PRÁCTICA 2.02 Más sintaxis con funciones

Normas de entrega

- En cuanto al código:
 - en la presentación interna, importan los comentarios, la claridad del código, la significación de los nombres elegidos; todo esto debe permitir considerar al programa como autodocumentado. No será necesario explicar que es un if un for pero sí su funcionalidad. Hay que comentar las cosas destacadas y, sobre todo, las funciones y clases empleadas. La ausencia de comentarios será penalizada,
 - en la presentación externa, importan las leyendas aclaratorias, información en pantalla y avisos de ejecución que favorezcan el uso de la aplicación,
 - si no se especifica lo contrario, la información resultante del programa debe aparecer en la consola del navegador console.log(información),
 - los ejercicios deben realizarse usando JavaScript ES6 y usar el modo estricto (use strict)
 No se podrá utilizar jQuery ni cualquier otra biblioteca (si no se especifica lo contrario en el enunciado),
 - para el nombre de variables, constantes y funciones se utilizará lowerCamelCase,
- En cuanto a la **entrega** de los archivos que componen los ejercicios:
 - todos los ejercicios en una carpeta (creando las subcarpetas necesarias para documentación anexa como imágenes o estilos) cuyo nombre queda a discreción del discente.
 - el nombre de los ficheros necesarios para resolver el ejercicio será el número de ejercicio que contenga,
 - el código contendrá ejemplos de ejecución, si procede, y
 - · la carpeta será comprimida en formato ZIP y será subida a Aules de forma puntual.

Ejercicio 1 - Sumando

Crea una función que permita sumar todos los números pasados como parámetros. Deberá seguir estas premisas:

- comprobará si son de tipo numérico antes de realizar la operación, si no lo es, lanzará un mensaje de error por consola y detendrá la ejecución del programa,
- debe tener como mínimo dos parámetros o no se ejecutará. Debe insultar informar al usuario.
- la salida debe estar debidamente formateada.

Nota: usa el pseudoarray arguments.

Ejercicio 2 - Multiplicando

Genera una función que imprima por consola las tablas de multiplicar de los números que van desde el indicado como parámetro (entero positivo mayor a dos) hasta dos. Para ello recibe dos parámetros: un número y una función.

- La función pasada como parámetro será la que calcule la tabla de multiplicar. Habrá que diseñar primero esa función y luego pasarla como parámetro. Esta función recibe un número como parámetro e imprime su tabla de multiplicar (debidamente formateada).
- Por ejemplo, si se ejecuta **tablas(5**, **multiplicar)** imprimirá por consola las tablas del dps. La del tres, la del cuatro y la del cinco, eso sí, debidamente formateadas.

Ejercicio 3 - Sumando vectores

Escribe la función **calcular()** para que maneje tres arrays (no serán pasados como parámetros a esta función sino declarados dentro de ella). Lo que sí se pasará como parámetros serán tres funciones con el siguiente comportamiento:

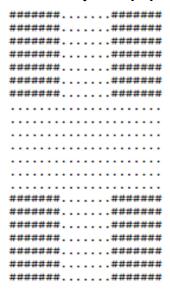
- una para generar un array con números aleatorios comprendidos entre 1 y 100, la cantidad de números generados será pasado como parámetro,
- otra que recibe dos arrays como parámetros y genera un array que contenga la suma de las cifras que ocupen posiciones contrarias, es decir, la primera posición del primero sumada a la última posición del segundo (los arrays deben tener la misma longitud),
- y la última para imprimir un array por consola debidamente formateado.

La ejecución del programa se realizará a través de la función **calcular()** del siguiente modo: **calcular(funcion1, funcion2, funcion3)**;

Nota: para generar números aleatorios en JavaScript: (Math.floor (Math.random () *100) +1);

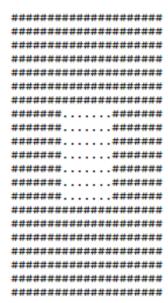
Ejercicio 4 - Cruz

Crea una función que reciba como parámetro el lado de un cuadrado y dibuje una cruz griega como la de la imagen. Asegúrate antes de dibujar que el número pasado es múltiplo de tres (para simplificar el dibujo). El resultado de **dibujarCruz(21)**; es el siguiente:



Ejercicio 5 - Marco

Basándote en el ejercicio anterior, crea una función que recibe dos parámetros. El primero es el tamaño exterior del marco (tamaño del lienzo) y el segundo es el tamaño interior (tamaño del área de puntos). El resultado de **dibujarMarco(21,7)**; es el siguiente:



Nota: Recuerda que la consola del navegador agrupa las filas iguales. Añade un contador al principio o al final de la línea para que el dibujo se imprima.