**Para saber mais: além do append - Outros métodos do DOM**

A manipulação do DOM é uma habilidade fundamental no desenvolvimento web. E mesmo que o .append() seja uma das funções mais populares, existem várias outras que podem ser muito úteis. A seguir, vou detalhar algumas delas pra ficar familiarizado com outras formas de manipular o DOM:

1 - **.append():**

Bastante utilizado durante o curso, esse método permite adicionar múltiplos nós e/ou strings de texto ao final de um elemento. Se você passar uma string, o método a trata como um texto e a adiciona diretamente.

2 - **.appendChild():**

É um método mais antigo e é específico para inserir um único nó no final do elemento selecionado. Diferentemente do .append(), o .appendChild() não aceita strings diretamente. Se você quiser adicionar uma string, precisa criar um nó de texto primeiro.

Exemplo:

**let** div = document.**querySelector**("div");

**let** novoSpan = document.**createElement**("span");

div.**appendChild**(novoSpan);

Por que é útil conhecer ambos?

Enquanto .append() oferece mais flexibilidade, permitindo inserir múltiplos elementos e/ou texto, .appendChild() tem amplo suporte e é mais específico na sua tarefa. Para desenvolvedores que estão focados no front-end, é fundamental conhecer ambos os métodos, para poder decidir qual é o mais adequado conforme a situação e o ambiente em que estão trabalhando.

3 - **.prepend():**

Esse método insere conteúdo no início de um elemento selecionado.

Por que é útil?

Se você deseja adicionar algo no topo de uma lista ou no começo de um container, esta é a função a ser usada.

Exemplo:

**let** lista = document.**querySelector**("ul");

lista.**prepend**("Item 0"); // Este item será inserido no começo da lista.

4 - **.insertBefore():**

Permite inserir um elemento, especificamente, antes de outro elemento.

Por que é útil?

É muito útil quando você precisa inserir um item em uma posição específica, e não apenas no começo ou fim.

Exemplo:

**let** novoItem = document.**createElement**("li");

novoItem.textContent = "Item intermediário";

**let** lista = document.**querySelector**("ul");

**let** itemEspecifico = document.**querySelector**("#itemEspecifico");

lista.**insertBefore**(novoItem, itemEspecifico); // Insere o novo item antes do item especificado.

5  **.insertAdjacentElement()**, **.insertAdjacentHTML()** e **.insertAdjacentText()**:

Oferecem controle preciso sobre onde você deseja inserir um novo elemento, seja ele um texto, HTML ou elemento, em relação a um elemento existente.

Por que é útil?

Permitem inserções em quatro posições diferentes: antes, depois, no início ou no final do elemento selecionado.

Exemplo:

**let** div = document.**querySelector**("div");

div.**insertAdjacentHTML**('beforebegin', '<p>Antes da div</p>');

div.**insertAdjacentHTML**('afterend', '<p>Depois da div</p>');

div.**insertAdjacentHTML**('afterbegin', '<p>No começo da div</p>');

div.**insertAdjacentHTML**('beforeend', '<p>No final da div</p>');

6 - **.replaceChild():**

Substitui um elemento filho por outro.

Por que é útil?

Se você precisa atualizar ou trocar um elemento por um novo, você pode usar este método.

Exemplo:

**let** lista = document.**querySelector**("ul");

**let** antigoItem = document.**querySelector**("#antigoItem");

**let** novoItem = document.**createElement**("li");

novoItem.textContent = "Item substituto";

lista.**replaceChild**(novoItem, antigoItem); // Substitui o item antigo pelo novo.

 7 - **.cloneNode():**

Duplica um elemento.

Por que é útil?

Se você precisa de uma cópia exata de um elemento (com ou sem seus filhos), este método é perfeito.

Exemplo:

**let** item = document.**querySelector**("li");

**let** itemDuplicado = item.**cloneNode**(**true**); // Com "true" ele copia todos os filhos do item.

document.**querySelector**("ul").**appendChild**(itemDuplicado);