

Actividad No.3: JavaScript para principiantes (Video)



Alumno: Silva Bedolla Emiliano

Profesor: Iván Isay Guerra López

Materia: Desarrollo Apps Móviles

Grupo: TIID-211

Tema: JavaScript para principiantes

Actividad

En esta actividad lo que nos toco fue ver un video que nos proporciono el maestro acerca del lenguaje de programación de JavaScript, que para empezar a tomar en cuenta este lenguaje es uno en conjunto con HTML y CSS para crear páginas de WEB o aplicaciones móviles, además de que se ejecuta mediante algún navegador web (Chrome o Edge).

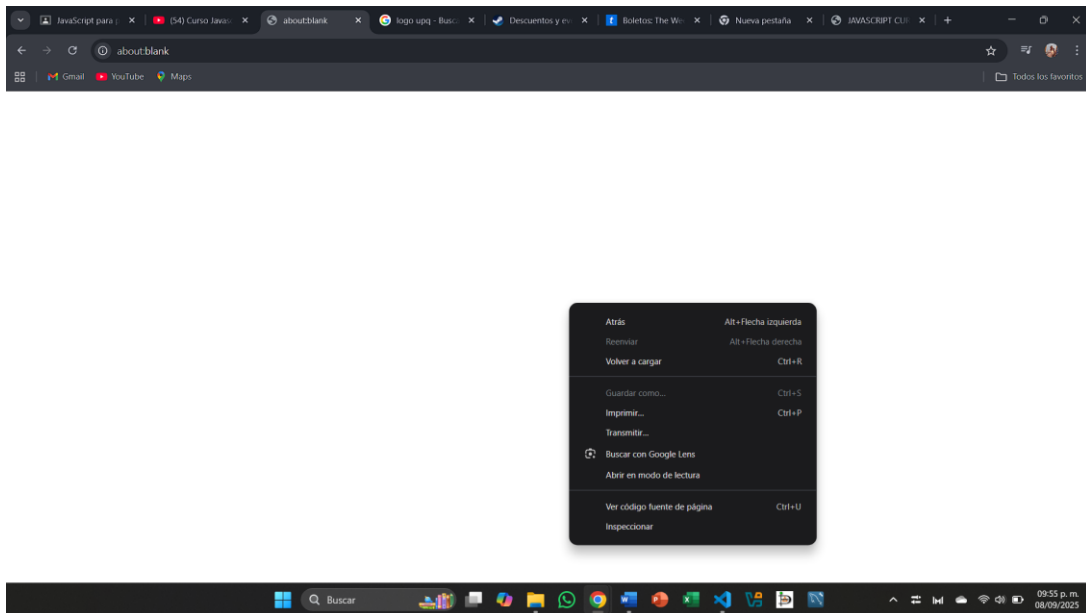
Dentro del video nos explico un poco acerca de lo que puedo hacer dentro de JavaScript como lo es lo siguiente:

- Poner contenido en una página ya que dentro de este código se pueden hacer modificaciones para que el texto se vea de cierta forma en nuestra página o aplicación.
- Hacer que nuestros sitios o aplicaciones se pueda un poco adaptar a los diferentes dispositivos que el usuario maneje (ser responsivos).
- Saber que tipos de navegadores utilice el usuario para visualizar el contenido de nuestra página.
- Validar ciertos tipos de formularios que tengamos dentro de nuestra página.
- Crear aplicaciones web en conjunto con HTML y CSS (que este último se encarga de todo lo visual o los colores y la distribución de espacios)

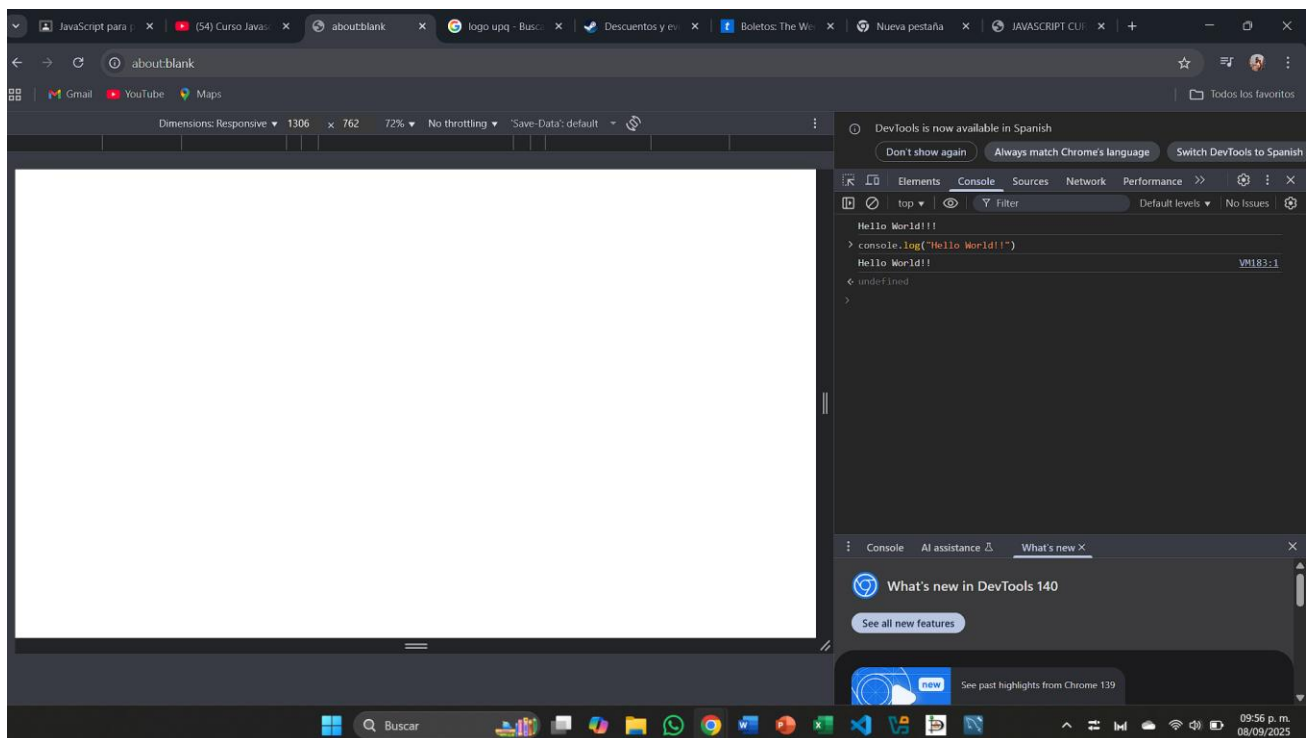
Después de eso nos comentaron acerca de las herramientas necesarias que en este caso es el navegador para poder ejecutar el código, un editor de código que en este caso es Visual Studio Code (es el que utiliza el del video y en mi caso es el mismo).

