Investigar y repasar la sintaxis básica de Javascript a saber:

1- Control de flujo y manejo de errores
    a. if...else

El ***if…else*** en javascript al igual que en varios lenguajes de programacion es una de las funciones más básicas que le dan la lógica al programa, en el caso del ***if*** a este le damos un variable a comparar con un criterio (por ejemplo, revisar si un numero es mayor a 20), si el criterio se cumple este mismo lo que hace es ejecutar una serie de comandos o funciones que se estipulan dentro de sus llaves. El ***else*** también es útil ya que nos sirve si se quisiera ejecutar una serie de funciones u operaciones si no se cumple el ***if***. Un ejemplo de esto podríamos hacer una función que acepte la nota de un estudiante, y este con el ***if*** revise si es mayor o igual a 70, si es correcto el mismo responde que aprobó, si no con uso del ***else***, este nos devuelve que reprobo

    b. switch

El ***switch*** al igual que ***if…else*** es un comando lógico, en este caso el ***switch*** lo que hace es recibir un valor, y de ahí revisa si hay un caso que corresponda al valor que se colocó. Un ejemplo de esto podríamos decir que tenemos una aplicación que nos muestra el listado de materias que lleva cada carrera, entonces nosotros nada mas colocamos el nombre de la carrera (este siendo la entrada del ***switch***) y de ahí el ***switch*** si encuentra un caso con ese valor insertado realiza la función de devolvernos el listado de materias de esa carrera.

    c. try/catch/throw

El ***try/catch/throw*** es bastante útil cuando manejamos entrada de datos de parte del usuario, ya que este mismo puede ser que este usuario coloque información que la computadora no pueda procesar, por ejemplo que el usuario entre letras en una espacio de números, con el ***try*** lo que se puede colocar son los cálculos que se van a realizar y este si encuentra errores (en este caso de que entrada no es numérica), con el ***catch*** este obtiene el error que se produjo en el ***try*** y se le colocan funciones para así mostrarle al usuario que hubo un error y el motivo porque se produjo. El ***throw*** también pudo haber sido usado en el caso anterior, pero este es utilizado para prevenir errores de ejecución, ya que se puede colocar un ***if*** al principio del ***try*** que revise el tipo si no es número, y si no lo es active el ***throw*** que nos va a mandar directamente al catch sin necesidad de que haya un error de ejecución.

    d. Objetos Error

Los ***Objetos Error*** son bastante utilizados en el ***try/catch/throw*** ya que con esto no estaríamos usando errores “genéricos” sino estaríamos usando errores que ya están establecidos dentro de JavaScript, estos errores son: ***EvalError***, ***InternalError***, ***RangeError***, ***ReferenceError***, ***SyntaxError***, ***TypeError***, ***URIError***

2- Bucles e iteración
    a. for

El bucle ***for*** es uno bastante util en los casos que estemos trabajando con ***listados*** o ***arrays***, ya que digamos que ocupamos revisar cada uno de los ítems de un ***array*** y revisar si estos son mayores a cierto número, y si lo son estos colocarlos en un array secundario.

    b. while

El bucle ***while*** es utilizado en el caso que queramos correr un código de manera continua hasta un punto que nosotros establezcamos, digamos que tenemos un numero bastante grande como 200, y queremos dividirlo por 2 hasta que nos de un numero menor a 1, entonces nada mas colocamos en el **while** la condición a cumplir para que se corra su código que seria revisar si el numero es menor o igual a 1, y dentro de las llaves nada más colocamos la división a realizar sobre la variable que tiene el número.

    c. do...while

El ***do…while*** es básicamente lo mismo que un ***while***, nada mas que tiene una única diferencia bastante grande, y es que el ***do…while*** va a correr su código por lo menos 1 vez, sin importar que se cumpla o no la condición del ***while***, esta condición del ***while*** lo que va a realizar es si se va a correr por segunda vez o más veces el código.

3- Funciones
    a. Definir funciones

La ***definición de funciones*** es una de las cosas que mas ayuda a la hora de programar, ya que con esto podemos crear código reutilizable es cual solo se debe colocar una única vez.

