

Python Programming Methods

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)



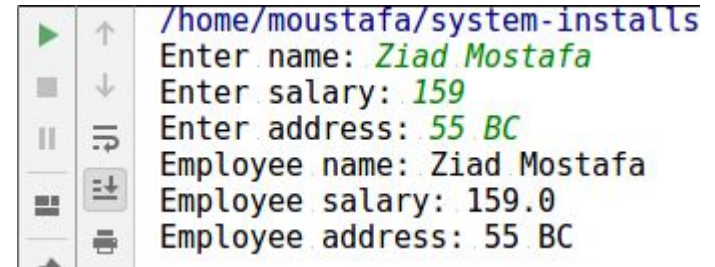
Functions with Objects

- We can create several functions to work over objects
- Read, Write, Process, etc!

```
2
3 class Employee:...
4
5
6
7
8 # Function takes object
9 # print using .
10 def print_empl(object):
11     print('Employee name:', object.name)
12     print('Employee salary:', object.salary)
13     print('Employee address:', object.address)
14
15 x = 1
16 mostafa = Employee()
17 mostafa.name = 'mostafa saad'
18 mostafa.salary = 1000
19 mostafa.address = 'Lovely Canada'
20
21 print_empl(mostafa)
22
23 """
24 Employee name: mostafa saad
25 Employee salary: 1000
26 Employee address: Lovely Canada
27 """
```

Read and Write

```
9 def print_empl(object):
0     print('Employee name:', object.name)
1     print('Employee salary:', object.salary)
2     print('Employee address:', object.address)
3
4 def read_empl():
5     obj = Employee()
6     obj.name = input('Enter name: ')
7     obj.salary = float(input('Enter salary: '))
8     obj.address = input('Enter address: ')
9
0     return obj
1
2 mostafa = read_empl()
3 print_empl(mostafa)
```

A terminal window with a dark background and light text. On the left is a vertical toolbar with icons for running, stepping through, and other debugging actions. The terminal text shows the program's execution path, including prompts for user input and the final printed output of an Employee object.

/home/moustafa/system-installs
Enter name: Ziad Mostafa
Enter salary: 159
Enter address: 55 BC
Employee name: Ziad Mostafa
Employee salary: 159.0
Employee address: 55 BC

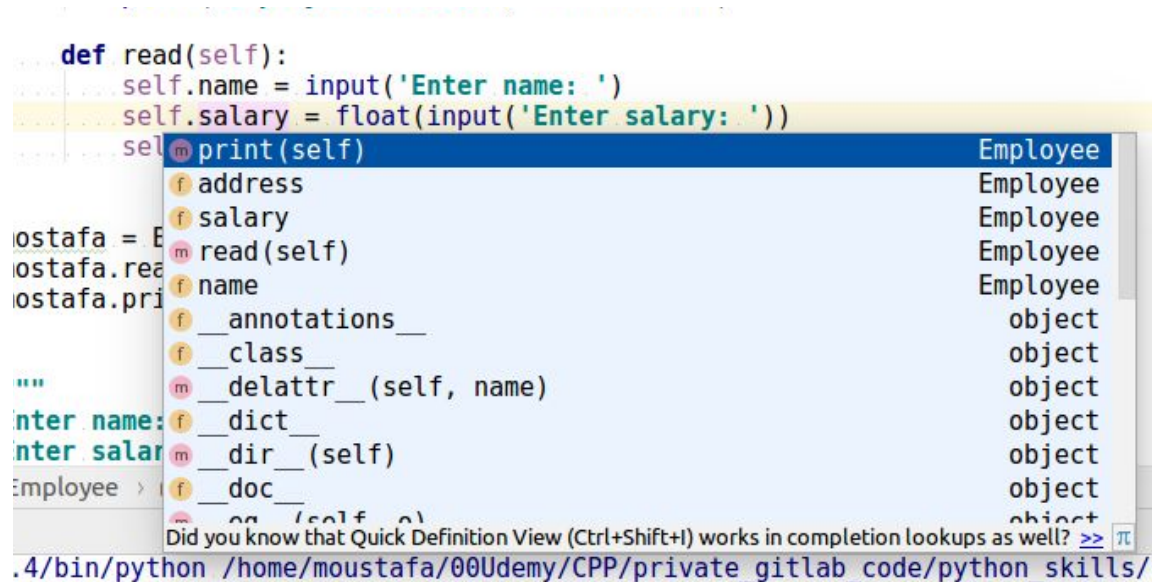
Methods

- Methods are functions defined **inside** the body of a class.
- The **first parameter** is referring to the object itself
 - Convention to name as **self**
 - Automatically added in a call
- Again using (.) operator we can call the methods

```
3 class Employee:
4     name = None
5     salary = None
6     address = None
7
8     def print(self):
9         print('Employee name:', self.name)
10        print('Employee salary:', self.salary)
11        print('Employee address:', self.address)
12
13    def read(self):
14        self.name = input('Enter name: ')
15        self.salary = float(input('Enter salary: '))
16        self.address = input('Enter address: ')
17
18
19 mostafa = Employee()
20 mostafa.read()
21 mostafa.print()
22
```

Ctrl + Space

- In the IDE, after the . press: Ctrl+Space to get menu



```
def read(self):  
    self.name = input('Enter name: ')  
    self.salary = float(input('Enter salary: '))  
    self.print(self)  
  
moustafa = Employee()  
moustafa.read()  
moustafa.print()  
  
"""  
Enter name:   
Enter salary:   
Employee >
```

| | |
|---------------------------|----------|
| m print(self) | Employee |
| f address | Employee |
| f salary | Employee |
| m read(self) | Employee |
| f name | Employee |
| f __annotations__ | object |
| f __class__ | object |
| m __delattr__(self, name) | object |
| f __dict__ | object |
| m __dir__(self) | object |
| f __doc__ | object |
| m __eq__(self, o) | object |

Did you know that Quick Definition View (Ctrl+Shift+I) works in completion lookups as well? >> π

.4/bin/python /home/moustafa/00Udemy/CPP/private_gitlab_code/python_skills/

Encapsulation

- Encapsulation is the **grouping** of **variables and functions** of a specific concept in a single component, named class
 - In C++/Java/C#: the concept is more loaded with **hiding** things from outsiders
 - But Python has another philosophy: trust the other programmers ... More later in OOP
- As you see the class now is very convenient
 - All our related variables are inside the class ... we called them attributes
 - All our related functions are inside the class ... we called them methods

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”