Ahora sigue la parte practica

Dentro del video nos enseñaron dos formas de hacer un "Hello World" la primera forma que nos mostro fue en hacer por medio de la consola de nuestro buscador que dentro de esto lo primero que hicimos fue buscar en nuestro navegador (en mi caso es Chrome por que se me hace más fácil de usar) la palabra "about:blank"

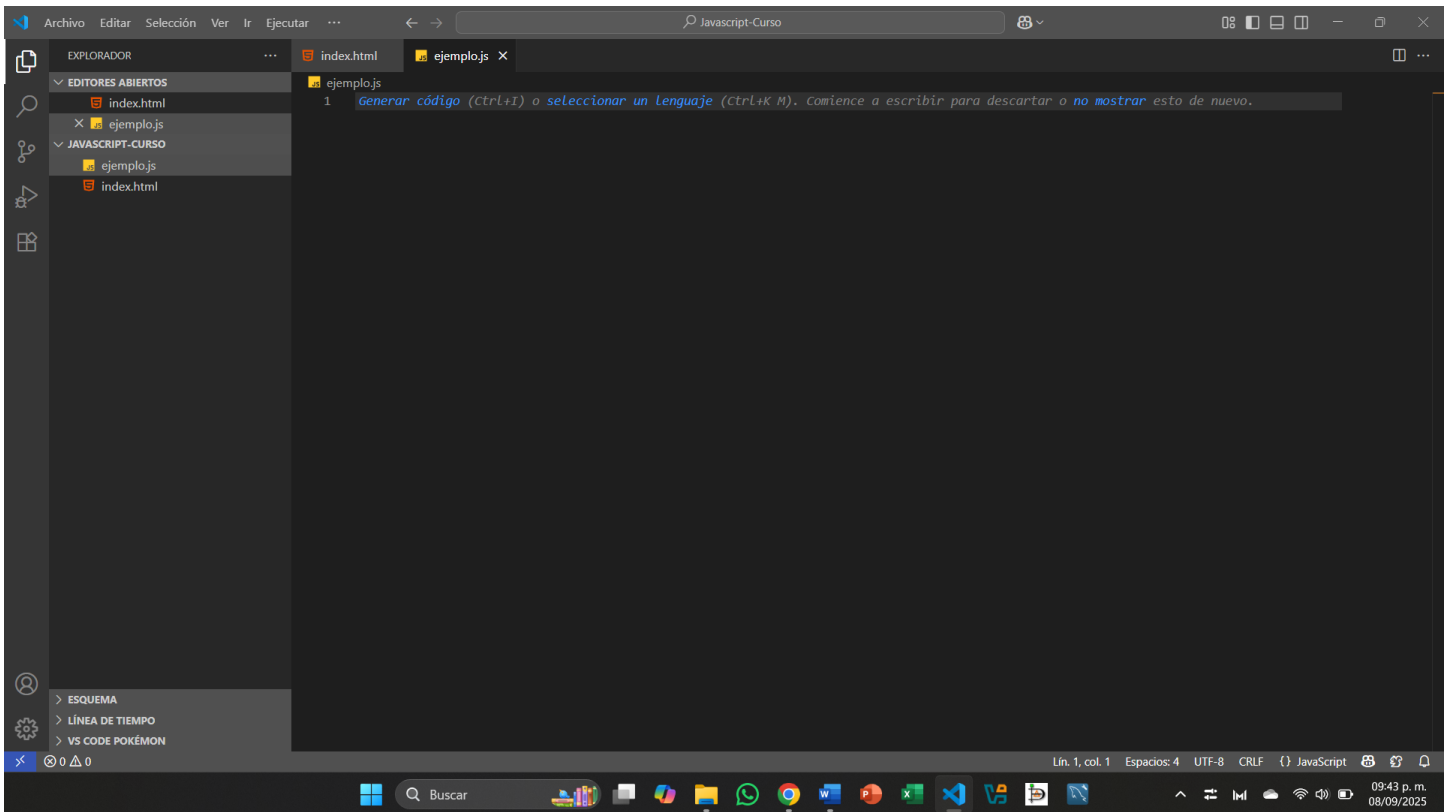


Dentro de esta página le damos clic derecho y nos vamos a la parte de "Inspeccionar" para que se nos abra una sección como de tipo "desarrollador" y una vez abierta nos iremos a la sección de "Console" para hacer esa actividad y escribiremos en la consola el comando "Console.log ("Hello World!!!")" que este comando sirve en JavaScript solo para que este mensaje de Hello World solo se muestre dentro de la consola del buscador y no en la página WEB como tal

