

---

## **TÉCNICO** SUPERIOR EN DESARROLLO DE APLICACIONES WEB

### **PRUEBA PRÁCTICA** MÓDULO DESARROLLO WEB ENTORNO CLIENTE

#### **Evaluación** Primera

#### **Práctica:** Programación Básica

**Grupo:2°C**

Lee bien todo lo que se indica en este enunciado y procura respetarlo.

Si tienes alguna duda o crees que hay alguna incoherencia, avisa al profesor para que te lo solucione.

#### **Normas de Realización**

- a) El ejercicio se realizará en clase en el tiempo indicado para ello. No se va a dejar tiempo para acabarlo en casa.
- b) Se debe usar el IDE VS Code.
- c) No se pueden utilizar herramientas que generen código para completar la práctica. Si se detecta dicho uso, se considerará plagio.
- d) Sólo se permite usar las herramientas del lenguaje que se han dado hasta este momento. Usa funciones JS que estén en los apuntes y/o se hayan explicado en clase para todos. Ante cualquier duda: PREGUNTA
- e) Se valorará la legibilidad del código: sangrado de líneas correcto, uso de comentarios, nombre de variables adecuado...

#### **Otras consideraciones:**

Para la realización de este ejercicio, puedes consultar tus apuntes, pero NO fuentes de internet.

## Entrega

El ejercicio puntuará para la evaluación práctica del 1er Trimestre de la asignatura.

- Al finalizar el ejercicio debes entregar un **archivo comprimido en .zip .rar** que contenga **una carpeta con dos archivos: index.html y el fichero miScript.js**. Ese archivo comprimido debe nombrarse de la siguiente manera: Apellido1\_Apellido2\_Nombre.zip Ej: Paredes\_Delgado\_Mariceli.zip Sube ese archivo comprimido a la tarea indicada en el aula virtual
- **Recuerda:** se puntúa lo que se entrega en la plataforma.

## Copia o plagio

Tal como se indica en los estatutos del centro, está totalmente prohibido copiar código de otro/a compañero/a, de internet y/o de una IA. Si se detecta y se demuestra este tipo de comportamiento, la calificación de la actividad será de 0 puntos, pudiendo quedar el trimestre suspenso para junio si el profesor lo estima oportuno.

## DESARROLLO:

Genera un fichero index.html que contenga 4 botones, cada uno de ellos correspondiente a un ejercicio

```
<body>
  <button onclick="Ejercicio1()">Ejercicio 1</button>
  <button onclick="Ejercicio2()">Ejercicio 2</button>
  <button onclick="Ejercicio3()">Ejercicio 3</button>
  <button onclick="Ejercicio4()">Ejercicio 4</button>
</body>
```

Genera un fichero miScript.js y crea una función para cada ejercicio. Además, puedes añadir tantas funciones como necesites para cada ejercicio.

```
function Ejercicio1 () {
  |
}
```

**Ejercicio 1 (1 punto).** Desarrolla un script que, al pulsar el botón correspondiente a este ejercicio, pida al usuario una cantidad en metros y muestre mediante un **alert** su equivalente en kilómetros, según el formato del siguiente ejemplo:

*5000 metros equivalen a 5 kilómetros*

Controla que la cantidad a convertir sea mayor a 0 y sea numérica. Si no es así, vuelve a pedir esa cantidad.

**Ejercicio 2 (2 puntos):** Desarrolla un script que, al pulsar el botón correspondiente a este ejercicio, pida 10 números enteros al usuario. Al finalizar, muestra un **alert** indicando cuántos de los números son menores de 10, cuántos están entre 10 y 20 (inclusive ambos), cuántos están entre 21 y 30 (inclusive ambos), y cuántos son mayores de 30. Asegúrate de que los valores introducidos sean números enteros válidos. Si no lo son, vuelve a pedir ese número.

**No está permitido el uso de arrays.**

Ejemplo de números introducidos: 5,12,25,19,8,15,30,22,40,9

Alert → Menores de 10: 3. Entre 10 y 20: 3. Entre 21 y 30: 3. Mayores de 30: 1.

**Ejercicio 3 (3 puntos):** Desarrolla un script que, al pulsar el botón correspondiente a este ejercicio, pida al usuario 7 números enteros positivos mediante prompt y los guarde en un array. A continuación:

**Nota: Haz cada apartado por separado. Si no puedes resolver un apartado pasa al siguiente, aunque tengas que generar un array de forma manual en código.**

- a) (0,25 puntos) Muestra por consola el array completo y la suma de todos sus elementos.

- b) (0,25 puntos) Pide al usuario dos nuevos números y reemplaza en el array los elementos en la primera y última posición por estos dos números, respectivamente. Muestra el **array actualizado** por consola.

A partir de aquí se trabaja con el **array actualizado obtenido en el apartado b)**

- c) (0,5 puntos) Calcula y muestra por consola la media de los números en el **array actualizado** redondeando a 2 decimales.
- d) (1 punto) Muestra por consola todos los números en el array que son menores que la media y también aquellos mayores que la media. Muéstralo como cadenas separadas por comas.

Ejemplo suponiendo media = 4

*Numeros Mayores Media: 8,13,28,8*

*Numeros Menores Media: 3,6,1*

- e) (1 punto) Pide al usuario un número adicional y verifica si está presente en el array. Si está, muestra por consola todas las posiciones en las que aparece como cadena separada por comas. Si no está, indica que el número no se encuentra.

Ejemplo

*El número x aparece en las posiciones: 3,4*

*El número x no se encuentra*

**Ejercicio 4 (4 puntos):** Desarrolla un script que, al pulsar el botón correspondiente a este ejercicio, defina un array bidimensional de tamaño NxN ( $2 \leq N \leq 5$ ), donde N es introducido por el usuario mediante prompt. Este array se llamará matriz. A continuación, realiza las siguientes operaciones:

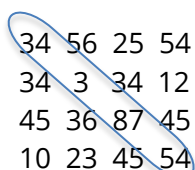
- a) (0,5 puntos) Rellena la matriz con valores aleatorios entre 1 y 99.
- b) (0,75 puntos) Muestra por consola el contenido de la matriz respetando el número de filas y columnas. Es decir, todos los valores de una fila deben aparecer en horizontal separados por espacios, y debajo la siguiente fila. Todos los valores deben ocupar 2 espacios; si un valor ocupa solo 1 espacio, añádele un 0 delante.

Ejemplo (3x3)

```
03 56 78
47 08 24
78 34 05
```

- c) (0,5 puntos) Calcula la suma de todos los elementos de la matriz y muestra un mensaje por consola indicando ese valor.
- d) (1,25 puntos) Crea una copia de la matriz llamada "matrizCopia", donde todos los valores de la matriz original menores o iguales a 50 deben ser reemplazados por 50 en la matriz Copia. Muestra matrizCopia por consola siguiendo las reglas del apartado b. **Nota: Para hacer la copia no puedes usar slice(), concat() ni el operador spread (...), sino que debes usar bucles.**
- e) (1 punto) Calcula la suma de las diagonales de ambas matrices (matriz y matrizCopia) y muestra un mensaje indicando cuál de las dos matrices tiene una suma diagonal mayor. Si ambas tienen la misma suma diagonal, indícalo.

En la siguiente imagen tienes una representación de la diagonal de una matriz:



```
34 56 25 54
34 3 34 12
45 36 87 45
10 23 45 54
```

Suma de la diagonal = 178 (34 + 3 + 87 + 54)