**Para saber mais: acessando dados com propriedades no DOM**

No projeto *Fokus*, estamos utilizando algumas das propriedades mais conhecidas para acessar elementos no DOM, como o classList e o innerHTML, porém, existem outros igualmente importantes.

Vamos conhecê-los?

## **Propriedade**[parentNode](https://developer.mozilla.org/pt-BR/docs/Web/API/Node/parentNode" \t "_blank)

A propriedade parentNode é utilizada para acessar o nó pai de um elemento no DOM. Por meio dela, podemos navegar pela árvore do DOM em direção ao nó pai do elemento atual.

**Exemplo de uso:**

Suponha que temos o seguinte código HTML:

<div id="container">

<p>Este é um parágrafo</p>

</div>

Agora, podemos usar o ‘parentNode’ para acessar o elemento pai do parágrafo:

**const** paragraph = document.**querySelector**('p');

**const** parentElement = paragraph.parentNode;

console.**log**(parentElement.id); // Saída: "container"

## **Propriedade**[childNodes](https://developer.mozilla.org/pt-BR/docs/Web/API/Node/childNodes" \t "_blank)

A propriedade childNodes é utilizada para acessar todos os nós filhos de um elemento no DOM. Ela retorna uma lista de nós, incluindo nós de texto e elementos HTML.

**Exemplo de uso:**

Considerando o mesmo HTML do exemplo anterior, podemos usar o **childNodes** para obter todos os nós filhos do elemento com o ID "container":

**const** container = document.**getElementById**('container');

**const** childNodes = container.childNodes;

console.**log**(childNodes.length); // Saída: 1 (o nó de texto "\n " é considerado um nó filho)

console.**log**(childNodes[0].nodeName); // Saída: "#text"

console.**log**(childNodes[1].nodeName); // Saída: "P"

## **Propriedade**[nextElementSibling](https://developer.mozilla.org/en-US/docs/Web/API/Element/nextElementSibling" \t "_blank)

A propriedade nextElementSibling permite acessar o próximo irmão (nó adjacente) de um elemento no DOM.

**Exemplo de uso:**

Vamos considerar o seguinte HTML:

<**ul**>

<**li**>Item 1</**li**>

<**li**>Item 2</**li**>

<**li**>Item 3</**li**>

</**ul**>

Agora, podemos usar o **nextElementSibling** para acessar o próximo irmão de um elemento **<li>:**

**const** item1 = document.**querySelector**('li:first-child');

**const** item2 = item1.nextElementSibling;

console.**log**(item2.textContent); // Saída: "Item 2"

## **Propriedade**[previousElementSibling](https://developer.mozilla.org/en-US/docs/Web/API/Element/previousElementSibling" \t "_blank)

A propriedade previousElementSibling é semelhante ao nextElementSibling, mas permite acessar o irmão anterior (nó adjacente) de um elemento no DOM.

**Exemplo de uso:**

Continuando o exemplo anterior, vamos usar o previousElementSibling para acessar o irmão anterior do elemento <li> que selecionamos:

**const** item3 = document.**querySelector**('li:last-child');

**const** item2 = item3.previousElementSibling;

console.**log**(item2.textContent); // Saída: "Item 2"

Com essas propriedades, é possível navegar, acessar e modificar elementos HTML em uma página da web, tornando a manipulação do DOM uma tarefa poderosa para pessoas desenvolvedoras web.

É importante continuar praticando e explorando o DOM para aprofundar seu conhecimento e habilidades em JavaScript.