

1 Ejercicios (ahora siempre hay que testear)

- 1.1 Crea una jerarquía teórica de tres niveles de clases, la que quieras, sobre un caso que podría ser utilizado en un programa. Ejemplo: Animales -> Mamíferos -> Perros. Después, impleméntalo de manera práctica.
- 1.2 Realiza todos los ejercicios que quieras de la página:
<https://www.w3resource.com/javascript-exercises/javascript-object-exercises.php>
- 1.3 Realiza los siguientes ejercicios
(<https://medium.com/@andrey.igorevich.borisov/10-javascript-exercises-with-objects-8942cc502754>):
 - 1.3.1 isPlainObject – Crea una función que comprueba si un objeto es sólo un diccionario, y descarte que sea un array o null.
 - 1.3.2 MakePairs – Crea una función que convierte un objeto en un array de arrays por clave valor ({a: "suerte"} -> [["a", "suerte"]])
 - 1.3.3 Without – Crea una función en la que metas el string de una propiedad de un objeto y el objeto y te devuelva el objeto sin esa propiedad.
 - 1.3.4 IsEqual – Crea una función que compare si dos objetos son iguales
 - 1.3.5 Intersection – Crea una función que coja las propiedades que se repiten (sólo la llave) de varios objetos
- 1.4 De la página web: <https://github.com/Klod19/JS-Classes-Exercises>, haz los ejercicios que quieras. Uno que sí que hay que hacer es este:
 - 1) Create a TV class with properties like brand, channel and volume. Specify brand in a constructor parameter. Channel should be 1 by default. Volume should be 50 by default.
 - 2) Add methods to increase and decrease volume. Volume can't never be below 0 or above 100.
 - 3) Add a method to set the channel. Let's say the TV has only 50 channels so if you try to set channel 60 the TV will stay at the current channel.
 - 4) Add a method to reset TV so it goes back to channel 1 and volume 50. (Hint: consider using it from the constructor).
 - 5) It's useful to write a status, that returns info about the TV status like: "Panasonic at channel 8, volume 75".
- 1.5 Suponiendo que tienes esta clase (escrita en Python, la tienes que convertir a javascript):

```
class Vehicle:
    def __init__(self, name, max_speed, capacity):
        self.name = name
        self.max_speed = max_speed
        self.mileage = mileage

    def seating_capacity(self, capacity):
        return f"The seating capacity of a {self.name} is {capacity} passengers"
```

Crea una clase llamada Bus que hereda de esta y que:

- Tenga como valor por defecto de la propiedad seating_capacity a 50.
- También debe tener un método que sirva para imprimir todas las direcciones que tiene en su itinerario (mínimo 10 direcciones)
- Si ahora quisiéramos instanciar varios objetos Bus con las direcciones de pamplona de la 16 y la 4, por ejemplo, ¿Cómo harías para meterle las direcciones? Recrea el ejemplo

1.6 Crea una clase para transformar números naturales a números romanos y viceversa (máximo que admite 1000, si no, da un error)

1.7 Crea una clase para trabajar diferentes transformaciones de un string inicial con diferentes métodos:

- Conversión de todo el string a array de caracteres uno por uno
- La ordenación de los caracteres de manera aleatoria
- La inversión del orden de caracteres
- Quitar las vocales
- Quitar las consonantes.
- Array de palabras
- Inversión del orden de las palabras del string

1.8 Crea una clase para encontrar todos los grupos de tres elementos dado un set de n números reales que sumen zero. Ejemplo input:

Input array : [-25, -10, -7, -3, 2, 4, 8, 10]

Output : [[-10, 2, 8], [-7, -3, 10]]

4.9 Crea una clase llamada círculo que tenga como entrada el radio. Debería tener dos métodos, uno para calcular el perímetro y otro para calcular el área. También debería poder calcular el volumen del cilindro que quedaría dada una altura.

BIBLIOGRAFÍA

1. <https://www.w3resource.com/javascript-exercises/javascript-object-exercises.php>, 2022/06/18
2. <https://medium.com/@andrey.igorevich.borisov/10-javascript-exercises-with-objects-8942cc502754>, 2022/06/19
3. <https://github.com/Klod19/JS-Classes-Exercises>, 2022/06/19

4. <https://pynative.com/python-object-oriented-programming-oop-exercise/#h-oop-exercise-4-class-inheritance>, 2022/06/19
5. <https://www.w3resource.com/python-exercises/class-exercises/>, 2022/06/19