# Ejercicios sobre estructuras de datos

## Meses

Crear un array llamado meses y que almacene el nombre de los doce meses del año. Mostrar por pantalla los doce nombres utilizando la función `console.log()`.

## Operaciones con elementos

A partir del siguiente array que se proporciona: `var valores = [true, 5, false, "hola", "adios", 2];`

1. Determinar cuál de los dos elementos de texto es mayor
2. Utilizando exclusivamente los dos valores booleanos del array, determinar los operadores necesarios para obtener un resultado true y otro resultado false
3. Determinar el resultado de las cinco operaciones matemáticas realizadas con los dos elementos numéricos

## Letra del DNI

El cálculo de la letra del Documento Nacional de Identidad (DNI) es un proceso matemático sencillo que se basa en obtener el resto de la división entera del número de DNI y el número 23. A partir del resto de la división, se obtiene la letra seleccionándola dentro de un array de letras.

El array de letras es:

```
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', 'T'];
```

Por tanto si el resto de la división es `0`, la letra del DNI es la `T` y si el resto es `3` la letra es la `A`. Con estos datos, elaborar un pequeño script que:

1. Almacene en una variable el número de DNI indicado por el usuario y en otra variable la letra del DNI que se ha indicado.

2. En primer lugar (y 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.
3. Si el número es válido, se calcula la letra que le corresponde según el método explicado anteriormente.
4. 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.

## Factorial

El factorial de un número entero n es una operación matemática que consiste en multiplicar todos los factores `n x (n-1) x (n-2) x ... x 1`. Así, el factorial de `5` (escrito como `5!`) es igual a: `5! = 5 x 4 x 3 x 2 x 1 = 120`

Utilizando la estructura `for`, crear un script que calcule el factorial de un número entero.

## Par o Impar

Escribir el código de una función a la que se pasa como parámetro un número entero y devuelve como resultado una cadena de texto que indica si el número es par o impar. Mostrar por pantalla el resultado devuelto por la función.

## Mayúsculas/Minúsculas

Definir una función que muestre información sobre una cadena de texto que se le pasa como argumento. A partir de la cadena que se le pasa, la función determina si esa cadena está formada sólo por mayúsculas, sólo por minúsculas o por una mezcla de ambas.

## Palíndromos

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 aportó otro paso natural".

## Trabajando con objetos (1)

Definir la siguiente jerarquía de objetos, haciendo uso de los prototipos de JavaScript:

- Objeto `Persona` con las propiedades `nombre`, `edad` y género, y el método `obtDetalles()`, que muestra por pantalla las propiedades de la persona.
- Objeto `Estudiante`, que hereda de `Persona`, e incluye las propiedades `curso` y `grupo` y el método `registrar()`.
- Objeto `Profesor`, que hereda de `Persona`, e incluye las propiedades `asignatura` y `nivel` y el método `asignar()`.

Crear los objetos y casos de prueba necesarios para comprobar el correcto funcionamiento de la jerarquía.

## Trabajando con objetos (2)

Repetir el ejercicio anterior usando clases en JS

## Trabajando con objetos (y 3)

Queremos hacer una aplicación en JavaScript para gestionar edificios con la información de la situación del edificio y de los propietarios de cada piso. Para ello queremos almacenar la siguiente información de cada edificio:

- nombre.
- calle.
- número.
- código postal.
- plantas del edificio (dentro de cada planta tendremos un número de puertas y para cada puerta almacenaremos el nombre del propietario).

Queremos:

- [ ] Crear un objeto que nos permita instanciar edificios. Cada vez que instanciamos un edificio le pasaremos la calle, número y código postal como parámetros. Se pide además crear los siguientes métodos para el objeto Edificio:

    - [ ] agregarPlantasYPuertas(numplantas, puertas) // Se le pasa el número de plantas que queremos crear en el piso y el número de puertas por planta. Cada vez que se llame a este método, añadirá el número de plantas y puertas indicadas en los parámetros, a las que ya están creadas en el edificio.
    - [ ] modificarNumero(numero) // Se le pasa el nuevo número del edificio para que lo actualice.
    - [ ] modificarCalle(calle) // Se le pasa el nuevo nombre de la calle para que lo actualice.
    - [ ] modificarCodigoPostal(codigo) // Se le pasa el nuevo número de código postal del edificio.
    - [ ] imprimeCalle // Devuelve el nombre de la calle del edificio.
    - [ ] imprimeNumero // Devuelve el número del edificio.
    - [ ] imprimeCodigoPostal // Devuelve el código postal del edificio.
    - [ ] agregarPropietario(nombre,planta,puerta) // Se le pasa un nombre de propietario, un número de planta y un número de puerta y lo asignará como propietario de ese piso.
    - [ ] imprimePlantas // Recorrerá el edificio e imprimirá todos los propietarios de cada puerta.

- [ ] Cada vez que se crea un edificio que muestre automáticamente un mensaje del estilo: ***"construido nuevo edificio en calle: xxxxxx, no: xx, CP: xxxxx."***

- [ ] Cada vez que se añada un propietario a un piso de un edificio que muestre un mensaje del estilo: ***"xxxxxxxx es ahora el propietario de la puerta x de la planta x."***