Realiza una investigación en Internet y de acuerdo a lo que leíste, responde y completa lo siguiente:

1. ¿Qué es el DOM?  
   El Modelo de Objetos del Documento (DOM) es una interfaz de programación de aplicaciones (API) para documentos válidos HTML y bien construidos XML. Define la estructura lógica de los documentos y el modo en que se accede y manipula. En la especificación DOM, el término "documento" es utilizado en un sentido amplio - the term "document" is used in the broad sense - cada vez más XML es utilizado como un medio de representar muchas clases diferentes de información que puede ser almacenada en sistemas diversos, y mucha de esta información se vería, en términos tradicionales, más como datos que como documentos. Sin embargo, XML presenta estos datos como documentos, y se puede utilizar DOM para manejar estos datos.
2. ¿Cuáles formas posee JavaScript para seleccionar los elementos HTML?  
   querySelector()   
   getElementById()  
   querySelectorAll()  
   getElementsByTagName()
3. ¿Qué diferencia existe en cada una de las formas anteriores?  
     
   **querySelector()**   
   Devuelve el primer elemento del documento (utilizando un recorrido primero en profundidad pre ordenado de los nodos del documento) que coincida con el grupo especificado de selectores.  
     
   **getElementById()**  
   GetElementById () es un método DOM utilizado para devolver el elemento que tiene el atributo ID con el valor especificado. Este es uno de los métodos más comunes en el DOM HTML y se usa casi cada vez que queremos manipular un elemento en nuestro documento.  
     
   **querySelectorAll()**  
   Un DOMString que contiene uno o más selectores para buscar coincidencias. Esta cadena de texto debe ser una cadena CSS selector válida; si no lo es, se lanzará una excepción SyntaxError. Vea Locating DOM elements using selectors para más información acerca de utilizar selectores para identificar elementos. Se pueden especificar varios selectores separándolos utilizando comas.  
     
   **getElementsByTagName()**  
   El método getElementsByTagName () devuelve una colección de todos los elementos en el documento con el nombre de etiqueta especificado, como un objeto HTMLCollection.

1. Muestra un ejemplo de uso de las formas más usadas para seleccionar elementos.

let elParrafo = document.getElementById('parrafo-ejemplo').

1. ¿Cuáles son las propiedades y métodos más comunes del objeto **document**?  
     
   **getElemenById (id)** - Obtener nodo (elemento) con el id especificado

**appendChild (nodo)** - insertar un nuevo nodo secundario (elemento)  
**removeChild (nodo)** - Elimina el nodo hijo (elemento)

1. ¿Qué es un evento en JavaScript?  
   En la programación tradicional, las aplicaciones se ejecutan secuencialmente de principio a fin para producir sus resultados. Sin embargo, en la actualidad el modelo predominante es el de la programación basada en eventos. Los scripts y programas esperan sin realizar ninguna tarea hasta que se produzca un evento. Una vez producido, ejecutan alguna tarea asociada a la aparición de ese evento y cuando concluye, el script o programa vuelve al estado de espera.
2. Menciona un listado de los eventos más comunes en JavaScript.

* onclick
* onkeyup
* onfocus
* onchange

1. ¿Cómo podemos agregar un evento a un objeto HTML?  
   
2. Realiza un ejemplo sencillo en donde se muestre el uso de los eventos con JavaScript.



1. Mira los videos de las carpetas “Introducción al DOM y Eventos” y “Selección de elementos y CSS” para que realices lo siguiente:
2. En la carpeta practicas de tu repositorio, crea un archivo **ocultar.html** y **script.js**
3. En el archivo HTML coloca un **h1**, un **div** y un **button.**
4. Puedes colocar el texto que desees en el **h1** y en el **div**.
5. Cuando el usuario haga doble click en el button el **div** debe desaparecer y el **h1** debe decir “*Haz hecho doble click*” en color rojo y en cursiva.

**Instrucciones:**

1. Realiza el ejercicio en Visual Studio Code.

2. Guarda tu trabajo en tu repositorio.

3. Coloca la URL de tu repositorio en esta asignación.

**Recuerda**:

\* Nombrar las variables apropiadamente.

\* Comentar el código (ya sea con comentarios de línea o multilínea).

\* El código debe estar organizada.