

JavaScript

Tecnologia em Análise e Desenvolvimento de Sistemas

Senac - Portão - Curitiba/PR

Prof. Msc. Guilherme Matos Barbosa



Introdução ao DOM

O que é o DOM (Document Object Model)

O DOM (Document Object Model) é um padrão que define como a linguagem Javascript pode acessar e modificar um documento HTML dentro do navegador;

Em outras palavras, o DOM pode ser entendido como uma API de acesso ao documento exibido pelo navegador.



Introdução ao DOM

Estrutura Básica do DOM

A árvore DOM consiste em nós de diferentes tipos:

- Elementos: Representam tags HTML (<div>, <p>, etc.);
- Atributos: Propriedades dos elementos (class, id, etc.);
- Texto: O conteúdo textual dos elementos.

Introdução ao DOM

Exemplo visual





Introdução ao DOM

Manipulação do DOM

Programas escritos em Javascript podem, facilmente, acessar e modificar quaisquer características de um documento HTML;

Dessa forma, os programas Javascript rodados em navegadores possuem acesso à uma variável global com o nome de document;

E, a partir dela, a todas operações possíveis sobre o documento.



Introdução ao DOM

Seleção de Elementos

Em geral, quando é necessário manipular um documento HTML a partir do DOM, temos que identificar o elemento a ser alterado;

Um elemento nada mais é do que um componente que faz parte do documento (como um parágrafo, um título, um link, etc.).



Introdução ao DOM

Exemplo:

Existem algumas formas de seleção de elementos permitidos pelo DOM:

```
// Seleção por id
var x = document.getElementById("teste");

// Seleção por nome de Tag
var x = document.getElementsByTagName("p");

// Seleção por classe
var x = document.getElementsByClassName("intro");

// Seleção por seletores CSS
var x = document.querySelectorAll("p.intro");
```



Introdução ao DOM

Exemplo:

- Selecionar e alterar o texto de um parágrafo com ID "paragrafo":

```
let paragrafo = document.getElementById("paragrafo");  
paragrafo.innerText = "Novo texto do parágrafo";
```




Introdução ao DOM

Exemplo:

Cada elemento do DOM possui atributos e métodos utilizados para alterar as propriedades do elemento no documento.

Por exemplo, os comandos abaixo realizam alterações em elementos do DOM:



Introdução ao DOM

```
// Altera todo o body
document.write(Date());

// Altera o conteúdo do elemento com id igual a "p1"
document.getElementById("p1").innerHTML = "Novo conteúdo!";

// Cada elemento pode ser atribuído em uma variável e alterado posteriormente
var element = document.getElementById("header");
element.innerHTML = "Novo texto";

// Altera propriedades CSS
document.getElementById("p2").style.color = "blue";
```

Manipulação de Elementos

Alteração de Conteúdo:

- Alterar o texto e o HTML interno de um elemento:

```
let elemento = document.getElementById("exemploId");  
elemento.innerText = "Novo texto";  
elemento.innerHTML = "<strong>Texto em negrito</strong>";  
elemento.textContent = "Texto sem formatação HTML";
```



Manipulação de Elementos

Exemplo Prático:

- Alterar o conteúdo de um elemento <div>:

```
let div = document.querySelector(".exemploDiv");  
div.innerHTML = "<p>Parágrafo dentro da div</p>";
```



Manipulação de Elementos

Estilo de Elementos:

- Alterar estilos diretamente:

```
let elemento = document.getElementById("exemploId");  
elemento.style.color = "blue";  
elemento.style.fontSize = "20px";
```



Manipulação de Elementos

Exemplo prático

- Alterar múltiplos estilos de um elemento:

```
let div = document.querySelector(".exemploDiv");  
div.style.backgroundColor = "lightgray";  
div.style.padding = "10px";  
div.style.borderRadius = "5px";
```



Criação e Remoção de Elementos

Criação de Novos Elementos:

- Criar e configurar novos elementos:

```
let novoElemento = document.createElement("div");  
novoElemento.innerText = "Novo Elemento";
```



Criação e Remoção de Elementos

Exemplo Prático:

- Adicionar elementos à árvore DOM:

```
let novoParagrafo = document.createElement("p");
novoParagrafo.innerText = "Este é um novo parágrafo.";
let div = document.getElementById("exemploDiv");
div.appendChild(novoParagrafo);
```


Criação e Remoção de Elementos

Exemplo Prático:

- Inserir um novo elemento antes de um elemento específico:

```
let novoElemento = document.createElement("li");
novoElemento.innerText = "Novo item da lista";
let lista = document.getElementById("lista");
let primeiroItem = lista.firstChild;
lista.insertBefore(novoElemento, primeiroItem);
```



Criação e Remoção de Elementos

Remoção de Elementos:

- Remover elementos da árvore DOM:

```
let elementoParaRemover = document.getElementById("exemploId");  
elementoParaRemover.parentNode.removeChild(elementoParaRemover);
```

Eventos no DOM

Adição de Eventos a Elementos:

- Adicionar eventos usando **addEventListener**:

```
let botao = document.getElementById("meuBotao");  
botao.addEventListener("click", function() {  
    alert("Botão clicado!");  
});
```

Eventos no DOM

Exemplo Prático:

- Alterar o texto de um parágrafo ao clicar em um botão:

```
let botao = document.getElementById("meuBotao");
botao.addEventListener("click", function() {
  let paragrafo = document.getElementById("meuParagrafo");
  paragrafo.innerText = "Texto alterado pelo clique no botão!";
});
```

Eventos no DOM

Exemplo Prático:

- Exemplos de eventos comuns:

```
let div = document.getElementById("exemploDiv");
div.addEventListener("mouseover", function() {
  div.style.backgroundColor = "yellow";
});
div.addEventListener("mouseout", function() {
  div.style.backgroundColor = "";
});
```

Eventos no DOM

Exemplo Prático:

- Adicionar um evento de teclado:

```
let input = document.getElementById("meuInput");
input.addEventListener("keyup", function(event) {
  console.log("Tecla pressionada: " + event.key);
});
```