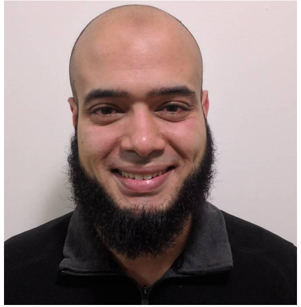
Python Programming

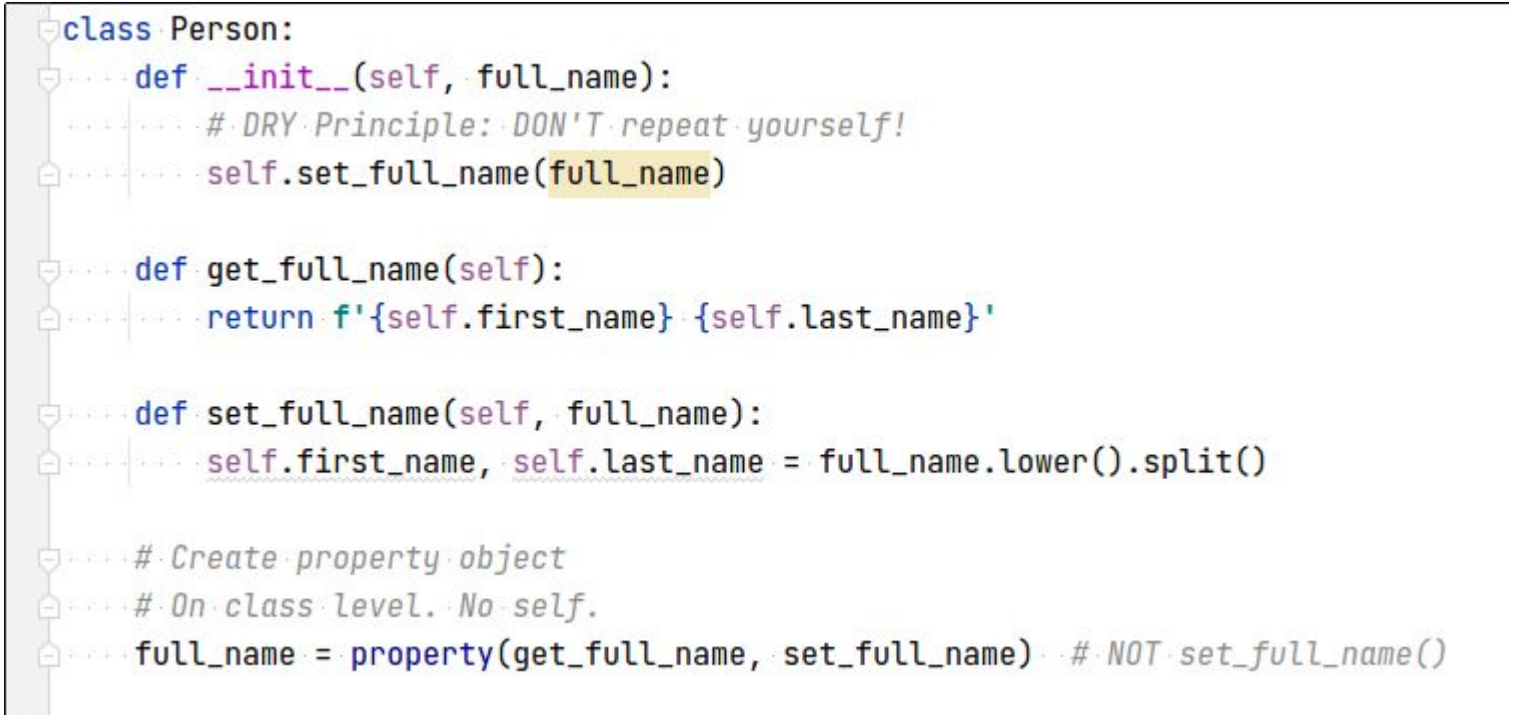
Property Class 2

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

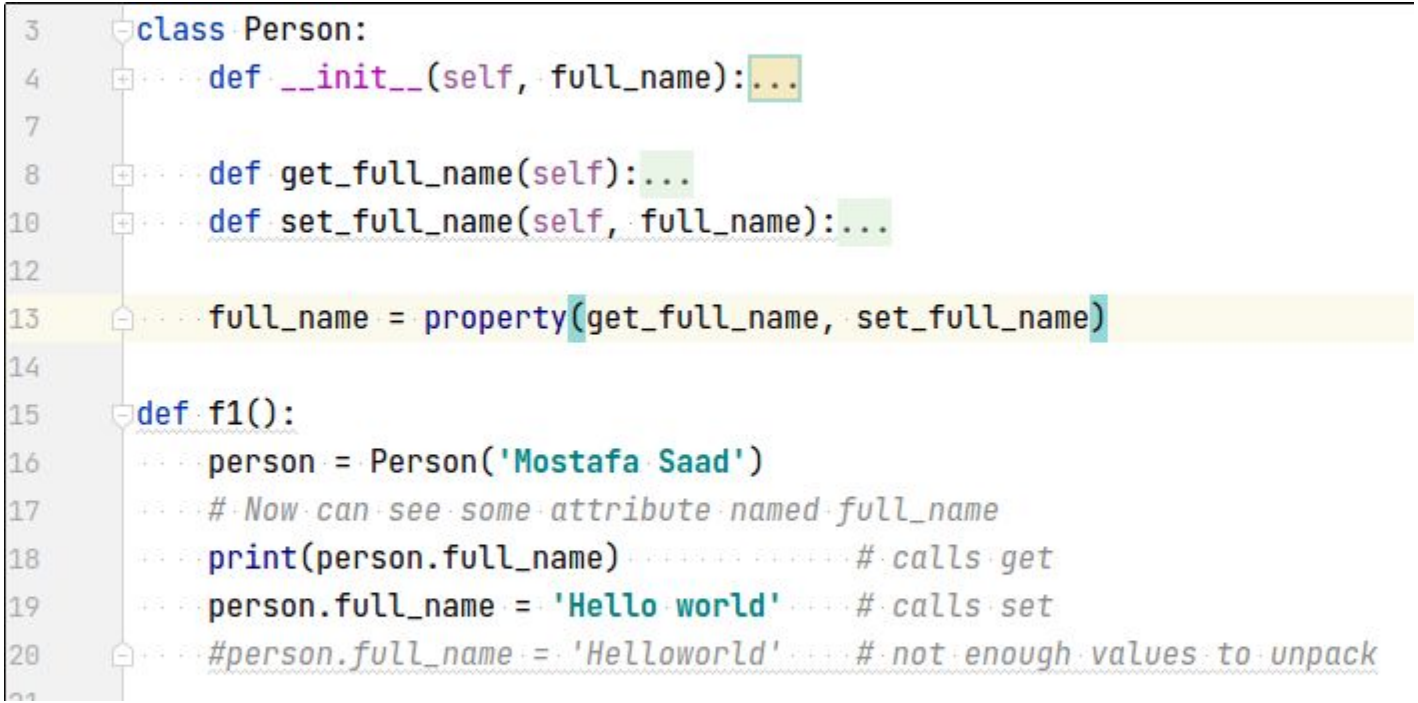


The property class



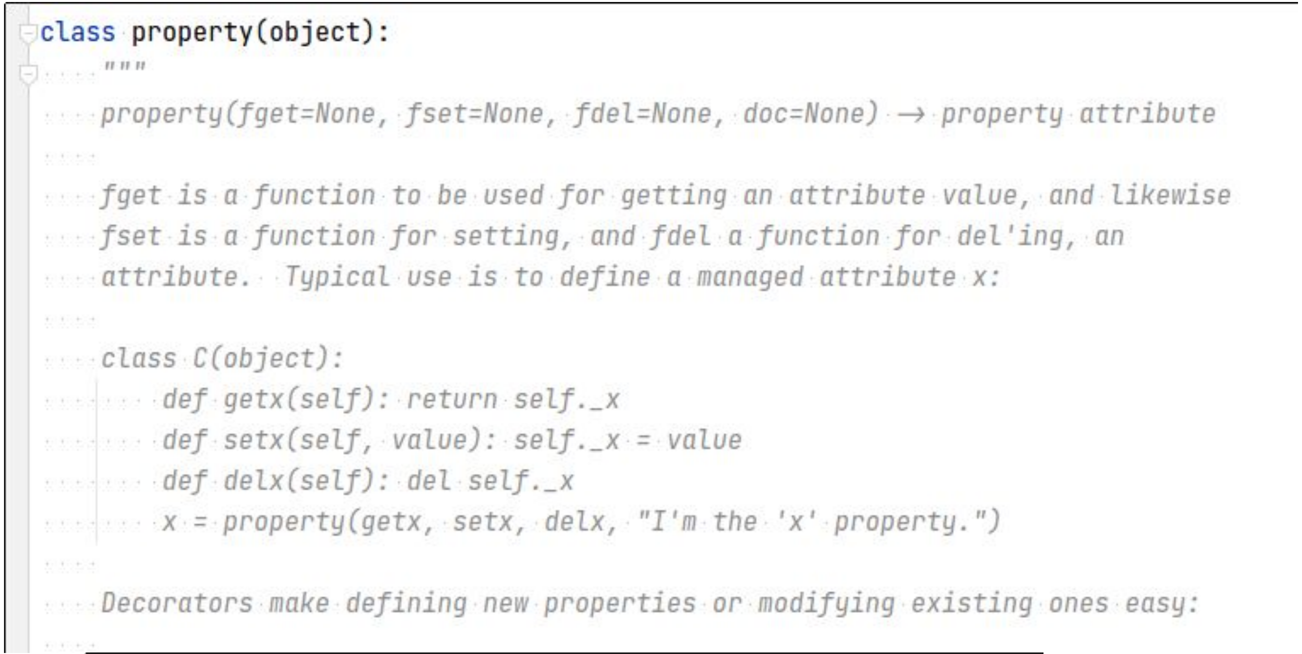
```python
class Person:
    def __init__(self, full_name):
        # DRY Principle: DON'T repeat yourself!
        self.set_full_name(full_name)

    def get_full_name(self):
        return f'{self.first_name} {self.last_name}'

    def set_full_name(self, full_name):
        self.first_name, self.last_name = full_name.lower().split()

    # Create property object
    # On class level. No self.
    full_name = property(get_full_name, set_full_name)  # NOT set_full_name()
```

