

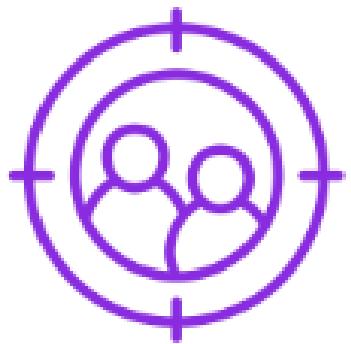
Curso: Programación Orientada a Objetos



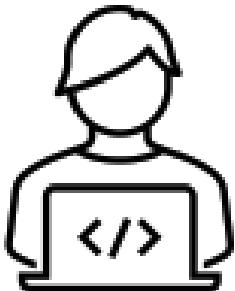
Tema:
Atributos Privados, métodos
get y set



Atributos privados y métodos get y set



¿Qué es un atributo privado?



Un atributo privado es aquel que es accesible solo desde la clase.

Ejemplo: los atributos privados en Python se escriben con doble subguion seguido del nombre que se le quiere atribuir al atributo de la siguiente manera:

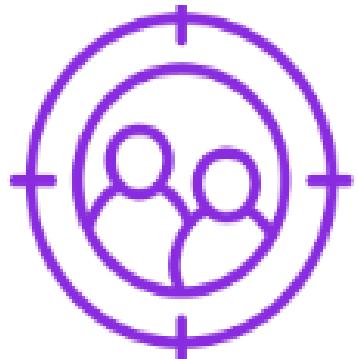
__nombre

__codigo

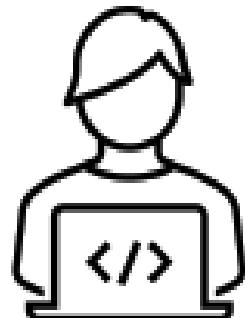
__dirección

Ejemplo 1

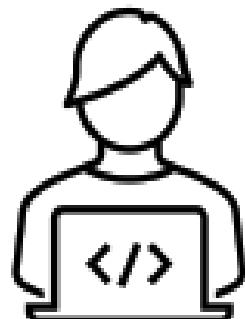
```
class Persona:  
    def __init__(self, nombre, apellido, edad):  
        self.__nombre = nombre  
        self.apellido = apellido  
        self.edad = edad  
  
    def mostrar_detalle(self):  
        print(f'Persona: {self.__nombre} {self.apellido} {self.edad}')  
  
personal = Persona('Juan', 'Perez', 28)  
personal.__nombre = 'Juan Carlos'  
personal.mostrar_detalle()
```



¿Qué son los métodos get y set?



Los métodos get son aquellos que permiten acceder al valor de un atributo privado.



Los métodos set son aquellos que permiten asignar un valor a un atributo privado.

Ejemplo decoradores get y set

```
1  class Persona:
2      def __init__(self, nombre="Pablo"):
3          self.__nombre = nombre
4
5      @property
6      def nombre(self):
7          return self.__nombre
8
9      @nombre.setter
10     def nombre(self, nombre):
11         self.__nombre = nombre
12
13 p = Persona()
14 print("Nombre: ", p.nombre)
15 p.nombre="Josefo"
16 print("Nombre: ", p.nombre)
```

```
1  class Persona:  
2      def __init__(self, nombre, apellido, edad):  
3          self.__nombre = nombre  
4          self.apellido = apellido  
5          self.edad = edad  
6  
7      @property  
8      def nombre(self):  
9          return self.__nombre  
10  
11     @nombre.setter  
12     def nombre(self, nombre):  
13         self.__nombre = nombre  
14  
15     def mostrar_detalle(self):  
16         print(f'Persona: {self.__nombre} {self.apellido} {self.edad}')  
17  
18     personal = Persona('Juan', 'Perez', 28)  
19     personal.nombre = 'Juan Carlos'  
20     print(personal.nombre)
```

Resultado:

```
Juan Carlos
```

```
class Persona:  
    def __init__(self, nombre, apellido, edad):  
        self._nombre = nombre  
        self._apellido = apellido  
        self._edad = edad  
  
    @property  
    def nombre(self):  
        return self._nombre  
  
    @nombre.setter  
    def nombre(self, nombre):  
        self._nombre = nombre  
  
    @property  
    def apellido(self):  
        return self._apellido  
  
    @apellido.setter  
    def apellido(self, apellido):  
        self._apellido = apellido  
  
    @property  
    def edad(self):  
        return self._edad  
  
    @edad.setter  
    def edad(self, edad):  
        self._edad = edad  
  
    def mostrar_detalle(self):  
        print(f'Persona: {self._nombre} {self._apellido} {self._edad}')  
  
persona1 = Persona('Juan', 'Perez', 28)  
persona1.nombre = 'Juan Carlos'  
persona1.apellido = 'Lara'  
persona1.edad = 30  
persona1.mostrar_detalle()
```

Resultado:

https://www.udemy.com/encapsulation/02_07

Persona: Juan Carlos Lara 30