Ejercicios Tema 2: Sintaxis del lenguaje JavaScript

# Actividad 2.1

Crea una función con la siguiente estructura:

function verCalificacion(nota) ; //nota es un número entero entre 0 y 10

La función debe usar internamente una estructura de tipo Switch y debe devolver un string que se corresponda con la nota que se le ha pasado como argumento: 0-4 (devuelve “INSUFICIENTE”), 5 (devuelve “SUFICIENTE”), 6 (devuelve “BIEN”), 7-8 (devuelve “NOTABLE”), 9-10 (devuelve “SOBRESALIENTE”). Si el valor está fuera del rango 0-10 debe devolver “VALOR INCORRECTO”.

Crea una función de testeo llamada test() que invoque a la función verCalificacion con un número entero del -1 al 11 y que vuelque el resultado por la consola.

# Actividad 2.2

Crea una función con la siguiente estructura:

function verCalificacionDecimal(nota) ; //nota es un número decimal entre 0 y 10

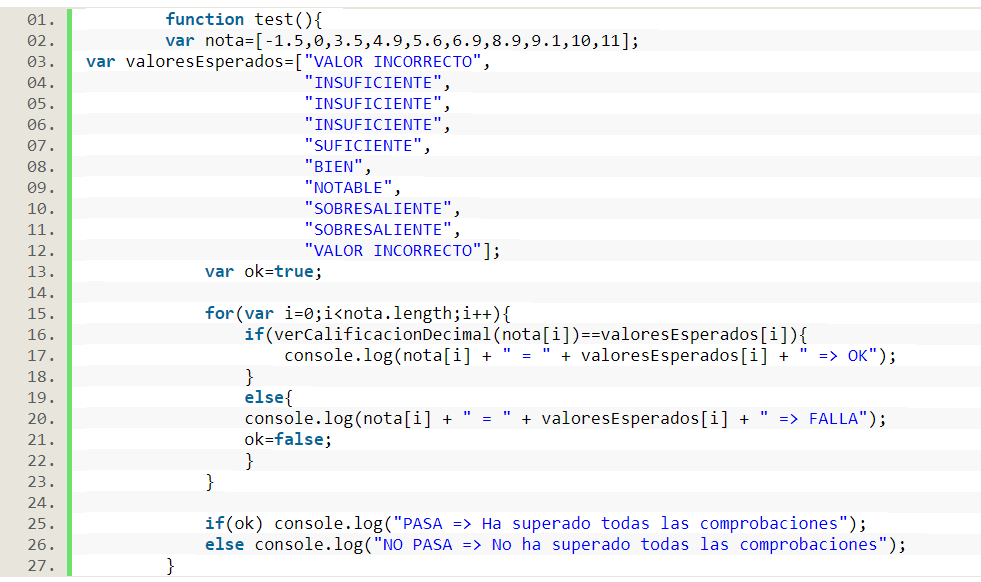
La función debe devolver un string que se corresponda con la nota que se le ha pasado como argumento: menor que 5 (devuelve “INSUFICIENTE”), entre 5 y menor que 6 (devuelve “SUFICIENTE”), entre 6 y menor que 7 (devuelve “BIEN”), entre 7 y menor que 9 (devuelve “NOTABLE”), entre 9 y 10 (devuelve “SOBRESALIENTE”). Si el valor está fuera del rango 0-10 debe devolver “VALOR INCORRECTO”.

Crea una función de testeo llamada test() que invoque a la función verCalificacionDecimal con una serie de notas y con una serie de valores esperados. Usa un par de arrays para introducir la información:

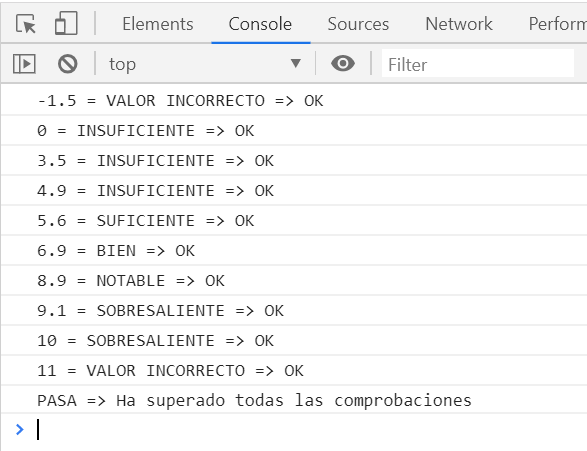
var nota=[-1.5,0,3.5,4.9,5.6,6.9,8.9,9.1,10,11];

var valoresEsperados=[“VALOR INCORRECTO”,”INSUFICIENTE”,”INSUFICIENTE”,”INSUFICIENTE”,”SUFICIENTE”,”BIEN”,”NOTABLE”,”SOBRESALIENTE”,”SOBRESALIENTE”,”VALOR INCORRECTO”];

Usa el siguiente código para testear tu función:



La salida por consola tras la ejecución del test debería tener un aspecto similar al de la captura:



