

JavaScript: Fundamentos para Desenvolvimento Web Interativo - Turma 2025B

4.4 Criando e adicionando elementos

O JavaScript permite você criar elementos HTML dinamicamente e inseri-los na página, possibilitando a interação em tempo real sem a necessidade de recarregar a página.

Método `document.createElement`

O método **`document.createElement`** permite os desenvolvedores criarem um novo elemento HTML dinamicamente, que pode então ser inserido no documento atual em qualquer posição desejada. O novo elemento é um objeto **Element**, que pode ser manipulado usando JavaScript antes de ser adicionado ao DOM visível da página.

A sintaxe para criar um novo elemento é bastante simples, conforme demonstrado no quadro abaixo:

```
const novoElemento = document.createElement(tagName);
```

Onde **tagName** é uma string que especifica o tipo do elemento a ser criado. Este deve ser o nome de uma tag HTML válida, como **'div'**, **'p'**, **'span'**, **'img'**, etc.

Para exemplificar o uso deste método, suponha que você pretenda criar um novo elemento **<div>** em sua página, conforme quadro abaixo:

```
// Criar um novo elemento <div>:  
const novaDiv = document.createElement('div');
```

Após criar um elemento, você pode definir suas propriedades, como **id**, **className** (atributo class no HTML), e **textContent** (o texto dentro do elemento). Veja um exemplo no quadro abaixo:

```
// Definindo propriedades do elemento  
novaDiv.id = 'minhaNovaDiv';  
novaDiv.className = 'classeDiv';  
novaDiv.textContent = 'Olá, sou uma nova div!';
```

Você também pode adicionar atributos usando **setAttribute**, que é útil para atributos não padronizados ou para garantir que o atributo seja corretamente configurado como HTML. Veja um exemplo no quadro abaixo:

```
// Adicionando um atributo 'data'  
novaDiv.setAttribute('data-custom', 'valorCustomizado');
```

Para modificar a aparência do elemento, você pode definir propriedades CSS diretamente através da propriedade **style** do elemento. Veja um exemplo no quadro abaixo:

```
// Adicionando estilos ao elemento  
novaDiv.style.color = 'red';  
novaDiv.style.backgroundColor = 'black';  
novaDiv.style.padding = '10px';  
novaDiv.style.marginTop = '20px';
```

OBS: Um elemento criado com **createElement** não é automaticamente visível no documento. Ele precisa ser explicitamente adicionado ao DOM. Isso é feito usando métodos como **appendChild** ou **insertBefore**. Assim, percebemos que você pode manipular completamente o elemento (atribuir estilos, adicionar atributos, etc.) antes de inseri-lo no DOM.

Método appendChild

O método **appendChild** é usado para adicionar um nó ao final da lista de filhos de um nó pai especificado. Se o nó que está sendo adicionado já existe em outro local do DOM, ele será removido de sua posição atual e então adicionado ao novo local.

A sintaxe básica para `appendChild` está demonstrada no quadro abaixo:

```
parentNode.appendChild(childNode);
```

Onde:

- **parentNode**: O nó DOM ao qual o novo filho será adicionado.
- **childNode**: O nó que será adicionado como um filho do **parentNode**.

Para exemplificar o uso deste método, observe o seguinte código, conforme quadro abaixo:

```
// Cria um novo elemento
const novoElemento = document.createElement('div');

// Adiciona conteúdo ao novo elemento
novoElemento.textContent = 'Este é um novo elemento';

// Adiciona o novo elemento ao DOM
document.body.appendChild(novoElemento);
```

Método `insertBefore`

O método **`insertBefore`** insere um nó diretamente antes de outro nó filho dentro de um elemento pai específico. Se o nó a ser inserido já existe em outro lugar no DOM, ele será movido para a nova localização (não será duplicado). Esse método é útil quando você precisa inserir elementos em posições específicas dentro do DOM, em vez de simplesmente adicioná-los ao final de um nó pai, como faz o **`appendChild`**.

A sintaxe para usar **`insertBefore`** está demonstrada no quadro abaixo:

```
parentNode.insertBefore(newNode, referenceNode);
```

Onde:

- **parentNode**: O elemento pai que contém o nó de referência.
- **newNode**: O novo nó que você deseja inserir.

- **referenceNode**: O nó existente antes do qual o novo nó será inserido. Se esse parâmetro for **null**, **insertBefore** funcionará como **appendChild**, e **newNode** será adicionado ao final do **parentNode**.

Para exemplificar o uso deste método, suponha que você tenha o seguinte HTML:

```
<ul id="minhaLista">
  <li id="item1">Item 1</li>
  <li id="item2">Item 2</li>
</ul>
```

Agora, considere que você deseja inserir um novo item entre "Item 1" e "Item 2":

```
const lista = document.getElementById('minhaLista');

const novoItem = document.createElement('li');
novoItem.textContent = 'Item Novo';

const item2 = document.getElementById('item2');
lista.insertBefore(novoItem, item2);
```

Observe, na figura abaixo, que isso adicionará "Item Novo" diretamente antes de "Item 2".

- Item 1
- Item Novo
- Item 2

Figura 1 - Exemplo de uso do método insertBefore

Descrição da imagem: No navegador, observamos uma lista não ordenada () com o id "minhaLista", contendo dois itens de lista (). Os itens são "Item 1" e "Item 2". Após a execução do JavaScript, um novo item de lista com o texto "Item Novo" é adicionado antes do "Item 2". Assim, a lista final no navegador aparece como "Item 1", "Item Novo" e "Item 2".

◀ 4.3 Modificando elementos

Seguir para...

4.5 Removendo e substituindo elementos ▶

[Baixar o aplicativo móvel.](#)