<codestudio/>

Pesquisar...

</> Estudo de JavaScript

<> O que é JavaScript?

JavaScript é a linguagem de programação responsável por adicionar **interatividade** às páginas web.

Com ele, você pode manipular o HTML, alterar estilos, reagir a eventos (como cliques) e muito mais.

🎯 Onde o JavaScript é usado?

- Alterar elementos do HTML com DOM
- Criar interações (cliques, animações, formulários)
- Validar dados antes de enviar para o backend
- Consumir APIs e mostrar dados dinamicamente
- Criar jogos, aplicativos web e até mobile

🧠 O que é Sintaxe no JavaScript?

Sintaxe é o conjunto de regras que definem como o código deve ser escrito para que o JavaScript entenda e execute corretamente. Assim como uma frase precisa de verbo e ponto final, o código também precisa seguir uma estrutura.

Um erro de sintaxe impede o código de funcionar — por exemplo: esquecer ponto e vírgula, chaves, escrever palavras-chave erradas ou não fechar parênteses.

✅ Sintaxe correta:

```
// Declaração de variável
let nome = "Adriano";

// Função simples
function saudacao() {
console.log("Olá, " + nome);
}

// Chamando a função
saudacao();
```

❌ Sintaxe incorreta:

```
let nome = "Adriano"          // ❌ Faltou ponto e vírgula

function saudacao( {          // ❌ Parêntese e chave juntos
console.log("Oi");
}                             // ❌ Chave sem fechar a função

saudacao(                     // ❌ Faltou fechar parêntese e ponto
```

✅ **Dica:** Erros de sintaxe aparecem no console do navegador (F12 → Aba "Console"). Sempre leia atentamente a linha do erro para corrigir mais rápido.

🎯 Eventos no JavaScript

Eventos são ações que acontecem na página — como clicar, digitar ou passar o mouse. Podemos usar JavaScript para "escutar" esses eventos e reagir a eles.

📌 Outros eventos comuns:

✅ **click** → quando o usuário clica em algo

✅ **input** → quando o usuário digita em um campo

✅ **mouseover** → quando o mouse passa por cima de um elemento

✅ **keydown** → quando alguma tecla do teclado é pressionada

💻 Exemplos:

📍 Evento: `click`

```
<button id="btn">Clique aqui</button>
                <p id="msg"></p>

                <script>
                document.getElementById("btn").addEventListener("click", () => {
                    document.getElementById("msg").textContent = "Você clicou no botão
                });
                </script>
```

📍 Evento: `input` (ao digitar)

```
<input type="text" id="nome" placeholder="Digite seu nome">
                <p id="resultado"></p>

                <script>
                document.getElementById("nome").addEventListener("input", (event) => {
                    document.getElementById("resultado").textContent = "Digitando: " +
                });
                </script>
```

📍 Evento: `mouseover` (passar o mouse)

```
<div id="box" style="width:200px; height:100px; background:lightgray;">
                Passe o mouse aqui
                </div>

                <script>
                document.getElementById("box").addEventListener("mouseover", () => {
                    document.getElementById("box").style.background = "lightgreen";
                });
                </script>
```

📍 Evento: `keydown` (tecla pressionada)

```
<p>Pressione qualquer tecla no teclado...</p>
                <p id="tecla"></p>

                <script>
                document.addEventListener("keydown", (event) => {
                    document.getElementById("tecla").textContent = "Você apertou: " +
                });
                </script>
```

🪟 Abrindo nova aba ou outra página com JavaScript

Com JavaScript, podemos abrir **outra página**, carregar **outro arquivo HTML**ou abrir um link em **nova aba** usando eventos como `onclick`.

