

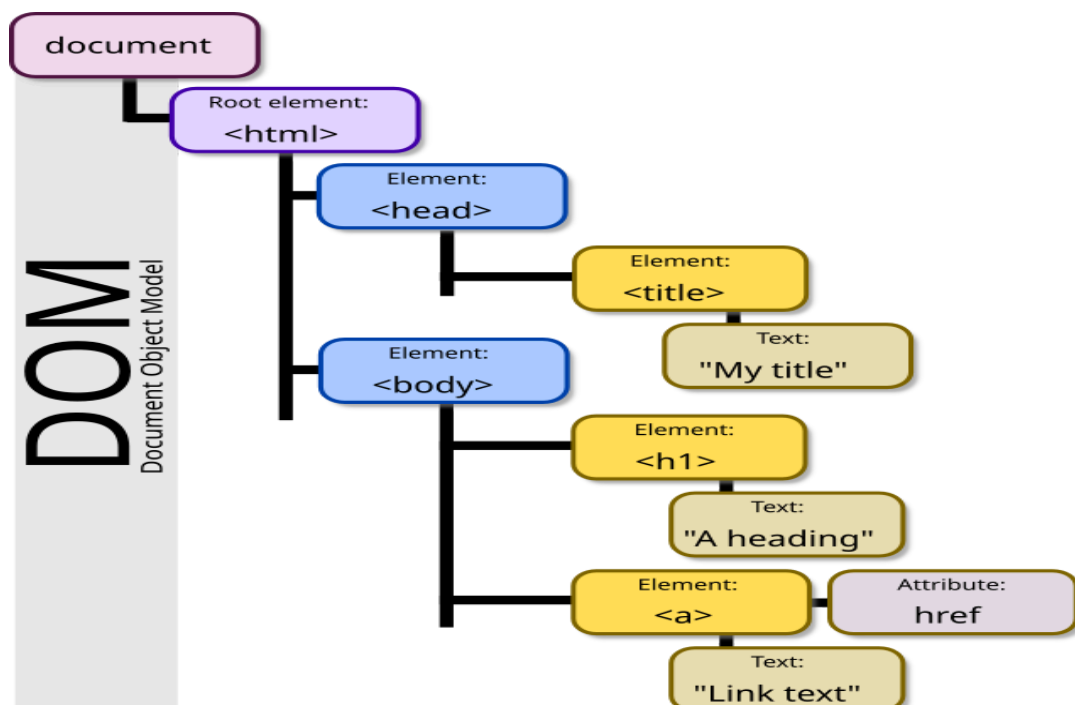


O QUE É DOM E JAVASCRIPT?

O **DOM (Document Object Model)** é uma interface de programação para documentos HTML e XML. Ele representa a estrutura de um documento de forma hierárquica, onde cada elemento, atributo e texto do documento é tratado como um objeto que pode ser manipulado via JavaScript.

Em termos simples, o DOM permite que você acesse e modifique o conteúdo, estrutura e estilo de uma página web dinamicamente. Cada parte de uma página (como tags HTML, atributos, texto, etc.) pode ser acessada e alterada através de comandos JavaScript.

O DOM é uma representação em árvore da página. Cada tag HTML é um nó na árvore e pode ser acessada, manipulada ou alterada. O DOM também permite que você adicione, remova ou modifique elementos e até crie eventos interativos.



APLICAÇÃO DO DOM

Uma das aplicações mais comuns do DOM é a **interatividade** em páginas web. Ele permite que a página responda a ações do usuário, como cliques em botões, movimentos do mouse, digitação de texto, entre outras interações, de forma dinâmica e sem a necessidade de recarregar a página. Isso resulta em uma experiência mais fluida e agradável para o usuário.

Exemplos de Aplicação do DOM:

1.Alteração Dinâmica do Conteúdo: O DOM permite que você altere o conteúdo da página instantaneamente. Por exemplo, ao clicar em um botão, é possível modificar o texto de um parágrafo, exibir novas informações ou ocultar elementos sem que a página precise ser recarregada. Esse tipo de funcionalidade é comum em páginas de busca, onde os resultados são atualizados dinamicamente conforme o usuário interage com os filtros ou opções.

