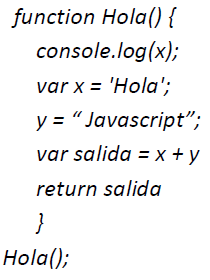
1. **Utiliza ‘use strict’ para detectar los errores que encuentras en el código siguiente y corrige esos errores.**

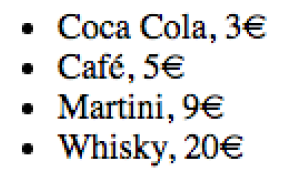


* <https://pgmonreal.github.io/DWEC/U1/P4/E1/index.html>

1. **Crea una función anónima que:**
2. Tome como parámetro un número al que debe asignar el valor por defecto de 18
3. Si el número es mayor o igual a 18 debe devolver “si”
4. Si el número es menor de 18 debe devolver “no”
5. Asigna la función anónima a la variable var MayorEdad
6. Prueba la función anterior con 3 llamadas distintas para probar el valor por defecto, la salida si y la salida no.

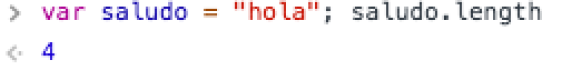
* <https://pgmonreal.github.io/DWEC/U1/P4/E2/index.html>

1. **Crea una función que solicite al usuario que introduzca su edad, la convierta a número y que emplee MayorEdad (ejercicio anterior) para mostrar por pantalla todos los artículos o sólo aquellos que no contienen alcohol, en caso de que sea menor de edad. Ejemplo artículos:**



* <https://pgmonreal.github.io/DWEC/U1/P4/E3/index.html>

1. **Crea una función que reciba 2 cadenas por parámetro.**
2. Dicha función imprimirá por consola qué cadena tiene mayor longitud. Para ello puedes usar la propiedad length tal y como muestra el ejemplo:



1. Si el tipo de algún parámetro no es string (typeof param !== "string"), debes imprimir un error.

**Llama a la función 3 veces con diferentes parámetros. En una de esas llamadas pásale por parámetro un valor que no sea string.**

* <https://pgmonreal.github.io/DWEC/U1/P4/E4/index.html>

1. **Empleando las funciones predefinidas que hemos visto, crea la función calculadora que:**
2. Solicite al usuario que introduzca la operación que quiere calcular (empleando operadores sencillos).
3. Calcule esa operación
4. Compruebe que el resultado no es infinito y que sí es un número y en ese caso, indique que se ha producido un error.
5. Muestre por pantalla el resultado si se trata de un resultado correcto.

* <https://pgmonreal.github.io/DWEC/U1/P4/E5/index.html>

1. **Crea un array con 10 elementos de distinto tipo (booleano, numeric y string). Recorre el array y muestra por pantalla el índice del elemento, el contenido y el tipo de dato que contiene y la longitud del array. A partir del array anterior, crea un nuevo array. Para ello convierte los elementos a números empleando Number(ar[i]) y añadiendo al array sólo aquellos que no sean NaN. Muestra los índices, elementos y longitud de este segundo array.**

* <https://pgmonreal.github.io/DWEC/U1/P4/E6/index.html>

1. **Dado un array con los días de la semana:**

var diasSemana = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"];

1. Crea una función que muestre los días de la semana por pantalla empleando for in
2. Crea una función que solicite al usuario el número de la semana y que le devuelva qué día es en letras. Asignado como valor por defecto el 4 en caso de que no se introduzca ningún valor.
3. Prueba la función del apartado b) pero comprueba primero que el valor que ha pasado el usuario no es infinito y es un número.

* <https://pgmonreal.github.io/DWEC/U1/P4/E7/index.html>

1. **Realiza la función concatena que dado dos valores, compruebe que son strings y que devuelva un único string en el que concatene los dos strings con un carácter en blanco entre ellos. Pasa la función anterior como parámetro junto a un nombre para crear la función saludador que debe mostrar el mensaje “Buenos Días XX”, siendo XX el nombre introducido como parámetro.**

* <https://pgmonreal.github.io/DWEC/U1/P4/E8/index.html>