**El DOM (Document Object Model)**

**👋 Introducción**

**¡Hola! 👋 Te damos la bienvenida a este nuevo paso.** El DOM (Document Object Model) es el esqueleto invisible detrás de cada página web, una estructura jerárquica que representa todos los elementos y su interrelación en un documento HTML.

Desde la creación de nodos hasta la navegación a través de árboles DOM, exploraremos las herramientas y técnicas esenciales para entender y modificar la estructura de un documento web

 ¡Prepárate para descubrir un nuevo nivel de control sobre tus aplicaciones web!

**📖 Material Descargable**

Aquí te dejamos acceso al material teórico pertinente que te acercará a los conceptos que pondrás en práctica en los próximos pasos.

👉[**Teoría DOM (Document Object Model)**](https://drive.google.com/file/d/1eEK6ZJ6GJhx4I20smi8YgxaD_-3ltfmv/view?usp=drive_link)

# Práctica Iniciando Manejo del DOM

Cómo vimos en la teoría, el DOM (Document Object Model) es la representación interactiva de la página en la web.

En esta sección ejercitaremos cómo Javascript interactúa con HTML y CSS generando dinamismo en las webs o aplicaciones que realizamos.

📌 Para practicar vamos a utilizar como base el siguiente archivo html - [**Link**](https://drive.google.com/drive/folders/1ylSiJIe4TROnoLQRcqCuOVmRQ-npkIbe?usp=drive_link) Les pedimos que descarguen los archivos y los corran en su compu. Deberían ver la siguiente imagen



Para hacer los ejercicios deben crear un archivo **script.js** y linkearlo al **index.html**. En ese archivo es en el cual haremos los ejercicios.

### ✏️ Actividad Modificar elementos I

Supongamos que tienes un arreglo de objetos que representa una lista de productos en una tienda en línea, con la siguiente estructura:

En el siguiente ejercicio se brinda un código javascript que se deberá modificar para que seleccione al botón CHECK!. Si funciona, el mismo deberá cambiar de CHECK! A OFF El código.

// Asignar el elemento correcto a la variable

const buttonElem = //AQUI DEBEN COLOCAR EL SELECTOR

buttonElem.addEventListener("click", () => {

  let oldText = buttonElem.innerText;

  const result = (buttonElem.innerText =

    oldText == "CHECK!" ? "OFF" : "CHECK!");

  return result;

});

Cómo se ve:



**✏️ Actividad Modificar elementos II**

Agregar un id al elemento h1 que contiene el texto “ADIVINA EL NUMERO” y tomando como referencia el ejercicio anterior, cambiar el texto por “JUEGO DE ADIVINAR”. Deberás seleccionar el elemento a través del ID

### ✏️ Actividad Cambiar color de fondo

En este ejercicio, crearemos una función que cambie el color de fondo de un elemento div al hacer clic en un botón, y luego restaure el color de fondo al hacer clic en otro botón.

<!DOCTYPE html>

<html>

<head>

    <title>Ejercicio: Cambio de Color de Fondo</title>

</head>

<body>

    <div id="miDiv" style="width: 200px; height: 100px; background-color: pink;">

        Este es un div con un color de fondo inicial.

    </div>

    <button id="cambiarColorBtn">Cambiar Color de Fondo</button>

    <button id="restaurarColorBtn">Restaurar Color de Fondo</button>

    <script>

        // Función para cambiar el color de fondo del div

        function cambiarColorDeFondo() {

            var miDiv = document.getElementById('miDiv');

            miDiv.setAttribute('style', 'background-color: yellow;');

        }

        // Función para restaurar el color de fondo del div al valor inicial

        function restaurarColorDeFondo() {

            var miDiv = document.getElementById('miDiv');

            miDiv.removeAttribute('style');

        }

        // Agregar eventos a los botones

        document.getElementById('cambiarColorBtn').addEventListener('click', cambiarColorDeFondo);

        document.getElementById('restaurarColorBtn').addEventListener('click', restaurarColorDeFondo);

    </script>

</body>

</html>

En este ejercicio, hemos creado un div con un color de fondo inicial "**pink**". Luego, hemos añadido dos botones: uno para cambiar el color de fondo del div a "**yellow**" y otro para restaurar el color de fondo al valor inicial.

Cuando hagas clic en el botón**"Cambiar Color de Fondo"**, se activará la función **cambiarColorDeFondo()**, que utiliza **setAttribute**() para cambiar el estilo de fondo del div a "**yellow**". Luego, cuando hagas clic en el botón **"Restaurar Color de Fondo**", se activará la función **restaurarColorDeFondo**(), que utiliza **removeAttribute**() para eliminar el estilo de fondo, restaurando así el color de fondo al valor inicial ("**pink**").