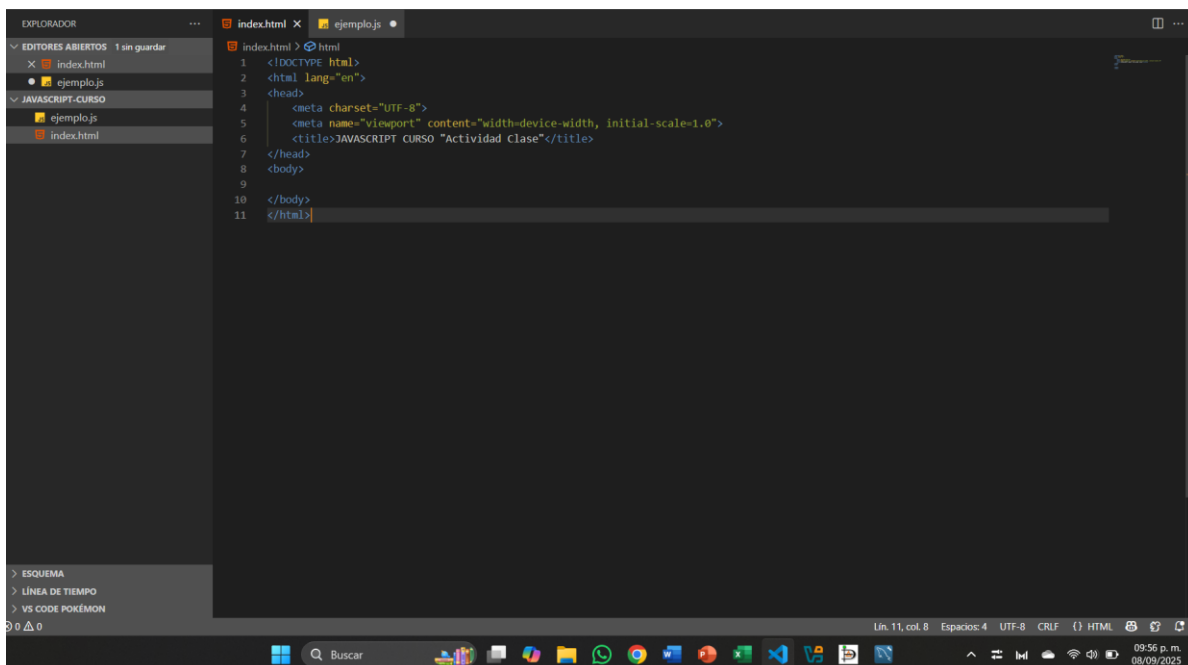


Una vez esto el chico del video nos enseñó ahora poder hacer esto dentro de un archivo de .js y .html, lo primero que hicimos fue abrir nuestro editor de código

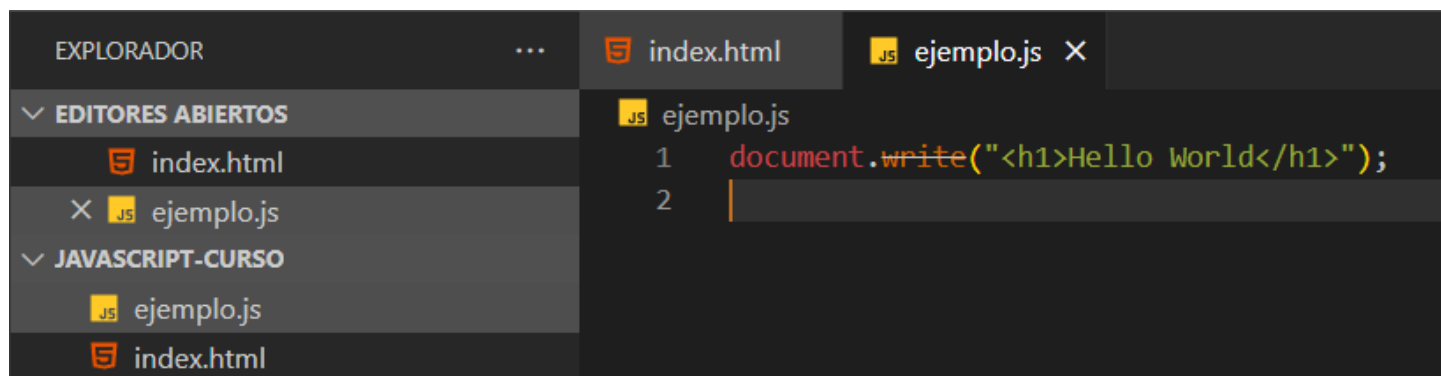
(Visual Studio Code en mi caso) y crear dos archivos uno llamado "index" en la extensión de .html y otro archivo llamado "ejemplo" en extensión .js



Una vez creado estos dos archivos, en el archivo "index" vamos a poner en segmento de código esencial en un archivo html (para que todo lo que se ponga dentro del archivo de JavaScript se logre ver en el cuerpo de nuestra página)



