Depuración en JavaScript

1.- Usa la consola de JavaScript para comprobar el valor de una variable:

let a = 5;
a++;
console.log(a);
	Utiliza el método console.log() para imprimir el valor de la variable a.

2.- Entendiendo las diferencias entre la consola de freeCodeCamp y la del navegador:

let output = "Get this to show once in the freeCodeCamp console and not at all in the browser console";
console.log(output);
console.clear();
	Abre la consola de tu navegador. Muestra en consola el valor de la variable output. Finalmente, utiliza console.clear() para borrar la consola del navegador.

3.- Utiliza typeof para comprobar el tipo de una variable:

let seven = 7;
let three = "3";
console.log(seven + three);
 	
console.log(typeof seven);
console.log(typeof three);
	Agrega dos sentencias console.log() para comprobar el typeof de cada una de las variables en el código.

4.- Captura nombres de variables y funciones mal escritas:

let receivables = 10; let payables = 8; let netWorkingCapital = receivables - payables; console.log(`Net working capital is: ${netWorkingCapital}`);	Corrige los dos errores ortográficos en el código para que funcione el cálculo de netWorkingCapital.

5.- Captura paréntesis, corchetes, llaves y comillas sin cerrar:

let myArray = [1, 2, 3]; let arraySum = myArray.reduce((previous, current) => previous + current); console.log(`Sum of array values is: ${arraySum}`);	Corrige el código.

6.- Captura el uso mixto de comillas simples y dobles:

let innerHtml = " Click here to return home "; console.log(innerHtml);	Corrige el código.

7.- Captura el uso del operador de asignación en lugar del operador de igualdad:

let x = 7;
let y = 9;
let result = "to come";
 	
if(x == y) {
  result = "Equal!";
} else {
  result = "Not equal!";
}
 	
console.log(result);
	Corrige el código.

8.- Captura los paréntesis de apertura y cierre que faltan después de una llamada a una función.

function getNine() {
  let x = 6;
  let y = 3;
  return x + y;
}
 	
let result = getNine();
console.log(result);
	Corrige el código.

9.- Captura argumentos pasados en el orden incorrecto al llamar a una función:

function raiseToPower(b, e) {
  return Math.pow(b, e);
}
 	
let base = 2;
let exp = 3;
let power = raiseToPower(base, exp);
console.log(power);
	Corrige el código.

10.- Captura los errores por uno al utilizar indexación:

function countToFive() {
  let firstFive = "12345";
  let len = firstFive.length;
  for (let i = 0; i < len; i++) {
  console.log(firstFive[i]);
  }
}
countToFive();
	Corrige el código.

11.- Ten cuidado al reinicializar variables dentro de un bucle:

function zeroArray(m, n) {
  // Crea un arreglo de 2 dimensiones con m filas y n columnas de ceros
  let newArray = [];
  let row = [];
  for (let i = 0; i < m; i++) {
  // Agrega la fila número m a newArray
  let row =[];
  for (let j = 0; j < n; j++) {
  // Inserta n ceros a la fila actual para crear las columnas
  row.push(0);
  }
  // Inserta la fila actual, que ahora contiene n ceros, al arreglo
  newArray.push(row);
  }
  return newArray;
}
let matrix = zeroArray(3, 2);
console.log(matrix);
	Corrige el código para que devuelva una matriz 3x2 de ceros correcta, tal qué [[0,0],[0,0],[0,0]].

12.- Prevenir bucles infinitos con una condición terminal válida:

function myFunc() {
  for (let i = 1; i <= 4; i += 2) {
  console.log("Still going!");
  }
}
	Corrige el código para evitar el bucle infinito.