 + 5;  
int subtr = 15 - 4;  
int multi = 4 * 6;  
int divi = 9 / 3;
```

## //Relacionais

```
boolean maior = 5 > 3;  
boolean menor = 1 < 9;  
boolean igual = 4 == 4;
```

## //Lógicos

```
boolean sim = !false;  
boolean nao = !true;  
boolean and = true && true;  
boolean or = true || false;
```



**Nota:** Deve-se notar que para comparar duas Strings, o comparador "==" não é implementado, mas um método especial chamado equals é usado.  
**Por exemplo:** String prova = "olá";  
boolean igual = prova.equals("olá");



# Operações de leitura e gravação

```
//Escrita  
System.out.println("olá");
```

*//dependendo do tipo de dados  
que queremos ler, usamos  
diferentes funções next*

```
//Leitura  
Scanner teclado = new Scanner(System.in);  
  
int num = teclado.nextInt();  
String nome = teclado.next();  
double media = teclado.nextDouble();
```



# Obrigado

IT BOARDING

**BOOTCAMP**

