

Se pide:

1. Desarrollar una función que reciba un texto y lo devuelva con la inicial en mayúscula y el resto en minúscula.
2. Definir un objeto "Propietario" que permita almacenar los datos de un propietario de una mascota.
  - a. Con las siguientes propiedades:
    - i. Nombre del propietario
    - ii. Tipo de mascota que tiene. (Perro, Gato, Iguana, Cobaya....)
    - iii. Edad de su mascota.
    - iv. Nombre de su mascota.
  - b. Con los siguientes métodos:
    - i. Aquellos que permiten dar valor a sus propiedades.
    - ii. Aquellos que permiten obtener los valores de sus propiedades.
  - c. El objeto inicializa sus propiedades a través de sus argumentos.
3. Definir un array "mascotas" que almacena los datos de 5 mascotas. En dicho array, cada fila corresponde a una mascota y cada columna corresponde a cada uno de los datos de cada mascota:
  - a. Tipo de mascota. (Perro, Gato, Iguana, Cobaya....)
  - b. Edad de la mascota (valor numérico comprendido entre 1 y 175).
  - c. Nombre de la mascota.
  - d. Nombre del propietario
4. Crear un array "propietarios", paralelo al de mascotas, a partir de los datos almacenados en el array "mascotas". Dicho array contendrá tantos objetos de tipo "Propietario" como mascotas haya en el array "mascotas".
5. Solicitar al usuario los datos para nuevas mascotas hasta que el usuario decida finalizar con un \*:
  - a. El usuario debe introducir todos los datos de cada una de las mascotas **de una sola vez** con el siguiente formato: *[Nombre del propietario],[Nombre de la mascota],[Tipo de mascota],[Edad de la mascota]*.
  - b. Controlar que no haya duplicidad mascotas. Es decir, no puede haber 2 mascotas con el mismo nombre, edad, tipo y propietario.
  - c. Si se omite cualquiera de los datos se asumirá que coincide con el dato correspondiente a la última mascota introducida. No se pueden omitir todos los datos ya que no puede haber mascotas duplicadas.
  - d. El nombre del propietario, el nombre de la mascota y el tipo de mascota se podrán introducir en mayúsculas y/o minúsculas, pero se almacenará con la inicial en mayúscula y el resto en minúsculas.
  - e. Controlar mediante el uso de una expresión regular que la edad de la mascota esté comprendida entre 1 y 175 años.
  - f. El usuario deberá ser informado de los siguientes errores:
    - i. Se han omitido todos los datos.
    - ii. Se ha intentado cancelar.
    - iii. Se ha introducido una mascota duplicada.
    - iv. La edad de la mascota no está comprendida entre 1 y 175.
6. Ampliar los arrays "mascotas" y "propietarios" con los datos de cada una de las mascotas introducidas siempre y cuando se trate de una mascota diferente.
7. Al finalizar la entrada de datos con el \*, hacer un listado de todas las mascotas:
  - a. El listado deberá salir ordenando por tipo de mascota.
  - b. En el listado deberá salir en cada línea los datos de una mascota.
  - c. La salida de cada línea tendrá el siguiente formato:  
Tipo – Nombre – Edad – Año de nacimiento – Propietario

Valoración apartados		Puntuación	Observaciones
1	1		
2	0,10		
2.a	0,40 (0,1 cada una)		
2.b	0,80 (0,4 cada una)		
2.c	0,20		
3	0,50 (0,1 cada una)		
4	1 + 0,50 (si se usa una función)		
5	0,25		
5.a	0,25		
5.b	0,50		
5.c	0,50		
5.d	0,50		
5.e	0,50		
5.f	1 (0,25 cada 1)		
6	0,50		
7	0,20 + 0,30 (si se usa una función)		
7.a	0,30 + 0,30 (si se usa una función)		
7.b	0,20		
7.c	0,20		

La falta de comentarios se penaliza con un descuento de un 10% de la nota obtenida.

La incorrecta estructura del código se penaliza con un descuento de un 10% de la nota obtenida.

El enlace del archivo js fuera de la cabecera del documento se penaliza con un descuento de un 10% de la nota obtenida.