

O que é o DOM?

O **DOM (Document Object Model)** é uma interface de programação que permite que os documentos HTML, XML ou SVG sejam manipulados de forma dinâmica. Ele representa a estrutura de um documento como uma árvore de objetos, onde cada nó na árvore corresponde a um elemento, atributo ou texto presente no documento.

O DOM é uma API que conecta scripts (como JavaScript) ao conteúdo de uma página, permitindo que você:

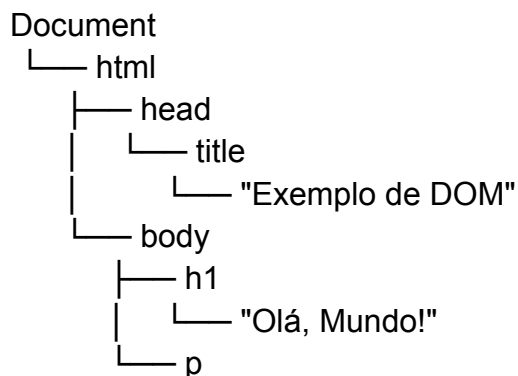
1. Leia e modifique a estrutura da página.
 2. Reaja a eventos como cliques ou interações do usuário.
 3. Alterne estilos, atributos e comportamentos em tempo real.
-

Estrutura do DOM

Considere o seguinte código HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo de DOM</title>
  </head>
  <body>
    <h1>Olá, Mundo!</h1>
    <p>Bem-vindo ao DOM.</p>
  </body>
</html>
```

O DOM para esse documento pode ser representado como uma **árvore hierárquica**:



└─ "Bem-vindo ao DOM."

Cada **nó** da árvore representa uma parte do documento, como:

- **Elemento:** `<html>`, `<head>`, `<body>`, `<h1>`, etc.
 - **Texto:** O conteúdo dentro dos elementos, como `"Olá, Mundo!"`.
-

Principais Propriedades e Métodos do DOM

Acessar Elementos

1. `document.getElementById(id)`

- Retorna um elemento com o `id` especificado.

```
const titulo = document.getElementById('meuTitulo');
```

2. `document.querySelector(selector)`

- Retorna o primeiro elemento que corresponde ao seletor CSS.

```
const paragrafo = document.querySelector('p');
```

3. `document.querySelectorAll(selector)`

- Retorna todos os elementos que correspondem ao seletor CSS.

```
const itens = document.querySelectorAll('.item');
```

Alterar Conteúdo

`element.textContent`: Altera o texto de um elemento.

```
const titulo = document.querySelector('h1');  
titulo.textContent = 'Texto Atualizado!';
```

`element.innerHTML`: Altera o conteúdo HTML de um elemento.

```
const div = document.querySelector('div');  
div.innerHTML = '<p>Conteúdo novo!</p>';
```

Estilizar Elementos

element.style.property: Modifica estilos inline.

```
const paragrafo = document.querySelector('p');  
paragrafo.style.color = 'blue';
```

Adicionar/Remover Elementos

Criar elementos:

```
const novoElemento = document.createElement('div');  
novoElemento.textContent = 'Novo Conteúdo';  
document.body.appendChild(novoElemento);
```

Remover elementos:

```
const elemento = document.querySelector('p');  
elemento.remove();
```

Eventos

Os eventos são ações que ocorrem no navegador (como cliques ou movimento do mouse) e podem ser capturados e tratados no DOM.

Exemplo de evento de clique:

```
const botao = document.querySelector('button');  
botao.addEventListener('click', () => {  
  alert('Botão clicado!');  
});
```

Por Que o DOM é Importante?

1. **Interatividade:** Permite criar páginas dinâmicas e interativas, reagindo a ações do usuário.
2. **Manipulação em Tempo Real:** Os elementos podem ser modificados sem recarregar a página.
3. **Acessibilidade:** Permite trabalhar com atributos e propriedades para melhorar a experiência de navegação.
4. **Compatibilidade:** Funciona em praticamente todos os navegadores modernos.

Limitações e Cuidados

1. **Manipulação Excessiva:** Alterar elementos em massa pode prejudicar o desempenho.
 2. **Cross-Site Scripting (XSS):** Usar `innerHTML` sem cuidado pode expor a aplicação a ataques.
 3. **Compatibilidade:** Certifique-se de que os métodos usados são suportados nos navegadores-alvo.
-

Resumo

O **DOM** é o coração da interação entre JavaScript e HTML, fornecendo uma estrutura programável para acessar e manipular páginas. Dominar o DOM é essencial para desenvolvedores web que desejam criar experiências ricas e responsivas na web.