💡Deberás copiar el código en Visual Studio Code para probarlo

### ✏️ Actividad Manipulando el DOM

Dado los siguientes códigos:

**HTML**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

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

    <title>Document</title>

  </head>

  <h1>

    TITULO DE TU PAGINA

  </h1>

  <div id="container">

   <div class="content">

    </div>

  </div>

</html>

**Javascript**

const container = document.querySelector('#container');

const content = document.createElement('div');

content.classList.add('content');

content.textContent = 'Este es mi nuevo contenido!';

container.appendChild(content);

Deberás copiar los códigos provistos en archivos en tu propia computadora. Para que funcione, deberás enlazar tu archivo de JavaScript, o colocar el JavaScript en una etiqueta de script en la página. ¡Asegúrate de que todo esté funcionando antes de continuar!

📌 Antes que nada revisa el funcionamiento del código anterior e identifica qué métodos utiliza.

Añade los siguientes elementos al contenedor utilizando SOLAMENTE JavaScript y los métodos del DOM mostrados arriba.

* Un <p> con texto rojo que diga "¡Hey, soy rojo!"
* Un <h3> con texto azul que diga "¡Soy un h3 azul!"
* Un <div> con un borde negro y un fondo de color rosa con los siguientes elementos dentro de él:
  + Otro <h1> que diga "¡Estoy dentro de un div!"
  + Un <p> que diga "¡YO TAMBIÉN!"

💡 Pista para este último: después de crear el <div> con createElement, agrega el <h1> y el <p> a él antes de agregarlo al contenedor.

**Práctica Manejo de DOM 2.0**

En esta sección vamos a continuar con la práctica de javascript para acceder al HTML y manipularlo. Es muy importante que se familiaricen con estas opciones para que puedan manipular el DOM sin inconvenientes.

**✏️ Actividad Seleccionar elemento y cambiar contenido**

Volvemos a utilizar el juego brindado anteriormente, y vamos a practicar cambiar diferentes partes del juego con javascript.



Deberás seleccionar los elementos que se señalan a continuación y modificar el contenido de los mismos.

**📌Tip: Si deseas puedes utilizar console.log() para ver en consola el contenido**

1. Selecciona el elemento con la clase **.message** y modifica el mensaje. Las opciones que puedes utilizar son:
   1. **"⛔️ Ingrese Numero!"**
   2. **"🎉 Es el número!"**
   3. **"📈 Muy Alto!"**
   4. **"📉 Muy bajo!"**
   5. **"💥 Perdiste!"**
2. Selecciona el elemento con la clase**.number**y modifica el número
3. Selecciona el elemento con la clase **.guess** y modifica el valor
4. Selecciona el body del html y modifica el color de fondo por el que desees

**✏️ Actividad Cambiar el color de fondo al hacer clic en un botón**

Crear una página web sencilla con un botón que, al hacer clic en él, cambie el color de fondo del cuerpo de la página a un color aleatorio. Para esto, utiliza el método **addEventListener**para capturar el evento de clic en el botón y una función que genere un color aleatorio. Cada vez que se haga clic en el botón, el color de fondo debe cambiar a uno diferente.

**📌Tip: Te dejamos la función para obtener el color random (pero si la piensan ustedes mucho mejor!)**

function getRandomColor() {

      const letters = '0123456789ABCDEF';

      let color = '#';

      for (let i = 0; i < 6; i++) {

        color += letters[Math.floor(Math.random() \* 16)];

      }

      return color;

   }

**✏️ Actividad Contador de clics en un botón**

Crear una página web que contenga un botón y un párrafo que muestre la cantidad de veces que se ha hecho clic en el botón. Al cargar la página, el contador debe estar en 0. Cada vez que el usuario haga clic en el botón, el contador debe aumentar en 1 y mostrar el valor actualizado en el párrafo. Utiliza el método addEventListener para capturar el evento de clic en el botón y una variable para llevar el seguimiento del número de clics.

**✏️ Actividad Mostrar una alerta cuando el mouse se mueve sobre un elemento**

Crear una página web con un elemento de tu elección (por ejemplo, un div) que muestre un mensaje cuando el mouse se mueva sobre él. Utiliza el método addEventListener para capturar el evento de entrada del mouse (mouseenter) en el elemento seleccionado. Cuando el mouse se encuentre sobre el elemento, debe aparecer una alerta con un mensaje predeterminado. Puedes personalizar el mensaje según tu preferencia.

**⏭️ En el paso siguiente** pondrás en practica integral lo aprendido en tus pasos anteriores, y comenzarás a darle dinamismo a tu proyecto HTML.