Clase 8

Sitio:	Aprende Programando	Imprimido por:	Mayra Pinciroli
Curso:	Desarrollo Web 2	Día:	domingo, 16 de junio de 2024, 20:57
Libro:	Clase 8		

Tabla de contenidos

1. 📝 **Clase N°8**

2. **¿Qué hace document.createElement()?**

3. **AppendChild:**

3.1. Metodo remove

4. **Ejercicios**

5. **Pregunta de reflexión**

6. **Desafío #8**

1. Clase N°8

¿Qué vamos a trabajar en esta clase?

- Conoceremos en profundidad .createElement.
- Append Child.
- Conoceremos el método remove.

Resultados de aprendizaje:

Al final de esta clase serás capaz de:

- Crear elementos HTML dinámicamente.
- Agregar elementos como hijos de otros.
- Eliminar elementos del DOM.

2. ¿Qué hace document.createElement()?

Es un **método en JavaScript** que se utiliza para **crear un nuevo elemento HTML.** Este método permite crear un elemento específico en el **DOM (Modelo de Objetos del Documento)** y luego agregarlo a la página web.

Es útil cuando necesitas crear elementos dinámicamente en respuesta a eventos o para manipular la estructura del **DOM** durante la ejecución de tu aplicación web.

Si necesitamos crear un nuevo elemento, en HTML, ya no necesitamos recurrir a un archivo HTML y modificar el código. Lo podemos hacer desde JavaScript, como en el siguiente ejemplo, donde:

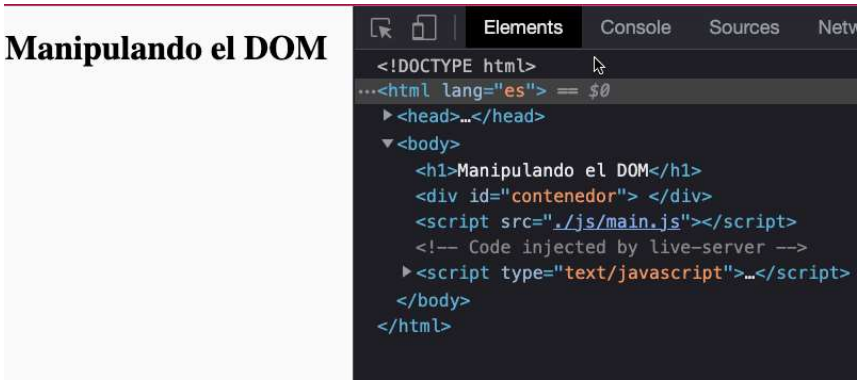
En primer lugar, creamos el elemento al cual llamamos nuevoTexto mediante el método de **createElement**.

En la siguiente línea de código, modificamos el elemento y le asignamos texto, mediante el método innerText. Y, por último,mostramos el elemento creado por la consola.

En código, se ve de la siguiente manera:

```
//Creación del Elemento:
let nuevoTexto = document.createElement("p");
//Personalización del Elemento:
nuevoTexto.innerText = "Este es un Nuevo Parrafo";
//Mostrar el elemento creado en la Consola
console.log(nuevoTexto);
```

Resultado es el siguiente:



El párrafo que creamos todavía no está insertado en el DOM, aunque sí lo podemos ver en la consola. Se dice que el elemento está “desconectado del DOM”: existe pero no tiene conexión con él.

Para comprobar esto utilizamos el método **isConnected** que devuelve true o false.