Exemplo: Um botão que altera o conteúdo de um parágrafo:
javascript

```
document.getElementById("meuBotao").addEventListener("click", function() {  
    document.getElementById("meuParagrafo").textContent = "Texto alterado  
dinamicamente!";  
});
```

2.Atualização de Campos de Formulário: Outro exemplo de aplicação do DOM é a atualização de campos de formulário com base na interação do usuário. Isso pode ser feito alterando os valores dos campos ou exibindo informações adicionais à medida que o usuário preenche um formulário. Por exemplo, ao selecionar uma opção em um menu suspenso, outros campos podem ser preenchidos automaticamente ou validados dinamicamente.

Exemplo: Atualizando um campo de texto com base na seleção de um dropdown:
javascript

```
document.getElementById("meuDropdown").addEventListener("change", function() {  
    let valorSelecionado = this.value;  
    document.getElementById("campoTexto").value = "Você  
selecionou: " + valorSelecionado; });
```

3.Exibição de Mensagens de Erro ou Confirmação: A manipulação do DOM também é usada para mostrar mensagens de erro ou confirmação de forma dinâmica. Isso é especialmente útil em formulários, onde os dados inseridos pelo usuário podem ser validados em tempo real. Quando o usuário comete um erro, uma mensagem de alerta pode ser exibida imediatamente na página, sem necessidade de uma nova requisição ao servidor ou de um recarregamento da página.

Exemplo: Exibindo uma mensagem de erro quando um campo obrigatório não é preenchido:

javascript

```
document.getElementById("meuFormulario").addEventListener("submit", function(event) {
    let campoEmail = document.getElementById("email").value;
    if (campoEmail === "") {
        event.preventDefault(); // Impede o envio do formulário
        document.getElementById("erroEmail").textContent = "O e-mail é obrigatório!";
    }
});
```

4.Criação e Remoção de Elementos: O DOM também permite adicionar novos elementos à página e remover os existentes conforme necessário. Isso é útil para criar listas dinâmicas, inserir novos itens em uma tabela ou até mesmo exibir notificações temporárias.

Exemplo: Adicionando um novo item em uma lista:

javascript

Copiar código

```
let novaLista = document.createElement("li");
novaLista.textContent = "Novo item na lista";
document.getElementById("minhaLista").appendChild(novaLista);
```

Exemplo prático de manipulação do DOM:

Vamos criar um exemplo simples onde, ao clicar em um botão, o conteúdo de um parágrafo será alterado dinamicamente. Neste caso, vamos alterar o texto de um parágrafo na página.

HTML:

html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Exemplo DOM</title>
</head>
<body>
  <p id="paragrafo">Este é o texto original.</p>
  <button id="alterarTextoBtn">Alterar Texto</button>

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

JavaScript (em um arquivo `script.js`):

javascript

```
// Acessa o botão e o parágrafo usando seus IDs
const botao = document.getElementById("alterarTextoBtn");
const paragrafo = document.getElementById("paragrafo");

// Função para alterar o texto do parágrafo
function alterarTexto() {
  paragrafo.textContent = "O texto foi alterado!";
}

// Adiciona um evento de clique no botão
botao.addEventListener("click", alterarTexto);
```

CONCLUSÃO

Observamos que o DOM se aplica muito à ordenação de estrutura de códigos que ficam tanto no HTML quanto no XML. Fornecendo uma estrutura que possibilita acessar e modificar elementos da página, alterar estilos e responder a eventos do usuário em tempo real, deixando a experiência muito mais interativa e dinâmica. Além disso, a capacidade de responder a eventos do usuário torna as páginas mais responsivas e envolventes. Utilizando diversos elementos interativos como um botão que altera o texto de um parágrafo, exibindo novas informações ou apenas ocultar elementos sem que a página precise ser atualizada.

