

# Módulo 5 – Funções em JavaScript

## Objetivo do módulo

Aprender a criar funções para organizar o código em blocos reutilizáveis e facilitar a manutenção e a leitura do programa.

## 1. O que é uma função?

Uma função é um **bloco de código que realiza uma tarefa específica**.

Você pode "chamar" a função várias vezes sem precisar reescrever o código.

Exemplo simples:

```
function saudacao() {  
    console.log("Olá! Bem-vindo ao minicurso de JavaScript!");  
}
```

```
saudacao(); // chama a função
```

## 2. Funções com parâmetros

Parâmetros são valores que você passa para a função, tornando-a mais flexível.

Exemplo:

```
function saudacao(nome) {  
    console.log("Olá, " + nome + "!");  
}
```

```
saudacao("Ana");  
saudacao("Carlos");
```

### 3. Funções com retorno

Funções podem **retornar valores** para serem usados em outras partes do código.

Exemplo:

```
function soma(a, b) {  
    return a + b;  
}  
  
let resultado = soma(5, 3);  
console.log("Resultado: " + resultado); // Resultado: 8
```

### 4. Funções anônimas

Funções podem ser armazenadas em variáveis sem ter nome, chamadas de **funções anônimas**.

Exemplo:

```
const multiplicar = function(x, y) {  
    return x * y;  
}  
  
console.log(multiplicar(4, 5)); // 20
```

### 5. Arrow Functions

Sintaxe curta para funções anônimas, muito usada em JavaScript moderno.

Exemplo:

```
const dividir = (a, b) => a / b;  
  
console.log(dividir(10, 2)); // 5
```

## 6. Boas práticas

- Dê nomes **claros e significativos** para suas funções.
- Cada função deve **fazer apenas uma coisa**, para facilitar a manutenção.
- Use parâmetros e retorno quando precisar reaproveitar a função com valores diferentes.

## 7. Próximos passos

No próximo módulo, você aprenderá sobre **arrays e objetos**, que permitem **armazenar e organizar vários dados** de forma eficiente.

Crie funções para automatizar tarefas repetitivas e teste passando diferentes valores para perceber como o código se adapta.