

JS DOM

1. removeChild

O elemento filho é removido, mas não permanente, pode-se utilizar ele mais tarde fazendo referência a ele.

“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).” developer.mozilla.

```
<!--Código HTML de exemplo-->
```

```
<div id="topo" align="center">
  <div id="interno"></div>
</div>
```

JS

```
// 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);
```

JS

```
// Removendo um elemento específico sem precisar especificar seu pai
var no = document.getElementById("interno");
if (no.parentNode) {
  no.parentNode.removeChild(no);
}
```

JS

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

2. replaceChild

Substitui o elemento filho que foi especificado por outro.

- newChild é o novo elemento que será inserido no lugar do oldChild. Se já existir no DOM, será removido primeiro para depois ser inserido.
- oldChild é o elemento existente que será substituído.
- replacedNode é elemento substituído. É o mesmo elemento que oldChild.

```
// <div>
//   <span id="childSpan">foo bar</span>
// </div>

// Cria um novo elemento vazio
// sem ID, atributos ou conteúdo
var sp1 = document.createElement("span");

// Adiciona um ID 'newSpan' para o elemento
sp1.setAttribute("id", "newSpan");

// Adiciona conteúdo para o elemento
var sp1_content = document.createTextNode("new replacement span element.");

// Coloca o conteúdo no elemento
sp1.appendChild(sp1_content);

// Procura o elemento que será substituído
var sp2 = document.getElementById("childSpan");
var parentDiv = sp2.parentNode;

// Substituí o elemento que já existe (sp2) por o novo elemento (sp1)
parentDiv.replaceChild(sp1, sp2);

// resultado:
// <div>
//   <span id="newSpan">new replacement span element.</span>
// </div>
```

3. InsertBefore

Insere um nó antes do nó de referência como um filho de um nó pai especificado. Se o filho especificado for uma referência a um nó existente no documento, insertBefore() o moverá de sua posição atual para a nova posição (não há necessidade de remover o nó de seu nó pai antes de anexá-lo a outro nó).

```
// Guarda a referências do elemento no quela nóe queremos inserir o novo nó
var elementoPai = document.getElementById("elementoPai");
// Guarda a referência do primeiro filho
var primeiroFilho = elementoPai.firstChild;

// Cria um novo elemento
var novoElemento = document.createElement("div");

// Insere o novo elemento antes do primeiro filho
elementoPai.insertBefore(novoElemento, primeiroFilho);
```