**Aplicación de Tecnologías Web**

Tema n.° 7

Manipulación del DOM

**Índice**

**Pág.**

[Introducción 3](#_Toc161321525)

[1.7. DOM 3](#_Toc161321526)

[1.7.1. Estructura jerárquica 3](#_Toc161321527)

[1.7.2. Selección de elementos 4](#_Toc161321528)

[1.7.3. Eventos del DOM 6](#_Toc161321529)

[1.7.4. Manipulación de contenido 8](#_Toc161321530)

[Recursos complementarios 10](#_Toc161321531)

[Referencias 10](#_Toc161321532)

[Autoevaluación 12](#_Toc161321533)

# Introducción

La manipulación del Document Object Model (DOM) es esencial en el desarrollo web moderno. El DOM proporciona una interfaz estructurada para representar documentos HTML y XML, permitiendo a los desarrolladores interactuar dinámicamente con los elementos de una página.

En esta unidad, exploraremos a fondo los métodos, elementos y eventos del HTML DOM en JavaScript.

# DOM

El DOM es una representación en memoria de la estructura de un documento HTML. Cada elemento en la página es un nodo en este árbol de objetos, y el DOM proporciona una interfaz que permite a los programadores manipular estos nodos con JavaScript.

## Estructura jerárquica

Cada documento tiene un nodo raíz que representa todo el documento (document). Los nodos se organizan en una estructura de árbol, donde cada nodo tiene padres, hijos y hermanos.

**Figura 1.**

Ejemplo de estructura del DOM.



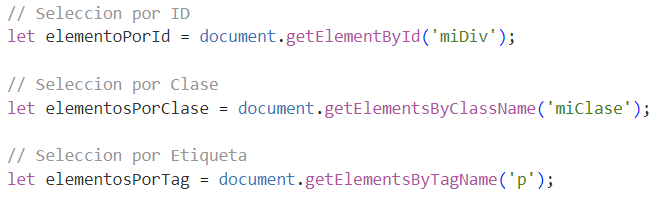
Cada parte de un documento HTML, como etiquetas, atributos y texto, se representa como un nodo en el DOM. Existen diferentes tipos de nodos, incluyendo elementos, atributos, y texto.

## Selección de elementos

Para interactuar con el DOM, es crucial seleccionar elementos de la página. Se utilizan métodos como getElementById, getElementsByClassName, y getElementsByTagName para esto.

**Figura 2.**

Selección de Elementos del DOM.



* **getElementById:** Este método permite seleccionar un elemento por su ID.
* **getElementsByClassName:** Selecciona elementos por su clase y devuelve una colección de nodos.
* **getElementsByTagName:** Selecciona elementos por su nombre de etiqueta y devuelve una colección de nodos.
* **querySelector:** Selecciona el primer elemento que coincida con un selector CSS.
* **querySelectorAll:** Selecciona todos los elementos que coincidan con un selector CSS y devuelve una NodeList.
* **createElement:** Crea un nuevo elemento HTML.
* **removeChild:** Elimina un hijo de un elemento.
* **innerHTML y textContent:** Permite obtener o establecer el contenido HTML o texto de un elemento.
* **setAttribute y removeAttribute:** Añade o elimina atributos de un elemento.

En la siguiente figura se presentan los métodos utilizados para la manipulación del DOM.

**Figura 3.**

Funciones para la manipulación del DOM.



## Eventos del DOM

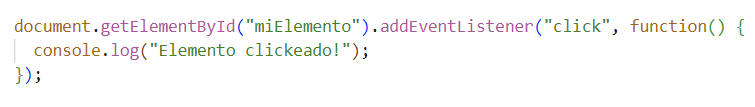
Los eventos en el DOM son acciones o sucesos que ocurren en los elementos HTML y que pueden ser detectados y manejados mediante JavaScript. Estos eventos permiten que una página web reaccione a la interacción del usuario o a cambios en el estado de la aplicación.