The property class

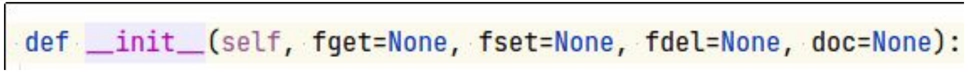


```python
class Person:
    def __init__(self, full_name):...

    def get_full_name(self):...
    def set_full_name(self, full_name):...

    full_name = property(get_full_name, set_full_name)

def f1():
    person = Person('Mostafa Saad')
    # Now can see some attribute named full_name
    print(person.full_name)             # calls get
    person.full_name = 'Hello world'    # calls set
    #person.full_name = 'Helloworld'     # not enough values to unpack
```

Inside the property class



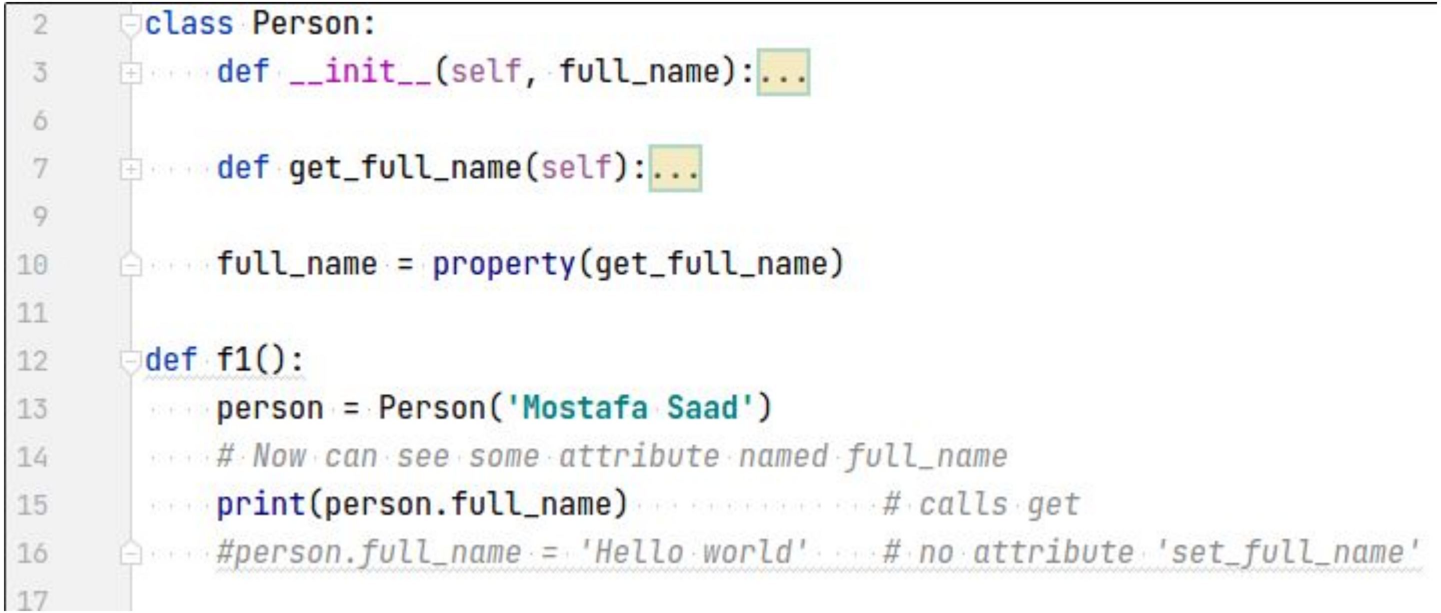
```
class property(object):
    """
    property(fget=None, fset=None, fdel=None, doc=None) → property attribute

    fget is a function to be used for getting an attribute value, and likewise
    fset is a function for setting, and fdel a function for del'ing, an
    attribute.  Typical use is to define a managed attribute x:

    class C(object):
        def getx(self): return self._x
        def setx(self, value): self._x = value
        def delx(self): del self._x
        x = property(getx, setx, delx, "I'm the 'x' property.")

    Decorators make defining new properties or modifying existing ones easy:

