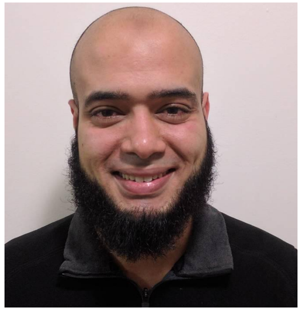
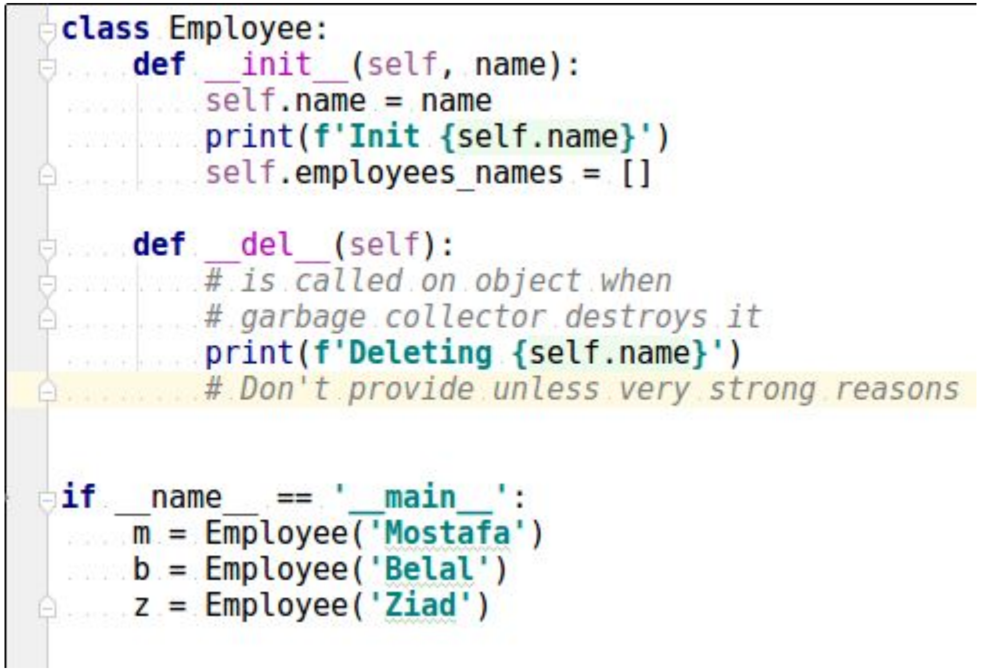
Python Programming
Del Special Method

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)



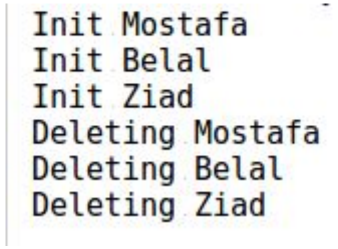
Dunder Del



```python
class Employee:
    def __init__(self, name):
        self.name = name
        print(f'Init {self.name}')
        self.employees_names = []

    def __del__(self):
        # is called on object when
        # garbage collector destroys it
        print(f'Deleting {self.name}')
        # Don't provide unless very strong reasons


if __name__ == '__main__':
    m = Employee('Mostafa')
    b = Employee('Belal')
    z = Employee('Ziad')
```



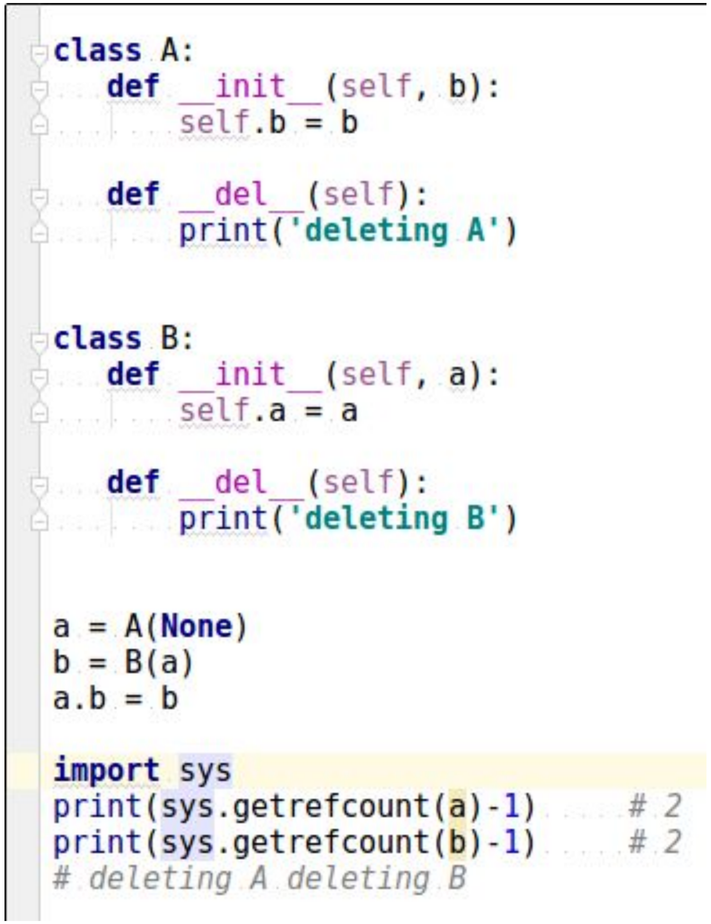
```
Init Mostafa
Init Belal
Init Ziad
Deleting Mostafa
Deleting Belal
Deleting Ziad
```

Memory leak

- In languages like C++, you can create the memory by yourself
  - Then you must free also by yourself
  - If you forgot, they will be there as long as the program is running
  - We call this memory leak: neither used or released
  - If your program allocated a lot of it, the machine memory will be consumed ⇒ Machine hangs
- In python, garbage collector handles the memory for us
  - E.g. using Reference counting, as we learned before
- Most of the cases, your python code is good in terms of memory
  - If you are calling some other language (e.g. C++), there could be memory leak in it
  - In python: be careful from creating dictionary/lists that hold many references without clearing
    - GC won’t clear, as there is a reference

Cyclic References

- Python's standard reference counting mechanism cannot free cycles
  - Supplemental garbage collection facility does (maybe to some extent)
  - Future reading: weakref
- In some special scenarios, we may disable GC
- Future readings: link link link



```
class A:
    def __init__(self, b):
        self.b = b

    def __del__(self):
        print('deleting A')


class B:
    def __init__(self, a):
        self.a = a

    def __del__(self):
        print('deleting B')


a = A(None)
b = B(a)
a.b = b

import sys
print(sys.getrefcount(a)-1)     # 2
print(sys.getrefcount(b)-1)     # 2
# deleting A deleting B
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."