Bucles

---------

- For

- https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/for

Arrays

--------

- Métodos y propiedades de los arreglos

- https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global\_Objects/Array

RESUMEN DE CLASES

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

For

----------

\*\*For:\*\* al igual que el while y el do While este bucle me permite ejecutar código de manera repetida siempre que cumpla una condición pero tiene la particularidad de que esta puede recibir dentro de la evaluación los paréntesis la variable que hara la iteracion de numero de ejecuciones y la cantidad

~~~

for(let index=0 ; index < array.length ; index++){

index: toma el rol de iterador el que nos permitirá hacer el bucle

array.length: puede o no ser un array o puede ser un numero o algo especifico

vendría a ser la condición a cumplir

}

~~~

\*\*Arrays\*\*

-----------

Un arreglo es una lista ordenada de elementos de cualquier tipo estos van dentro de [ ] corchetes y van separados por , comas.

\*\*Para mostrar un elemento:\*\*La posición de un elemento de un Array/Arreglo sirve para obtener un elemento y se la puede obtener pasando le la posición del elemento que queremos obtener array[1] devolverá el segundo elemento (porque siempre empieza a contar desde 0).

~~~

let myArray = ["Marco","Maria","Pedro"]

myArray.length = 3

~~~

ForOf()

-------

Me permite recorrer un array de elementos , yo defino el nombre de la variable que recibe cada elemento en este caso nombre

~~~

let myArray = ["Marco","Maria","Pedro"]

for(const nombres of myArray){

nombres = "Cambiado"

}

Resultado: myArray = ["Cambiado","Cambiado","Cambiado"]

~~~

join()

-------

Me permite unificar todos los elementos de un array en una sola cadena de texto

~~~

let myArray = ["Marco","Maria","Pedro"]

Resultado: myArray = "Marco Maria Pedro"

~~~

Push()

-------

\*\*Para agregar un elemento:\*\* Con array.push se le puede introducir un nuevo elemento al array en su ultima posición

\*\*Para Reemplazar un elemento:\*\* Se le puede pasar la el elemento nuevo a ingresar en la posición en la que queremos reemplazar.

array[1] = "Germán"; // reemplazamos el elemento en la posición 1

~~~

let myArray = ["Marco","Maria","Pedro"]

myArray.push("Fer")

Resultado: myArray = ["Marco","Maria","Pedro","Fer"]

~~~

Splice()

--------

\*\*Para colocar en una posición especifica:\*\* Utilizamos el método splice() en donde recibe 3 parámetros 1ro es la posición, 2do es para borrar elementos, 3ro el elemento que queremos agregar splice(0,0,"nuevo elemento")

~~~

let myArray = ["Marco","Maria","Pedro"]

myArray.splice(0,0,"Fer")

Resultado: myArray = ["Fer","Marco","Maria","Pedro"]

~~~

Splice()

---------

\*\*Para Borrar un elemento:\*\* Usando Splice() en el segundo parámetro recibe la cantidad de elementos a borrar y en el primero la posición del elemento que queremos borrar

~~~

let myArray = ["Marco","Maria","Pedro"]

myArray.splice(0,2)

Resultado: myArray = ["Pedro"]

~~~

pop()

------

Para quitar el ultimo elemento del array.

~~~

let myArray = ["Marco","Maria","Pedro"]

myArray.pop()

Resultado: myArray = ["Marco","Maria"]

~~~

unshift(nuevo elemento)

---------

Para agregar en la primera posición un nuevo elemento.

~~~

let myArray = ["Marco","Maria","Pedro"]

myArray.unshift("Juan")

Resultado: myArray = ["Juan","Marco","Maria","Pedro"]

~~~

shift()

--------

Elimina el primer elemento del array.

~~~

let myArray = ["Marco","Maria","Pedro"]

myArray.shift()

Resultado: myArray = ["Maria","Pedro"]

~~~

indexOf(nombre del elemento)

----------

Me devuelve la posición, el indice del elemento. Si no existe el elemento me devuelve un -1

~~~

let myArray = ["Marco","Maria","Pedro"]

myArray.indexOf("Maria")

Resultado: myArray = 1

~~~

concat(array)

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

Sirve para juntar dos arreglos en uno solo. En un arreglo nuevo

~~~

let myArray = ["Marco","Maria","Pedro"]

let myArray2 = ["Pablo","Javier"]

myArray.concat(myArray2)

Resultado: myArray = ["Marco","Maria","Pedro","Pablo","Javier"]

~~~

includes()

--------

Me permite encontrar un valor si existe o no en el array

~~~

let myArray = ["Marco","Maria","Pedro"]

myArray.includes("Pedro")

Resultado: myArray = true

~~~

reverse()

----------

Me invierte el orden de el array

~~~

let myArray = ["Marco","Maria","Pedro"]

myArray.reverse()

Resultado: myArray = ["Pedro","Maria","Marco"]

~~~

sort()

------

Me ordena el Array. Ordenación no es necesariamente estable. El modo de ordenación por defecto responde a la posición del valor del string de acuerdo a su valor Unicode ( universalidad, uniformidad y unicidad.).

- compareFunction: Opcional. Especifica una función que define el modo de ordenamiento. Si se omite, el array es ordenado atendiendo a la posición del valor Unicode de cada caracter, según la conversión a string de cada elemento.

- firstEl: El primer elemento a comparar.

- secondEl: El segundo elemento a comparar.

~~~

let myArray = ["Marco","Maria","Pedro"]

myArray.sort()

Resultado: myArray = ["Marco","Maria","Pedro"]

~~~