

JAVASCRIPT DE BOLSO

TUDO OUE VOCE PRECISA SABER



LEONARDO LUUIAN

JavaScript de Bolso: Tudo o Que Você Precisa Saber

Bem-vindo ao "JavaScript de Bolso: Tudo o Que Você Precisa Saber". Este eBook é uma fonte completa e prática de conhecimento sobre JavaScript, ideal tanto para iniciantes quanto para desenvolvedores experientes.

Por que JavaScript?

JavaScript é a espinha dorsal da web moderna, trazendo interatividade e dinamismo aos sites.

Dominar JavaScript é essencial para qualquer desenvolvedor devido à crescente demanda por aplicações web robustas e interativas.

Este eBook é uma referência prática que você pode levar para qualquer lugar. Cada capítulo oferece explicações claras e exemplos práticos. Navegue pelos capítulos conforme suas necessidades, seja para aprender do zero ou aprofundar conhecimentos específicos.

01

Introdução ao JavaScript

Introdução ao JavaScript

JavaScript é a linguagem de programação que traz interatividade às páginas web. É usada tanto no front-end quanto no back-end. Com JavaScript, você pode criar elementos dinâmicos e responsivos em um site, como botões que mudam de cor ao serem clicados, formulários que validam dados em tempo real e muito mais.



JS DE BOLSO

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplo de JavaScript</title>
</head>
<body>
  <h1 id="titulo">Olá, Mundo!</h1>
  <script>
    document.getElementById('titulo').innerText = 'Olá, JavaScript!';
  </script>
</body>
</html>
```

Sugestão: Experimente alterar o texto no exemplo acima e veja a mudança no navegador.

02

Variáveis e tipos de dados

Variáveis e Tipos de Dados

As variáveis armazenam dados que podem ser usados e manipulados pelo programa. Os principais tipos de dados são string, number, boolean, null, undefined, object, e symbol. Compreender como declarar e usar variáveis é fundamental para criar qualquer programa em JavaScript.



JS DE BOLSO

```
let nome = 'Pedro'; // string
let idade = 30;    // number
let casado = true; // boolean

console.log(`Nome: ${nome}, Idade: ${idade}, Casado: ${casado}`);
```

Sugestão: Tente mudar os valores das variáveis e observe como o output se altera.

03

Operadores

Operadores

Os operadores são usados para realizar operações em variáveis e valores. Os principais tipos são aritméticos (como +, -, *, /), de atribuição (como =, +=), de comparação (como ==, !=, ===), e lógicos (como &&, ||, !). Eles são essenciais para realizar cálculos, tomar decisões e controlar o fluxo do programa.



JS DE BOLSO

```
let a = 10;  
let b = 5;  
  
console.log(a + b); // 15  
console.log(a > b); // true  
console.log(a === 10 && b === 5); // true
```

Sugestão: Explore outros operadores como -, *, /, !=, || e veja os resultados.

04

Estruturas de Controle

Estruturas de Controle

As estruturas de controle, como if, else, for, while e switch, ajudam a tomar decisões no código e a repetir blocos de código várias vezes. Elas são usadas para controlar o fluxo da aplicação, tornando possível a execução de diferentes blocos de código com base em diferentes condições.



JS DE BOLSO

```
let nota = 85;

if (nota >= 90) {
  console.log('A');
} else if (nota >= 80) {
  console.log('B');
} else {
  console.log('C');
}
```

Sugestão: Altere a nota para ver diferentes saídas.

05

Funções

Funções

Funções são blocos de código que executam uma tarefa específica e podem ser reutilizados. Elas ajudam a organizar o código, tornando-o mais modular e fácil de manter. Você pode definir funções com ou sem parâmetros e retornar valores com `return`.



JS DE BOLSO

```
function saudacao(nome) {  
    return `Olá, ${nome}!`;  
}  
  
console.log(saudacao('Maria'));
```

Sugestão: Crie funções para outras tarefas, como somar números ou verificar idade.

05

Objetos e Arrays

Objetos e Arrays

Objetos são coleções de propriedades, e arrays são listas ordenadas de valores. Ambos são usados para armazenar múltiplos valores em uma única variável, facilitando o gerenciamento de dados complexos.



JS DE BOLSO

