

MATERIAL EXTRA

Manipulando Elementos



growdev

MANIPULANDO ELEMENTOS

Apesar de ser possível criar elementos HTML dinamicamente através de atribuições no `innerHTML`, a atribuição é feita por strings e pode dificultar o processo. Dentro da interface de `document`, além da seleção de elementos já existentes, também é possível realizar a criação de **novos elementos**. A função usada neste caso é o `document.createElement`.

A função `createElement` recebe uma string como parâmetro que indica qual será o elemento HTML criado (p.ex. uma `<div>` ou um `<h1>`). O retorno é o próprio elemento criado. A partir de então, o elemento pode ser incrementado com as propriedades `innerHTML`, `style` e demais. Considere o HTML abaixo.

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <title>JS - DOM</title>
  </head>
  <body>
    <h1>Growdev</h1>
    <div id="content"></div>

    <script src="./script.js"></script>
  </body>
</html>
```

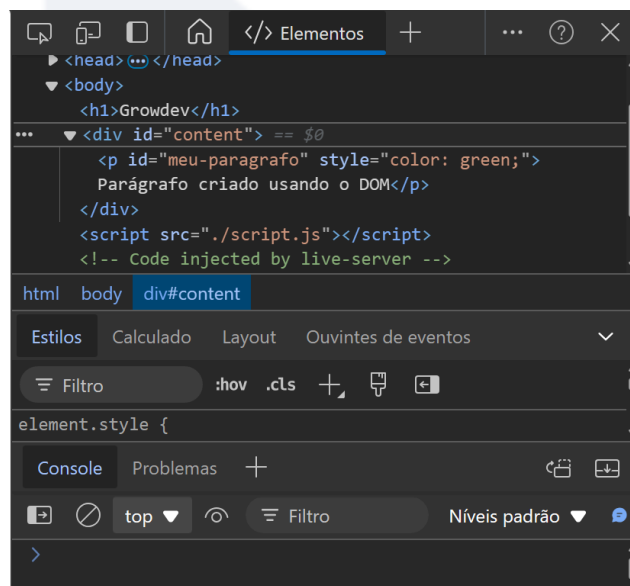
No documento acima separamos o título do conteúdo da página a partir de dois elementos HTML (um `<h1>` e uma `<div>`), sendo que a `<div>` de conteúdo recebe o ID "content". Podemos, por exemplo, criar um novo elemento que irá se colocar dentro desta `<div>`.

```
const novoParagrafo = document.createElement("p");
novoParagrafo.style.color = "green";
novoParagrafo.innerHTML = "Parágrafo criado usando o DOM";
novoParagrafo.setAttribute("id", "meu-paragrafo");

const divContent = document.querySelector("#content");
divContent.appendChild(novoParagrafo);
```

Growdev

Parágrafo criado usando o DOM



Para a criação do elemento chamamos a função `createElement` usando "p" como parâmetro, pois definimos que o elemento é uma tag `<p>`. Logo em seguida, setamos a propriedade `color` do CSS, inserimos o conteúdo HTML do elemento e finalmente definimos o seu ID.

Neste ponto, o elemento estará criado no código mas em momento algum definimos em qual lugar da estrutura DOM ele deveria se encaixar. Portanto, cabe ressaltar que o `createElement` sozinho **não gera modificações na página**.

Para adicionar o novo elemento no DOM, é preciso especificar seu lugar na estrutura. No exemplo acima, buscamos a <div> com ID content e usamos a função appendChild que serve para **adicionar** um **elemento filho** em sua hierarquia. Para esta função, o parâmetro é o elemento criado anteriormente que desejamos incluir no DOM.

Existem outras duas ações referente a atualização de nós filhos em um elemento, além do appendChild. O removeChild recebe um elemento como parâmetro e **remove** ele da estrutura de filhos do elemento alvo. Já o replaceChild recebe dois elementos como parâmetros e **substitui** o primeiro – caso exista – pelo segundo.

REFERÊNCIAS

1. [O que é DOM \(Document Object Model\)? Trabalhando com DOM em JavaScript](#)
2. [Introdução ao DOM - APIs da Web | MDN](#)
3. [JavaScript DOM Tutorial](#)
4. [HTML DOM Style object](#)
5. [Document - Web APIs | MDN](#)

BONS ESTUDOS