

EJERCICIO CLASES Y OBJETOS

Vamos a crear una clase llamada 'motocicleta' y haremos un par de instancias de ella.

- ✚ Crea una clase vacía llamada "motocicleta".
- ✚ Todas las motocicletas del concesionario son nuevas. Por lo tanto, vamos a añadir un atributo de clase para especificar ese valor siempre en todas las motos.
- ✚ Ahora, crea el método `__init__` vacío, con el que le daremos unos valores a las nuevas instancias.
- ✚ Añade los siguientes parámetros al `__init__`:
 - color.
 - matricula.
 - combustible_litros.
 - numero_ruedas.
 - marca.
 - modelo.
 - fecha_fabricacion.
 - velocidad_punta.
 - peso.
- ✚ Añade otro atributo de clase. Este va a ser "motor". Lo utilizaremos con un valor booleano para especificar si el motor está en marcha o detenido. True, en marcha. False, detenido. Por defecto, quiero que todos los motores vengan detenidos.
- ✚ Crea dos métodos inteligentes, arrancar y detener que representarán estas dos acciones con las motocicletas. Estos deben ser capaces de informar en la consola de las siguientes cosas.
 - Método `arrancar()`:
 - Si el motor está detenido, se indica que el motor ha arrancado.
 - Si el motor está ya en marcha y se intenta arrancar de nuevo, se indica que el motor ya estaba en marcha.
 - Método `detener()`:
 - Si el motor está en marcha, se indica que el motor se ha detenido.
 - Si el motor está ya detenido, y se intenta detener de nuevo, se indica que el motor ya estaba detenido.
- ✚ Instancia una motocicleta. La mayoría de argumentos son libres, pero quiero que algunos, tengan los siguientes valores:
 - combustible litros = 10
 - numero_ruedas = 2
- ✚ Prueba los dos métodos de arranque y detención con una o dos motocicletas. Haz las pruebas que quieras.
- ✚ El concesionario ya tiene precio para una de las motocicletas. Añade, desde fuera de la clase, este nuevo atributo con un valor para uno de los dos objetos.

✚ Imprime el valor que acabas de añadir desde fuera de la clase con una frase como esta:

"El precio de la motocicleta 'x (marca y modelo)' es de 'x_precio' \$."

✚ Ahora, quiero que añadas una forma de consultar el precio desde la clase con un método (lo mismo, que en el ejercicio 11, pero con un método).

✚ Llama al nuevo método con las dos motocicletas. ¿Qué ocurre con una de ellas?

```
class Motorcycle:
    is_new = True # Atributo de clase para todas las motos son nuevas
```

```
    def __init__(self, colour, number_plate, combustible_liters,
number_wheel=2, brand=None, model=None, fabrication_date=None, speed=None,
weight=None):
```

```
        self.colour = colour
        self.number_plate = number_plate
        self.combustible_liters = combustible_liters
        self.number_plate = number_wheel
        self.brand = brand
        self.model = model
        self.fabrication_date = fabrication_date
        self.speed = speed
        self.weight = weight
        self.motor = False # Por defecto, el motor está detenido
```

```
    def start(self):
        if self.motor:
            print("El motor ya estaba en marcha.")
        else:
            self.motor = True
            print("El motor ha arrancado.")
```

```
    def stop(self):
        if self.motor:
            self.motor = False
            print("El motor se ha detenido.")
        else:
            print("El motor ya estaba detenido.")
```

```
    def consult_price(self, price):
        self.price = price
```

```
# Instanciar una motocicleta
moto1 = Motorcycle("Rojo", "123ABC", 10, brand="Honda", model="CBR1000RR")
```

```
# Prueba de arranque y detención
```

```
moto1.start()  
moto1.start()  
moto1.stop()  
moto1.stop()
```

```
# Añadir precio desde fuera de la clase
```

```
moto2 = Motorcycle("Azul", "456XYZ", 15, brand="Harley Davidson",  
model="CBR1777RR")  
moto2.consult_price(5000)
```

```
# Consultar precio desde la clase
```

```
moto1.consult_price(4000)  
print(f"El precio de la motocicleta {moto1.brand} {moto1.model} es de  
{moto1.price} $.")  
print(f"El precio de la motocicleta {moto2.brand} {moto2.model} es de  
{moto2.price} $.")
```