

## Python Classes

```
In [1]: 1 class Pelicula:
2
3     # Constructor de clase
4     def __init__(self, titulo, duracion, lanzamiento):
5         self.titulo = titulo
6         self.duracion = duracion
7         self.lanzamiento = lanzamiento
8         print('Se ha creado la película:', self.titulo)
9
10    def __str__(self):
11        return '{} ({} {}) minutos'.format(self.titulo, self.lanzamiento, self.duracion)
12
13
14    class Catalogo:
15
16        peliculas = [] # Esta lista contendrá objetos de la clase Pelicula
17
18        def __init__(self, peliculas=[]):
19            self.peliculas = peliculas
20
21        def agregar(self, p): # p será un objeto Pelicula
22            self.peliculas.append(p)
23
24        def mostrar(self):
25            print("\nCatalogo: -----")
26            for p in self.peliculas:
27                print(p) # Print toma por defecto str(p)
28
29
30    p = Pelicula("El Padrino", 175, 1972)
31    c = Catalogo([p]) # Añado una lista con una película desde el principio
32    c.mostrar()
33    c.agregar(Pelicula("El Padrino: Parte 2", 202, 1974)) # Añadimos otra
34    c.mostrar()
```

Se ha creado la película: El Padrino

Catalogo: -----

El Padrino (1972) 175 minutos

Se ha creado la película: El Padrino: Parte 2

Catalogo: -----

El Padrino (1972) 175 minutos

El Padrino: Parte 2 (1974) 202 minutos

```
In [21]: 1 class Car:
2
3     def __init__(self, marca, modelo, color, matricula):
4         self.marca = marca
5         self.modelo = modelo
6         self.color = color
7         self.matricula = matricula
8
9     def __str__(self):
10        return f'{self.marca}, {self.modelo}, {self.color}, {self.matricula}'
11
12    class Cars:
13
14        def __init__(self, coches=[]):
15            self.coches = coches
16
17        def llista_tots(self):
18            print("\nCoches: -----")
19            for c in self.coches:
20                print(c) # Print toma por defecto str(c)
21
22        def cerca_matricula(self, matricula_cercar):
23            for coche in self.coches:
24                if matricula_cercar == coche.matricula:
25                    print("trobat: ", coche)
26                    return
27
28            print('No trobat')
29
30    c1 = Car ("TOYOTA", "YARIS", "ROJO", "6789-CYR")
31    c2 = Car ("TOYOTA", "YARIS", "VERDE", "333-CYR")
32
33    #print (c1)
34    #print (c1.marca)
35
36    c = Cars([c1,c2])
37    c.llibra_tots()
38    c.cerca_matricula('333-CYR')
```

Coches: -----

TOYOTA, YARIS, ROJO, 6789-CYR

TOYOTA, YARIS, VERDE, 333-CYR

trobat: TOYOTA, YARIS, VERDE, 333-CYR

```
In [22]: 1 #----- Definición de clases
2 class Perro:
3     def sonido(self):
4         print('Guauuuuu!!!')
5
6 class Gato:
7     def sonido(self):
8         print('Miaaaaauuu!!!')
9
10 class Vaca:
11     def sonido(self):
12         print('Múuuuuuuu!!!')
13
14 #----- Definición de funciones
15 def a_cantar(animales):
16     for animal in animales:
17         animal.sonido()
18
19 #----- Programa principal : main
20 perro1 = Perro()
21 perro2 = Perro()
22 gato1 = Gato()
23 gato2 = Gato()
24 vaca = Vaca()
25
26 granja = [perro1, gato1, vaca, gato2, perro2]
27 a_cantar(granja)
28
Guauuuuu!!!
Miaaaaauuu!!!
Múuuuuuuu!!!
Miaaaaauuu!!!
Guauuuuu!!!
```

```
In [24]: 1 class Producto:
2     def __init__(self, referencia, nombre, pvp, descripcion):
3         self.referencia = referencia
4         self.nombre = nombre
5         self.pvp = pvp
6         self.descripcion = descripcion
7
8     def __str__(self):
9         return f"REFERENCIA\t {self.referencia}\n" \
10             f"NOMBRE\t\t {self.nombre}\n" \
11             f"PVP\t\t\t {self.pvp}\n" \
12             f"DESCRIPCIÓN\t {self.descripcion}\n"
13
14
15 class Adorno(Producto):
16     pass
17
18 adorno = Adorno(2034, "Vaso adornado", 15, "Vaso de porcelana")
19 print(adorno)
20
21
REFERENCIA      2034
NOMBRE          Vaso adornado
PVP             15
DESCRIPCIÓN     Vaso de porcelana
```

```

In [13]: 1 class Person :
2         def __init__(self, name, surname, school):
3             self.name = name
4             self.surname = surname
5             self.school = school
6
7         def __str__(self):
8             return(f'{self.name}, {self.surname}, {self.school}')
9
10        def print_name(self):
11            print(self.name, self.surname)
12
13        def print_school(self):
14            print(self.school)
15
16        personas = []
17        personas.append(Person('Laura', 'Martinez', 'UB'))
18        personas.append(Person('Alex', 'Navarro', 'UAB'))
19        personas.append(Person('Vsevolod', 'Vasylets', 'UPC'))
20
21        for i in personas:
22            i.print_name()
23
24        print()
25
26        for y in personas:
27            print(y)
28

```

Laura Martinez  
 Alex Navarro  
 Vsevolod Vasylets

Laura, Martinez, UB  
 Alex, Navarro, UAB  
 Vsevolod, Vasylets, UPC

```

In [18]: 1 class Complejo:
2
3         def __init__(self, real, imaginario):
4             self.r = real
5             self.i = imaginario
6
7         def imprimeReal(self):
8             print("Parte real: ", self.r)
9
10        def imprimeImaginario(self):
11            print("Parte imaginaria: ", self.i)
12
13        def __str__(self):
14            print(f'{self.r}+{self.i}')
15
16        equ = Complejo(3,5)
17        equ.imprimeReal()
18        equ.imprimeImaginario()
19
20
21        print(equ.i)
22

```

Parte real: 3  
 Parte imaginaria: 5  
 5