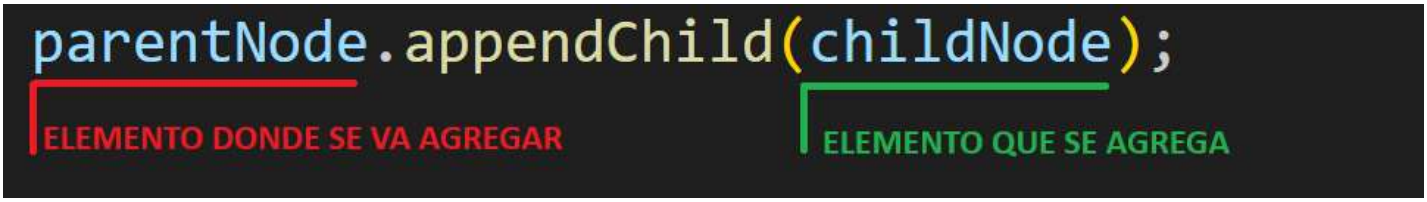
```
//Creación del Elemento:
let nuevoTexto = document.createElement("p");
//Personalización del Elemento:
nuevoTexto.innerText = "Este es un Nuevo Parrafo";
//Mostrar el elemento creado en la Consola
console.log(nuevoTexto);
//Compruebo si el Elemento creado esta Conectado
console.log(nuevoTexto.isConnected);
```

Como vemos nos devuelve **false** porque no está conectado, entonces para conectarlo al DOM tenemos que añadirlo dentro de algún lugar dentro del body. Para esto usaremos el método **appendChild**

3. AppendChild:

Este método se utiliza para **agregar un nuevo elemento (nodo) como hijo de otro elemento existente** en el DOM (Documento de Objetos del Modelo). Lo importante a tener en cuenta con este método es que el nuevo elemento que se agregue se suma al final de una lista de elementos contenidos en un “padre".

Vamos a ver como es la estructura:




```
parentNode.appendChild(childNode);
```

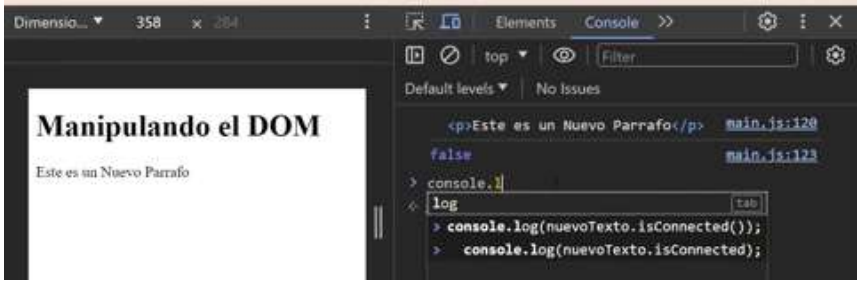
Si queremos agregar el elemento directamente dentro del body:

```
//Creación del Elemento:
let nuevoTexto = document.createElement("p");

//Personalización del Elemento:
nuevoTexto.innerText = "Este es un Nuevo Parrafo";

//Mostrar el elemento creado en la Consola
console.log(nuevoTexto);

//Compruebo si el Elemento creado esta Conectado
console.log(nuevoTexto.isConnected);

//Agregar al DOM (ACA LO AGREGAMOS DENTRO DEL BODY):
document.body.appendChild(nuevoTexto);
```

```
//Creación del Elemento:
let nuevoTexto = document.createElement("p");
//Personalización del Elemento:
nuevoTexto.innerText = "Este es un Nuevo Parrafo";
//Mostrar el elemento creado en la Consola
console.log(nuevoTexto);
//Compruebo si el Elemento creado esta Conectado
```

```
console.log(nuevoTexto.isConnected);
//Agregar al DOM (ACA LO AGREGAMOS DENTRO DEL BODY) :
document.body.appendChild(nuevoTexto);
```

Vemos el resultado y comprobamos si está conectado:



Otro ejemplo si añadimos un elemento dentro de un lugar que no sea el body, en este caso lo añadiremos dentro de un div:

