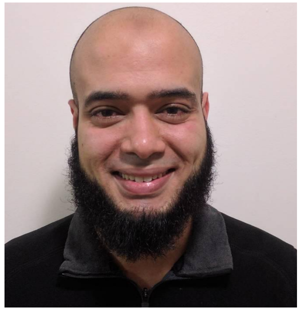
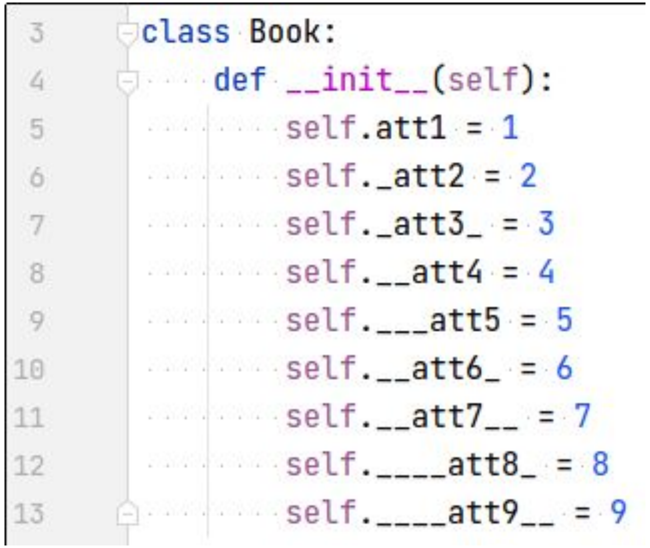
Python Programming
Name Mangling

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

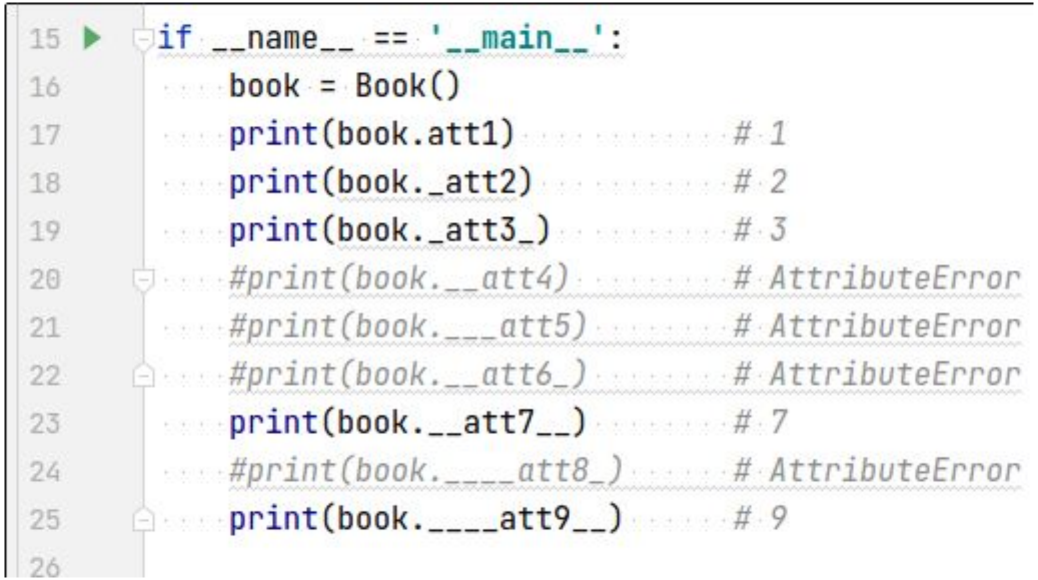
Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Underscores!

