

Guía Teórica N°04

Creación, Obtención, Modificación y Eliminación de elementos

Ha llegado el turno de agregarle más dinamismo a nuestra página web y para ello necesitamos saber cómo crear, modificar y eliminar los diferentes elementos que vayan conformando a nuestro documento de manera dinámica.

Si el usuario, por ejemplo, presiona un botón, ¿qué sucederá exactamente? Basándonos en esta idea, hablaremos próximamente sobre los eventos y los manejadores de eventos, pero antes de llegar a ello, vamos a crear base ante el dinamismo de los elementos y para ello contamos con un conjunto de herramientas que nos ayudarán en esta tarea.

Creación de elementos:

Vamos primero con un detalle muy interesante y cabe mencionar que este orden no será el mismo que se encuentra en los vídeos explicativos, así que puede que te estés adelantando un poco con relación a las clases, aprendamos a cómo crear un nuevo elemento y nos basaremos de los siguientes métodos:

- `createElement()`: Como su nombre lo indica, nos permite crear un nodo nuevo de tipo **element**, cuyo parámetro de entrada es una cadena de caracteres en MAYUSCULA haciendo referencia al tipo de elemento que queremos crear, por ejemplo, "H1", "P", "FORM", "INPUT", entre otros.
- `createTextNode()`: Nos permite crear un nuevo nodo de texto cuyo parámetro de entrada será una cadena de caracteres que indicará el texto a colocar dentro del nodo.

- `createDocumentFragment()`: La creación de muchos elementos a la vez puede ser una carga muy pesada al momento de ingresar a nuestra página web, para ello, en las últimas versiones del estándar **ECMAScript 6**, nos encontramos con este nuevo método que nos ayuda a aligerar esta carga de trabajo.
- `appendChild()`: Ya teniendo el elemento creado, lo que falta es agregarlo a nuestro documento como tal y para ello utilizamos el método `appendChild()`.

¿Qué te parece si vemos un ejemplo que nos ayude a entender todo esto?

- Código HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript y el DOM</title>
</head>
<body>
  <div id="contenedor">
    <!--Aquí agregaremos un párrafo-->
  </div>
<script src="./script.js"></script>
</body>
</html>
```

- Código JavaScript:

```
//Seleccionamos el contenedor
const contenedor = document.getElementById('contenedor');
//Creamos un nuevo nodo elemento de tipo P (párrafo)
const nuevoElemento = document.createElement('P');
//Creamos un nuevo nodo de texto para el párrafo creado
const texto_nuevoElemento = document.createTextNode('Hola Mundo');
//Agregamos el nodo de texto al párrafo creado
nuevoElemento.appendChild(texto_nuevoElemento);
//Finalmente, agregamos el párrafo al contenedor de nuestro HTML
contenedor.appendChild(nuevoElemento);
```

Modificación de elementos:

Ya vimos la manera que tenemos para crear nuevos elementos, pero ahora es turno de saber cómo podemos modificar un poco esos elementos nuevos o incluso los ya creados directamente en nuestro documento HTML. Para ello, haremos uso de algunos atributos importantes luego de seleccionar el elemento que queremos modificar.

- **textContent:** Este representa el contenido del texto de un nodo y sus descendientes.
- **innerText:** La propiedad `innerText` de la interfaz `HTMLElement` representa el contenido de texto "renderizado" de un nodo y sus descendientes. Como captador, se aproxima al texto que obtendría el usuario si resaltara el contenido del elemento con el cursor y luego lo copiara en el portapapeles.
- **innerHTML:** Devuelve o establece la sintaxis HTML describiendo los descendientes del elemento.

- `outerHTML`: El atributo `outerHTML` de la interfaz DOM `element` obtiene el fragmento HTML serializado que describe el elemento incluyendo sus descendientes. Puede ser asignado para reemplazar el elemento por los nodos resultantes de examinar de la cadena proporcionada.

Elementos Hijos (Childs):

La interfaz que nos ofrece JavaScript para la manipulación del Modelo de Objetos del Documento nos permite también manejar los elementos hijos de los diferentes nodos. Recordemos que los elementos hijos son aquellos descendientes que nos encontramos presentes dentro de una etiqueta HTML.

Para esta manipulación veremos algunos atributos y métodos que nos ayudarán para esta tarea.

- `firstChild`: Atributo que nos retorna el primer hijo de un nodo.
- `lastChild`: Atributo que nos retorna el último hijo de un nodo.
- `firstElementChild`: Atributo que nos indica el primer nodo de tipo **element** de un nodo padre.
- `lastElementChild`: Atributo que nos indica el último nodo de tipo **element** de un nodo padre.
- `childNodes`: Nos retorna todos los nodos hijos pertenecientes a un nodo padre.
- `children`: Este nos retorna todos los elementos hijos de un elemento padre en forma de un arreglo y podemos recorrerlo a través de un **for...of** o un **for...in**
- `replaceChild()`: Método que nos proporciona la posibilidad de reemplazar un elemento hijo por un nuevo elemento creado dinámicamente.
- `removeChild()`: Método que nos permite eliminar un elemento hijo de un elemento padre.