

Creado por:

Isabel Maniega

Encapsulamiento

¿Qué es?

Sirve para proteger a las variables de modificaciones no deseadas.

VARIABLES PRIVADAS / MÉTODOS PRIVADOS

= SOLO ACCESIBLES DESDE LA CLASE DONDE SE ENCUENTREN

en java y c++ se consigue la encapsulación de otra manera en python se hace con:

- (`__nombre`)
- Doble barra baja antes de la variable (atributo)
- Doble barra baja antes de la función (método)

Ejemplo 1

```
In [42]: class Clase1:
          def __metodo1(self):
              print("Estamos en método1")
          class Clase2:
              def metodo2(self):
                  print("estamos en metodo2")
```

```
In [43]: # AttributeError: type object 'Clase1' has no attribute '__metodo1'
          # Clase1.__metodo1()
```

```
In [44]: # AttributeError: type object 'Clase1' has no attribute 'metodo1'
          # Clase1.metodo1()
```

```
In [49]: obj = Clase1()

          print(obj._Clase1__metodo1())
```

Estamos en método1
None

```
In [7]: Clase2.metodo2()
```

estamos en metodo2

```
In [8]: class Clase1:
          def __metodo1():
```

```

        print("Estamos en método1")
        # __metodo1() # Si lo declaramos dentro de la clase podemos hacer al

class Clase2:
    def metodo2():
        print("estamos en metodo2")

```

Ejemplo 2

```

In [ ]: class Clase1:
        def __metodo1():
            print("Estamos en método1")
            print("llamada a: metodo2")
            Clase2.metodo2()

        class Clase2:
            def metodo2():
                print("estamos en metodo2")
                print("llamada a: -metodo1-")
                Clase1.__metodo1()

```

```

In [ ]: # Clase1.__metodo1()

        # AttributeError: type object 'Clase1' has no attribute '__metodo1'

```

```

In [ ]: # Clase1.metodo1()

        # AttributeError: type object 'Clase1' has no attribute '__metodo1'

```

```

In [ ]: # Clase2.metodo2()

        # AttributeError: type object 'Clase1' has no attribute '_Clase2__metodo1'

```

Ejemplo 3

1ª Parte

```

In [10]: class Fecha_albaran:
        def __init__(self, dia, mes, año):
            self.dia = dia
            self.mes = mes
            self.año = año
        albaran1 = Fecha_albaran(15, 1, 2015)

```

```

In [11]: albaran1.dia

```

```

Out[11]: 15

```

```

In [12]: albaran1.mes

```

```

Out[12]: 1

```

```
In [13]: albaran1.año
```

```
Out[13]: 2015
```

2ª Parte

```
In [14]: class Fecha_albaran:
    def __init__(self, dia, mes, año):
        self.__dia = dia # atributo privado
        self.mes = mes
        self.año = año
albaran1 = Fecha_albaran(15, 1, 2015)
```

```
In [15]: albaran1.mes
```

```
Out[15]: 1
```

```
In [16]: albaran1.año
```

```
Out[16]: 2015
```

```
In [19]: # albaran1.dia

# AttributeError: 'Fecha_albaran' object has no attribute 'dia'
```

```
In [20]: # albaran1.__dia

# AttributeError: 'Fecha_albaran' object has no attribute 'dia'
```

3ª Parte

```
In [21]: class Fecha_albaran:
    def __init__(self):
        self.__dia = 15 # atributo privado
        self.__mes = 4
        self.__año = 2015

    def setMes(self, mes):
        if mes > 0 and mes < 13:
            self.__mes = mes
        else:
            print("mes no valido - No está entre Enero y Diciembre")

    def getMes(self):
        return self.__mes

albaran1 = Fecha_albaran()
```

```
In [22]: albaran1.setMes(45)
```

```
mes no valido - No está entre Enero y Diciembre
```

```
In [23]: albaran1.getMes()
```

Out[23]: 4

In [24]: albaran1.setMes(6)