✅ 1) Abrir link em nova aba (target="_blank")

```
<!-- HTML -->
                <button onclick="window.open('https://google.com', '_blank')">
                Abrir Google em nova aba
                </button>
```

✅ 2) Abrir outro HTML do seu site

```
<!-- HTML -->
                <button onclick="window.location.href = 'sobre.html'">
                Ir para página Sobre (mesma aba)
                </button>

                <button onclick="window.open('contato.html', '_blank')">
                Abrir página Contato em nova aba
                </button>
```

✅ 3) Usando JavaScript separadamente (sem onclick no HTML)

```
<button id="btnNovaAba">Abrir GitHub</button>

                <script>
                const botao = document.getElementById("btnNovaAba");

                botao.addEventListener("click", () => {
                    window.open("https://github.com", "_blank");
                });
                </script>
```

✅ 4) Enviando dados para outra página (com parâmetros)

```
// Exemplo: abre outra página e passa nome pela URL
                const nome = "Adriano";
                window.open(`perfil.html?nome=${nome}`, "_blank");

                // Na outra página (perfil.html), pegar valor assim:
                const params = new URLSearchParams(window.location.search);
                console.log(params.get("nome")); // "Adriano"
```

✅ **Dica:** Use `window.open()` para nova aba e`window.location.href` para trocar de página na mesma aba.

🔢 Variáveis e Operações Matemáticas no JavaScript

No JavaScript usamos **variáveis** para guardar informações. As palavras mais usadas são `let` e `const`.

✅ let vs const

```
// let → pode ser alterada depois
let nome = "Ana";
nome = "João"; // ✅ permitido

// const → NÃO pode ser alterada depois
const idade = 20;
idade = 25; // ❌ ERRO: não é possível mudar o valor de uma const
```

➕ Operações Matemáticas no JavaScript

JavaScript também permite realizar contas matemáticas de forma simples, como soma, subtração, multiplicação e divisão.

```
let a = 10;
let b = 5;

let soma = a + b;        // 15
let subtracao = a - b;   // 5
let multiplicacao = a * b; // 50
let divisao = a / b;     // 2

console.log("Soma:", soma);
console.log("Subtração:", subtracao);
console.log("Multiplicação:", multiplicacao);
console.log("Divisão:", divisao);
```

💻 Exemplo interativo com botão:

```
<h2>Calculadora Simples</h2>
<button onclick="calcular()">Clique para calcular</button>
<p id="resultado"></p>

<script>
function calcular() {
    let x = 8;
    let y = 2;
    const soma = x + y;
    const multiplica = x * y;
```

```
        document.getElementById("resultado").textContent =
        "Soma: " + soma + " | Multiplicação: " + multiplica;
    }
    </script>
```

✅ **Dica:** use `let` quando o valor pode mudar e `const` para valores fixos ou funções.

⚖️ Condições: if, else

Condicionais permitem que o JavaScript tome decisões. Exemplo: se o usuário for maior de idade, mostrar uma mensagem diferente.

✅ Exemplo:

```
const idade = 17;

if (idade >= 18) {
console.log("✅ Pode entrar!");
} else {
console.log("❌ Menor de idade.");
}
```

✨ Operador ternário (forma mais curta):

```
const idade = 20;
const mensagem = idade >= 18 ? "✅ Maior de idade" : "❌ Menor de id
console.log(mensagem);
```

✅ **Dica:** Use `if` para lógica mais complexa e `? :` (ternário) para decisões simples.

🔁 Loops (Repetições) no JavaScript

Loops servem para repetir uma ação várias vezes sem precisar escrever o mesmo código repetidamente. Eles são muito usados para percorrer listas, repetir cálculos, mostrar itens na tela, etc.

