**el método appendChild() se utiliza para agregar un nodo como hijo a otro nodo existente en el DOM. Es especialmente útil para añadir elementos HTML dinámicamente a una página web.**

**<body>**

<div id="miDiv">

<!-- Aquí se quiere añadir un elemento -->

</div>

<script src="mi\_script.js"></script>

**</body>**

mi\_script.js

// 1.- Obtener una referencia al div en el que se quiere agregar el párrafo

var miDiv = document.getElementById('miDiv');

// 2.- Crear un nuevo elemento de párrafo:

var nuevoParrafo = document.createElement('p');

// 3.- Crear un nodo de texto (contenido del párrafo)

var texto = document.createTextNode('Este es el párrafo añadido con appendChild');

// 4.-Agregar el nodo de texto(3) al párrafo(2)

nuevoParrafo.appendChild(texto);

// 5.-Agregar el párrafo como hijo al div

miDiv.appendChild(nuevoParrafo);

**replaceChild() es un método que se utiliza para reemplazar un nodo hijo existente por otro nodo.**

**<body>**

<div id="miDiv">

<p id="parrafoInicial">Este es el párrafo inicial.</p>

</div>

<button onclick="reemplazarParrafo()">Reemplazar Párrafo</button>

<script src="mi\_script.js"></script>

**</body>**

mi\_script.js

function reemplazarParrafo() {

// 1.- Obtener una referencia al div contenedor

var miDiv = document.getElementById('miDiv');

// 2.- Crear un nuevo párrafo

var nuevoParrafo = document.createElement('p');

nuevoParrafo.textContent = 'Este es el párrafo nuevo';

// 3.- Obtener una referencia al párrafo inicial

var parrafoInicial = document.getElementById('parrafoInicial');

// 4.- Reemplazar el párrafo antiguo con el nuevo párrafo

miDiv.replaceChild(nuevoParrafo, parrafoInicial);

}

**removeChild() es un método que se utiliza para eliminar un nodo hijo específico de un nodo padre.**

<body>

<div id="miDiv">

<p id="parrafoAEliminar">Este es el párrafo que será eliminado.</p>

</div>

<button onclick="eliminarParrafo()">Eliminar Párrafo</button>

<script src="mi\_script.js"></script>

</body>

mi\_script.js

function eliminarParrafo() {

// 1.- Obtener una referencia al div contenedor

var miDiv = document.getElementById('miDiv');

// 2.- Obtener una referencia al párrafo que se eliminará

var parrafoAEliminar = document.getElementById('parrafoAEliminar');

// 3.- Eliminar el párrafo del div contenedor

miDiv.removeChild(parrafoAEliminar);

}

**Puntos importantes que considerar al trabajar con estos métodos :**

1. Las imprescindibles referencias se pueden obtener con métodos como `getElementById()`, `querySelector()`, `getElementsByClassName()`…

2. Son métodos aplicados a los nodos padres para manipular sus nodos hijos.

3. Creación de nuevos elementos: Antes de añadir o reemplazar elementos, es necesario crear los nuevos nodos que se desean insertar en el DOM. Por ejemplo, dinámicamente con `document.createElement()`.

4. Se deben guardar referencias a los nodos agregados, reemplazados o eliminados para poder trabajar con ellos más adelante.

5. Impacto en la estructura del DOM. Estos métodos afectan la estructura del DOM y, por tanto, pueden influir en el flujo y presentación de la página.

6. Optimización y rendimiento: Realizar manipulaciones en el DOM puede afectar al rendimiento. En caso de realizar múltiples cambios en el DOM, es aconsejable manipular los elementos fuera del árbol DOM utilizando fragmentos (`DocumentFragment`) para reducir la cantidad de manipulaciones reales en el DOM y mejorar el rendimiento.