

# DOMINE JAVASCRIPT



**UMA JORNADA PROGRAMADA**

CARLA SILVEIRA

# 01

## CONHEÇA O JAVASCRIPT

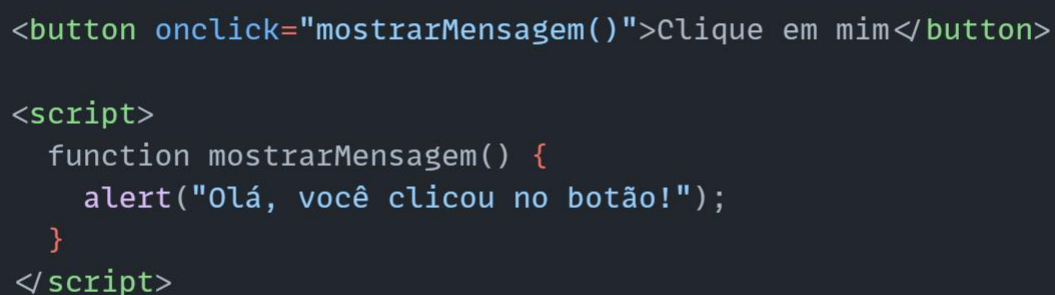
---

# 1. Fundamentos do JavaScript

## 1.1 O que é JavaScript?

JavaScript é uma linguagem de programação usada para criar páginas web interativas. Enquanto o HTML cria a estrutura e o CSS cuida da aparência, o JavaScript inclui interatividade nas páginas, permitindo coisas como animações, validação de formulários e até jogos online.

**Exemplo:** Aqui está um exemplo simples que exibe uma mensagem quando o usuário clica em um botão:



```
<button onclick="mostrarMensagem()">Clique em mim</button>

<script>
  function mostrarMensagem() {
    alert("Olá, você clicou no botão!");
  }
</script>
```

snappify.com

Neste exemplo, o código JavaScript faz com que uma mensagem de alerta apareça quando o usuário clica no botão. Isso é apenas o começo do que podemos fazer com JavaScript!

## 1.2 Por que aprender JavaScript?

JavaScript é uma das linguagens mais populares e poderosas do mundo. Ao aprender JavaScript, você pode criar sites mais dinâmicos, desenvolver aplicativos móveis, criar jogos e até trabalhar no *back-end* com *Node.js*. Além disso, a demanda por desenvolvedores JavaScript é alta no mercado de trabalho.

**Exemplo:** Imagine que você está criando um site para uma loja online. Com o JavaScript, você pode adicionar funcionalidades como estas:

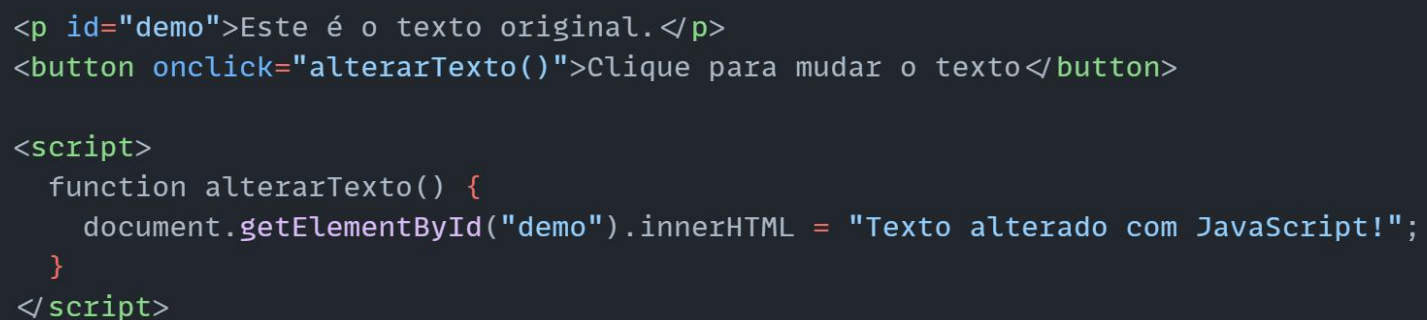
- ➔ Carrinho de compras interativo
- ➔ Calculadora de desconto
- ➔ Filtros dinâmicos para busca de produtos

JavaScript é, portanto, essencial para criar experiências web modernas e funcionais.

## 1.3 Como o JavaScript se encaixa no ambiente Web?

JavaScript é a linguagem que dá vida a páginas web. Trabalha com o **HTML** (estrutura) e **CSS** (design) para tornar a navegação interessante e interativa. Uma das ferramentas usadas pelo JavaScript é o **DOM** (**Document Object Model**), esse modelo ou mapa da estrutura de uma página web permite alterar o conteúdo e a estrutura da página dinamicamente no navegador.

