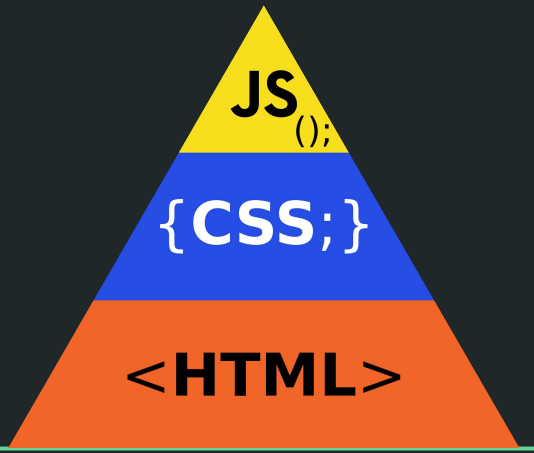
Programação WEB



Bruno Torres

Introdução

- JavaScript é uma linguagem de programação essencial para adicionar interatividade e dinamismo a páginas web.

- Combinado com HTML e CSS, o JavaScript permite criar interfaces de usuário complexas, animações e aplicações web completas.

O que você precisa saber:

- Noções básicas de HTML e CSS;

- Estrutura básica de um arquivo HTML.

```html
<!DOCTYPE html>
<html>
<head>
    <title></title>
    <link rel="stylesheet" type="text/css" href="estilo.css">
</head>
<body>
</body>
</html>
```

Inserindo JavaScript em HTML

- **Interno:** código JavaScript dentro do arquivo HTML, entre as tags `<script>` e `</script>`.

- **Externo:** código JavaScript em um arquivo separado (.js), referenciado no HTML com a tag `<script src="nome_do_arquivo.js">`.

Inserindo JavaScript em HTML

Interno: código JavaScript dentro do arquivo HTML, entre as tags <script> e </script>.

```html
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Exemplo JavaScript Interno</title>
</head>
<body>
  <p id="texto">Este texto muda de cor!</p>

  <script>
    // Função para mudar a cor do texto
    function mudarCor() {
      const texto = document.getElementById("texto");
      texto.style.color = "red";
    }

    // Registrar a função ao evento "mouseover" do elemento
    const elementoTexto = document.getElementById("texto");
    elementoTexto.addEventListener("mouseover", mudarCor);
  </script>
</body>
</html>
```

Inserindo JavaScript em HTML

Externo: código JavaScript em um arquivo separado (.js), referenciado no HTML com a tag:

<script src="nome_do_arquivo.js">.

```html
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Exemplo JavaScript Externo</title>
</head>
<body>
  <p id="texto">Este texto muda de cor!</p>

  <script src="script.js"></script>
</body>
</html>
```

Acessando elementos HTML

- `document.getElementById("id_do_elemento")`: obtém um elemento pelo seu ID.

- `document.getElementsByClassName("nome_da_classe")`: obtém uma lista de elementos pela classe.

- `document.getElementsByTagName("nome_da_tag")`: obtém uma lista de elementos pela tag.

Acessando elementos HTML

- document.getElementById("id_do_elemento"):

```html
<button id="meu-botao">Clique aqui!</button>
```

```
const botao = document.getElementById("meu-botao");

// Altera a cor do botão
botao.style.backgroundColor = "red";
```

Acessando elementos HTML

- document.getElementsByClassName("nome_da_classe"):

```html
<div class="caixa">Caixa 1</div>
<div class="caixa">Caixa 2</div>
<div class="outro">Caixa 3</div>
```

```js
const caixas = document.getElementsByClassName("caixa");

// Altera a cor de todas as caixas
for (const caixa of caixas) {
  caixa.style.border = "1px solid blue";
}
```

Acessando elementos HTML

- document.getElementsByTagName("nome_da_tag"):

```html
<h1>Título da página</h1>
<h2>Subtítulo</h2>
<p>Este é um parágrafo.</p>
```

```javascript
const titulos = document.getElementsByTagName("h1");

// Altera o texto do primeiro título
titulos[0].textContent = "Novo título";
```

Manipulando elementos

- innerHTML: altera o conteúdo textual de um elemento.
- style.propriedade: altera propriedades de estilo CSS de um elemento.
- classList.add("nome_da_classe"): adiciona uma classe a um elemento.
- classList.remove("nome_da_classe"): remove uma classe de um elemento.

Manipulando elementos

- innerHTML: altera o conteúdo textual de um elemento.

```html
<p id="meu-paragrafo">Este é um parágrafo.</p>
```

```javascript
const paragrafo = document.getElementById("meu-paragrafo");
paragrafo.innerHTML = "Este parágrafo foi atualizado!";
```

Manipulando elementos

- style.propriedade: altera propriedades de estilo CSS de um elemento.

```
<div id="meu-div"></div>
```

```
const div = document.getElementById("meu-div");
div.style.backgroundColor = "red";
div.style.width = "200px";
```

Manipulando elementos

- classList.add("nome_da_classe"): adiciona uma classe a um elemento.

```
<button id="meu-botao">Clique aqui</button>
```

```
const botao = document.getElementById("meu-botao");
botao.classList.add("ativo");
```

Manipulando elementos

- classList.remove("nome_da_classe"): remove uma classe de um elemento.

```html
<button id="meu-botao">Clique aqui</button>
```

```
const botao = document.getElementById("meu-botao");
botao.classList.remove("ativo");
```

Eventos

- Ações do usuário que podem ser detectados e manipulados pelo JavaScript.
- Exemplos: clique, mouseover, submit, keydown.
- addEventListener("nome_do_evento", function_callback): registra uma função para ser executada quando o evento ocorrer.

Eventos

```html
<button id="meu-botao">Clique aqui</button>
```

```javascript
const botao = document.getElementById("meu-botao");

botao.addEventListener("click", function() {
  alert("O botão foi clicado!");
});
```

O objeto document

- Representa o documento HTML completo.
- Permite acessar elementos, propriedades e métodos do documento.
- Exemplos:
  - document.title: obtém ou define o título da página.
  - document.location.href: obtém ou define a URL da página.