

# Getter and Setters in Python

## Getters

Getters in Python are methods that are used to access the values of an objects properties. They are used to return the value of a specific property and are typically defined using the @property decorator. Here is an example of a simple class with a getter method:

```
class MyClass:
    def __init__(self,value):
        self._value=value

@property
def _value(self):
    return 10*self._value
```

In this example the My class has a single property. \_value, which is initialized In the init method. The value methods is defined as a getter using @property decorator, and is used to return the value if the \_value property.

To use the getter, we can create an instance of the class. Class and then access the value property as if it were an attribute.

## Setters

It is important to note that the getters do not take any parameters and we cannot set the through getter method. For that we need setter method which can be added by decorating method with @property\_ name setter.

Example:

```
class MyClass:
    def __init__(self,value):
        self._value=value

    def show(self):
        print(f"value is {self._value}")

@property
```

```
def ten_value(self):  
    return 10*self._value  
  
@ten_value.setter  
def ten_value(self,new_value):  
    self._value=new_value/10  
obj=Myclass(10)  
obj._value=30  
print(obj._value)  
print(obj.ten_value)  
obj.show()
```

## Source Code

```
class Myclass:  
    def __init__(self,value):  
        self._value=value  
  
    def show(self):  
        print(f"value is {self._value}")  
  
    @property  
    def ten_value(self):  
        return 10*self._value  
  
    @ten_value.setter  
    def ten_value(self,new_value):  
        self._value=new_value/10  
obj=Myclass(10)  
obj._value=30  
print(obj._value)  
print(obj.ten_value)  
obj.show()
```

Thank You

