

JavaScript: Fundamentos para Desenvolvimento Web Interativo - Turma 2025B

4.3 Modificando elementos

Uma vez que você tem acesso a um elemento, pode modificar seu conteúdo, atributos e estilo. Vamos explorar como você pode efetivamente modificar elementos do DOM usando JavaScript.

Modificando HTML interno

A propriedade **innerHTML** é usada para obter ou definir o conteúdo HTML de um elemento. Ela permite acessar o conteúdo HTML dentro de um elemento e pode ser usada para inserir ou modificar HTML dinamicamente. Veja um exemplo no quadro abaixo:

```
const container = document.getElementById('container');  
container.innerHTML = '<p>Este é um novo parágrafo</p>';
```

Modificando o conteúdo de texto

Para obter ou modificar apenas o texto de um elemento, sem afetar qualquer marcação interna, você pode usar a propriedade **textContent**. Veja um exemplo no quadro abaixo:

```
const elemento = document.getElementById('meuElemento');  
elemento.textContent = 'Novo conteúdo de texto aqui!';
```

Alterando atributos

Você pode manipular atributos de elementos, como **id**, **class**, **src**, **href**, e muitos outros, utilizando **setAttribute**, **getAttribute**, e **removeAttribute**.

Método **setAttribute**

O método **setAttribute** é usado para definir ou adicionar um novo atributo a um elemento HTML. Ele recebe dois argumentos: o nome do atributo a ser definido e o valor do atributo.

O código demonstrado no quadro abaixo definirá o atributo **"src"** do elemento com o ID **"meuElemento"** com o valor **"imagem.jpg"**:

```
const elemento = document.getElementById("meuElemento");  
  
// Definindo o atributo "src" com o valor "imagem.jpg"  
elemento.setAttribute("src", "imagem.jpg");
```

Método **getAttribute**

O método **getAttribute** é usado para obter o valor de um atributo específico de um elemento HTML. Ele recebe o nome do atributo como argumento e retorna o valor desse atributo.

O código demonstrado no quadro abaixo irá imprimir o valor do atributo **"src"** do elemento com o ID **"meuElemento"**:

```
const elemento = document.getElementById("meuElemento");  
  
// Obtendo o valor do atributo "src"  
const valorSrc = elemento.getAttribute("src");  
  
console.log(valorSrc); // Saída: "imagem.jpg"
```

Método **removeAttribute**

O método **removeAttribute** é usado para remover um atributo específico de um elemento HTML. Ele recebe o nome do atributo a ser removido como argumento.

O código demonstrado no quadro abaixo removerá o atributo **"src"** do elemento com o ID **"meuElemento"**:

```
const elemento = document.getElementById("meuElemento");  
  
// Removendo o atributo "src"  
elemento.removeAttribute("src");
```

Modificando estilos CSS

Modificar estilos CSS com JavaScript é uma técnica comum e útil para criar interações dinâmicas em uma página da web. Isso permite modificar qualquer propriedade CSS de forma programática.

Alterando diretamente as propriedades de estilo

Você pode acessar as propriedades de estilo de um elemento HTML diretamente através da propriedade **style**.

Veja um exemplo no quadro abaixo:

```
const elemento = document.getElementById("meuElemento");  
  
// Alterando a cor do texto para vermelho  
elemento.style.color = "red";  
  
// Modificando o tamanho da fonte  
elemento.style.fontSize = "20px";
```

Adicionando ou removendo classes CSS

Você pode adicionar ou remover classes CSS de um elemento para aplicar estilos definidos em uma folha de estilo externa ou interna.

Veja um exemplo no quadro abaixo:

```
const elemento = document.getElementById("meuElemento");

// Adicionando uma classe CSS
elemento.classList.add("minhaClasse");

// Removendo uma classe CSS
elemento.classList.remove("outraClasse");
```

Aplicando estilos com `setAttribute`

Você também pode aplicar estilos usando o método **`setAttribute`** para definir o atributo **`style`** diretamente. Isso permite uma manipulação mais detalhada, embora menos conveniente.

Veja um exemplo no quadro abaixo:

```
const elemento = document.getElementById("meuElemento");

// Definindo diretamente uma propriedade de estilo
elemento.setAttribute("style", "color: blue; font-size: 16px;");
```

◀ 4.2 Acessando elementos

Seguir para...

4.4 Criando e adicionando elementos ►