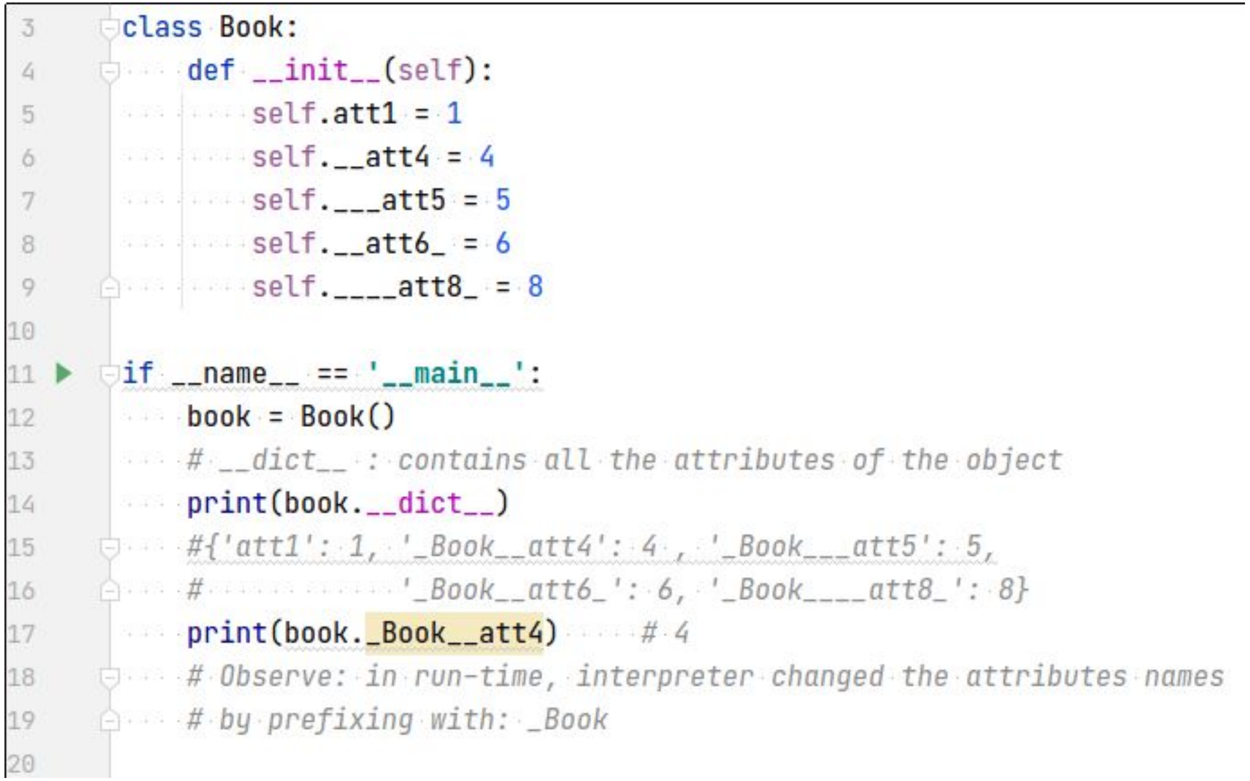


```python
class Book:
    def __init__(self):
        self.att1 = 1
        self._att2 = 2
        self._att3_ = 3
        self.__att4 = 4
        self.___att5 = 5
        self.__att6_ = 6
        self.__att7__ = 7
        self.____att8_ = 8
        self.____att9__ = 9
```



```python
if __name__ == '__main__':
    book = Book()
    print(book.att1)            # 1
    print(book._att2)           # 2
    print(book._att3_)          # 3
    #print(book.__att4)         # AttributeError
    #print(book.___att5)        # AttributeError
    #print(book.__att6_)        # AttributeError
    print(book.__att7__)        # 7
    #print(book.____att8_)      # AttributeError
    print(book.____att9__)      # 9
```

Not that strict!



