



Relación I de ejercicios BÁSICOS

- 1) Realizar un programa que nos pida números usando el método `prompt` hasta que escribamos fin o FIN. Seguidamente usando el método `confirm(mensaje)` nos preguntará si estamos seguros de aseguraremos que al pulsar en aceptar detenemos la petición de números. Si pulsamos en el botón cancelar del método `confirm` entonces continuaremos solicitando números. En caso de introducir texto mostrará mensaje de error con `alert`. Una vez concluida la petición de números realizará los siguientes cálculos que introducirá dentro del documento html usando:

```
document.write("<h1>cálculo pedido:</h1>");
```

Cálculos:

- Suma total de todos los números introducidos.
 - Media Aritmética de los números introducidos.
 - Máximo y mínimo.
 - Mediana. (La mediana de un conjunto de números es el número que se encuentra en la mitad dentro del conjunto de números (después que los números han sido arreglados del menor al mayor) - o, si hay un número par de datos, la mediana es el promedio de los dos números medios.
 - Moda. (Valor que más veces se repite o que tiene mayor frecuencia de aparición).
- 2) Crear el objeto ***trabajador*** con las propiedades:
- a) DNI, nombre, apellidos, tfno, sueldo_Bruto, retenciones(%), Sueldo_Neto.
 - b) La propiedad `Sueldo_Neto` debe ser una función que calcule el sueldo después de descontar el tanto por ciento de retención al sueldo bruto.
 - c) Solicitar a través de un formulario los datos para almacenarlos.
 - d) Utilizar `Object.keys(nombre_objeto)` , `Object.values(nombre_objeto)` , `for ... in` , etc para mostrar en el propio documento HTML los datos almacenados.
 - e) Mejorar el proyecto creando un ARRAY de Objetos trabajadores, realizando las modificaciones o mejoras necesarias para poder añadir trabajadores y buscar datos de un trabajador (por ejemplo: buscar por DNI o por tfno mostrando los datos del trabajador en la pantalla)
 - f) Calcular el coste total de sueldos (brutos) que ha de pagar la empresa a los trabajadores.



- 3) Crear una función que calcule el índice de masa corporal. Para ello a través de un formulario en HTML realizar las preguntas necesarias:
- Se pedirá el Peso en Kgr.
 - Se pedirá la altura en cm.
 - Comprobaremos que hemos introducido números.
 - Pasaremos los cm a metros.
 - Aplicaremos la Fórmula: **peso (kg) / [estatura (m)]²**
 - Con el número obtenido y fijándonos en la tabla siguiente le indicaremos cuál es su nivel de masa corporal y si se encuentra en niveles buenos o malos.
 - Dicha información la mostraremos en la propia página web a través del comando:

```
document.write("<h1>Su Índice de masa corporal es:</h1>");
```

IMC	Nivel de peso
Por debajo de 18.5	Bajo peso
18.5 – 24.9	Normal
25.0 – 29.9	Sobrepeso
30.0 o más	Obesidad

- 4) Crear un array que almacene los meses del año. Adicionalmente crear un array que almacene el nombre de las cuatro estaciones. Generar un número aleatorio entre 1 y 12 para seleccionar uno de los meses del año. Seguidamente utilizando las estructuras de control (if, switch, etc) asociar el mes obtenido con la estación que le corresponde.
- Como mejoras usar funciones / funciones flecha, etc.
- 5) Dado el siguiente array ["Nombre", 54 , true, "Apellidos", "Teléfono", 29, true, "Edades", "Datos", "999"]
- Crear una función que recorra el array, sumando todos los números que se encuentre.
 - Crear una función que cree un array formado sólo por las palabras que encuentre del array de partida. Dichas palabras serán almacenadas en el array en orden según el número de caracteres que posea cada palabra, de menor a mayor. En caso de tener el mismo número de caracteres dos palabras, se ordenarán por orden alfabético.
- 6) Crear un proyecto donde introduzcamos a través de un formulario en HTML un campo de texto y un botón de enviar. En el campo de texto escribiremos un DNI (8 dígitos y letra). Seguidamente al pulsar el botón enviar, calculará la letra del DNI.

El cálculo de dicha letra se realiza a través del siguiente algoritmo: Teniendo un array con las siguientes letras:

```
var Letras = ['T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E', 'I'];
```



Calculamos el resto de dividir nuestro número de DNI entre 23. En función del resto obtenido así elegimos la letra. (Si el resto es 0, entonces la letra es 'T').

Condicionantes a aplicar:

- a) En primer lugar, se debe de comprobar que el usuario ha introducido 8 dígitos y una letra al final en una sola instrucción.
 - b) También en una sola instrucción se debe comprobar si el número es menor que 0 o mayor que 99999999. Si ese es el caso, se muestra un mensaje al usuario indicando que el número proporcionado no es válido y el programa no muestra más mensajes.
 - c) Si el número es válido, se calcula la letra que le corresponde según el método explicado anteriormente.
 - d) Una vez calculada la letra, se debe comparar con la letra indicada por el usuario. Si no coinciden, se muestra un mensaje al usuario diciéndole que la letra que ha indicado no es correcta. En otro caso, se muestra un mensaje indicando que el número y la letra de DNI son correctos.
 - e) No aplicar aún Expresiones Regulares para Gestionar las restricciones al texto utilizado por el usuario.
- 7) Haciendo uso del bucle **for** calcular el factorial de un número. Recuerdo que el factorial de un número $n! = n \times (n-1) \times \dots \times 2 \times 1$. Por tanto el $5! = 5 \times 4 \times 3 \times 2 \times 1$.
- Nota:** Como versión dos, crear una función recursiva que permita realizar el cálculo. Recuerdo que una función recursiva es aquella que en su construcción se llama a sí mismo.
- 8) Realizar una función flecha a la que se le pase una cadena de texto con letras en mayúscula y minúscula. Seguidamente mostrará por pantalla aquellas **letras** que se encuentran en **Mayúscula**, y preguntará si queremos convertirla a minúscula. Dicha pregunta la realizará por cada letra mayúscula que encuentre. Una vez finalizado, me mostrará la cadena resultante con las letras mayúsculas que el usuario ha decidido que permanezcan o no.
- 9) Definir una función que determine si la cadena de texto que se le pasa como parámetro es un palíndromo, es decir, si se lee de la misma forma desde la izquierda y desde la derecha. Ejemplo de palíndromo complejo: **"La ruta nos aporoto otro paso natural"**.
- 10) Realizar un programa en JavaScript que simule el lanzamiento de dos dados. Repetir dicha acción 1000 veces, almacenando en un array la suma de ambos dados. Por último, indicar cuántas veces han salido los ojos de serpiente (1 – 1) y el seis doble (6 – 6).