***Eventos de Ratón:***

**click:** Se desencadena cuando se hace clic en un elemento.

**Figura 4**

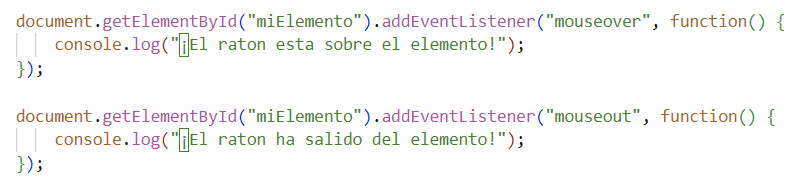
Eventos del ratón sobre el DOM.



**mouseover y mouseout:** Se desencadenan cuando el cursor del ratón entra o sale de un elemento.

**Figura 5**

Eventos mouseover y mouseout

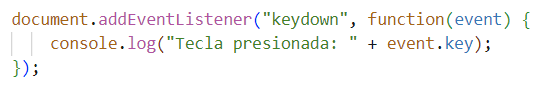


**Eventos de Teclado**

**keydown, keyup y keypress:**Se desencadenan cuando una tecla del teclado es presionada, liberada o pulsada.

**Figura 6**

Eventos keydown, keyup y keypress

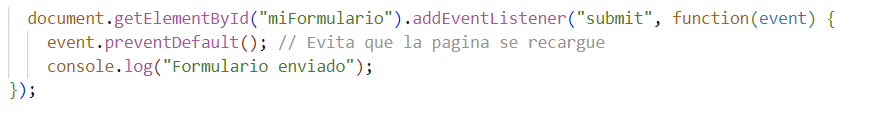


**Eventos de Formulario**

**submit:**Se desencadena cuando se envía un formulario.

**Figura 7**

Evento submit

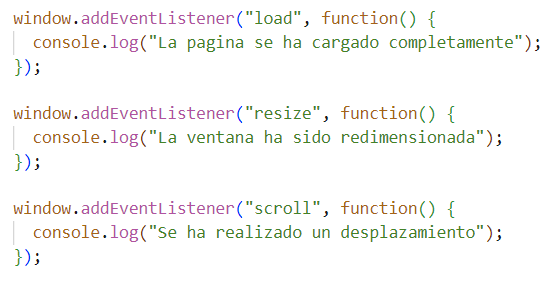


**Eventos de Ventana**

**load, resize, scroll:** Se desencadenan cuando la ventana se carga, se redimensiona o se realiza un desplazamiento.

**Figura 8.**

Eventos load, resize, scroll

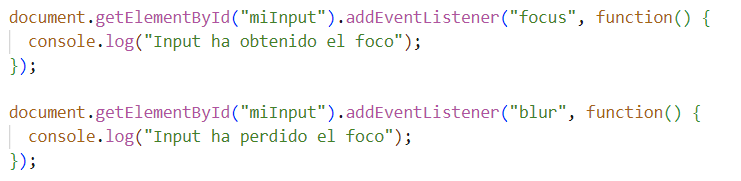


**Eventos de Interacción**

**focus y blur:**Se desencadenan cuando un elemento obtiene o pierde el foco.

**Figura 9**

Eventos focus y blur

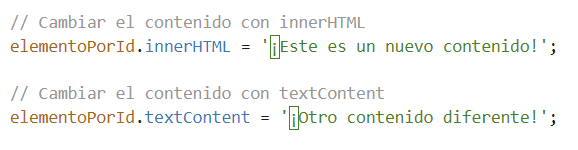


## Manipulación de contenido

Podemos cambiar el contenido de un elemento utilizando propiedades como innerHTML, textContent, o innerText.

**Figura 10.**

Ejemplo de manipulación de contenido del DOM.