**Exemplo:** Veja como JavaScript pode alterar o conteúdo de uma página HTML em tempo real. Este código altera o texto de um parágrafo quando o usuário clica em um botão:



```
<p id="demo">Este é o texto original.</p>
<button onclick="alterarTexto()">Clique para mudar o texto</button>

<script>
  function alterarTexto() {
    document.getElementById("demo").innerHTML = "Texto alterado com JavaScript!";
  }
</script>
```

snappify.com

Quando o botão é clicado, o JavaScript altera o conteúdo do parágrafo, mudando o texto que aparece na tela.

## 1.4 O que você vai aprender neste ebook?

Vamos explorar conceitos essenciais do JavaScript e aprender como usá-los na prática. Os tópicos deste ebook serão:

- ➔ **Variáveis e Tipos de Dados:** Como armazenar e manipular informações.
- ➔ **Funções:** Como criar blocos de código reutilizáveis.
- ➔ **Condicionais:** Como fazer decisões no código (se algo acontecer, faça isso).
- ➔ **Eventos:** Como responder a interações do usuário, por exemplo, cliques e digitação.

**Exemplo:** Aqui está um exemplo simples de como usamos variáveis e funções para armazenar e manipular dados:



```
let nome = "João"; // Armazenando um valor em uma variável
let idade = 25;    // Outra variável

function saudacao() {
  alert("Olá " + nome + ", você tem " + idade + " anos!");
}

saudacao(); // Chama a função e exibe a mensagem
```

snappify.com

Este código cria duas variáveis e uma função simples que exibe uma saudação personalizada com base nas informações armazenadas. É assim que você começa a construir a lógica de programação no JavaScript!



Com esse primeiro capítulo, você já teve um gostinho de como JavaScript pode ser poderoso e divertido de aprender. Agora, vamos avançar para conceitos mais profundos e começar a programar de verdade!

# 02

## **VARIÁVEIS E TIPOS DE DADOS**

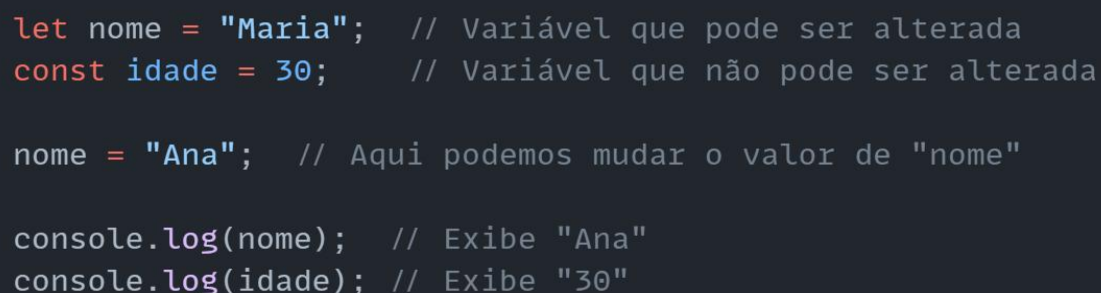
---

# 2. Uso de variáveis e tipos de dados

## 2.1 O que são variáveis?

Em JavaScript, variáveis são como "caixas" para armazenar informações, como números, textos ou até valores mais complexos. São essenciais para manipular dados no código. Você cria uma variável usando a palavra-chave ***let*** ou ***const***.

**Exemplo:**



```
let nome = "Maria"; // Variável que pode ser alterada
const idade = 30;   // Variável que não pode ser alterada

nome = "Ana"; // Aqui podemos mudar o valor de "nome"

console.log(nome); // Exibe "Ana"
console.log(idade); // Exibe "30"
```

snappify.com

Neste último caso, podemos observar que:

➔ ***let***: Cria uma variável que pode ser alterada.

➔ ***const***: Cria uma variável cujo valor não pode ser alterado após sua definição.

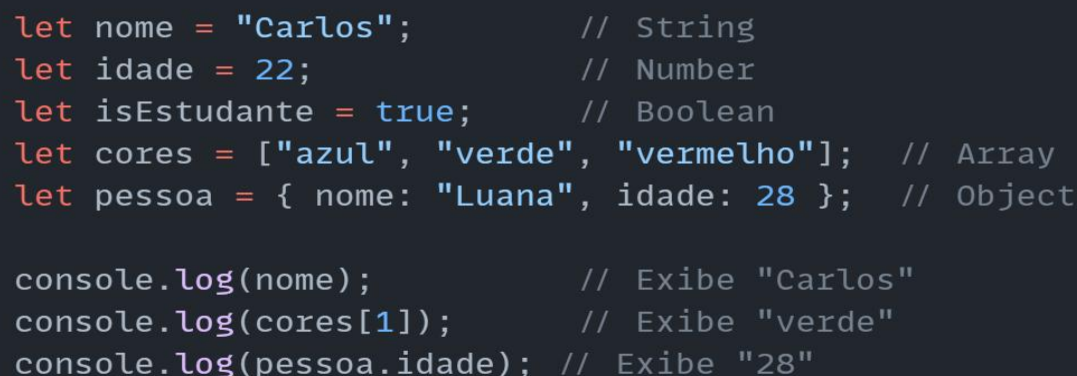


## 2.2 Tipos de dados em JavaScript

JavaScript tem vários tipos de dados, mas os mais comuns são:

