

Para este proyecto, quiero que crees una clase llamada «motocicleta» y que hagas, al menos, un par de instancias de ella.

Quiero que te imagines un concesionario de motos. Esta clase será para representar sus motocicletas.

1. Crea una clase vacía llamada «motocicleta».
2. 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.
3. Ahora, crea el método `__init__` vacío, con el que le daremos unos valores a las nuevas instancias.
4. Añade los siguientes parámetros al `__init__`:
 - `color`.
 - `matricula`.
 - `combustible_litros`.
 - `numero_ruedas`.
 - `marca`.
 - `modelo`.
 - `fecha_fabricacion`.
 - `velocidad_punta`.
 - `peso`.

Opcionalmente, puedes añadir más atributos para describir cualquier cosa sobre las motocicletas, por ejemplo, cilindrada, tipo de transmisión y tantas otras cosas que podrían describir una motocicleta.

5. 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.
6. 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.

7. Instancia una motocicleta. La mayoría de argumentos son libres, pero quiero que algunos, tengan los siguientes valores:
 - combustible litros = 10
 - numero_ruedas = 2
8. Crea otra instancia de Motocicleta. Esta vez, quiero que utilices los argumentos de clave en lugar de los posicionales. Vimos este tema con las funciones. Funciona igual en las instanciaciones que en las llamadas de función. Además, quiero que el orden no sea el mismo que necesitas para los argumentos posicionales. Esta nueva motocicleta tiene el depósito vacío.
9. Prueba los dos métodos de arranque y detención con una o con las dos motocicletas. Haz las pruebas que quieras. El requisito es solo saber llamar a un método.
10. 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. Soy consciente de que sería mejor añadir el atributo precio a la clase y que lo tuvieran ya todos los objetos, pero es para que practiques diferentes cosas. No borres en ningún momento de los otros ejercicios la asignación de este precio.
11. 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' \$.»

12. 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).
13. Llama al nuevo método con las dos motocicletas. ¿Qué ocurre con una de ellas?
14. Para finalizar, crea un nuevo método en la clase, que sea capaz de repostar las motocicletas. Para esto, necesitarás lo siguiente:
 - Crea un método para comprobar la cantidad de combustible que tienen las motocicletas. Este servirá para informarnos del estado antes de iniciar el repostaje.
 - En este método, se deberá indicar la cantidad de litros que tiene de combustible, la capacidad máxima del tanque de combustible y los litros que faltan para llenar el depósito.
 - La salida de este método debe ser una especie de reporte. Pon todo lo anterior y añade un título personalizado con el nombre de la motocicleta que se consulta. Por ejemplo, → Reporte del depósito de «marca x y modelo x de motocicleta» <—.
 - Establecer un tope de depósito que indicaremos especialmente para cada motocicleta con un nuevo atributo. Por ejemplo, la primera tiene una capacidad máxima de 17 litros de combustible. La otra de 20.
 - Crear un método que se utilice para poner la cantidad de litros que se quieren repostar. Esto se indicará en la consola por un input().
 - Si la cantidad de litros es superior a la de la capacidad que hay en el depósito, se deberá advertir de que no se puede repostar esa cantidad y se le dejará intentarlo de nuevo las veces que haga falta.

- En cambio, si el repostaje es correcto, se indicará en consola la cantidad de litros que tiene la motocicleta.
- No solo hay que indicar la cantidad de combustible que tendrá la motocicleta, tiene que ser efectivo el cambio.

En el vídeo correspondiente a este día 11 del curso, podrás ver un ejemplo orientativo de la salida en la consola, seguro que así consigues hacerlo más fácil.

15. Crea un diagrama que describa la clase Motocicleta y los dos objetos instanciados en el ejercicio 14. Para ver como es este diagrama, te recomiendo ver el ejemplo del vídeo.

Espero que con estos ejercicios resueltos, te hayan venido bien para dominar un poco más la programación orientada a objetos.