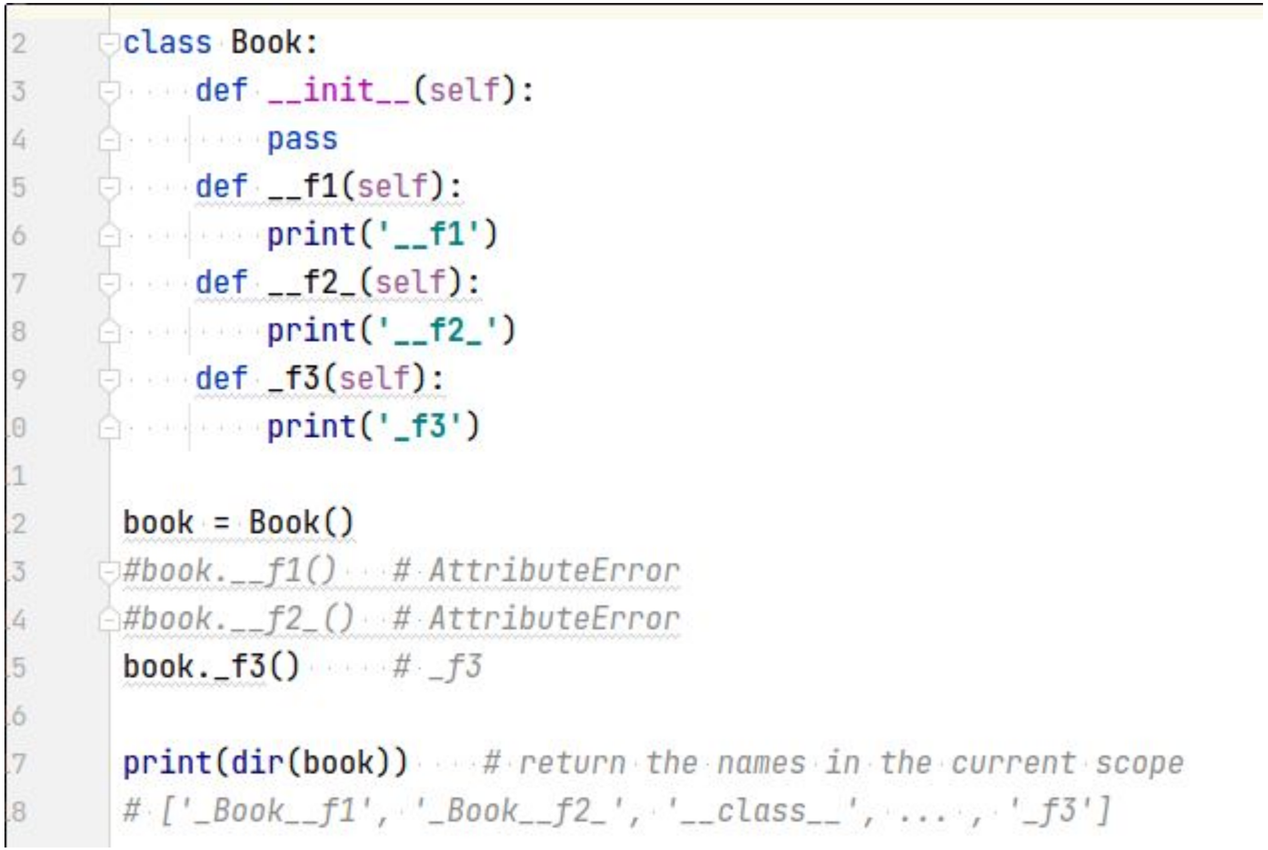
```
class Book:
    def __init__(self):
        self.att1 = 1
        self.__att4 = 4
        self.___att5 = 5
        self.__att6_ = 6
        self.____att8_ = 8

if __name__ == '__main__':
    book = Book()
    # __dict__ : contains all the attributes of the object
    print(book.__dict__)
    #{'att1': 1, '_Book__att4': 4 , '_Book___att5': 5,
    #              '_Book__att6_': 6, '_Book____att8_': 8}
    print(book._Book__att4)    # 4
    # Observe: in run-time, interpreter changed the attributes names
    # by prefixing with: _Book
```

Name Mangling

- Prefixing specific attributes with **_classname**
- If they have
  - at least 2 leading (before) underscores __
  - and at most 1 trailing (after) _
- Examples, for a book class:
  - __var ⇒ _book__var
  - __var_ ⇒ _book__var_
- So by default, the user can't access them
  - unless the coder wanna really use them ⇒ then use the mangled name
- Same rules for functions!

