

Guia Completo de JavaScript

1. Introdução ao JavaScript

JavaScript é uma linguagem de programação interpretada, usada principalmente para criar interatividade em páginas web. Com ele, podemos manipular o conteúdo HTML e CSS, responder a ações do usuário e criar aplicações dinâmicas completas.

```
<script>
    document.write("Olá, mundo!");
</script>
```

2. Variáveis e Tipos de Dados

As variáveis são usadas para armazenar informações. Existem três formas de declarar variáveis: var, let e const. - var: escopo global ou de função. - let: escopo de bloco, recomendada para uso atual. - const: usada para valores constantes.

```
let nome = "Gabriel";
const idade = 25;
var ativo = true;

console.log("Nome:", nome);
console.log("Idade:", idade);
```

3. Operadores

Os operadores permitem realizar cálculos e comparações. Existem operadores aritméticos (+, -, *, /), de comparação (==, ===, !=, <, >) e lógicos (&&, ||, !).

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

4. Estruturas Condicionais

Permitem tomar decisões com base em condições.

```
let idade = 18;
if (idade >= 18) {
    console.log("Maior de idade");
} else {
    console.log("Menor de idade");
}
```

5. Laços de Repetição

São usados para executar blocos de código várias vezes: for, while e do...while.

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

6. Funções

Funções permitem reutilizar código. Podem receber parâmetros e retornar valores.

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

7. Arrays e Objetos

Arrays armazenam listas de valores; objetos armazenam pares de chave e valor.

```
let frutas = ["Maçã", "Banana", "Laranja"];  
let pessoa = { nome: "Ana", idade: 22, cidade: "São Paulo" };  
  
console.log(frutas[1]); // Banana  
console.log(pessoa.nome); // Ana
```

8. Manipulação do DOM

O DOM representa a estrutura HTML como uma árvore de objetos, permitindo acessar e modificar elementos da página.

```
document.getElementById("titulo").innerText = "Novo Título";  
document.querySelector(".texto").style.color = "blue";
```

9. Eventos

Eventos permitem reagir a ações do usuário, como cliques e digitação.

```
<button id="btn">Clique aqui</button>  
  
<script>  
document.getElementById("btn").addEventListener("click", function() {  
    alert("Botão clicado!");  
});  
</script>
```

10. JSON e LocalStorage

O JSON (JavaScript Object Notation) é usado para armazenar e transferir dados. O LocalStorage salva informações no navegador do usuário.

```
let usuario = { nome: "Lucas", idade: 28 };
localStorage.setItem("usuario", JSON.stringify(usuario));

let dados = JSON.parse(localStorage.getItem("usuario"));
console.log(dados.nome);
```

11. Boas Práticas

- Use 'let' e 'const' no lugar de 'var'.
- Nomeie variáveis de forma clara.
- Evite funções muito longas.
- Comente o código quando necessário.
- Teste e valide dados de entrada.

12. Exemplo Completo HTML + JS

Um exemplo de página simples com interatividade em JavaScript. Abra este código em um arquivo .html e veja o resultado no navegador.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<title>Exemplo JS</title>
</head>
<body>
<h1 id="titulo">Bem-vindo!</h1>
<button id="mudar">Mudar Título</button>

<script>
document.getElementById("mudar").addEventListener("click", () => {
    document.getElementById("titulo").innerText = "Título Alterado com JS!";
});
</script>
</body>
</html>
```