✅ Tipos mais comuns:

- 🏷️ **for** – repete algo um número definido de vezes
- 🏷️ **while** – repete enquanto uma condição for verdadeira
- 🏷️ **do...while** – executa pelo menos uma vez, depois testa

🏷️ **for...of** – percorre valores dentro de arrays

📌 Exemplo: for (o mais usado)

```
// Contar de 1 até 5
                for (let i = 1; i <= 5; i++) {
                console.log("Número:", i);
                }
```

📌 Exemplo: while

```
let contador = 1;

                while (contador <= 3) {
                console.log("Contando...", contador);
                contador++;
                }
```

📌 Exemplo: do...while

```
let numero = 1;

                do {
                console.log("Executa pelo menos uma vez:", numero);
                numero++;
                } while (numero <= 3);
```

📌 Exemplo: for...of (percorrer lista)

```
const frutas = ["🍎 Maçã", "🍌 Banana", "🍇 Uva"];

                for (const fruta of frutas) {
                console.log("Fruta:", fruta);
                }
```

✅ **Dica:** Loops + condicionais são a base de qualquer lógica de programação. Use `console.log()` para testar a execução passo a passo.

📦 Arrays (Listas)

Um **Array** é uma lista que pode guardar vários valores dentro de uma única variável. Cada item tem um número de posição chamado **índice** (começa do zero).

```
const frutas = ["Maçã", "Banana", "Uva"];

                // Acessando itens:
                console.log(frutas[0]); // "Maçã"
                console.log(frutas[2]); // "Uva"

                // Adicionando item:
                frutas.push("Laranja");

                // Removendo o último:
                frutas.pop();

                console.log(frutas); // ["Maçã", "Banana", "Uva"]
```

✅ **Dica:** Arrays são ótimos para listas de nomes, produtos, números etc.

⚙️ Funções em JavaScript

Funções são blocos de código que podemos "guardar" e usar quando quisermos. São úteis para evitar repetição de código.

✅ Exemplo básico:

```
function saudacao() {
                console.log("Olá! Seja bem-vindo!");
                }

                saudacao(); // chamando a função
```

📌 Função com parâmetros e retorno:

```
function somar(a, b) {
                return a + b;
                }

                const resultado = somar(5, 3);
                console.log("Resultado:", resultado); // 8
```

✅ **Dica:** Funções podem receber valores (parâmetros) e devolver algo com `return`.

🧱 Objetos em JavaScript

Objetos são usados para representar algo com várias informações juntas, como uma pessoa, produto, carro, usuário etc.

✅ Exemplo de objeto:

```
const pessoa = {
                    nome: "Ana",
                    idade: 25,
                    cidade: "São Paulo"
                    };

                    console.log(pessoa.nome);   // "Ana"
                    console.log(pessoa["idade"]); // 25
```

📌 Objeto com função dentro (método):

```
const usuario = {
                    nome: "Carlos",
                    saudacao: function() {
                        console.log("Olá, " + this.nome);
                    }
                    };

                    usuario.saudacao(); // "Olá, Carlos"
```

✅ **Dica:** Objetos são perfeitos para representar coisas do mundo real no código.

⚡ Funções Arrow (=>)

As **funções arrow** são uma forma mais curta e moderna de escrever funções no JavaScript.

✅ Comparação:

```
// Função tradicional
                function somar(a, b) {
                return a + b;
                }

                // Função arrow
                const somarArrow = (a, b) => {
                return a + b;
                };
```

```
// Arrow function ainda mais curta (retorno implícito)
const somarDireto = (a, b) => a + b;

console.log(somar(2, 3));        // 5
console.log(somarArrow(2, 3));   // 5
console.log(somarDireto(2, 3));  // 5
```

📦 Manipulação de Arrays

O JavaScript possui métodos poderosos para trabalhar com listas. Os mais usados são: **forEach, map e filter.**

🔁 forEach — percorre o array

```
const frutas = ["Maçã", "Banana", "Uva"];

frutas.forEach((item, indice) => {
console.log(indice, item);
});
// 0 Maçã, 1 Banana, 2 Uva
```

🧪 map — cria um novo array

```
const numeros = [1, 2, 3, 4];
const dobrados = numeros.map(num => num * 2);

console.log(dobrados); // [2, 4, 6, 8]
```

🎯 filter — filtra valores do array

```
const valores = [10, 3, 25, 8, 30];
const maiores = valores.filter(num => num > 10);

console.log(maiores); // [25, 30]
```

📁 O que é JSON?