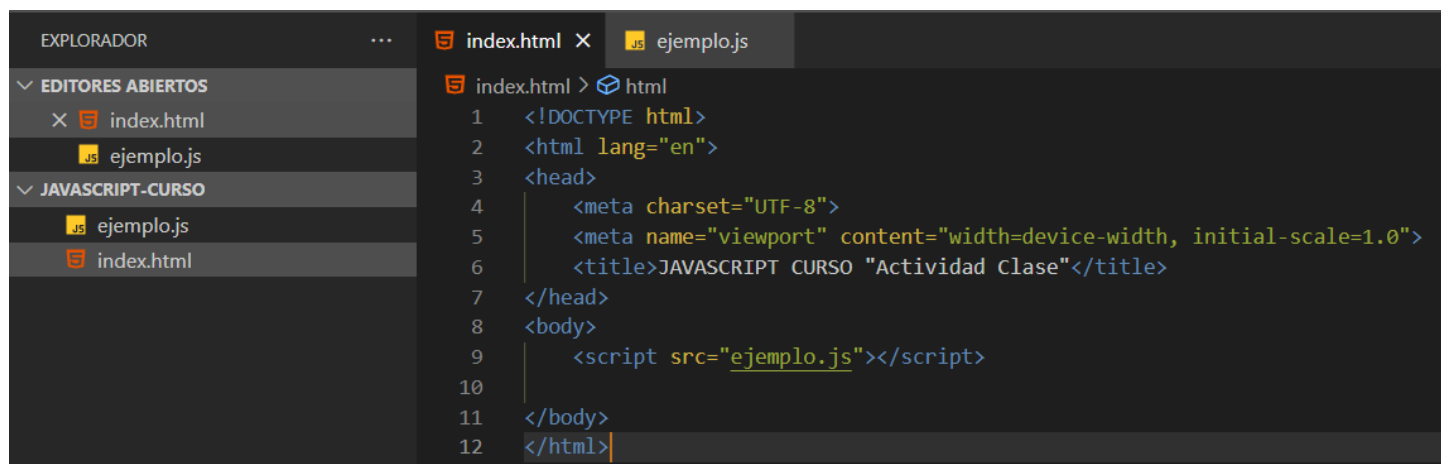
Ahora lo que sigue es que dentro del archivo de JavaScript (en mi caso es "ejemplo.js") la línea "document.write(<h1> Hello World</h1>)" para que cuando se haga un enlace del documento de JavaScript y el de HTML se logre ver como si se estuviera escribiendo solo en el de HTML



```
EXPLORADOR  ...  index.html  ejemplo.js X
EDITORES ABIERTOS
  index.html
  X ejemplo.js
JAVASCRIPT-CURSO
  ejemplo.js
  index.html

ejemplo.js
1  document.write("<h1>Hello World</h1>");
2  |
```

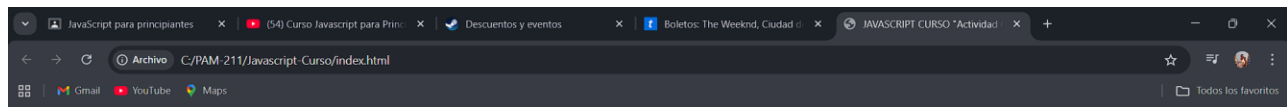
Ahora lo que haremos para que se logre ver lo que escribimos dentro de nuestro archivo de JavaScript dentro de nuestro archivo HTML ponemos la siguiente línea "<script src='ejemplo.js'></script>" (dentro de la sección de src="" dentro de esta parte debemos de poner el nombre de nuestro archivo de JavaScript) que esta línea lo que hace es como crear un tipo "vínculo" para que todos los cambios que hicimos dentro de nuestro archivo de JavaScript se logren ver dentro de nuestro ejecutable de la página



```
EXPLORADOR  ...  index.html X  ejemplo.js
EDITORES ABIERTOS
  index.html
  ejemplo.js
JAVASCRIPT-CURSO
  ejemplo.js
  index.html

index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  |   <title>JAVASCRIPT CURSO "Actividad Clase"</title>
7  | </head>
8  | <body>
9  |   <script src="ejemplo.js"></script>
10 |
11 | </body>
12 | </html>
```

Y una vez que tenemos todo guardado lo que haremos es ir a donde esta guardado nuestro documento "index.html" y darle doble clic para que se ejecute en nuestro navegador y poder ver nuestro archivo de la página con el texto de "Hello World"



