4.POO

August 10, 2021

1 POO

1.1 Creación de la clase

[6]:
```
class Empleado:
    # atributos
    ID = None
    salario = None
    departamento = None
```

Instanciando un objeto (Ricardo)

[2]:
```
Ricardo = Empleado()
```

[4]:
```
Ricardo.ID = 567812
```

[5]:
```
Ricardo.ID
```

[5]:
```
567812
```

1.1.1 Método constructor

[ ]:
```
class Empleado:
    # atributos
    def __init__(self, ID, salario, departamento):
        self.ID = ID
        self.salario = salario
        self.departamento = departamento
```

[7]:
```
Ricardo = Empleado(52689, 300, "Finanzas")
```

[9]:
```
print(Ricardo.ID)
print(Ricardo.salario)
print(Ricardo.departamento)
```

```
52689
300
Finanzas
```

Atributos opcionales

[10]:
```
class Empleado:
    # atributos
    def __init__(self, ID = 11111, salario = 200, departamento = "Marketing"):
        self.ID = ID
        self.salario = salario
        self.departamento = departamento
```

[11]:
```
Ricardo = Empleado()
```

[12]:
```
Ricardo.ID
```

[12]:
```
11111
```

1.1.2 Agregar métodos a la clase

[13]:
```
class Empleado:
    # atributos
    def __init__(self, ID = 11111, salario = 200, departamento = "Marketing"):
        self.ID = ID
        self.salario = salario
        self.departamento = departamento

    # métodos
    def salarioPorDia(self):
        return (self.salario/30)

    def impuestos(self):
        return (self.salario*0.2)
```

[14]:
```
Ricardo = Empleado(564464, 300, "Finanzas")
```

[15]:
```
print(Ricardo.ID)
print(Ricardo.salario)
print(Ricardo.departamento)
print(Ricardo.impuestos())
print(Ricardo.salarioPorDia())
```

```
564464
300
Finanzas
60.0
10.0
```

1.2 Herencia

[16]:
```
class Vehiculo:
    def __init__(self, marca, color, modelo):
        self.marca = marca
        self.color = color
        self.modelo = modelo
    # metodos
    def imprimirDetalles(self):
        print("Productor: ", self.marca)
        print("Color: ", self.color)
        print("Modelo:", self.modelo)
```

[17]:
```
obj1 = Vehiculo("Suzuki", "Gris", "2016")
```

[18]:
```
obj1.imprimirDetalles()
```

```
Productor:  Suzuki
Color:  Gris
Modelo: 2016
```

[21]:
```
class Carro(Vehiculo):
    def __init__(self, marca, color, modelo, puertas):
        self.marca = marca
        self.color = color
        self.modelo = modelo
        self.puertas = puertas

    def imprimirDetallesCarro(self):
        print("Productor: ", self.marca)
        print("Color: ", self.color)
        print("Modelo:", self.modelo)
        print("Puertas: ", self.puertas)
```

[22]:
```
obj1 = Carro("Suzuki", "Gris", "2016", 4)
```

[23]:
```
obj1.imprimirDetallesCarro()
```

```
Productor:  Suzuki
Color:  Gris
Modelo: 2016
Puertas:  4
```

Hacer uso de el constructor y los metodos de la clase padre

[27]:
```
class Carro(Vehiculo):
    def __init__(self, marca, color, modelo, puertas):
        Vehiculo.__init__(self, marca, color, modelo)
        self.puertas = puertas
```

```
    def imprimirDetallesCarro(self):
        self.imprimirDetalles()
        print("Puertas: ", self.puertas)
```

[28]:
```
obj1 = Carro("Suzuki", "Gris", "2016", 4)
```

[29]:
```
obj1.imprimirDetallesCarro()
```

```
Productor:  Suzuki
Color:  Gris
Modelo: 2016
Puertas:  4
```