JSON é um formato de texto usado para **armazenar e enviar dados**. É muito usado em APIs, banco de dados e troca de informações entre front-end e back-end.

```
// Exemplo de JSON (parecido com objetos JS)
{
"nome": "Ana",
"idade": 25,
"ativo": true,
"hobbies": ["Música", "Leitura"]
}
```

✅ Como usar JSON no JavaScript:

```
// Objeto normal em JS
const pessoa = {
nome: "Carlos",
idade: 30,
ativo: true
};

// Converter objeto em JSON (texto)
const json = JSON.stringify(pessoa);
console.log(json);
// {"nome":"Carlos","idade":30,"ativo":true}

// Converter JSON (texto) em objeto JS novamente
const objetoDeNovo = JSON.parse(json);
console.log(objetoDeNovo.nome); // Carlos
```

📁 JSON com vários dados (lista de objetos)

Em JSON, também podemos armazenar **vários registros ao mesmo tempo**. Para isso, usamos **colchetes [ ]** para criar uma **lista (array)** de objetos.

Isso é muito usado em APIs, bancos de dados e sistemas que precisam guardar listas de usuários, produtos, alunos, etc.

✅ Exemplo: lista de pessoas em JSON

```
[
{
    "nome": "Ana",
    "idade": 25,
    "cidade": "São Paulo"
},
{
    "nome": "Bruno",
    "idade": 30,
```

```
    "cidade": "Rio de Janeiro"
},
{
    "nome": "Carla",
    "idade": 22,
    "cidade": "Curitiba"
}
]
```

📌 Como usar esse JSON no JavaScript?

```
// Lista de pessoas (array de objetos)
const pessoas = [
{ nome: "Ana", idade: 25, cidade: "São Paulo" },
{ nome: "Bruno", idade: 30, cidade: "Rio de Janeiro" },
{ nome: "Carla", idade: 22, cidade: "Curitiba" }
];

// Exemplo: mostrar o nome de todas as pessoas
pessoas.forEach((pessoa) => {
console.log(pessoa.nome);
});

// Exemplo: pegar a cidade do segundo (índice 1)
console.log(pessoas[1].cidade); // "Rio de Janeiro"
```

💾 Salvando e carregando JSON

```
// Converter a lista para texto JSON (ex: enviar para API ou salvar)
const jsonTexto = JSON.stringify(pessoas);
console.log(jsonTexto);

// Converter texto JSON de volta para array de objetos
const listaConvertida = JSON.parse(jsonTexto);
console.log(listaConvertida[0].nome); // "Ana"
```

✅ **Dica:** JSON com vários objetos é ideal para listas como alunos, produtos, posts de blog, usuários de um sistema, etc.

💾 Salvando dados do formulário com JavaScript

Vamos criar um formulário no HTML, capturar os dados com JavaScript e exibir na tela sem recarregar a página.

✅ 1) Estrutura HTML do formulário

```
<h2>Cadastro de Pessoas</h2>

                <form id="form">
                <label>Nome:</label>
                <input type="text" id="nome" required>

                <label>Idade:</label>
                <input type="number" id="idade" required>

                <label>Cidade:</label>
                <input type="text" id="cidade" required>

                <button type="submit">Salvar</button>
                </form>

                <h3>Pessoas cadastradas:</h3>
                <ul id="lista"></ul>
```

