

Python Programming Assigning Functions

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)



Variable = Function

- Python is a flexible language
- We can assign functions to variables
- This can allow several flexible codes

```
4 def fun(a, b):  
5     return a+b, a-b  
6  
7  
8 if __name__ == '__main__':  
9     print(fun(10, 3)) # (13, 7)  
10  
11     # function as variable name  
12     my_fun = fun  
13     print(my_fun(10, 3)) # (13, 7)
```

Passing functions

```
2 def process(iterable, fun):
3     """Iterate on the iterable, apply function and return sum"""
4     sum = 0
5
6     for value in iterable:
7         sum += fun(value)
8
9     return sum
10
11 lst = [2, -4, 6]
12
13 print(process(lst, abs)) ... # 12
14
15 def sq(n):
16     return n*n
17
18 print(process(lst, sq)) ... # 56
19
20 funcs = [abs, sq] ... # list of functions
21 for f in funcs:
22     print(process(lst, f))
```

Key argument

```
3 lst = ['I', 'am', 'Mostafa', 'and', 'You']
4 print(sorted(lst)) # ['I', 'Mostafa', 'You', 'am', 'and']
5
6 # key: will be used to compare elements
7 print(sorted(lst, key = str.lower)) # ['am', 'and', 'I', 'Mostafa', 'You']
8
9 print(sorted(lst, key = len)) # ['I', 'am', 'and', 'You', 'Mostafa']
10
11 def fun(string):
12     if not string:
13         return ''
14     return string[-1].lower()
15
16 print(sorted(lst, key = fun)) # ['Mostafa', 'and', 'I', 'am', 'You']
17
18 n = len(max(lst, key=len)) # 7 = length of longest string in list!
19
20 def get_key(id):
21     if id == 1:
22         return str.lower
23     return len
24
```

Replacing methods

- It is not common, but we can now even replace a method with another

```
21
20
19     emp.print = hack
18
17     emp.print() # Mostafa
16     emp = Employee('Mostafa')
15     if __name__ == '__main__':
14
13         print('Hey!')
12     def hack():
11
10
9         print(self.name)
8     def print(self):
7
6
5         self.name = name
4     def __init__(self, name):
3     class Employee:
```

Everything is object! Even functions.

```
3 def fun():
4     fun.counter += 1
5     print(fun.counter)
6
7     print(type(fun)) # <class 'function'>
8
9 # everything in python is object: so function var is an object!
10 # this means it has attributes!
11
12 print(fun.__dict__) # {}
13 fun.counter = 0
14
15 fun() # 1
16 fun() # 2
17 fun() # 3
18
19 # we typically don't do that, just to administrate the idea!
20
21
```

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”