

Vocação - SENAI

Juliane dos Anjos Silva

18/11/2024

Professor: Ivan Paulino

Web Design

DOM

DOM (Document Object Model) é uma interface de programação que representa documentos HTML e XML como uma árvore de objetos. Essencialmente, ele permite que programas e scripts interajam com o conteúdo, estrutura e estilo de um documento web de maneira dinâmica. Aqui estão alguns pontos principais sobre o DOM:

Estrutura em Árvore

O DOM organiza o documento como uma árvore de nós, onde cada nó representa uma parte do documento (como um elemento HTML, atributo ou texto). A raiz da árvore é o elemento <html>, e a partir dele, outros elementos e nós são seus descendentes.

Acesso e Manipulação

Os desenvolvedores podem usar o DOM para acessar e manipular elementos e atributos HTML. Por exemplo, você pode alterar o texto de um parágrafo, adicionar novos elementos à página ou modificar estilos.

APIs de Navegadores

Os navegadores fornecem várias APIs baseadas no DOM, permitindo que os desenvolvedores interajam com o documento através de linguagens de programação como JavaScript. Algumas operações comuns incluem:

- Selecionar elementos (getElementById, querySelector)

- Modificar conteúdo (innerHTML, textContent)
- Alterar estilos (style)
- Adicionar ou remover classes (classList)
- Manipular atributos (setAttribute, getAttribute)

Tipos de Nós (Nodes)

O DOM é composto por vários tipos de nós, cada um representando uma parte diferente do documento:

- Element Nodes: Representam elementos HTML e XML. Exemplos: <div>, <p>, <a>, etc.
- Text Nodes: Representam o texto contido dentro dos elementos. Exemplo: o texto "Olá, mundo!" dentro de um <p> seria um nó de texto.
- Attribute Nodes: Representam os atributos de um elemento. Embora eles sejam geralmente acessados através dos próprios elementos, tecnicamente são nós separados.
- Comment Nodes: Representam comentários no código HTML ou XML. Exemplo: <!-- Este é um comentário -->.

Coleções de Nós

O DOM fornece várias coleções de nós que você pode utilizar para acessar e manipular grupos de nós:

- NodeList: Uma coleção de nós, muitas vezes retornada por métodos como querySelectorAll.
- HTMLCollection: Uma coleção de elementos HTML, muitas vezes retornada por métodos como getElementsByTagName.

Eventos DOM

O DOM permite a criação de eventos que respondem a interações do usuário, sendo ações que ocorrem no documento como cliques, digitação e

movimentação do mouse. Isso é essencial para criar páginas web interativas e dinâmicas:

- Mouse Events: click, dblclick, mousedown, mouseup, mouseover, mouseout, etc.
- Keyboard Events: keydown, keypress, keyup.
- Form Events: submit, reset, focus, blur, change, input.
- Window Events: load, resize, scroll, unload.

Interfaces DOM

O DOM oferece várias interfaces que definem métodos e propriedades para trabalhar com diferentes tipos de nós e eventos:

- Document: Representa o documento em si e serve como ponto de entrada principal para o conteúdo da página.
- Element: Representa um elemento na página e fornece métodos e propriedades para manipulá-lo.
- Node: A interface básica para todos os tipos de nós no DOM.
- Event: Representa um evento que pode ser disparado em um elemento.


Exemplos e aplicações do DOM:

Manipulação de Conteúdo

Você pode usar o DOM para alterar o conteúdo de elementos na página.

- Exemplo: Alterar o texto de um parágrafo quando um botão é clicado.

Html

 Copiar

```
<p id="texto">Texto original</p>
<button onclick="alterarTexto()">Clique aqui</button>

<script>
  function alterarTexto() {
    document.getElementById("texto").textContent = "Texto alterado!"
  }
</script>
```

Criação Dinâmica de Elementos

É possível adicionar novos elementos à página dinamicamente.

- Exemplo: Adicionar um novo item a uma lista quando um botão é clicado.

Html

 Copiar

```
<ul id="minhaLista">
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
<button onclick="adicionarItem()">Adicionar Item</button>

<script>
  function adicionarItem() {
    const lista = document.getElementById("minhaLista");
    const novoItem = document.createElement("li");
    novoItem.textContent = "Novo Item";
    lista.appendChild(novoItem);
  }
</script>
```

Manipulação de Estilos

Você pode alterar os estilos de elementos diretamente através do DOM.

- Exemplo: Mudar a cor de fundo de um elemento ao passar o mouse sobre ele.

```
Html Copiar  
  
<div id="meuDiv" onmouseover="mudarCor()" onmouseout="reverterCor()">  
  Passe o mouse sobre mim!  
</div>  
  
<script>  
  function mudarCor() {  
    document.getElementById("meuDiv").style.backgroundColor = "yellow"  
  }  
  
  function reverterCor() {  
    document.getElementById("meuDiv").style.backgroundColor = "";  
  }  
</script>
```

Conclusão

O DOM é uma interface de programação para documentos HTML e XML, com todos os elementos estruturados em uma espécie de árvore, sendo a raiz dela o HTML. Ele serve como uma ponte entre o HTML e outras linguagens de programação como o JavaScript, resultando em uma página de web mais dinâmica e interativa, utilizando códigos e até mesmo e interações do usuário. Permitindo assim, uma manipulação de dados e criação de eventos de maneira satisfatória.