✅ 2) JavaScript para salvar e mostrar os dados

```
const form = document.getElementById("form");
                const lista = document.getElementById("lista");
                let pessoas = []; // onde os dados serão armazenados

                form.addEventListener("submit", (event) => {
                event.preventDefault(); // evita recarregar a página

                // Pega os valores dos inputs
                const nome = document.getElementById("nome").value;
                const idade = document.getElementById("idade").value;
                const cidade = document.getElementById("cidade").value;

                // Cria um objeto com os dados
                const pessoa = { nome, idade, cidade };

                // Salva no array
                pessoas.push(pessoa);

                // Mostra na tela
                atualizarLista();

                // Limpa os campos
                form.reset();
                });

                function atualizarLista() {
                lista.innerHTML = ""; // limpa a lista antes de reexibir
```

```
pessoas.forEach((p) => {
    const item = document.createElement("li");
    item.textContent = `👤 Nome: ${p.nome} | 🎂 Idade: ${p.idade} |
    lista.appendChild(item);
});
}
```

✅ **Dica:** esse mesmo código pode ser adaptado para salvar no `localStorage`, enviar para uma API ou até transformar em tabela.

🖍️ Como Editar e Excluir Dados Salvos no JSON

Quando salvamos várias pessoas em um **array de objetos (JSON)**, também podemos**editar** e **excluir** esses cadastros usando JavaScript. Vamos aprender isso passo a passo!

✅ 1) Estrutura básica do formulário:

```
<form id="form">
                <input id="nome" type="text" placeholder="Nome" required />
                <input id="idade" type="number" placeholder="Idade" required />
                <input id="cidade" type="text" placeholder="Cidade" required />
                <button type="submit">Salvar</button>
                </form>

                <ul id="lista"></ul>
```

✅ 2) Salvando os dados em JSON (array de objetos):

```
let pessoas = []; // Aqui ficam os cadastros
                let indiceEdicao = null; // Guarda qual item está sendo editado

                form.addEventListener("submit", (event) => {
                event.preventDefault();

                const pessoa = {
                    nome: document.getElementById("nome").value,
                    idade: document.getElementById("idade").value,
                    cidade: document.getElementById("cidade").value
                };

                if (indiceEdicao === null) {
                    pessoas.push(pessoa); // Adiciona novo
                } else {
                    pessoas[indiceEdicao] = pessoa; // Salva edição
```

```
        indiceEdicao = null;
    }

    atualizarLista();
    form.reset();
    });
```

✅ 3) Mostrando os dados + botões de ação:

```
function atualizarLista() {
    lista.innerHTML = "";

    pessoas.forEach((p, index) => {
        const item = document.createElement("li");
        item.textContent = `👤 ${p.nome} | ${p.idade} anos | ${p.cidade}

        // Botão Editar
        const btnEditar = document.createElement("button");
        btnEditar.textContent = "🖍️ Editar";
        btnEditar.onclick = () => editarPessoa(index);

        // Botão Excluir
        const btnExcluir = document.createElement("button");
        btnExcluir.textContent = "🗑️ Excluir";
        btnExcluir.onclick = () => excluirPessoa(index);

        item.appendChild(btnEditar);
        item.appendChild(btnExcluir);
        lista.appendChild(item);
    });
    }
```

✅ 4) Funções para editar e excluir:

```
function editarPessoa(index) {
    document.getElementById("nome").value = pessoas[index].nome;
    document.getElementById("idade").value = pessoas[index].idade;
    document.getElementById("cidade").value = pessoas[index].cidade;
    indiceEdicao = index; // Ativa modo de edição
    }

    function excluirPessoa(index) {
    pessoas.splice(index, 1); // Remove do array
    atualizarLista();
    }
```

✅ **Dica:** Agora você consegue criar sistemas de cadastro simples com JavaScript, JSON e HTML. Você pode evoluir salvando no `localStorage` para não perder os dados ao recarregar a página!

<> Teste você mesmo:

🌐 HTML:

```
<h1>Clique no botão!</h1>
<button id="btn">Clique</button>
<p id="msg"></p>
```

⚙️ JavaScript:

```
const btn = document.getElementById("btn");
const msg = document.getElementById("msg");

btn.addEventListener("click", () => {
    msg.textContent = "✅ Funcionou com JavaScript!";
});
```

Pré-visualização:

Clique no botão!

Clique

Acesse nosso repositorio no GitHub e deixe sua contribuição