

Universidad Tecnológica Metropolitana

Aplicaciones Web

Mtra. Ruth Betsaida Martínez Domínguez

Práctica 1. Manejo y uso de los elementos con el DOM

Karla Patricia Cuevas Cab

3B

Fecha de entrega: 10 de junio de 2024

**Contenido**

[**Introducción** 3](#_Toc169093576)

[**Explicación** 4](#_Toc169093577)

[**Script** 5](#_Toc169093578)

[**Captura de pantalla** 8](#_Toc169093579)

[**Conclusión** 10](#_Toc169093580)

# **Introducción**

El DOM es una forma de representar una página web en un formato que JavaScript puede usar para interactuar con el contenido y la estructura del documento. Con el DOM, puedes cambiar texto, estilos, y agregar o eliminar elementos dinámicamente.

Primero, puedes seleccionar elementos usando métodos como getElementById o querySelector. Esto te permite encontrar elementos específicos en tu página. Una vez que los tienes, puedes cambiar su contenido con propiedades como innerHTML, lo que te permite actualizar texto sin recargar la página.

También puedes cambiar estilos CSS y atributos de los elementos, como cambiar el color de un encabezado o el src de una imagen, usando JavaScript. Esto hace que tu página sea más interactiva y dinámica.

Además, puedes agregar nuevos elementos al documento o eliminar los que ya no necesitas. Por ejemplo, podrías agregar nuevos ítems a una lista o crear un nuevo párrafo.

Otra cosa útil es validar entradas de formulario, como comprobar si un número ingresado está dentro de un rango específico, y dar retroalimentación instantánea.

# **Explicación**

Este código HTML y JavaScript tiene como objetivo principal demostrar varias técnicas y métodos de manipulación del DOM en una página web. A lo largo del documento, se presentan ejemplos prácticos que muestran cómo interactuar con elementos HTML utilizando JavaScript, lo cual es esencial para desarrollar aplicaciones web dinámicas y responsivas.

En la primera parte, se utiliza el método `getElementById` para cambiar el contenido de un párrafo identificado por su ID, mostrando así cómo actualizar el texto de un elemento específico. La segunda parte amplía esta idea al modificar el contenido de un encabezado `<h1>` y cambiar su estilo, lo que demuestra cómo se pueden aplicar estilos dinámicamente a través de JavaScript.

La tercera parte se enfoca en la búsqueda de elementos HTML por su nombre de etiqueta. Aquí, se obtiene una colección de todos los elementos `<p>` y se verifica si un contenedor específico con ID "main" contiene párrafos, ilustrando cómo navegar y acceder a subconjuntos específicos del DOM.

La cuarta parte muestra cómo encontrar elementos por su clase, utilizando `getElementsByClassName` para seleccionar todos los elementos con la clase "intro". La quinta parte avanza un paso más al utilizar selectores de CSS con `querySelectorAll` y `querySelector` para obtener conjuntos más específicos de elementos, como párrafos con una clase particular o elementos de una lista `<ul>`.

En la sexta parte, se explora cómo trabajar con formularios HTML, accediendo a sus elementos y mostrando los valores de entrada en la consola y en la página web. Esto es útil para manejar datos de usuario en aplicaciones web.

La séptima parte ilustra cómo agregar nuevos elementos al DOM, demostrando la creación de una lista y la adición de elementos a esta, lo que es crucial para generar contenido dinámico en respuesta a acciones del usuario.

La octava parte introduce la validación de entradas numéricas, mostrando cómo verificar que un número ingresado por el usuario esté dentro de un rango específico y proporcionando retroalimentación adecuada. La novena parte muestra cómo cambiar atributos de elementos HTML, como la fuente de una imagen.

Finalmente, la décima parte se centra en la generación de contenido dinámico, mostrando la fecha y hora actuales en un párrafo, lo que ejemplifica cómo utilizar JavaScript para actualizar contenido en tiempo real.

