

# Java<sup>TM</sup>



## *A Arte de Programar em Java: Fundamentos para Iniciantes*

# 01

## Declarando variáveis

Em Java, usamos tipos como int, double, boolean, entre outros, para armazenar dados.

```
int idade = 25;  
double preco = 19.99;  
boolean ativo = true;
```

# 02

## Estruturas condicionais

Utilizamos if, else if e else para tomar decisões no código.

```
int idade = 18;  
if (idade >= 18) {  
    System.out.println("Maior de idade");  
} else {  
    System.out.println("Menor de idade");  
}
```

# 03

## Laços e Repetição

Usamos for, while e do-while para repetição de código.

```
for (int i = 0; i < 5; i++) {  
    System.out.println("Repetição " + i);  
}
```



# 04

## Trabalhando com arrays

Arrays armazenam uma lista de valores do mesmo tipo.

```
int[] numeros = {1, 2, 3, 4, 5};  
System.out.println(numeros[0]); // Imprime 1
```

# 05

## Criando e chamando métodos

Métodos ajudam a organizar o código e evitar repetição.

```
public static void saudacao() {  
    System.out.println("Olá, seja bem-vindo!");  
}  
  
saudacao();
```

# 06

## Orientação a objetos

Criamos classes e objetos para estruturar melhor o código.

```
class Pessoa {  
    String nome;  
    int idade;  
  
    void apresentar() {  
        System.out.println("Meu nome é " + nome + " e tenho " + idade + " anos");  
    }  
}  
  
Pessoa p = new Pessoa();  
p.nome = "João";  
p.idade = 30;  
p.apresentar();
```

# 07

## Manipulando Listas com ArrayList

A classe ArrayList é mais flexível que arrays comuns.

```
import java.util.ArrayList;

ArrayList<String> nomes = new ArrayList<>();
nomes.add("Ana");
nomes.add("Carlos");
nomes.remove("Ana");
System.out.println(nomes);
```



# 08

## Estruturas Try-Catch (Tratamento de Exceções)

Usamos try-catch para tratar erros no código e evitar que o programa quebre.

```
try {  
    int resultado = 10 / 0;  
} catch (ArithmeticException e) {  
    System.out.println("Erro: divisão por zero!");  
}
```

# 09

## Entrada de Dados com Scanner

Podemos capturar dados digitados pelo usuário com Scanner.

```
import java.util.Scanner;

Scanner scanner = new Scanner(System.in);
System.out.print("Digite seu nome: ");
String nome = scanner.nextLine();
System.out.println("Olá, " + nome);
scanner.close();
```

# 10

## Trabalhando com arquivos

Podemos ler e escrever arquivos usando File e FileWriter.

```
import java.io.FileWriter;
import java.io.IOException;

try {
    FileWriter writer = new FileWriter("arquivo.txt");
    writer.write("Conteúdo do arquivo");
    writer.close();
} catch (IOException e) {
    System.out.println("Erro ao escrever no arquivo");
}
```