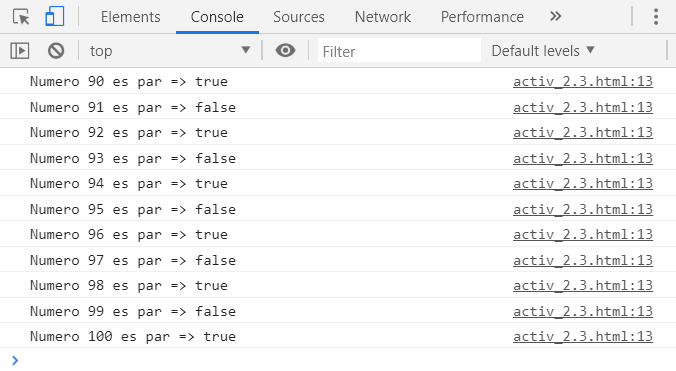
# Actividad 2.3 (activ\_2.3.html)

Crea una función con la siguiente estructura:

function comprobarEsPar(numero) ; //numero es un entero

La función tendrás como argumento un número entero y devolverá true si el número es para y false si es impar.

Crea una función test1() que muestre por consola qué números comprendidos entre 0 y 100 son pares. La función de testeo deberá mostrar un resultado similar al de la figura:



Prepara otra función de testeo llamada testeo2() de tipo caja negra con valores de entrada y valores esperados usando los siguiente arrays:

var valoresEntrada=[1,2,5,8,9,12,16];

var valoresEsperados=[false,true,false,true,false,true,true];

La función de testeo devolverá deberá mostrar si se ha superado el testeo para cada uno de los pares valor entrada – valor esperado. Se considerará que se ha superado el test si se superan todas las comprobaciones.

# Actividad 2.4

Crea una función con la siguiente estructura:

function europea2talla (europea);

La función anterior recibirá como entrada el argumento europea, que será un string con los siguientes posibles valores (“XS”, “S”, “M”, “L”, “XL”, “XXL”) y retornará una de las siguientes tallas (“Grande”, “Mediana”, “Pequeña”) según el siguiente criterio:

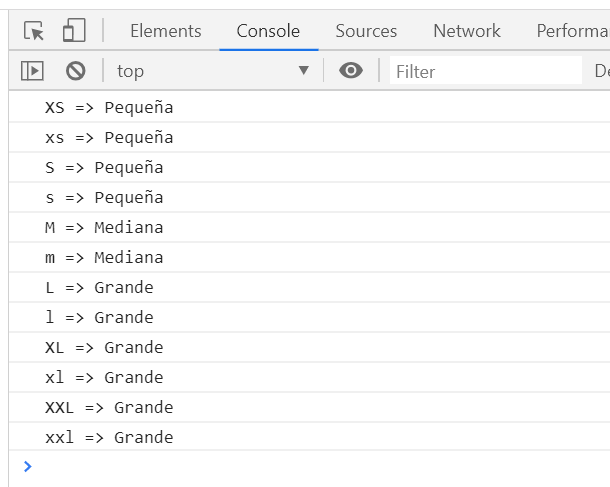
Pequeña = { “XS”, “S”}, Mediana = {“M”}, Grande = {“L”, “XL”, “XXL”}

Prepara la función para que la talla europea de entrada (“XS”, “L”, etc) pueda estar tanto en minúsculas como en mayúsculas.

Crea una función de testeo llamada test () que a partir de un vector con los siguientes valores testee por consola si la salida de función es la esperada:

var tallasEuropeas = [“XS”, “xs”, “S”, “s”, ”M”, ”m”, “L”, “l”, “XL”, “xl”, “XXL”, “xxl”];

La salida esperada por consola tras ejecutar la función test() es la siguiente:



Prepara una función de testeo similar a la utilizada en la actividad 2.11 de manera que a partir de los datos de entrada y de la salida esperada mostrados en la captura anterior testee de manera automática tu función indicando cuáles ha superado e indicando finalmente si ha superado o no todas las pruebas. Para ello usa los siguientes vectores:

var tallasEuropeas = [“XS”, “xs”, “S”, “s”,”M”,”m”, “L”, “l”, “XL”, “xl”, “XXL”, “xxl”];

var valoresEsperados =[“Pequeña”, ”Pequeña”, ”Pequeña”, ”Pequeña”, “Mediana”, ”Mediana” ”Grande”, ”Grande”, ”Grande”, ”Grande”, ”Grande”, ”Grande”];

# Actividad 2.5

Crea una función llamada hexa2decimal que reciba como argumento de entrada un número hexadecimal en formato string (p.e. “2FD”) y devuelva el correspondiente número convertido a decimal (en formato number). Los caracteres hexadecimales podrán estar tanto en mayúsculas como minúsculas (p.e. se podrá usar “2FD” o “2fd”). Prepara una función de testeo con dos arrays de valores de entrada y valores esperados (usa al menos 6 números hexadecimales de entrada mezclando tanto mayúsculas como minúsculas para los dígitos hexadecimales). Para preparar los arrays del testeo puedes usar la calculadora, conversores online, etc.