# **Script**

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title> Práctica</title>

</head>

<body>

    <p id="p1">¡Hola Mundo!</p>

    <script>

        document.getElementById("p1").innerHTML = "Nuevo Texto";

    </script>

    <h1 id="titulo">Encabezado Viejo</h1>

    <script>

        const elemento = document.getElementById("titulo");

        elemento.innerHTML = "Nuevo Encabezado";

        const h1 = document.querySelector('h1');

        h1.style.color = "blue";

    </script>

    <script>

        const elementosP = document.getElementsByTagName("p");

        console.log(elementosP);

        const main = document.getElementById("main");

        if (main) {

            const psEnMain = main.getElementsByTagName("p");

            console.log(psEnMain);

        } else {

            console.log("No se encontró el elemento con id 'main'");

        }

    </script>

    <script>

        const elementosIntro = document.getElementsByClassName("intro");

        console.log(elementosIntro);

    </script>

    <p class="intro">Este es un ejemplo</p>

    <p class="intro">Este es otro ejemplo</p>

    <ul>

        <li>Elemento 1</li>

        <li>Elemento 2</li>

        <li>Elemento 3</li>

        <li>Elemento 4</li>

        <li>Elemento 5</li>

    </ul>

    <a id="list" href="https://www.youtube.com/">Enlace</a>

    <script>

        const psIntro = document.querySelectorAll("p.intro");

        console.log(psIntro);

        const listas = document.querySelectorAll("ul > li");

        console.log(listas);

        const listElement = document.querySelector("#list");

        console.log(listElement);

        const listaItems = document.querySelectorAll(".list");

        console.log(listaItems);

        const listaItem = document.querySelector(".list");

        console.log(listaItem);

    </script>

    <script>

    let lista = document.createElement("ul");

    document.body.appendChild(lista);

    let elemento1 = document.createElement("li");

    elemento1.textContent = "Flor";

    lista.appendChild(elemento1);

    let elemento2 = document.createElement("li");

    elemento2.textContent = "Jaguar";

    lista.appendChild(elemento2);

    </script>

<form id="frm1">

    <p class="intro">Nombre y Apellido</p>

    <input type="text" name="nombre" id="nombre">

    <input type="text" name="apellido">

    <button type="button" onclick="validarNombre()">Enviar</button>

</form>

<script>

    function validarNombre() {

        var nombre = document.getElementById("nombre").value;

        if (nombre.trim() === "") {

            alert("Por favor, ingresa un nombre válido.");

        } else {

            alert("¡Nombre válido! Puedes proceder.");

        }

    }

</script>

    <script>

        const x = document.forms["frm1"];

        let texto = "";

        for (let i = 0; i < x.length; i++) {

            texto += x.elements[i].value + "<br>";

        }

        document.getElementById("demo").innerHTML = texto;

        console.log(texto);

    </script>

    <script>

        let lista = document.createElement("ul");

        document.body.appendChild(lista);

        let elemento1 = document.createElement("li");

        elemento1.textContent = "Nuevo elemento en la lista";

        lista.appendChild(elemento1);

    </script>

    <h2>Validación de Entrada Numérica</h2>

    <input type="number" id="numero">

    <button type="button" onclick="miFuncion()">Validar</button>

    <p id="demo"></p>

    <script>

        function miFuncion() {

            let x = document.getElementById("numero").value;

            let texto;

            if (isNaN(x) || x < 1 || x > 10) {

                texto = "Entrada no válida";

            } else {

                texto = "Entrada correcta";

            }

            document.getElementById("demo").innerHTML = texto;

        }

    </script>

    <h2>Cambiar el Valor de un Atributo</h2>

    <img id="miImagen" src="im4.jpg" alt="Imagen">

    <script>

        document.getElementById("miImagen").src = "im4.jpg";

    </script>

    <h2>Contenido Dinámico</h2>

    <p id="demo"></p>

    <script>

        document.getElementById("demo").innerHTML = "Fecha: " + new Date();