De la manera que se crea o define la función es simple, es únicamente de colocar ***function*** seguido con el nombre de la función, y por último abrimos y cerramos paréntesis y también abrimos y cerramos llaves, en el caso que ocupemos que en esa función se revisa algún tipo de variables, nada mas dentro de los paréntesis colocamos el nombre de esas variables las cuales únicamente van a poder ser accedidas dentro de esa función, dichas variables tienen que estar separadas de comas. Y por ultimo dentro de las llaves que creamos, aquí es donde colocaríamos el código que lo que realizaría esta función, por ejemplo, una función que revise si la entrada es numérica, y si lo es, que devuelva si es par o no.

    b. Llamar funciones

Con el paso anterior se vio como crear las funciones, pero también ocuparemos saber como ***utilizar o llamar estas funciones***, lo cual es bastante simple ya que únicamente lo que debemos de hacer es colocar el nombre de la función que le pusimos con anterioridad, y de ahí habrá que colocarle los paréntesis, si la función tiene variables, tendremos que colocar las variables en el paréntesis.

4- Expresiones y operadores

a. Asignación y Comparación

En JavaScript, la asignación de variables es bastante sencilla a comparación de otros lenguajes, ya que en esta no debemos de especificar cual es el tipo de variable, lo único que ocupamos definir es como es esta disponible, a la hora de declararlas tenemos ***var*** (esta siempre ha existido en JavaScript, el acceso de esta es global, ósea que se declara una única vez en todo el código), ***let*** (esta es mas reciente, el acceso es dependiente del lugar donde se declaró, si se declara en una función, únicamente la función tendrá acceso a la variable) y ***const*** (esta fue agregada junto a ***let***, esta es una variable constante que al momento de definirla, es el único momento que se le puede colocar un valor, después de eso no podrá ser cambiado.

A la hora que nosotros definimos o cambiamos el valor de alguna de las variables, estas pueden utilizar valores aritméticos, ya que estas guardarían el resultado de dicha operación.

Y dichas variables pueden ser utilizadas para operaciones lógicas, ósea utilizadas en ***if…else*** por ejemplo, esto es gracias a los operadores lógicos los cuales nos sirven para poder comparar una variable con un valor estático, o hasta podemos comparar una variable con otra variable.

b. Operadores aritméticos

En el caso de los ***Operadores Aritméticos***, estos son las operaciones básicas de matemáticas, como la ***suma (+)***, ***resta (-)***, ***multiplicación (*)***, ***división (/)*** y hasta la ***potencia(^)***. Estas a diferencia de los Operadores Lógicos es que pueden dar valores “dinámicos” ya que este estos se puede utilizar con números para poder realizar una gran cantidad de cálculos que van a devolver otros números, lo cual no nos deja sujetos a un rango de resultados.

c. Operadores lógicos

En el caso de los Operadores Lógicos, estos son lo que se utilizan para poder realizar las comparaciones, en la lista tenemos a: ***Igual (==)***, ***Distinto (!=)***, ***Menor (<)***, ***Menor o Igual (<=)***, ***Mayor (>)***, ***Mayor o Igual (>=)***, ***AND (&&)***, ***OR (||)***, ***NOT (!)***. Y como se menciono en los Operadores Aritméticos, estos si están sujetos a 2 únicas respuestas, en este caso tendríamos a ***Verdadero*** o ***Falso***, dándonos a saber si se esta cumpliendo o no.

5- Números y fechas
    a. Objeto Number

El objeto Number, es uno de los objetos más útiles a la hora de utilizar la entrada de un usuario que queramos utilizar como número, ya que con esta función de manera simple enviándole la entrada de usuario, este nos responde con el numero si es un numero o nos puede hasta responder con NaN (Not a Number) en el caso que el valor no sea un número.