```



```
def __init__(self, fget=None, fset=None, fdel=None, doc=None):
```

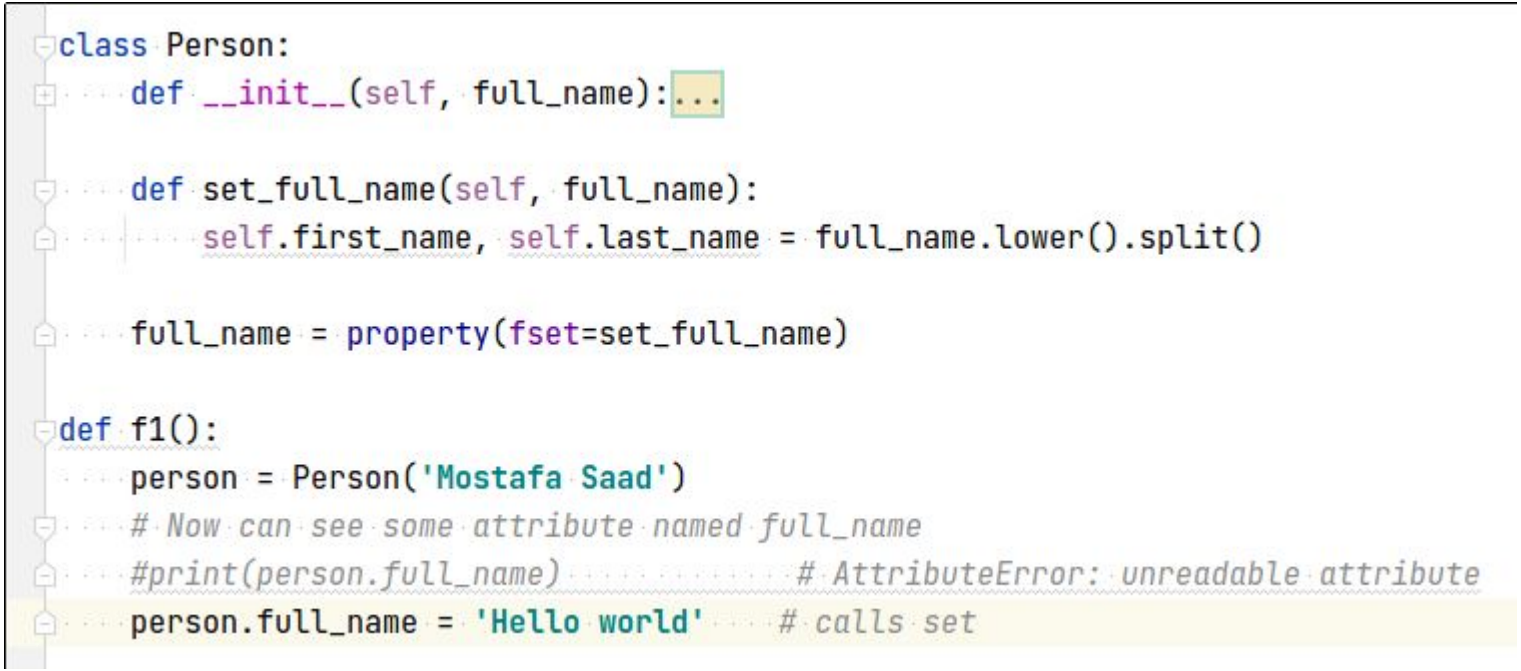
Observe You can write Docstring For a class

Provide access only!



```python
class Person:
    def __init__(self, full_name):...

    def get_full_name(self):...

    full_name = property(get_full_name)

def f1():
    person = Person('Mostafa Saad')
    # Now can see some attribute named full_name
    print(person.full_name)             # calls get
    #person.full_name = 'Hello world'   # no attribute 'set_full_name'
```

Provide set only!



```python
class Person:
    def __init__(self, full_name):...

    def set_full_name(self, full_name):
        self.first_name, self.last_name = full_name.lower().split()

    full_name = property(fset=set_full_name)

def f1():
    person = Person('Mostafa Saad')
    # Now can see some attribute named full_name
    #print(person.full_name)            # AttributeError: unreadable attribute
    person.full_name = 'Hello world'    # calls set
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."