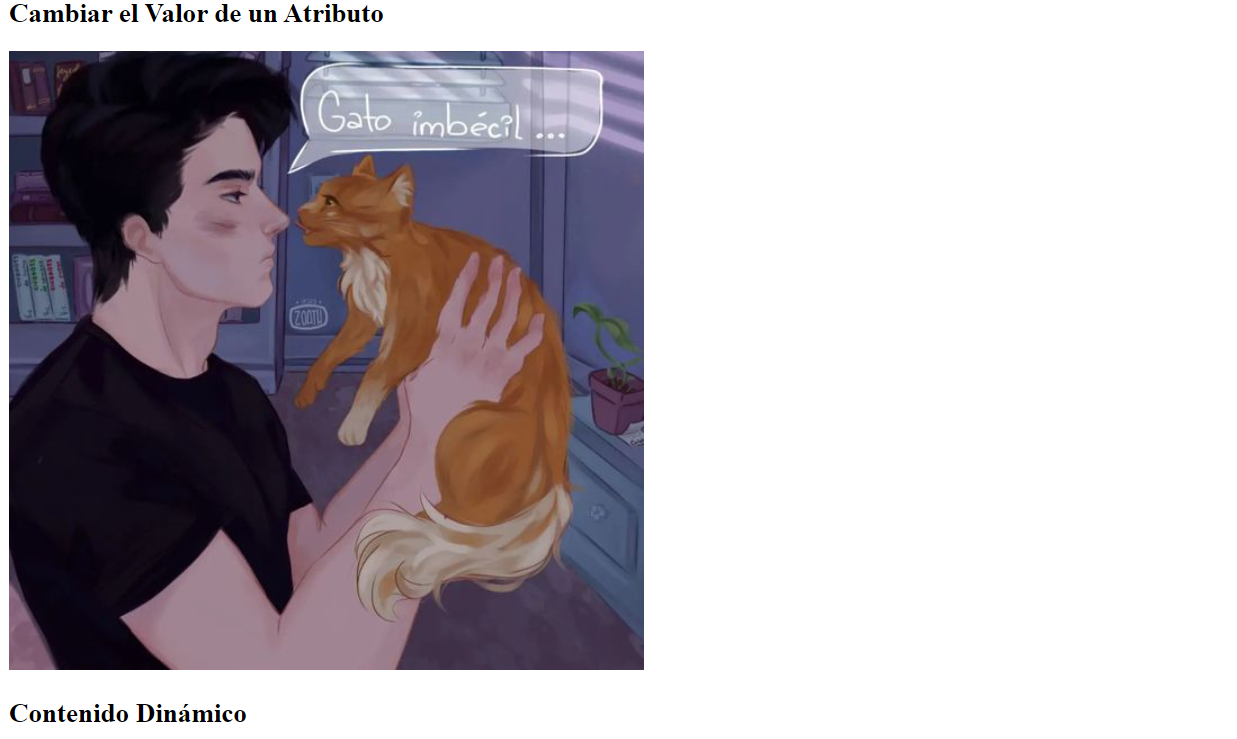
    </script>

</body>

</html>

# **Captura de pantalla**





# **Conclusión**

El manejo y uso de los elementos con el DOM es fundamental para cualquier desarrollador web que quiera crear páginas interactivas y dinámicas. A través de métodos como `getElementById`, `querySelector`, y `getElementsByClassName`, podemos seleccionar y manipular elementos específicos de una página web de manera eficiente. Cambiar contenido, actualizar estilos, agregar nuevos elementos y validar datos son solo algunas de las muchas cosas que podemos hacer usando el DOM. Estas habilidades no solo mejoran la experiencia del usuario, sino que también hacen que nuestras aplicaciones web sean más flexibles y responsivas. Con una comprensión sólida del DOM y su manipulación, los desarrolladores pueden llevar la interactividad y dinamismo de sus sitios web al siguiente nivel.