Introducción al DOM

1. ¿Qué es el DOM?

El Document Object Model (DOM) es una representación estructural de un documento HTML o XML. Permite a los lenguajes de programación, como JavaScript, acceder y manipular la estructura, el estilo y el contenido de las páginas web.

2. Estructura del DOM

El DOM se organiza en una jerarquía de nodos. Cada elemento HTML es un nodo en este árbol. Por ejemplo, un documento HTML básico tiene un nodo raíz <html>, con nodos hijos como <head> y <body>.

Métodos para Acceder a Elementos del DOM

3. getElementById

Este método permite acceder a un elemento específico mediante su atributo id.

const elemento = document.getElementById('miElemento');

4. querySelector y querySelectorAll

querySelector devuelve el primer elemento que coincide con un selector CSS.

querySelectorAll devuelve todos los elementos que coinciden con el selector.

const primerElemento = document.querySelector('.clase'); // Primer elemento con clase 'clase'

**const todosLosElementos = document.querySelectorAll('div'); // Todos los **

Manipulación de Elementos

5. Modificar Contenido

Cambiar el contenido de un elemento usando innerHTML o textContent.

elemento.innerHTML = 'Nuevo contenido'; // Cambia el HTML interno

elemento.textContent = 'Texto simple'; // Cambia solo el texto

6. Modificar Atributos

Cambiar atributos de un elemento utilizando setAttribute o directamente.

elemento.setAttribute('class', 'nueva-clase'); // Cambia la clase

elemento.id = 'nuevoId'; // Cambia el id directamente

7. Estilos CSS

Modificar estilos CSS directamente a través de la propiedad style.

elemento.style.backgroundColor = 'blue'; // Cambia el color de fondo

Creación y Eliminación de Elementos

8. Crear Nuevos Elementos

Usar createElement para crear nuevos nodos y appendChild para añadirlos al DOM.

```
const nuevoDiv = document.createElement('div');
nuevoDiv.textContent = 'Soy un nuevo div';
document.body.appendChild(nuevoDiv); // Añade el nuevo div al final del body
```

9. Eliminar Elementos

Eliminar un elemento del DOM usando removeChild o remove.

```
const divAEliminar = document.getElementById('divEliminar');
divAEliminar.parentNode.removeChild(divAEliminar); // Elimina usando parentNode
// O simplemente:
divAEliminar.remove(); // Elimina directamente
```

Manejo de Eventos

10. Agregar Eventos

Usar addEventListener para manejar eventos como clics.

```
const boton = document.querySelector('button');
boton.addEventListener('click', () => {
  alert('Botón clickeado!');
});
```

Ejemplo completo

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Manipulación del DOM</title>
</head>
<body>
  <h1 id="titulo">Hola Mundo</h1>
  <button id="cambiarTexto">Cambiar Texto</button>
```

```
  <script>
    const boton = document.getElementById('cambiarTexto');
    boton.addEventListener('click', () => {
      const titulo = document.getElementById('titulo');
      titulo.textContent = 'Texto Cambiado!';
      titulo.style.color = 'red';
    });
  </script>
</body>
</html>
```

Este ejemplo muestra cómo cambiar el texto y el color de un encabezado al hacer clic en un botón.

Ejercicios Prácticos

12. Ejercicio 1: Cambiar Estilo

Crear una página donde cambien el color de fondo al hacer clic en diferentes botones.

13. Ejercicio 2: Crear una Lista Dinámica

Crear una lista donde se añadan elementos mediante un formulario y se eliminen al hacer clic en ellos.