

Criação de nós no DOM



**Certified Tech
Developer**

The Ultimate Degree



Para criar nós no HTML com JavaScript no DOM, precisamos usar métodos específicos.



Métodos do objeto document

| Método | |
|---------------------|---|
| 1. createElement() | Cria um nó do elemento com o tipo de acordo com o nome da tag HTML que indicamos. |
| 2. createTextNode() | Cria um nó de texto explicando entre aspas. Não é exibido até que seja atribuído a um elemento DOM existente. |
| 3. appendChild() | Anexa um elemento filho a um elemento pai no DOM. Se o elemento pai já existia no documento, ele muda sua posição para o outro elemento pai indicado. Se não existir, criamos com o método 1. |

Sintaxe para criar um nó de elemento

Objeto

Aqui especificamos o objeto do documento como um prefixo. Sem esses dados, o nó não será criado, portanto deve sempre ser usado.

Elemento

Aqui, vamos colocar o nome do nó que queremos criar no Dom. Por exemplo: "button", "div", "section", etc.



```
js document.createElement("input");
```

Método

Aqui, vamos definir o método que usamos para criar o nó de acordo com o tipo que precisamos. Neste caso, createElement para criar uma tag de entrada.

Sintaxe para criar um nó de texto

Objeto

Como mencionamos no caso anterior, esse prefixo deve sempre ser incluído na sintaxe.

Texto

Aqui, vamos colocar um texto que será refletido em um elemento existente do DOM.



```
js document.createTextNode("Olá Mundo");
```

Método

Neste caso, o método que usamos é createTextNode para mostrar que será um nó de texto.

Sintaxe para anexar um elemento filho ao DOM

Objeto - elemento pai

Neste caso, é incorporado imediatamente após o corpo. Também poderíamos, por exemplo, seleccionar o elemento pai pelo seu ID.

Variável - elemento filho.

É essencial colocar a variável onde armazenamos anteriormente o elemento filho seleccionado ou criado para ser posicionado no documento sem aspas.

```
js document.body.appendChild("titulo");
```

Método

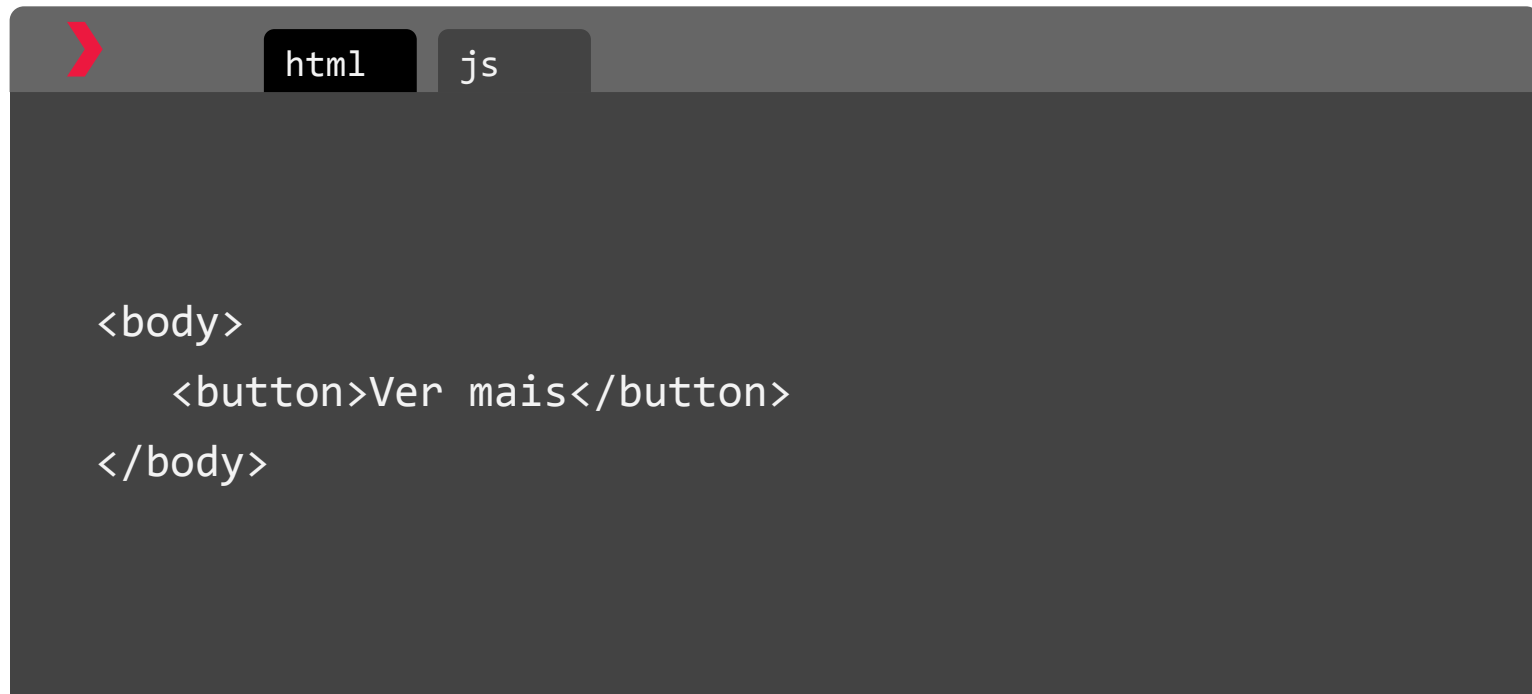
Graças a este método, podemos atribuir o elemento filho a um elemento pai e visualizar as alterações na tela do navegador.

Vamos ver um exemplo:



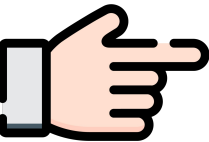
```
var botaoVerMais = document.createElement("button")
var botaoTexto = document.createTextNode("Ver mais")
botaoVerMais.appendChild(botaoTexto)
document.body.appendChild(botaoVerMais)
```

Resultado no DOM:

A code editor interface with a dark gray background. At the top, there is a tab bar with two tabs: 'html' (selected) and 'js'. To the left of the tabs is a red arrow icon pointing right. The main area of the editor displays the following HTML code:

```
<body>  
  <button>Ver mais</button>  
</body>
```


Vamos analisar por partes: createElement()



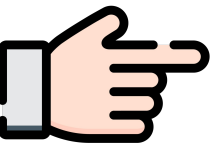
```
var botaoVerMais =  
document.createElement("button")
```

Armazenamos em uma variável **"botaoVerMais"** A criação de um botão.

```
var botaoTexto =  
document.createTextNode("Ver mais")  
  
botaoVerMais.appendChild(botaoTexto)  
  
document.body.appendChild(botaoVerMais)
```

createTextNode()

```
var botaoVerMais =  
document.createElement("button")
```



```
var botaoTexto =  
document.createTextNode("Ver mais")
```

Armazenamos na variável **"botaoTexto"** a criação de um nó de texto que diz "Ver mais".

```
botaoVerMais.appendChild(botaoTexto)
```

```
document.body.appendChild(botaoVerMais)
```

appendChild()

```
var botaoVerMais =  
document.createElement("button")
```

```
var botaoTexto =  
document.createTextNode("Ver mais")
```



```
botaoVerMais.appendChild(botaoTexto)  
document.body.appendChild(botaoVerMais)
```

Anexamos o nó de texto criado anteriormente ao nosso botão acessando com sua variável. E, em seguida, adicionamos o botão ao corpo do documento.

DigitalHouse>
Coding School