Hello World



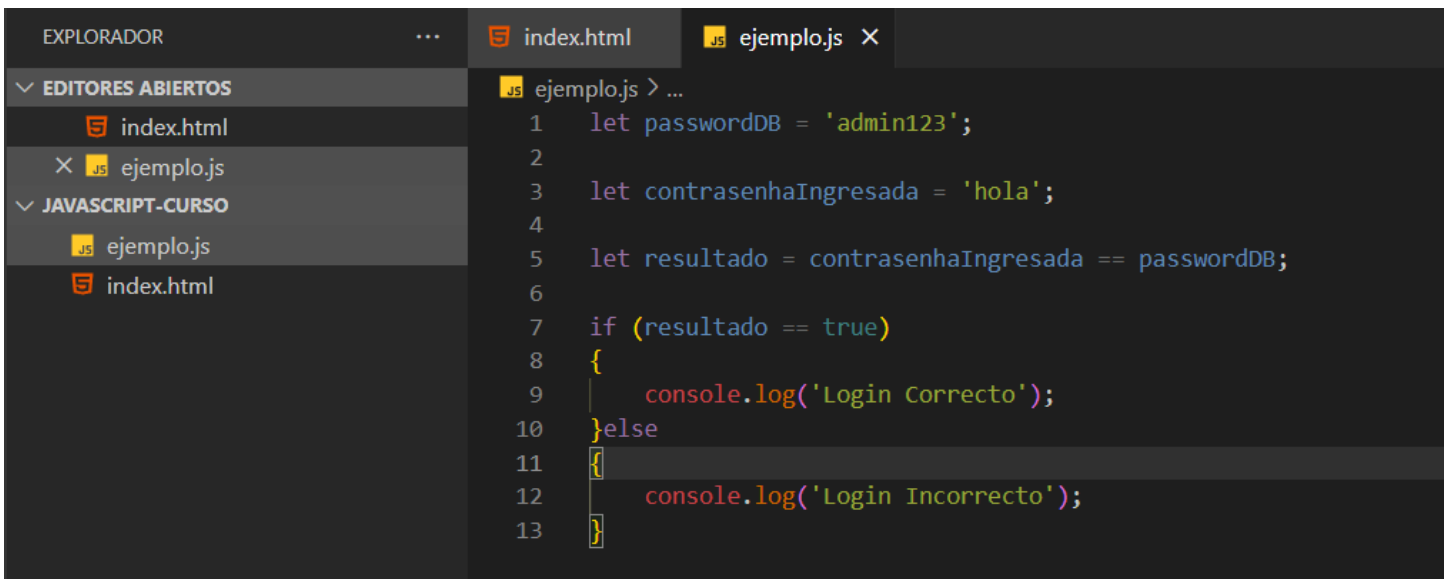
Una vez realizado esto el chico del video nos realizó una pequeña explicación acerca de temas sencillos que se ocupan para manejar este lenguaje que son los siguientes:

- **Tipos de datos:** Dentro de los tipos de datos son los mismo que hemos utilizado dentro de diferentes tipos de leguajes de programación como lo son:
 - **String:** Este tipo de dato es lo más conocido que son las cadenas de caracteres y se pueden guardar todo tipo de caracteres dentro de este dato.
 - **Number:** Este es utilizado más que nada para guardar todo tipo de números enteros, fraccionarios, etc.
 - **Boolean:** Este es el tipo booleano que son más que nada utilizados para la validación de cosas ya que solo aceptan dos tipos de datos (true y false).
 - **Arrays:** Los arrays o arreglos es como si fuera una lista ya que dentro de esta lista se pueden guardar diferentes datos en un solo lugar.
- **Variables:** Las variables son como por así decirlo un tipo de "contenedor" que guarda todo tipo de dato que se le asigne (con forme a los datos anteriores además de que puede cambiar su valor por eso de su nombre "variable") y para

su declaración es muy sencilla ya que solo se ocupan las palabras "Var" o "let" que son como la "palabra" clave para así saber que es una variable.

- **Constantes:** Las constantes son valores ya definidos que tenemos dentro de nuestro código estos solo tienen un solo valor y no cambian de valor, estas al declararlas se ocupa la palabra "const" para así diferenciarlas de las constantes.
- **Condicionales:** Estas permiten como por así decirlo controlar ciertas acciones de algunas acciones dentro del código por ejemplo en el video nos muestran un ejemplo de condicional con el if, else y elseif.

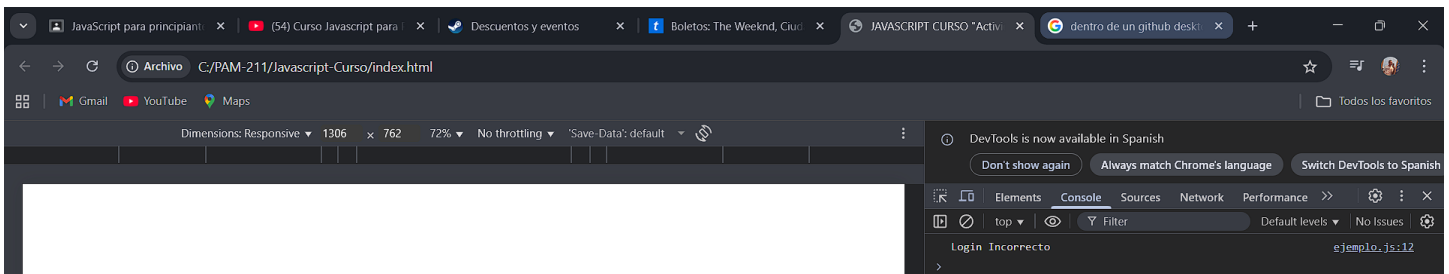
En el ejemplo que nos mostro el video es acerca de una contraseña de un Inicio de sesión, en donde la primera variable llamada "passwordDB" es por así decirlo la contraseña que se guarda dentro de nuestra base de datos de nuestra aplicación o página, después de eso declaramos otra variable de contraseña ingresada para así tener una contraseña incorrecta y dentro de nuestra condicional que es un if-else sencillo mostrar si la contraseña es incorrecta o correcta según sea el caso y lo que va dentro de los paréntesis del if es por así decirlo la acción que esta haciendo, por ejemplo en este caso es cuando declaramos la variable "resultado" estamos haciendo una comparación (las comparaciones aquí en JavaScript siempre se hacen con doble ==) se está comparando ambas contraseñas pero dentro del paréntesis del IF se está recibiendo una orden de "Si las dos contraseñas son iguales (true) va a mostrar que el Login es correcto" pero si no lo es así en la parte del else es cuando no se cumple esta condición y se muestra el mensaje de Login incorrecto.



