

# Manipulación del DOM con JavaScript

En esta práctica vamos a trabajar con distintos métodos para manipular el DOM (Document Object Model) en un archivo HTML. Exploraremos cómo añadir y eliminar elementos de una lista, modificar el contenido de un título, agregar o eliminar filas de una tabla, trabajar con clases, y cambiar estilos en línea. También aprenderemos a recorrer selectores que retornan múltiples elementos, y a usar los diferentes métodos de iteración en JavaScript.

## Teoría necesaria

## 1. Selectores

Función	Descripción	Elementos que puede devolver
<code>document.getElementById("id")</code>	Selecciona un elemento por su <code>id</code> .	Uno
<code>document.getElementsByTagName("tagName")</code>	Selecciona todos los elementos con la etiqueta indicada.	Varios
<code>document.getElementsByClassName("className")</code>	Selecciona todos los elementos con una clase dada.	Varios
<code>document.querySelector("selector")</code>	Selecciona el primer elemento que coincida con el selector CSS.	Uno
<code>document.querySelectorAll("selector")</code>	Selecciona todos los elementos que coincidan con el selector CSS.	Varios

## 2. Modificación de contenido HTML y texto

Método	Descripción
1. Observación directa	El investigador observa y registra el comportamiento de los sujetos en su entorno natural.
2. Entrevistas	Se realiza una conversación estructurada o no estructurada con los sujetos para obtener información sobre sus experiencias y opiniones.
3. Encuestas	Se distribuyen cuestionarios o cuestionarios para recoger datos cuantitativos de una muestra representativa de la población.
4. Experimentación	Se manipula una variable independiente en un entorno controlado para observar su efecto sobre una variable dependiente.
5. Análisis de contenido	Se analizan documentos, textos o grabaciones para identificar temas o categorías recurrentes.
6. Análisis de discurso	Se estudia el uso del lenguaje en contextos comunicativos para comprender los procesos de construcción de significado.
7. Análisis de redes	Se estudia la estructura de las relaciones entre individuos o entidades para comprender patrones de interacción.
8. Análisis de datos secundarios	Se utilizan datos ya existentes, como registros médicos o estadísticas gubernamentales, para responder a preguntas de investigación.
9. Análisis de datos cualitativos	Se utilizan técnicas como el análisis de temas o el análisis de discurso para interpretar datos no cuantitativos.
10. Análisis de datos cuantitativos	Se utilizan técnicas estadísticas para analizar datos numéricos y establecer relaciones entre variables.

<code>element.innerHTML = "nuevo contenido"</code>	Cambia el contenido HTML interno de un elemento.
<code>element.outerHTML = "nuevo contenido"</code>	Cambia el contenido HTML, incluyendo el propio elemento.
<code>element.innerText = "nuevo texto"</code>	Cambia el texto interno de un elemento.

---

### 3. Manipulación de clases

Método	Descripción
<code>element.classList.add("className")</code>	Añade una o más clases al elemento.
<code>element.classList.remove("className")</code>	Elimina una o más clases del elemento.
<code>element.classList.toggle("className")</code>	Añade la clase si no está, y la elimina si ya está.
<code>element.classList.contains("className")</code>	Devuelve <code>true</code> si el elemento tiene la clase, y <code>false</code> si no.

---

### 4. Modificación de atributos y estilos

Método	Descripción
<code>element.getAttribute("atributo")</code>	Obtiene el valor de un atributo.
<code>element.setAttribute("atributo", "valor")</code>	Establece o modifica el valor de un atributo.
<code>element.style.propiedadCSS = "valor"</code>	Modifica el estilo en línea de un elemento (e.g. <code>element.style.backgroundColor = "red"</code> ).

---

### 5. Métodos de iteración

Método	Descripción
<code>for (let i = 0; i &lt; elements.length; i++)</code>	Itera sobre un conjunto de elementos de forma clásica.
<code>for (let element of elements)</code>	Itera sobre un conjunto de elementos de forma simplificada.
<code>elements.forEach(function(element))</code>	Itera sobre un conjunto de elementos y ejecuta una función.

---

## Ejercicio práctico

### Tareas a realizar

#### Modificar el título principal

- Utilizando `document.getElementById`, selecciona el título con id `titulo-principal` y modifica su contenido usando `innerText` y luego `outerHTML`. Muestra los resultados en consola.

#### Agregar una tarea a la lista

- Selecciona la lista con `document.querySelector` y añade un nuevo elemento `li` a través de `innerHTML`. Verifica si el elemento se agregó correctamente con un `console.log`.

#### Agregar una fila a la tabla

- Selecciona la tabla y añade una nueva fila. Usa `classList` para agregar clases alternadas (par/impar) y luego muestra la tabla actualizada en consola.

#### Eliminar la última fila de la tabla

- Usa `querySelectorAll` para seleccionar las filas de la tabla y elimina la última. Muestra el resultado en consola.

## Marcar una fila de la tabla

- Modifica una fila específica para simular que está "marcada" usando `classList.toggle`.

## Recorrer y verificar múltiples elementos

Usa un bucle `forEach` para recorrer todos los elementos de la clase `tarea` en la lista y mostrar su contenido en consola:

Recuerda que los nodos devueltos por `querySelectorAll` y otros métodos similares pueden ser convertidos a arrays si necesitas aplicar más métodos de array como `map`.

## Inventa algún otro tipo de acción javascript

Usando lo aprendido en el curso durante el episodio “Acceso y modificación de los elementos”, realiza otros cambios que consideres oportunos y explícalos.

Puedes crear clases extra en el fichero `.css` o declarándolas inline, y también puedes añadir nuevos elementos al esqueleto `.html`.