With functions!

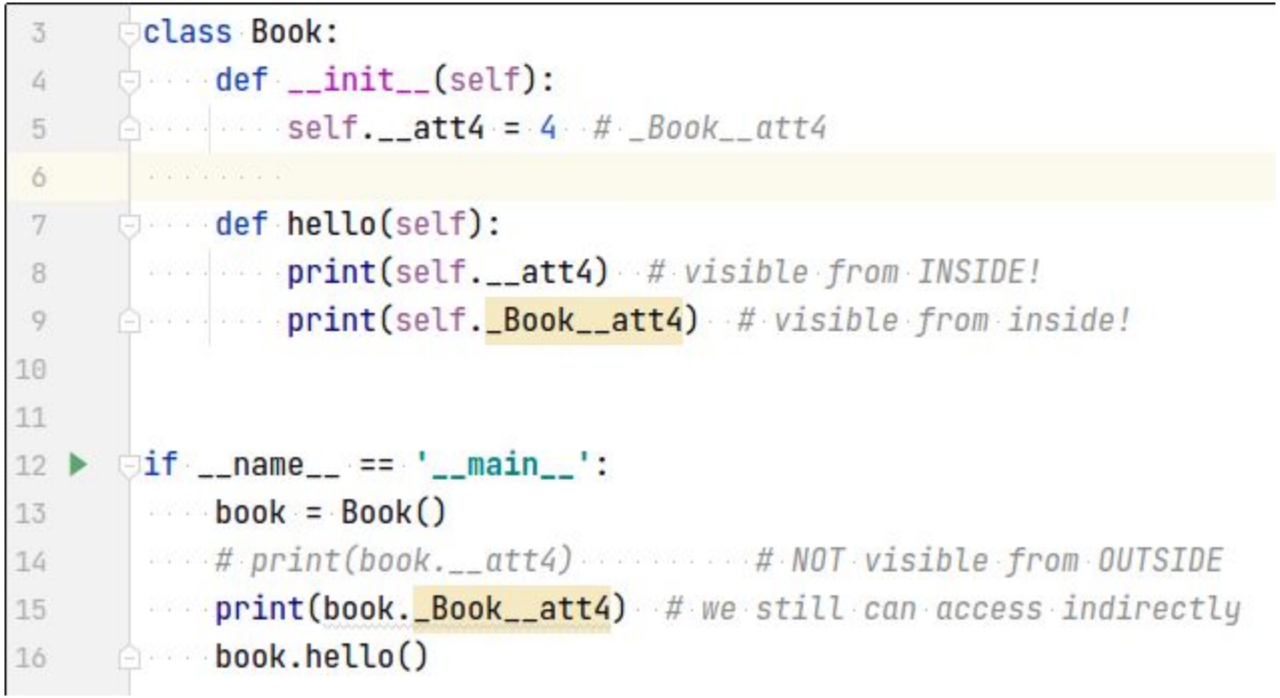


```python
class Book:
    def __init__(self):
        pass
    def __f1(self):
        print('__f1')
    def __f2_(self):
        print('__f2_')
    def _f3(self):
        print('_f3')

book = Book()
#book.__f1()   # AttributeError
#book.__f2_()  # AttributeError
book._f3()     # _f3

print(dir(book))    # return the names in the current scope
# ['_Book__f1', '_Book__f2_', '__class__', ... , '_f3']
```

Visible from inside!



```python
class Book:
    def __init__(self):
        self.__att4 = 4  # _Book__att4

    def hello(self):
        print(self.__att4)  # visible from INSIDE!
        print(self._Book__att4)  # visible from inside!


if __name__ == '__main__':
    book = Book()
    # print(book.__att4)          # NOT visible from OUTSIDE
    print(book._Book__att4)  # we still can access indirectly
    book.hello()
```

Next

- We will know more about **Data-hiding** and why do mangling?

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."