- ➔ **String**: Sequência de caracteres (texto).
- ➔ **Number**: Números, tanto inteiros quanto decimais.
- ➔ **Boolean**: Representa verdadeiro (*true*) ou falso (*false*).
- ➔ **Array**: Lista de valores.
- ➔ **Object**: Coleção de valores em pares chave-valor.

Exemplo:



```
let nome = "Carlos";           // String
let idade = 22;                // Number
let isEstudante = true;        // Boolean
let cores = ["azul", "verde", "vermelho"]; // Array
let pessoa = { nome: "Luana", idade: 28 }; // Object

console.log(nome);             // Exibe "Carlos"
console.log(cores[1]);         // Exibe "verde"
console.log(pessoa.idade);     // Exibe "28"
```

snappify.com

## 2.3 Prática com variáveis e tipos de dados

Experimente alterar os valores das variáveis e veja como o comportamento do código acima muda. Tente criar suas próprias variáveis com diferentes tipos de dados e use `console.log()` para verificar o que está armazenado nelas.

# 03

## FUNÇÕES

---

# 3. Simplificação do código com funções

## 3.1 O que são funções?

Funções em JavaScript são blocos de código que você pode definir uma vez e executar quantas vezes quiser. Tornam o código mais organizado, reutilizável e fácil de manter.

Você cria uma função usando a palavra-chave *function*.

**Exemplo:**

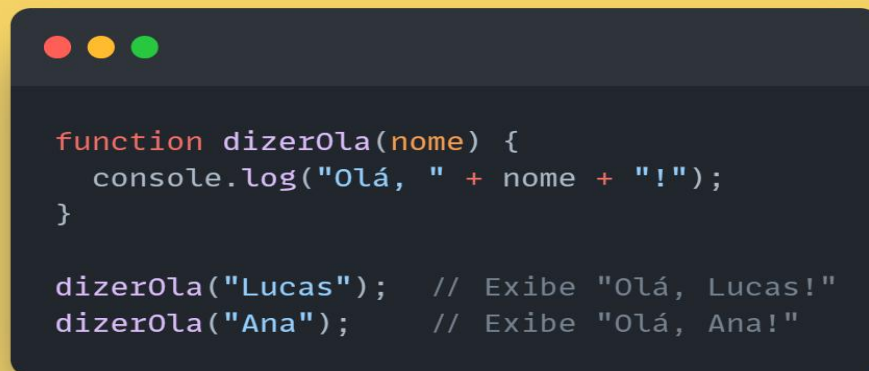


No exemplo acima, a função *saudacao* é criada e chamada duas vezes, exibindo a mesma mensagem cada vez.

## 3.2 Funções com parâmetros

Você também pode passar valores para a função por meio de parâmetros, o que torna a função mais flexível e dinâmica.

**Exemplo:**



```
function dizerOla(nome) {  
  console.log("Olá, " + nome + "!");  
}  
  
dizerOla("Lucas"); // Exibe "Olá, Lucas!"  
dizerOla("Ana");  // Exibe "Olá, Ana!"
```

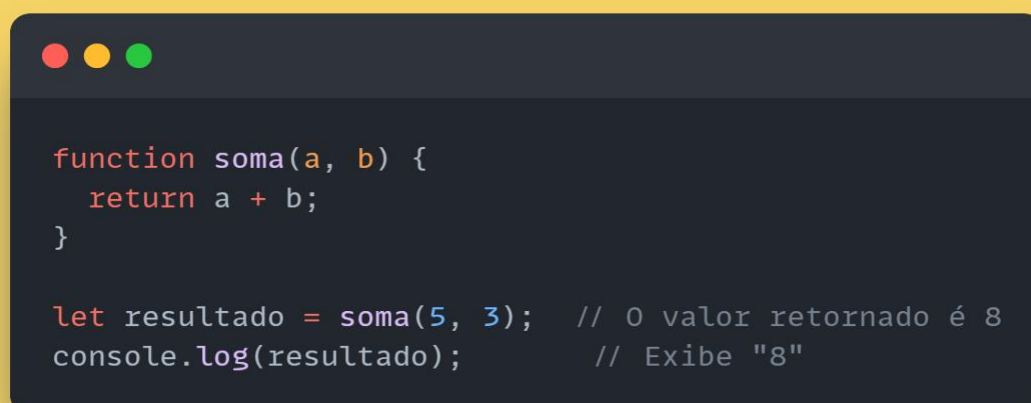
snappify.com

Aqui, a função *dizerOla* recebe um parâmetro chamado *nome*, e você pode passar diferentes valores para ela toda vez que a chamar no código.

