

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 [14]: class Clase1:
def __metodo1():
    print("Estamos en método1")
class Clase2:
def metodo2():
    print("estamos en metodo2")

Estamos en método1
```

```
In [11]: # AttributeError: type object 'Clase1' has no attribute '__metodo1'
# Clase1.__metodo1()
```

```
In [12]: # AttributeError: type object 'Clase1' has no attribute 'metodo1'
# Clase1.metodo1()
```

```
In [15]: Clase2.metodo2()

estamos en metodo2
```

```
In [14]: class Clase1:
def __metodo1():
    print("Estamos en método1")
    # __metodo1() # Si lo declaramos dentro de la clase podemos hacer al metodo, fuera NO.
class Clase2:
def metodo2():
    print("estamos en metodo2")
    print("llamada a: -metodo1-")
    Clase1.__metodo1()

Estamos en método1
```

Ejemplo 2

```
In [24]: 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 [19]: # Clase1.__metodo1()

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

```
In [20]: # Clase1.metodo1()

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

```
In [22]: # Clase2.metodo2()

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

Ejemplo 3

1ª Parte

```
In [25]: 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 [26]: albaran1.dia
```

```
Out[26]: 15
```

```
In [27]: albaran1.mes
```

```
Out[27]: 1
```

```
In [28]: albaran1.año
```

```
Out[28]: 2015
```

2ª Parte

```
In [31]: 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 [32]: albaran1.mes
```

```
Out[32]: 1
```

```
In [33]: albaran1.año
```

```
Out[33]: 2015
```

```
In [37]: # albaran1.dia

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

```
In [38]: # albaran1.__dia

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

3ª Parte

```
In [46]: 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 [47]: albaran1.setMes(45)

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

```
In [48]: albaran1.getMes()
```

```
Out[48]: 4
```

```
In [49]: albaran1.setMes(6)
```

```
In [50]: albaran1.getMes()
```

```
Out[50]: 6
```

Ejemplo 4

```
In [67]: 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():
    # SE PUEDE ACCEDER A SU INFORMACIÓN
    print("el método es público")

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

```
In [79]: # objeto
clase1 = Clase()
```

```
In [80]: # Acceso a elementos PÚBLICOS
```

```
In [81]: clase1.x
```

```
Out[81]: 10
```

```
In [82]: clase1.metodoPublico()

el método es público
el valor de y es  <__main__.Clase object at 0x7f859e3b1c10>
```

```
In [72]: # Acceso a los elementos PRIVADOS
```

```
In [73]: # Clase.y

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

```
In [74]: # Clase.__metodoPrivado()

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

```
In [75]: 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 [76]: 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 [78]: import numpy as np
```

```
In [83]: print(np.array(dir(clase1)))

['_Clase__metodoPrivado' '_Clase__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 [87]: # '_Clase__metodoPrivado'

# Veiamos que está ahí, pero no es accesible

# pero...

# SE PUEDE IMPRIMIR ASI:
```

```
In [88]: clase1.x
```

```
Out[88]: 10
```

```
In [89]: clase1._Clase__y
```

```
Out[89]: 20
```

```
In [90]: clase1.metodoPublico()

el método es público
el valor de y es  <__main__.Clase object at 0x7f859e3b1c10>
```

```
In [91]: clase1._Clase__metodoPrivado()

Este método es privado
el valor de x es :  <__main__.Clase object at 0x7f859e3b1c10>
```

```
In [94]: # visualizar el contenido del objeto
vars(albaran1)

{'_Fecha_albaran__dia': 15,
'_Fecha_albaran__mes': 1,
'_Fecha_albaran__año': 2015}
```

Creado por:

Isabel Maniega