

# Javascript

## O Poder dos Programadores



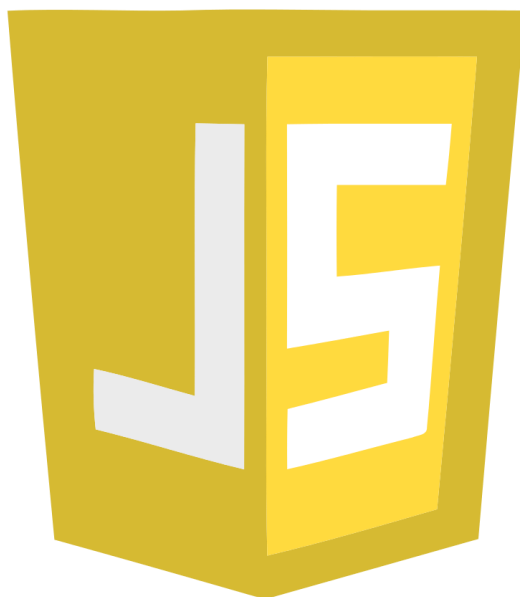


# Desbravando o mundo da Programação

Simplificando o JavaScript

Neste ebooks, exploraremos os conceitos fundamentais da linguagem JavaScript de uma maneira simples e acessível, fornecendo exemplos práticos para ajudá-lo a entender como essa poderosa linguagem está transformando a maneira como os desenvolvedores constroem aplicativos web dinâmicos.

## JavaScript



**Vamos Começar Nossa Jornada Pelo Mundo Do JS!**



# 01

# Intro ao JavaScript

---

JavaScript é uma linguagem de programação de alto nível amplamente utilizada para desenvolver aplicativos web dinâmicos e interativos. Neste primeiro tópico, vamos nos familiarizar com os conceitos básicos do JavaScript e entender como ele é executado nos navegadores.



# O Que é JavaScript?

JavaScript é uma linguagem de programação leve e versátil usada para adicionar interatividade e funcionalidade aos sites. Ela pode ser incorporada diretamente no código HTML de uma página web e executada no navegador do usuário.

## Por que Usar JavaScript?

Ele permite que os desenvolvedores criem sites dinâmicos, que respondem às ações dos usuários em tempo real, sem a necessidade de recarregar a página. Isso torna a experiência do usuário mais interativa e envolvente.

# Configurando o Ambiente de Desenvolvimento



Para começar a programar em JavaScript, você só precisa de um navegador web e um editor de texto, como o Visual Studio Code, Sublime Text ou Notepad++.

Escreva seu código JavaScript em um arquivo com a extensão ".js" e não se esqueça de "conecta-lo" no HTML

```
● ● ● código HTML com JavaScript
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de Conexão JavaScript</title>
  <!-- Conectando o arquivo JavaScript -->
  <script src="script.js"></script>
</head>
<body>

</body>

</html>
```

# 02

# Conceitos Básicos

---

conceitos fundamentais do JavaScript, incluindo variáveis, tipos de dados, operadores e estruturas de controle de fluxo.

# Variáveis e Tipos de Dados

as variáveis são usadas para armazenar dados temporários

## Exemplo Variáveis

```
var nome = "João";  
let idade = 30;  
const PI = 3.14;
```

Os tipos de dados incluem números, strings, booleanos, arrays e objetos.

## Exemplo Tipos de Dados

```
let numero = 10;  
let texto = "Olá, Mundo!";  
let booleano = true;  
let lista = [1, 2, 3];  
let objeto = { nome: "João", idade: 30 };
```

# Operadores e Expressões

JavaScript suporta uma variedade de operadores, incluindo aritméticos, de atribuição, de comparação e lógicos. Esses operadores são usados para realizar operações em variáveis e valores.

```
Operadores em JavaScript:

Aritméticos: +, -, *, /, %, ++, --
Atribuição: =, +=, -=, *=
Comparação: ==, ===, !=, !==, >, <, >=, <=
Lógicos: &&, ||, !
Ternário: ?:
```

```
Expressões em JavaScript

Combinações de operadores, valores e variáveis para realizar operações e obter resultados.

Exemplos :

5 + 3 * 2
(x > 10) && (y < 5)
z = (a === b) ? "igual" : "diferente"
```



# Estruturas de Controle

As estruturas de controle permitem controlar o fluxo de execução do código JavaScript. As principais estruturas são if, else, else if, for, while e switch

## Estruturas Condicionais:

- if/else: Permite executar um bloco de código se uma condição for verdadeira e outro bloco se for falsa.
- switch/case: Permite avaliar diversas condições e executar o código correspondente àquela que for verdadeira.

```
if/else

if (idade >= 18) {
  console.log("Você é maior de idade!");
} else {
  console.log("Você é menor de idade!");
}
```

```
switch/case

switch (fruta) {
  case "maçã":
    console.log("Você escolheu uma maçã!");
    break;
  case "banana":
    console.log("Você escolheu uma banana!");
    break;
  default:
    console.log("Você não escolheu nenhuma fruta das opções.");
}
```

# Estruturas de Controle

## 2. Estruturas de Repetição

- **for:** Permite executar um bloco de código repetidamente um número determinado de vezes
- **while:** Permite executar um bloco de código repetidamente enquanto uma condição for verdadeira.
- **do/while:** Similar ao while, mas o bloco de código é executado pelo menos uma vez antes da verificação da condição.

```
for (let i = 0; i < 5; i++) {  
  console.log("Número:", i);  
}
```

```
let i = 0;  
while (i < 10) {  
  console.log("Número:", i);  
  i++;  
}
```

```
let i = 0;  
do {  
  console.log("Número:", i);  
  i++;  
} while (i < 5);
```

# 03

## Funções em JavaScript

---

conceitos fundamentais do JavaScript, incluindo variáveis, tipos de dados, operadores e estruturas de controle de fluxo.

# 04

# Manipulação do DOM com JavaScript

---

conceitos fundamentais do JavaScript, incluindo variáveis, tipos de dados, operadores e estruturas de controle de fluxo.

# 05

# Lidando com Eventos

---

conceitos fundamentais do JavaScript, incluindo variáveis, tipos de dados, operadores e estruturas de controle de fluxo.



# 06

# **AJAX e Requisições Assíncronas**

---

conceitos fundamentais do JavaScript, incluindo variáveis, tipos de dados, operadores e estruturas de controle de fluxo.

# 07

## **Frameworks e Bibliotecas**

---

conceitos fundamentais do JavaScript, incluindo variáveis, tipos de dados, operadores e estruturas de controle de fluxo.

# Agradecimientos

---

# Obrigado Por Ler Até Aqui

Esse Ebook foi criado com ajuda da IA, e montado e editado por humano.

Criado por fins didáticos, pode conter erros gerados por uma IA pois não foi realizado uma análise cuidadosa humana

Esse documento é o Rascunho do Ebook em construção, que não está finalizado, ainda deve ser analisado e melhorado

Autora:

Gabriela Carrodegua Simões



<https://github.com/GabiCarrodegua>

# Modelo de pags

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.