```
let pessoa = {  
    nome: 'Ana',  
    idade: 25,  
    cidade: 'São Paulo'  
};  
  
let numeros = [1, 2, 3, 4, 5];  
  
console.log(pessoa.nome); // Ana  
console.log(numeros[2]); // 3
```

Sugestão: Crie funções para outras tarefas, como somar números ou verificar idade.

06

Manipulação do DOM

Manipulação do DOM

O DOM (Document Object Model) representa a estrutura da página web. JavaScript pode manipular o DOM para alterar o conteúdo e o estilo das páginas, permitindo criar experiências de usuário dinâmicas e interativas.



JS DE BOLSO

```
document.getElementById('titulo').style.color = 'blue';
```

Sugestão: Tente alterar outras propriedades de estilo, como backgroundColor ou fontSize.

07

Eventos

Eventos

Eventos são ações que ocorrem no navegador, como cliques ou teclas pressionadas. Podemos adicionar ouvintes de eventos para executar código quando um evento ocorre, tornando as páginas interativas.



JS DE BOLSO

```
document.getElementById('meuBotao').addEventListener('click', function() {
  alert('Botão clicado!');
});
```

Sugestão: Adicione eventos a outros elementos e veja como eles respondem.

08

Programação Assíncrona

Programação Assíncrona

A programação assíncrona permite que o código execute outras operações enquanto espera por tarefas demoradas, como chamadas de rede. Isso melhora a performance e a experiência do usuário.



JS DE BOLSO

```
setTimeout(() => {
  console.log('Essa mensagem aparece após 2 segundos.');
}, 2000);
```

Sugestão: Experimente diferentes tempos de espera e veja como isso afeta a execução do código.

09

Classes e POO

Classes e POO

Classes são moldes para criar objetos com propriedades e métodos. A POO facilita a organização e reutilização do código, permitindo a criação de estruturas mais complexas e reutilizáveis.

```
JS DE BOLSO

class Pessoa {
  constructor(nome, idade) {
    this.nome = nome;
    this.idade = idade;
  }

  saudacao() {
    return `Olá, meu nome é ${this.nome} e tenho ${this.idade} anos.`;
  }
}

let pessoa1 = new Pessoa('Carlos', 40);
console.log(pessoa1.saudacao());
```

Sugestão: Crie novas classes e métodos para diferentes objetos.

10

Módulos

Módulos

Módulos permitem a organização do código em arquivos separados, facilitando a manutenção e o reuso. Eles são uma forma de encapsular código relacionado, tornando-o mais fácil de gerenciar.



JS DE BOLSO

```
// arquivo1.js
export function saudacao(nome) {
  return `Olá, ${nome}!`;
}

// arquivo2.js
import { saudacao } from './arquivo1.js';

console.log(saudacao('Maria'));
```

Sugestão: Divida seu código em módulos menores e importe-os conforme necessário

1 1

Promises Async/Await

Promises e Async/Await

Promises e async/await simplificam o tratamento de operações assíncronas, tornando o código mais legível e gerenciável. Eles ajudam a lidar com tarefas que levam tempo para serem concluídas, como chamadas de API.



JS DE BOLSO

```
function esperar(ms) {
  return new Promise(resolve => setTimeout(resolve, ms));
}

async function executar() {
  console.log('Início');
  await esperar(2000);
  console.log('Fim após 2 segundos');
}

executar();
```

Sugestão: Experimente usar fetch para fazer uma chamada de API assíncrona.

12

Ferramentas e Bibliotecas

Ferramentas e Bibliotecas

Existem muitas ferramentas e bibliotecas que podem facilitar o desenvolvimento com JavaScript, como Node.js, React, Angular e Vue.js. Elas oferecem funcionalidades adicionais e estruturas para construir aplicações web robustas e escaláveis.



JS DE BOLSO

```
import React from 'react';
import ReactDOM from 'react-dom';

function App() {
  return <h1>Olá, React!</h1>;
}

ReactDOM.render(<App />, document.getElementById('root'));
```

Sugestão: Explore outras bibliotecas e frameworks para ver como podem ajudar em seus projetos.

Finalização

Obrigado por ler até aqui!

Projeto desenvolvido para o projeto do Santander
2024 com a DIO.

<https://github.com/felipeAguiarCode/prompts-recipe-to-create-a-ebook>