

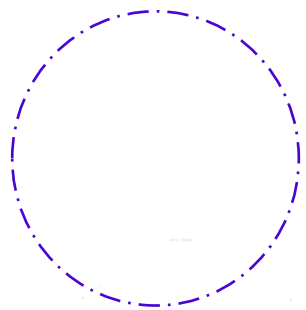


dev Senior JavaScript

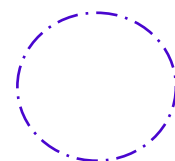
MEAN Mastery de cero a líder Full Stack

www.devseniorcode.com






dev Senior



Módulo 1: Fundamentos del Desarrollo Web y JavaScript (Meses 1 y 2)

Objetivo: Dominar los fundamentos de JavaScript, HTML y CSS, y sentar las bases del desarrollo web.

Clase 5: Manipulación del DOM con JavaScript.

 **Objetivo de la sesión:** Que los estudiantes comprendan qué es el DOM y cómo se puede manipular con JavaScript para crear interfaces web dinámicas e interactivas.

www.devseniorcode.com

🧠 1. Conceptos clave ¿Qué es el DOM?

El DOM (Document Object Model) es una representación en forma de árbol de los elementos HTML de una página web. JavaScript puede acceder y modificar estos elementos a través del DOM.

¿Por qué es importante?

Permite:

- Cambiar contenido dinámicamente.
- Controlar estilos y clases.
- Responder a eventos del usuario.
- Crear, mover o eliminar elementos HTML.

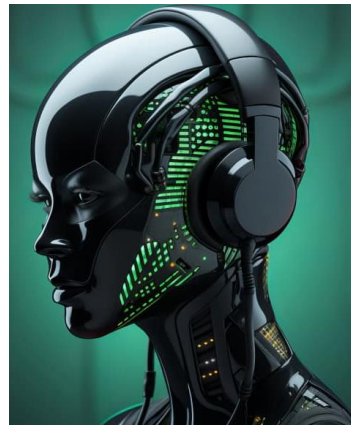




2. Métodos esenciales del DOM en JS

Acción	Método
Seleccionar un elemento	<code>getElementById</code> , <code>querySelector</code>
Seleccionar varios elementos	<code>querySelectorAll</code> , <code>getElementsByClassName</code>
Cambiar texto o HTML	<code>textContent</code> , <code>innerHTML</code>
Cambiar atributos	<code>setAttribute</code> , <code>getAttribute</code>
Cambiar estilos	<code>element.style</code>
Agregar clases	<code>classList.add</code> , <code>classList.remove</code> , <code>toggle</code>
Crear o eliminar elementos	<code>createElement</code> , <code>appendChild</code> , <code>removeChild</code>

www.devseniorcode.com





3. Ejemplo práctico #1: To-Do List interactiva (básica)

```
<!DOCTYPE html>
<html lang="en">

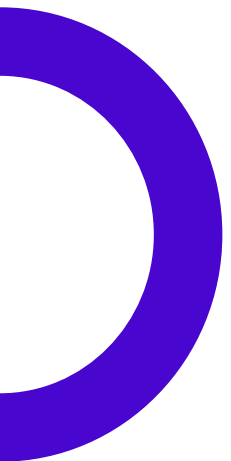
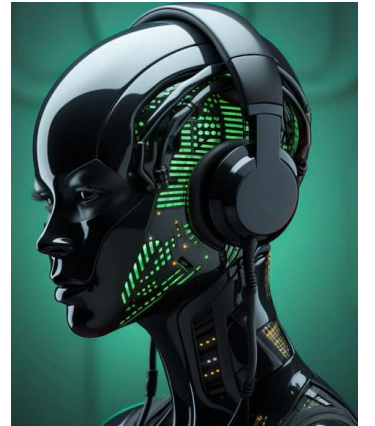
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>To-Do List interactiva (básica)</title>
</head>

<body>
  <input type="text" id="tareaInput" placeholder="Escribe una tarea">
  <button id="agregarBtn">Agregar</button>
  <ul id="listaTareas"></ul>

  <script>
    document.getElementById("agregarBtn").addEventListener("click", function () {
      const input = document.getElementById("tareaInput");
      const nuevaTarea = document.createElement("li");
      nuevaTarea.textContent = input.value;
      document.getElementById("listaTareas").appendChild(nuevaTarea);
      input.value = ""; // limpiar input
    });
  </script>

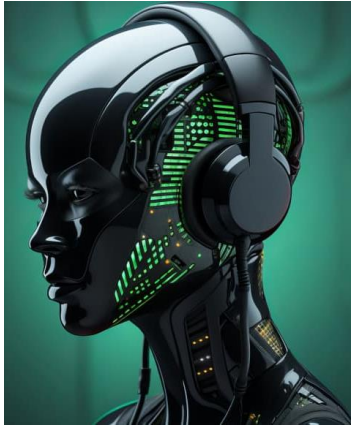
</body>

</html>
```





4. Ejemplo práctico #2: Contador con botones



```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Contador con Botones</title>
</head>

<body>
  <h2>Contador: <span id="contador">0</span></h2>
  <button id="incrementar">+</button>
  <button id="disminuir">-</button>

  <script>
    let valor = 0;
    const contadorSpan = document.getElementById("contador");

    document.getElementById("incrementar").addEventListener("click", function () {
      valor++;
      contadorSpan.textContent = valor;
    });

    document.getElementById("disminuir").addEventListener("click", function () {
      valor--;
      contadorSpan.textContent = valor;
    });
  </script>
</body>

</html>
```





5. Actividad en clase (20-30 min)

Desafío:

Crea una pequeña app que:

- Tenga un input para escribir nombres.
- Al hacer clic en “Añadir nombre”, lo agregue a una lista.
- Si el nombre ya está en la lista, muestre una alerta que diga "Este nombre ya fue agregado".





6. Conclusión y recomendaciones

- El DOM es la puerta de entrada para que JavaScript interactúe con el HTML.
- Todo lo que ves en una página puede ser modificado en tiempo real.
- Usar el DOM correctamente permite crear experiencias más ricas e interactivas.





¿ Dudas e Inquietudes ?



Gracias