</> CodeWithHarry

HTML CSS JS C C++ JAVA PYTHON PHP REACT JS

Getters and Setters in Python | Python Tutorial - Day #60

Overview Q&A Downloads Announcements

# Day 60 - Getters

Getters in Python are methods that are used to access the values of an object's 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 self._value
```

In this example, the MyClass class has a single property, _value, which is initialized in the **init** method. The value method is defined as a getter using the @property decorator, and is used to return the value of the _value property.
To use the getter, we can create an instance of the MyClass class, and then access the value property as if it were an attribute:

```
>>> obj = MyClass(10)
>>> obj.value
10
```

# Setters

It is important to note that the getters do not take any parameters and we cannot set the value through getter method.For that we need setter method which can be added by decorating method with @property_name.setter
Here is an example of a class with both getter and setter:

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

    @property
    def value(self):
        return self._value

    @value.setter
    def value(self, new_value):
        self._value = new_value
```

We can use setter method like this:

```
>>> obj = MyClass(10)
>>> obj.value = 20
>>> obj.value
20
```

In conclusion, getters are a convenient way to access the values of an object's properties, while keeping the internal representation of the property hidden. This can be useful for encapsulation and data validation.

For Next Lesson Please Click On The "Next" button Below.

Previous Next