Este objeto también tiene otras funciones como por ejemplo darnos el valor de EPSILON, darnos el numero más grande o pequeño que pueda trabajar JavaScript. Y también hay funciones de comprobación, para saber si un valor es NaN o si es un numero entero, etc.

    b. Objeto Math

Otro de los objetos que tiene Java sería el de Math, este es utilizado para realizar cálculos matemáticos más complejos, como los Logaritmos, Valores Absolutos, Trigonometría, Raíces numéricas, Potencias, Redondeo y también generación de números aleatorios.

    c. Objeto Date

Uno de los objetos bastante importantes es de las Fechas, con este podemos obtener la fecha actual, o podemos parsear la fecha que se puede obtener de la entrada de algún usuario.

6- Formateo de texto
    a. Cadenas literales

Las Cadenas literales son básicamente un string, con la única diferencia de que se pueden hacer un poco más simples de trabajar si dentro del string hay varios valores dependientes a variables.

Estos string en lugar de ser declarados con la comilla simple o la comilla doble, se utiliza la ***"tilde inversa" (`)***, con este tipo de string dentro del texto nada más ocuparíamos colocar un símbolo de dólar y llaves, dentro de las llaves puede ir cualquier variable o hasta una operación de cualquier tipo.

    b. Objeto String

Los objetos String, es básicamente una cadena de texto.

7- Colecciones indexadas
    a. Arreglos

En JavaScript una de las cosas que podemos tener serían los ***arreglos***, ***listas*** o ***array***. Estos son básicamente una compilación de variables, nada más que cada variable tiene un index o una posición asignada, la cual podemos utilizar para poder obtener rápidamente la variable de una lista según su posición.

    b. Arreglos tipados

Los Arreglos tipados son lo mismo que los arreglos, ya que tiene un conjunto de variables indexadas, la única diferencia es que están únicamente dejan entrar un tipo especifico, usualmente únicamente números. Los tipos de estos arreglos son: ***Int8Array, Uint8Array, Uint8ClampedArray, Int16Array, Uint16Array, Int32Array, Uint32Array, Float32Array, Float64Array, BigInt64Array, BigUint64Array***.

8- DOM

Con ayuda del DOM, se puede realizar las páginas webs un poco más dinámicas, ya que con esto en el script podemos encontrar un elemento especifico y editarlo en el mismo script, por ejemplo colocar la fecha actual en un elemento párrafo.

a. Encontrar elementos HTML por ID

Este nada más se puede realizar con él ***".getElementById(id)"***, entonces digamos que estamos buscando un div con el id "content" el cual es un hijo del body del HTML, entonces nada más ocupamos colocar ***"document.getElementById("content")"*** en el script.

b. Encontrar elementos HTML por nombre de etiqueta

Este se realiza con el ***".getElementsTagName(name)"***, con este obtenemos todos los Elementos según el tipo que queramos, por ejemplo tenemos una tabla, y con el ***".getElementById(id)"***, ya tenemos toda la tabla, entonces lo que podemos realizar es un ***".getElementsTagName("td")" para poder obtener los datos de cada una de las celdas***.

c. Encontrar elementos HTML por nombre de clase

Este es bastante similar al ID, se obtiene de la siguiente manera ***".getElementsByClass Name(name)"***, la diferencia que hay que tomar en cuenta es debido a los limites de HTML, nosotros podemos tener un elemento con un id, este mismo no se puede repetir por ende si lo buscamos con JavaScript, este nos va a devolver dicho elemento, pero en el caso del ClassName, nos va a devolver todos los elementos que encuentre que tengan ese nombre.

d. Encontrar elementos HTML mediante selectores CSS

e. Encontrar elementos HTML por colecciones de objetos HTML

Una vez investigada toda la sintaxis anteriormente lo que debe entregar es un ejemplo práctico de una página web que contenga el uso de todos los elementos vistos en la sintaxis, queda a criterio del estudiante como quiere presentar la página y no se solicita que tenga un estilo muy elaborado dado que lo que se busca es que se pongo en práctica todos los elementos vistos en la sintaxis.

Enlace de página web de ejemplo: https://baka69420.github.io/Trabajos-ISW-512/Trabajo-5/sintaxis-basico-JavaScript.html