



Aula 5 – Modificando elementos

Criando elementos dinamicamente

- Para criar um novo elemento que será exibido na página temos que:
 1. Criar este novo elemento pelo JavaScript
 2. Adicionar este elemento criado no DOM.



Aula 5 – Modificando elementos

Criando elemento

- O objeto **document** possui um método **createElement()** que recebe um nome de **tag** como parâmetro e retorna esse elemento. Por exemplo, poderíamos criar um novo item para uma lista de alimentos escrevendo o seguinte no console:

```
const banana = document.createElement('li');
```



Aula 5 – Modificando elementos

Criando elemento

- No momento, este elemento está vazio. Para adicionar algum conteúdo, usaremos a propriedade **innerHTML** para adicionar o conteúdo HTML:

```
banana.innerHTML = `Banana`;
```

- Como é um elemento de texto podemos utilizar **textContent**:

```
banana.textContent = 'Banana';
```



Aula 5 – Modificando elementos

Adicionando elemento no DOM

- Agora que aprendemos como criar elementos HTML, precisamos aprender como colocá-los na página. Há vários métodos para permitir que você insira nós no DOM:
- Vamos primeiro olhar para **appendChild()**. Cada nó DOM tem um método `appendChild()` que adicionará outro nó (dado como argumento) como um nó filho.



Aula 5 – Modificando elementos

Adicionando elemento no DOM

- Agora que aprendemos como criar elementos HTML, precisamos aprender como colocá-los na página. Há vários métodos para permitir que você insira nós no DOM:
- Vamos primeiro olhar para **appendChild()**. Cada nó DOM tem um método `appendChild()` que adicionará outro nó (dado como argumento) como um nó filho. O novo nó será inserido na **última** posição do elemento.

```
food.appendChild(banana);
```



Aula 5 – Modificando elementos

Adicionando elemento no DOM

- Agora que aprendemos como criar elementos HTML, precisamos aprender como colocá-los na página. Há vários métodos para permitir que você insira nós no DOM:
- Vamos primeiro olhar para **appendChild()**. Cada nó DOM tem um método `appendChild()` que adicionará outro nó (dado como argumento) como um nó filho. O novo nó será inserido na **última** posição do elemento.

```
food.appendChild(banana);
```



Aula 5 – Modificando elementos

Adicionando elemento no DOM

- Outro método para inserção de um elemento no DOM é o **insertBefore()**. Ele serve para colocar um novo elemento entre dois elementos existentes.
- É importante observar que esse método é chamado no nó pai. São necessários dois parâmetros: o primeiro é o novo nó a ser adicionado e o segundo é o nó que você deseja que ele vá antes.

```
food.insertBefore(banana,apple);
```



Aula 5 – Modificando elementos

Removendo elemento no DOM

- Um elemento pode ser removido de uma página usando o método **remove()**. Este método é chamado no nó a ser removido. Ele retorna uma referência ao nó removido. Por exemplo, se quiséssemos remover o elemento “banana”, usaríamos o seguinte código:

```
banana.remove();  
<< "<li>Banana</li>"
```



Aula 5 – Modificando elementos

Alterando elemento no DOM

- O método **replaceChild()** pode ser usado para substituir um nó por outro. Ele é chamado no nó pai e tem dois parâmetros: o novo nó e o nó que deve ser substituído. Por exemplo, se quisermos alterar o conteúdo do terceiro item da, podemos substituir o nó de texto por um novo, assim:

```
const lemon = document.createElement('li');  
lemon.innerHTML = `Lemon`;   
food.replaceChild(lemon,banana);
```



Obrigado!

Professor ||| `alert('Hello, World!');`
LAURO VICTOR