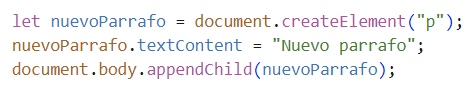
La manipulación del contenido en el DOM (Document Object Model) es esencial para crear páginas web dinámicas e interactivas. JavaScript proporciona varias funciones y métodos para acceder y modificar elementos, contenido, atributos y más. Aquí te presento algunas técnicas comunes de manipulación de contenido:

**Creación y Agregado de Elementos**

**createElement y appendChild:**Puedes crear nuevos elementos y agregarlos al DOM.

Figura 11

Creación y Agregado de Elementos

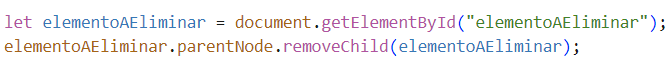


**Eliminación de Elementos**

**removeChild:** Permite eliminar un elemento hijo de otro elemento.

**Figura 12.**

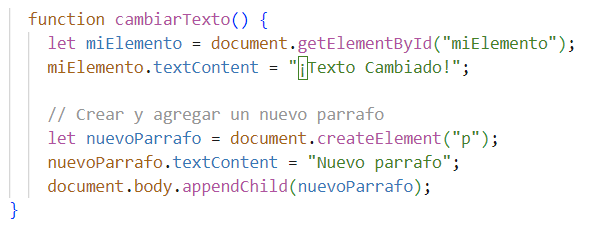
Eliminación de Elementos



**Ejemplo completo**

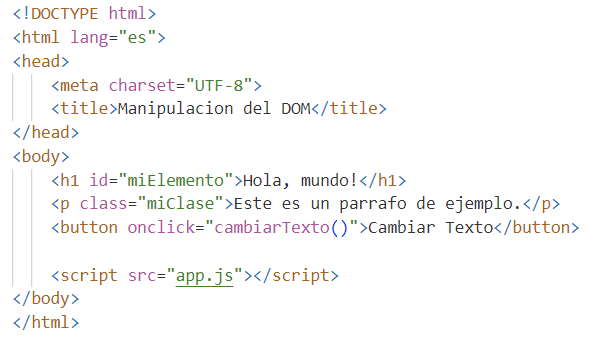
**Figura 13**

Código del archivo app.js



**Figura 14**

Código del archivo index.html



# Recursos complementarios

* Documentación de JavaScript

<https://developer.mozilla.org/es/docs/Web/JavaScript>

# Referencias

Duckett, J. (2014). *JavaScript and jQuery: The Missing Manual (3rd ed.).* O'Reilly Media.

Flanagan, D. (2011). *JavaScript: The Definitive Guide (6th ed.).* O'Reilly Media.

McFarland, D. (2011). *JavaScript & jQuery: The Good Parts.* O'Reilly Media.

Resig, J., & Bibeault, B. (2009). *Secrets of the JavaScript Ninja*. Manning Publications.

Zakas, N. C. (2012). *Professional JavaScript for Web Developers (3rd ed.).* Wrox.

# Autoevaluación

1. **¿Cuál de las siguientes funciones se utiliza para acceder a un elemento por su ID en el DOM?**

getElementByClass

selectElementById

getElementById

accessElementByID

1. **¿Qué propiedad se utiliza para cambiar el contenido HTML de un elemento en el DOM?**

innerText

textContent

innerHTML

outerHTML

1. **¿Cuál es el método que se utiliza para crear un nuevo elemento en el DOM?**

createNode

makeElement

createElement

newElement

1. **¿Cómo se agrega un nuevo elemento al final de un elemento padre en el DOM?**

appendChild

insertAfter

addToEnd

addElement

1. **¿Cuál de los siguientes eventos se desencadena cuando un usuario hace clic en un elemento?**

mouseover

keyup

click

submit

1. **Para obtener el valor de un atributo de un elemento en el DOM, ¿cuál método se utiliza?**

getAttribute

retrieveAttribute

fetchAttribute

obtainAttribute

1. **¿Cuál de las siguientes funciones se utiliza para eliminar un elemento del DOM?**

deleteElement

removeNode

removeChild

eraseElement

1. **¿Qué evento se desencadena cuando se envía un formulario en HTML?**

submit

click

change

keypress

1. **¿Cuál es la función principal de preventDefault() en un manejador de eventos?**

Detener la propagación del evento

Cancelar el evento

Prevenir la recarga de la página

Desactivar el evento temporalmente

1. **¿Qué método se utiliza para agregar un evento a un elemento en el DOM?**

addEventListener

attachEvent

setEvent

eventHandler