BOM

-----

- Location

- History

- Funciones de tiempo

- setInterval

- clearInterval

- setTimeout

DOM

-------

- Acceder a elementos del DOM

- Cambiar valores de los nodos

- Agregar nodos

RESUMEN DE CLASES

==================

\*\*BOM\*\*

-------

\*\*BOM:\*\* Browser Object Model Son todas las cosas relacionadas con el navegador. Son los objetos que están fuera de los documentos cargados

-No tiene un standart y no todos los navegadores poseen las mismas caracteristicas

-Sirve de contenedor y tiene elementos útiles como frame, iframe,popup,ventana,pestaña EJ

1-1 windows.navigator

1.2 windows.location etc

2- Para manejar el tiempo:

\*\*2.1 setInterval():\*\*

Sirve para ejecutar un intervalo repetidamente en un intervalo establecido,

~~~

segundero= setInterval(function(){

let seg = new Date().getSeconds()

console.log(seg)

},1000 ) <--------- se indica en mili-segundos el intervalo en el que se realizara la acción

~~~

\*\* 2.2 clearInterval():\*\*

Sirve para detener un bucle cronometrado que se inicio con el setInterval()

~~~

clearInterval(segundero) <------ es la variable del ejemplo 1

~~~

\*\* 2.3 setTimeout(): \*\*

Utiliza una función después de una determinada cantidad de tiempo, se ejecuta una vez

~~~

segundero= setTimeout(function(){

console.log("Me ejecuto una sola vez pasado 3 segundos")

},3000 ) <--------- se indica en mili-segundos el intervalo en el que se realizara la acción una sola vez

~~~

\*\*DOM\*\*

========

\*\*DOM:\*\* traer elementos del HTML

---------------------------------------

DOM: Document Object Model, hace referencia a todos los elementos que corresponden a la pagina y no al navegador a diferencia del DOM

- document.getElementById(): trae las etiquetas por el id que le colocamos.

- documen.getElementByTagName(): trae el elemento por el tag es decir la etiqueta como por ejemplo <p>.

- documen.getElementByClassName(): trae el elemento por la clase que lleva.

- documen.querySelector(): especificando en el parametro si es un id con un # o una clase con un . "punto" o la etiqueta podemos traer ese elemento pero solo uno.

- documen.querySelectorAll(): al igual que el anterior pero este me trae todo los que coincidan

\*\*DOM:\*\* crear contenido HTML

---------------------------------------

- innetText: injerta texto dentro de una etiqueta seleccionada, o al body

- innetHTML: Me permite crear código HTML e inyectarlo en el body o una etiqueta seleccionada

- classList="Permite agregar clases a la propiedad class de una etiqueta"

- document.creatElement(): para crear un elemento desde js hay que esta funcion recibe un parametro string donde le colocamos le nombre de la etiqueta que queremos ("div")

- document.body.appendChild(): esta funcion nos permite añadir al documento dentro de body pero al final del body y esta recibe de parametro algun elemento creado con createElement()

- document.body.appendChild("elElemento".cloneNode(true)): nos permite clonar un elemento con el contenido interno.

- document.body.appendChild("elElemento".cloneNode(false)): nos permite clonar un elemento si el contenido interno

- document.body.insertBefore(nuevo nodo, nodo): recibe dos parametros uno es el elemento que quiero colocar y el segundo es en que ubicacion anterior lo quiero

- document..bodyremoveChild(): recibe el elemento traido con id o query selector etc y lo elimina del body

- document.body.replaceChild(nuevo nodo, nodo a reemplazar): recibe dos parametros el primero es el que queremos que ocupe su lugar y el segundo cual queremos reemplazar