In [25]: albaran1.getMes()

Out[25]: 6

Ejemplo 4

```
In [31]: class Clase:
# Variables de clase (atributos en este caso)
x = 10 # Atributo ACCESIBLE
__y = 20 # SOLO ES ACCESIBLE DENTRO DE LA CLASE

# MÉTODO ACCESIBLE
def metodoPublico(self):
# SE PUEDE ACCEDER A SU INFORMACIÓN
print("el método es público")

# MÉTODO SÓLO ACCESIBLE EN LA CLASE
def __metodoPrivado(self):
print("Este método es privado")
```

```
In [32]: # objeto
clase1 = Clase()
```

```
In [33]: # Acceso a elementos PÚBLICOS
```

```
In [34]: clase1.x
```

Out[34]: 10

```
In [35]: clase1.metodoPublico()
```

el método es público

```
In [ ]: # Acceso a los elementos PRIVADOS
```

```
In [ ]: # Clase.y
```

```
# AttributeError: type object 'Clase' has no attribute 'y'
```

```
In [ ]: # Clase.__metodoPrivado()
```

```
# AttributeError: type object 'Clase' has no attribute '__metodoPrivado'
```

```
In [36]: class Clase:
# Variables de clase (atributos en este caso)
x = 10 # Atributo ACCESIBLE
__y = 20 # SOLO ES ACCESIBLE DENTRO DE LA CLASE

# MÉTODO ACCESIBLE
def metodoPublico(y):
# SE PUEDE ACCEDER A SU INFORMACIÓN
```

```

print("el método es público")
print("el valor de y es ", y)

# MÉTODO SÓLO ACCESIBLE EN LA CLASE
def __metodoPrivado(x):
    print("Este método es privado")
    print("el valor de x es : ", x)

__metodoPrivado(x)
# metodoPublico(__y)

```

Este método es privado
el valor de x es : 10

```

In [37]: class Clase:
# Variables de clase (atributos en este caso)
x = 10 # Atributo ACCESIBLE
__y = 20 # SOLO ES ACCESIBLE DENTRO DE LA CLASE

# MÉTODO ACCESIBLE
def metodoPublico(y):
    # SE PUEDE ACCEDER A SU INFORMACIÓN
    print("el método es público")
    print("el valor de y es ", y)

# MÉTODO SÓLO ACCESIBLE EN LA CLASE
def __metodoPrivado(x):
    print("Este método es privado")
    print("el valor de x es : ", x)

# __metodoPrivado(x)
metodoPublico(__y)

```

el método es público
el valor de y es 20

DIR

```
In [38]: import numpy as np
```

```
In [39]: print(np.array(dir(clase1)))
```

```

['_Class__metodoPrivado' '_Class__y' '_class__' '_delattr__' '_dict__'
 '_dir__' '_doc__' '_eq__' '_format__' '_ge__' '_getattr__'
 '_gt__' '_hash__' '_init__' '_init_subclass__' '_le__' '_lt__'
 '_module__' '_ne__' '_new__' '_reduce__' '_reduce_ex__' '_repr__'
 '_setattr__' '_sizeof__' '_str__' '_subclasshook__' '_weakref__'
 'metodoPublico' 'x']

```

```

In [ ]: # '_Class__metodoPrivado'

# Veíamos que está ahí, pero no es accesible

# pero...

```

```
# SE PUEDE IMPRIMIR ASI:
```

```
In [40]: clase1.x
```

```
Out[40]: 10
```

```
In [41]: clase1._Clase__y
```

```
Out[41]: 20
```

```
In [46]: clase1.metodoPublico()
```

el método es público

```
In [47]: clase1._Clase__metodoPrivado()
```

Este método es privado

```
In [48]: # visualizar el contenido del objeto  
vars(albaran1)
```

```
Out[48]: {'_Fecha_albaran__dia': 15,  
          '_Fecha_albaran__mes': 6,  
          '_Fecha_albaran__año': 2015}
```

Creado por:

Isabel Maniega