**Getter and Setter in Python**

In Python, getters and setters are not the same as those in other object-oriented programming languages. Basically, the main purpose of using getters and setters in object-oriented programs is to ensure data encapsulation. Private variables in python are not actually hidden fields like in other object oriented languages. Getters and Setters in python are often used when:

* We use getters & setters to add validation logic around getting and setting a value.
* To avoid direct access of a class field i.e. private variables cannot be accessed directly or modified by external user.

To achieve getters & setters property, if we define normal get() and set() methods it will not reflect any special implementation. To achieve such functionality Python has a special function property().

In Python property()is a built-in function that creates and returns a property object. A property object has three methods, getter(), setter(), and delete(). property() function in Python has four arguments property(fget, fset, fdel, doc), fget is a function for retrieving an attribute value. fset is a function for setting an attribute value. fdel is a function for deleting an attribute value. doc creates a docstring for attribute. A property object has three methods, getter(), setter(), and delete() to specify fget, fset and fdel individually.

Python @property is one of the built-in decorators. The main purpose of any decorator is to change your class methods or attributes in such a way so that the user of your class no need to make any change in their code.

For example-

# Python program showing the use of

# @property

class Geeks:

     def \_\_init\_\_(self):

          self.\_age = 0

     # using property decorator

     # a getter function

     @property

     def age(self):

         print("getter method called")

         return self.\_age

     # a setter function

     @age.setter

     def age(self, a):

         if(a < 18):

            raise ValueError("Sorry you age is below eligibility criteria")

         print("setter method called")

         self.\_age = a

Output-

setter method called

getter method called

19