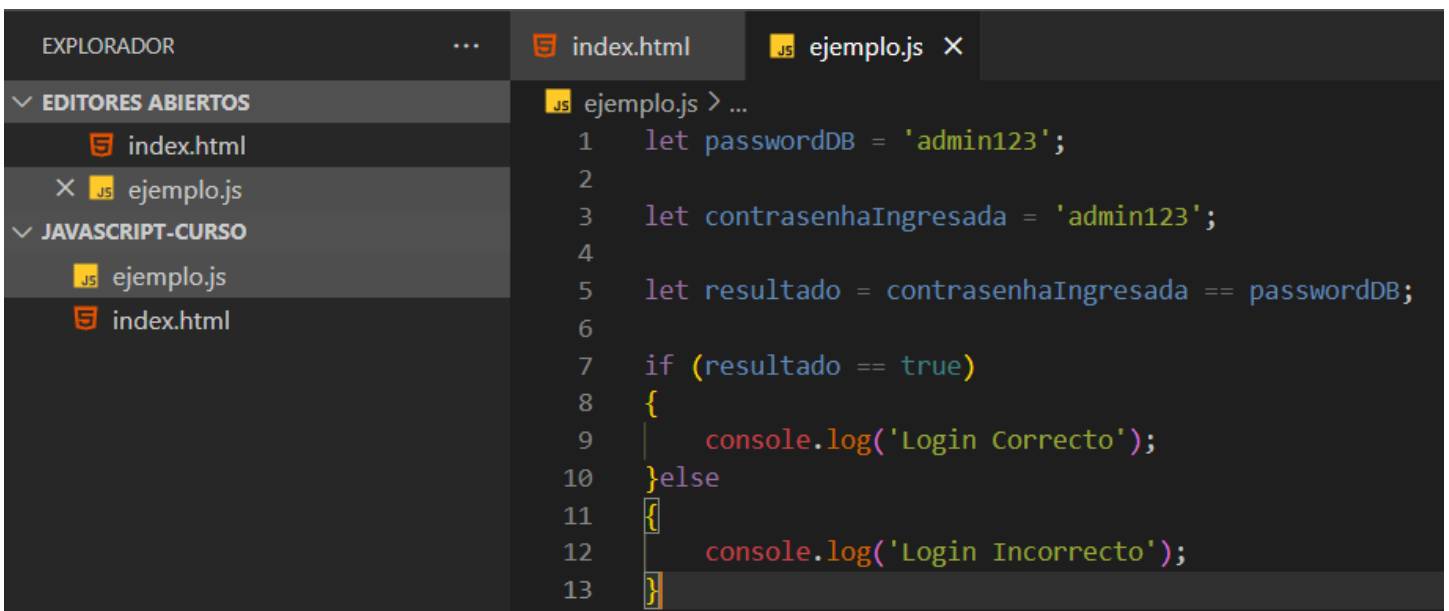
```
index.html | ejemplo.js X
EDITORES ABIERTOS
index.html
ejemplo.js
JAVASCRIPT-CURSO
ejemplo.js
index.html

ejemplo.js > ...
1 let passwordDB = 'admin123';
2
3 let contrasenhaIngresada = 'hola';
4
5 let resultado = contrasenhaIngresada == passwordDB;
6
7 if (resultado == true)
8 {
9     console.log('Login Correcto');
10 }else
11 {
12     console.log('Login Incorrecto');
13 }
```

Todo esto se ejecuta dentro de la consola de nuestra página del buscador, en este primer ejemplo se realizo un inicio de sesión incorrecto



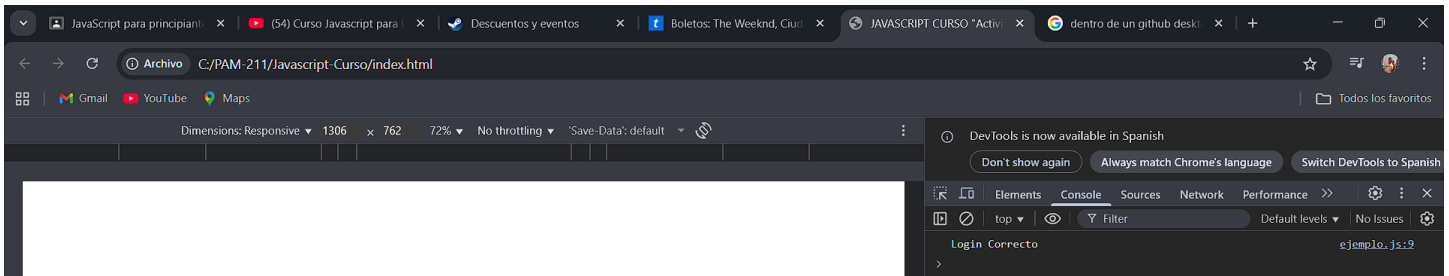
Después de eso hicimos lo mismo, pero ahora solo con un inicio de sesión correcto cambiando la contraseña ingresada por la misma que estará definida en nuestra base de datos



```
index.html | ejemplo.js X
EDITORES ABIERTOS
index.html
ejemplo.js
JAVASCRIPT-CURSO
ejemplo.js
index.html

ejemplo.js > ...
1 let passwordDB = 'admin123';
2
3 let contrasenhaIngresada = 'admin123';
4
5 let resultado = contrasenhaIngresada == passwordDB;
6
7 if (resultado == true)
8 {
9     console.log('Login Correcto');
10 }else
11 {
12     console.log('Login Incorrecto');
13 }
```