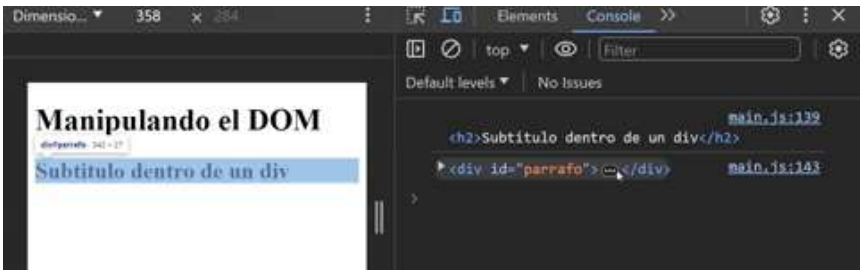
HTML

```
</head>
<body>
    <h1>Manipulando el DOM</h1>
    <div id="parrafo"></div>
    <script src="main.js"></script>
</body>
</html>
```

JS

```
//Creación del Elemento:
let subtitulo = document.createElement("p");
//Personalización del Elemento:
subtitulo.innerText = "Subtitulo dentro de un div";
console.log(subtitulo); //Mostrar el elemento creado en la Consola
//Seleccion del Elemento
let caja = document.querySelector("#parrafo");
console.log(caja); //Mostrar elemento por Consola
//Agregar hr dentro del div #parrafo:
caja.appendChild(subtitulo);
```

Este es el resultado:

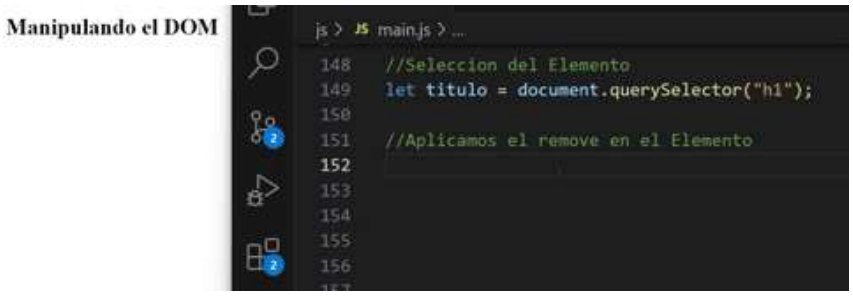


3.1. Metodo remove

Se utiliza para eliminar un elemento del DOM. Este método **permite eliminar un elemento específico** de la página web. Aquí tienes un ejemplo básico de cómo puedes usar el método **remove():**

```
//Seleccion del Elemento
let titulo = document.querySelector("h1");
//Aplicamos el remove en el Elemento
titulo.remove();
```

Este es el resultado:



4. Ejercicios

- **Multiple Choice:**
- **¿Qué hace el método document.createElement() en JavaScript?**
  - **A) Elimina un elemento del DOM.**
  - **B) Crea un nuevo elemento HTML.**
  - **C) Modifica el contenido de un elemento existente.**
- **Respuesta Correcta: B) Crea un nuevo elemento HTML.**
- **Multiple Choice:**
- **¿Cuál de los siguientes métodos se utiliza para agregar un nuevo elemento como hijo de otro elemento existente en el DOM?**
  - **A) createElement**
  - **B) appendChild**
  - **C) remove**
- **Respuesta Correcta: B) appendChild**
- **Verdadero o Falso:**
- **El método remove() se utiliza para eliminar un elemento específico del DOM.**
  - **A) Verdadero**
  - **B) Falso**
- **Respuesta Correcta: A) Verdadero**

5. Pregunta de reflexión

Durante la clase 8 conocimos los métodos de **createElement()**, **appendChild()** y **remove()**.

- ¿Cómo aplicamos estos métodos con el fin de modificar el DOM y mejorar tu página web? 🤔

- ¿Cuándo sería más útil utilizar createElement en lugar de modificar directamente el HTML en un archivo estático? 🤔

6. Desafío #8

Declarar un array con nombres de productos y recorrerlo.

Por cada producto, crear una etiqueta **<h2>** que contenga el nombre, y agregarla al body.

Se recomienda recorrer el array con for of.

En cada repetición se crea un elemento con **createElement**, se modifica su **innerHTML**, y se lo agrega con **appendChild**.