

Node.removeChild

Sumário

Remove um nó filho do DOM. Devolve o nó removido.

Sintaxe

```
var filhoRemovido = elemento.removeChild(filho);  
elemento.removeChild(filho);
```

- `filho` é o nó filho a ser removido do DOM.
- `elemento` é o nó pai de `filho`.
- `filhoRemovido` contém uma referência ao nó filho removido. `filhoRemovido === filho`.

O nó filho removido ainda existe em memória, mas não é mais parte do DOM. Você pode reutilizar o nó removido mais tarde no seu código por meio da referência `filhoRemovido`.

Se `filho` não for um filho do nó `elemento`, o método lança uma exceção. Isto também acontecerá se `filho` era, de fato, um filho de `elemento` no momento da chamada, mas foi removido por um manipulador de eventos invocado enquanto o `elemento` estava sendo removido (por exemplo, `blur`).

Exemplos

```
<!--Código HTML de exemplo-->  
  
<div id="topo" align="center">  
  <div id="interno"></div>  
</div>
```

```
// Removendo um elemento específico quando se conhece seu pai  
var d = document.getElementById("topo");  
var d_interno = document.getElementById("interno");  
var noRemovido = d.removeChild(d_interno);
```



```
// Removendo um elemento específico sem precisar especificar seu r
```

```
// Removendo todos os nós filhos de um elemento
var elemento = document.getElementById("topo");
while (elemento.firstChild) {
  elemento.removeChild(elemento.firstChild);
}
```



Especificação

- [DOM Level 1 Core: removeChild](#)
- [DOM Level 2 Core: removeChild](#)
- [DOM Level 3 Core: removeChild](#)

Veja também

- [Node.replaceChild](#)
- [Node.parentNode](#)

Last modified: 8 de jul. de 2021, [by MDN contributors](#)