### 3.3 Funções com retorno

Funções também podem retornar valores e permitem usar o resultado da execução da função em outros pontos do código.

**Exemplo:**



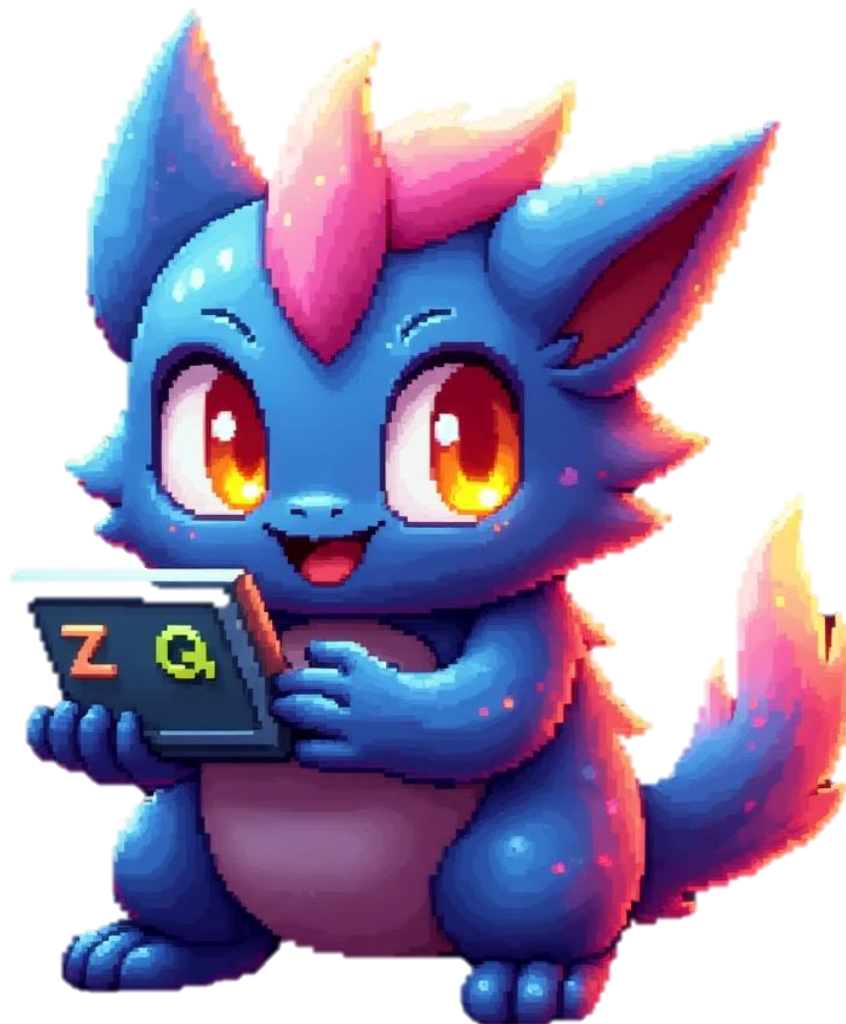
```
function soma(a, b) {  
  return a + b;  
}  
  
let resultado = soma(5, 3); // O valor retornado é 8  
console.log(resultado);    // Exibe "8"
```

snappify.com

No exemplo anterior, a função `soma` recebe dois parâmetros e retorna a soma deles, que é armazenada na variável `resultado`.

## 3.4 Prática com funções

Crie suas próprias funções com parâmetros e experimente retornar valores. Tente chamá-las com diferentes entradas e veja o que acontece. Por exemplo, faça uma função que calcula o dobro de um número e exibe o resultado.



Os capítulos 2 e 3 são essenciais para entender como armazenar dados e reutilizar código em JavaScript. No próximo capítulo, vamos explorar como tomar decisões no código com condicionais e criar lógicas mais complexas!

# AGRADECIMENTOS

---



# Agradeço seu interesse!



Este Ebook foi gerado por IA e diagramado por humano.

Este conteúdo foi gerado com fins didáticos e, embora tenha passado por validação cuidadosa humana, ainda pode conter erros gerados por uma IA.



<https://github.com/rosacarla>  
Continue aprendendo JavaScript!