

---

### Examen práctico

1.- Colocar las condiciones necesarias para que se cumpla la salida en consola. **(10 pts)**

```
var numero1 = 5;
var numero2 = 8;

if(...) {
    console.log("numero1 no es mayor que numero2");
}

if(...) {
    console.log("numero2 es positivo");
}

if(...) {
    console.log("numero1 es negativo o distinto de cero");
}

if(...) {
    console.log("Incrementar en 1 unidad el valor de numero1 no lo hace mayor o igual que numero2");
}
```

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

Utilizando la estructura crear un script que calcule la factorial de un número entero.

**(10 pts)**

3.- 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. **(20 pts)**

---

---

**4.-** Definir una función 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 pts)**

**5.-** Definir la siguiente jerarquía de objetos: **(50 pts)**

1. Clase `Persona` con las propiedades `nombre` y `edad`
    - 1.1. Crear el método `obtDetalles()` este método imprimirá en consola todos los atributos de la persona.
    - 1.2. El constructo recibirá `nombre` y `edad` y los asignará a los atributos internos.
  2. Clase `Estudiante`, hereda de `Persona`, e incluye las propiedad de `calificación`
    - 2.1. Sobrecribir el método `obtDetalles()` para imprimir en consola los detalles de persona y la calificación
  3. Clase `Profesor`, que hereda de `Persona`, e incluye las propiedades `asignatura` y `nivel`.
    - 3.1. Inicializar en el constructor todos los parámetros de persona así como asignatura y nivel.
    - 3.2. El constructor tendrá que estar habilitado para que en caso de que no se especifique la asignatura esta sea igual a JS y el nivel igual a básico.
    - 3.3. Sobrecribir el método `obtener detalles` para imprimir todas las propiedades de persona más todas las propiedades de profesor.
  4. Clase `grupo` que contendrá las propiedades `profesor` `promedio` y `estudiantes`.
    - 4.1. Crear un constructor que acepte y asigne el objeto profesor y el arreglo de estudiantes que pertenecen al grupo.
    - 4.2. Crear el método `calificar` que itere sobre el arreglo de estudiantes pertenecientes a ese grupo. En cada iteración se deberá asignar una calificación al estudiante aleatoria (`Math.random() * 10`)
    - 4.3. Crear un método que calcule el promedio de todos los estudiantes en ese curso.
    - 4.4. Crea el método `obtdetalles()` que imprime los detalle del profesor y los detalles de todos los estudiante.
  5. Todo junto.
    - 5.1. Crea un arreglo llamado `estudiantes`.
      - 5.1.1. Agrega 6 estudiantes con diferentes nombres y edades.
    - 5.2. Crea un objeto de tipo profesor el cual solo inicializarás con nombre y edad
    - 5.3. Crea un grupo al cual agregas el arreglo de estudiantes
-

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

**Bonus:** Escribe una function que genere todas las posibles combinaciones con las letras de dicho string. (La entrada la puede limitar a 3 letras) **(15 puntos)**

Ejemplo entrada : casa

Ejemplo salida: c,ca,cas,casa,a, ac, as,...

---