

Bruna Pereira Sanches - 5º SI

Pesquisa de Programação Web

Estes são termos de manipulação de elementos HTML, utilizando o JavaScript.

InnerText

Permite acessar e modificar um texto visível de um elemento HTML. Incluindo o texto e os espaços que ficam visíveis, mas exclui qualquer conteúdo HTML ou elementos filhos.

InnerHTML

Pode acessar e modificar tanto o conteúdo de texto quanto o conteúdo HTML dentro de um elemento, ou seja, ao invés do innerText, o innerHTML inclui todos os elementos HTML e formatação dentro desse elemento.

TextContent

Sendo semelhante ao innerText, permite acessar e modificar o texto dentro de um elemento. Mas, o textContent retorna o conteúdo de texto exato, sem interpretar ou considerar elementos HTML internos.

É importante observar certos pontos de segurança, em desenvolvimento web é comum que os sites permitam que os usuários insiram texto ou outros tipos de conteúdo, como comentários, mensagens, entradas de formulários, etc. Esses dados fornecidos pelos usuários podem ser inseridos dinamicamente em páginas da web usando propriedades como innerHTML ou textContent. No entanto, isso pode representar um risco de segurança se não for tratado corretamente.



Sobre os conceitos de **getElementById** e **querySelector**, os dois são métodos que pertencem ao objeto document.

getElementById:

Ele permite selecionar um elemento HTML específico pelo seu ID único. Ele pertence ao objeto document, que representa o documento HTML em JavaScript. O método busca no documento o elemento com o ID especificado e o retorna. Se nenhum elemento com o ID for encontrado, ele retorna null. Quando é preciso o **getElementById** é mais rápido e direto ao selecionar um elemento de id único.

Exemplo: document.getElementById('id-do-elemento')

querySelector:

É mais flexível e permite selecionar elementos HTML usando seletores CSS. Ele também pertence ao objeto document. O método procura no documento o primeiro elemento que corresponde ao seletor CSS especificado e o retorna. Se nenhum elemento for encontrado, ele retorna null. E o **querySelector** é mais flexível e podendo ser usado com seletores CSS complexos para selecionar elementos com base em classes, tipos e hierarquia.

Exemplo: document.querySelector('seletor-css')

Normalmente, desenvolvedores escolhem entre usar **querySelector** ou **getElementById** para selecionar elementos, pois não há a necessidade de utilizar os dois ao mesmo tempo para o mesmo elemento, mas isso não são regras.