Python Programming

Function

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

- This is a major topic that we will study later
- Let’s know little about it for now
- Imagine that we wrote a code for some task
  - The code is ~50 lines of code for a common task
  - Another team member needed the same task, so repeated the 50 lines of code
  - In another company, again, they wrote the same 50 lines of code
  - Time waste in repeating our self with such a **common task**
- Function
  - Contains this code
  - We write it **once**
  - Others **call** it to get results

Print Function

- We need to print to the screen what we write!
  - Every programmer needs that
    - Write something
    - Go to next line and so on
- Python provides us a function to do this task
  - Its syntax: print(something)
  - Usage:
    - print(’hello’)
    - print(123)
    - etc

Arguments

- We pass to the function several values
- print(‘mostafa’, ‘is’, 33)
  - We passed 3 arguments. We use comma to separate them
  - The first 2 are of type string. The last is integer

Return

- Sometimes the function returns nothing
  - Such as print()
- But sometimes it returns the **result** of the function
- answer = min(6, 3)
  - The function name is: min
  - It took 2 arguments: 6 and 3
  - The purpose: find the minimum value among the passed arguments
  - The return: the minimum value: 3 in our case
  - answer = min(6, 3)
    - Compute the value of min(6, 3), which is 3
    - Assign it to answer

Min and Max function

```
3   answer = min(3, 6)
4   print(answer)           # 3
5
6   answer = min(3, 6, -2)
7   print(answer)           # -2
8
9   answer = max(9, 6, -2, 15)
10  print(answer)           # 15
11
12  # nested Function calls
13  # max(4, 7) computed first => 7
14  # print(7)  => 7
15  print( max(4, 7) )      # 7
16  print( max(4, 7) + 2)   # 9
17
```

Type function

```
2
3   # We can use _ with int/float
4   # to write numbers in easier wau
5   age = 1000_000_000      # int
6
7   print( type(15) )       # <class 'int'>
8   print( type(age) )      # <class 'int'>
9
10  print( type(20.5) )     # <class 'float'>
11
12  print( type('20.5') )   # <class 'str'>
13
```

len function

```
3   str = 'mostafa'
4
5   # Compute length of the string
6   str_len = len(str)
7
8   print(str_len)      # 7
9   print(len(str))     # 7
10
11  # Don't use variable name same as function name
12  len = len(str)      # Don't
13  #len = len(str)     # Now error!
14
```

Conversions

```
msg_str = '10'
xint = int(msg_str)
xflo = float(msg_str)

print(msg_str, type(msg_str))   # 10 <class 'str'>
print(xint, type(xint))         # 10 <class 'int'>
print(xflo, type(xflo))         # 10.0 <class 'float'>

my_float = 20.7
my_int = int(my_float)
msg = str(my_float)


print(my_float, type(my_float)) # 20.7 <class 'float'>
print(my_int, type(my_int))     # 20 <class 'int'>   observe loss of .7
print(msg, type(msg))           # 20.7 <class 'str'>

# Tip: Don't use variable name same as function name
# such as int, str, len, min, max, etc
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."