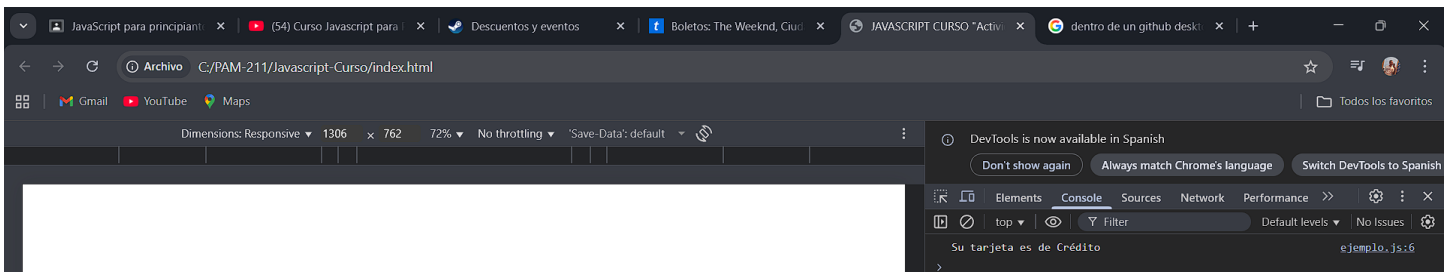

Y se podrá mostrar un mensaje de Login correcto



Y también aquí se puede ocupar un **Switch** que es lo mismo que el **if** pero se vería mas que nada como un tipo "menú" y de igual forma se hizo un ejemplo de este uso en este caso se hizo un ejemplo de una tarjeta de crédito que mientras que el la variable que tenemos declarada dentro de nuestro programa tiene como ejemplo ya la opción de "tarjeta de crédito" dentro del switch se hace una tipo comparación para saber que mensaje mostrar (para que no se muestren ambos mensajes se debe de agregar el **break** en cada caso para que solo se muestre y ejecute solo la opción que hemos elegido y el **case** del switch es para saber que acciones se van a ejecutar todo dependiendo de las opciones que hallamos escrito o elegido)

```
index.html  ejemplo.js X
ejemplo.js > ...
1  let typecard = 'Tarjeta de Crédito';
2
3  switch (typecard)
4  {
5      case 'Tarjeta de Crédito':
6          console.log('Su tarjeta es de Crédito');
7          break;
8      case 'Tarjeta de Débito':
9          console.log('Su tarjeta es de Débito');
10         break;
11     }
12
```

Y así se vería dentro de la consola del navegador



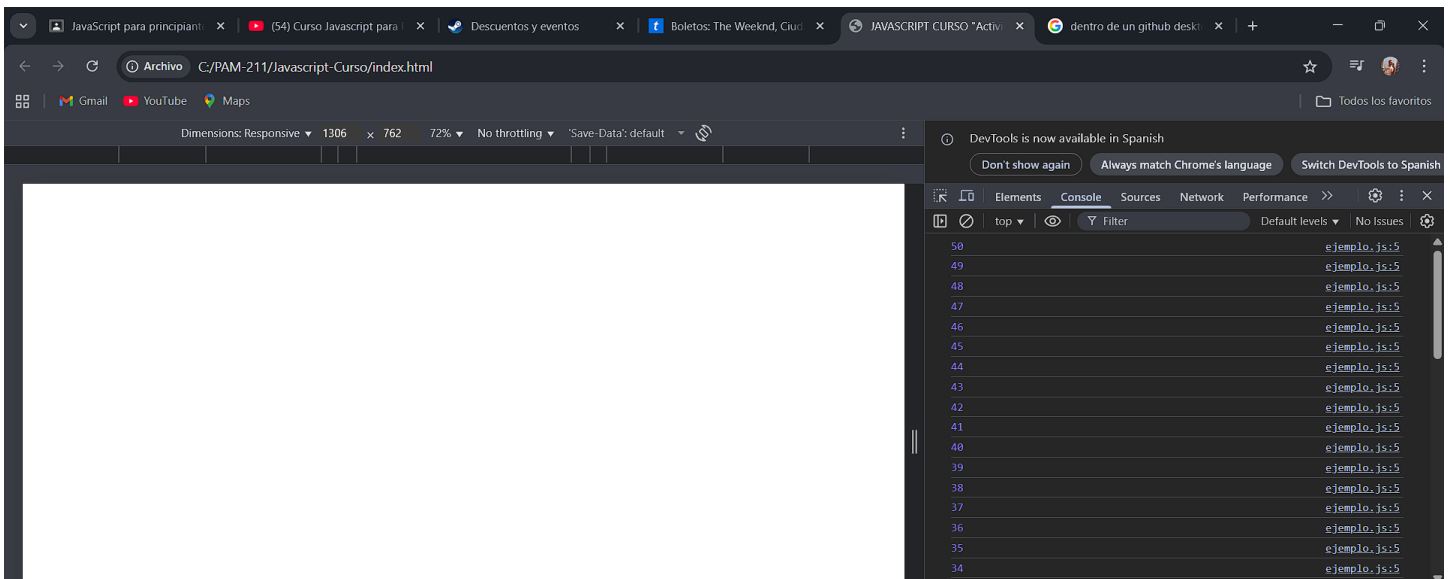
- **Bucles:** Esta sección es mas que nada cuando se tienen que repetir cierta cantidad de veces dentro de este mismo código, pero sin escribir la misma línea "n" cantidad de veces que se necesita

Aquí se realizo un ejemplo de una sucesión de numero para solo mostrar cierta cantidad de numero y esto se logro mediante el uso de un while lo que hicimos fue declarar una variable "count" que se inicio desde 50 y mediante el uso del while lo que se va a hacer es que mediante la variable "count" siga siendo mayor a cero se va a repetir y va a mostrar el numero y con cada repetición se va a ir restando 1 para que al final hasta que sea mayor a 0 se deje de repetir y mostrar el número

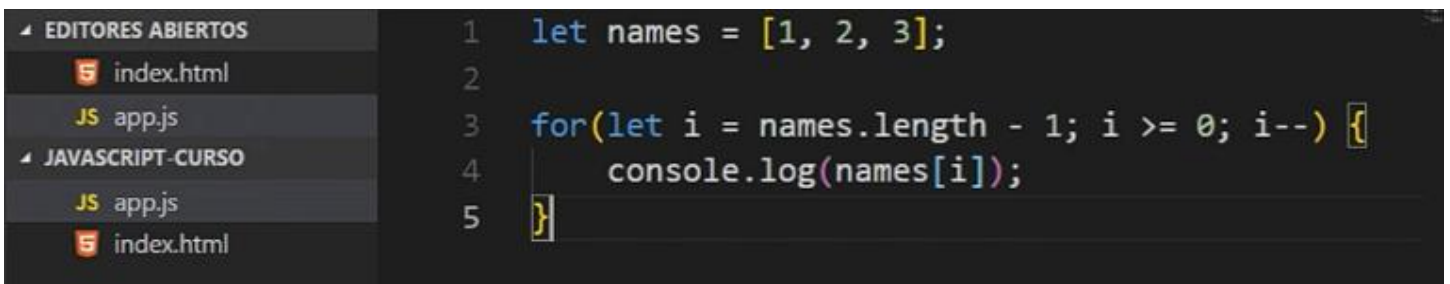
A screenshot of the Visual Studio Code editor. The Explorer panel on the left shows the file structure with 'index.html' and 'ejemplo.js' under 'EDITORES ABIERTOS' and 'JAVASCRIPT-CURSO'. The main editor area shows the code in 'ejemplo.js':

```
1 let count = 50;
2
3 while(count > 0)
4 {
5     console.log(count);
6     count = count - 1;
7 }
8
9
```

Y así se ve mostrando en la consola en este caso se mostro del número 50 al numero 1



Y para los bucles también se mostraron el uso del "FOR" para que dentro de un arreglo (arrays) muestre todo los datos que estén guardados dentro de este mismo arreglo (imagen tomada del video)



➤ **Funciones:** Estas funciones son segmentos de códigos que se pueden reutilizar en algunas otras cosas dentro de nuestro mismo código y por así decirlo son funciones que nosotros hacemos dentro de nuestro código, pero si no las mandamos a llamar no se ejecutan

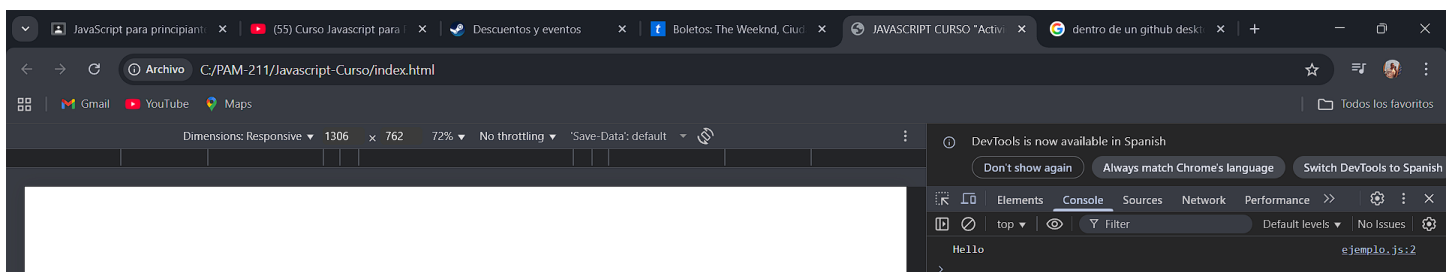
El ejemplo que se hizo fue para mostrar en consola un "Hello", lo primero que hicimos y es **obligatorio** para declarar una función es poner la palabra "Function" que se utiliza mas que nada para decir y avisar que se está creando una función y después de esta palabra se pondrá su nombre que en este caso es "greeting" con ella se acompañan unos paréntesis y una llaves que dentro de estas llaves ira todo lo que quieres que se haga dentro de esta función que en este caso es solo mostrar en la consola un "hello" y una vez ya definido todo lo que nosotros estemos realizando dentro de la función se va a tener que llamar la función para que se ejecute ya que si no se manda a llamar no se va a ejecutar y se podría

decir que es "código muerto". Dentro de la función siempre se debe de llamar como nosotros declaramos su nombre en este caso se mandará a llamar como "greetind();"



```
1 function greeting(){
2   console.log("Hello");
3 }
4
5 greeting();
```

Una vez llamado podremos ver su ejecución dentro de nuestra consola de nuestro navegador.



Conclusión

Dentro de lo que llevo de esta actividad es mas que nada aprender como se utilizan datos que ya conocía dentro de la programación (ya que los hemos utilizado dentro de todos los cuatrimestres) pero dentro de un nuevo lenguaje que en este caso es JavaScript