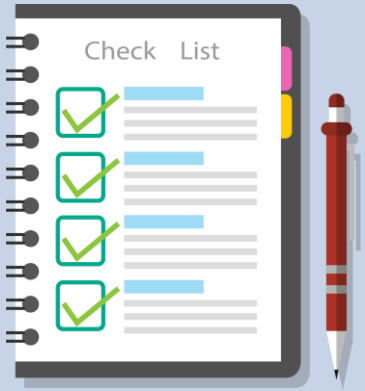
Python – Functions

Digital Lync

INNOVATION - EDUCATION - INCUBATION

Shah Ayub Quadri
aquadri@digital-lync.com

Index



Python Basics

- Function Advanced
    - Arguments
    - Recursion
    - Anonymous Function

Functions

Arguments:

- The parameters which we passed to a function are called as arguments.
- There are different way how we can pass arguments in python.

Types of Arguments:

Default Arguments:

»Function arguments can have default values in Python.

»This default value to an argument done by using the assignment operator (=)

```
1  def message(name, msg = "is God of cricket"):
2      print(name + ', ' + msg)
3
4  message("Sachin")
5  message("Dhoni","The great captain of India")
```

```
C:\Users\gsanjeevareddy\Desktop>python functions.py
Sachin, is God of cricket
Dhoni, The great captain of India
```

» In example we provide default value to msg argument and to name during function call.

» If we provide msg value during function call it will overwrite the default value.

Arbitrary arguments:

» When we do not know in the number of arguments that will be passed into a function. Python allows us to handle this kind of situation through function calls with arbitrary number of arguments.

» In the function definition we use an asterisk (*) before the parameter name to denote this kind of argument.

```
1  def students(*names):
2
3      # names is a tuple with arguments
4      for name in names:
5          print(name)
6
7  students('ravi','raju','ramesh','venkatesh','sunil')
```

```
C:\Users\gsanjeevareddy\Desktop>python functions.py
ravi
raju
ramesh
venkatesh
sunil
```

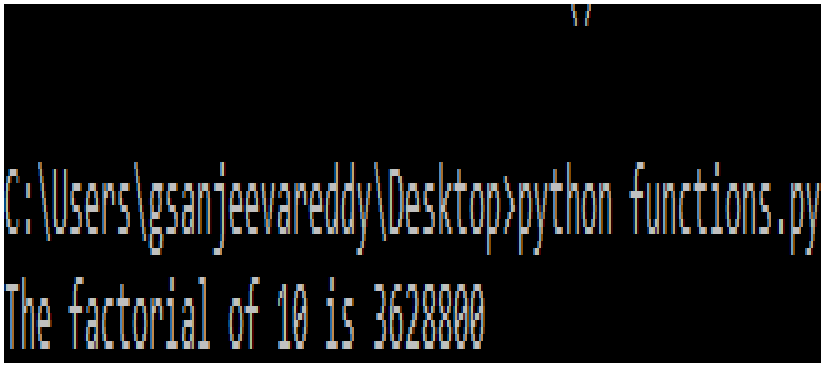
»We have called the function with multiple arguments. These arguments get wrapped up into a tuple before being passed into the function. Inside the function, we use a for loop to retrieve all the arguments back.

Functions

Recursive Function:

» A function can call other functions. It is even possible for the function to call itself. These type of construct are termed as recursive functions.

```
1   def factorial(x):
2       """This is a recursive function
3       to find the factorial of an integer"""
4
5       if x == 1:
6           return 1
7       else:
8           return (x * factorial(x-1))
9
10  num=10
11  print("The factorial of", num, "is", factorial(num))
```



» Here factorial factorial is a recursive function as it calls itself.
» It will stop when the value of x=1.

Functions

Anonymous functions:

» Anonymous function is a function that is defined without a name.

» Normal functions are defined using the def keyword, in Python anonymous functions are defined using the lambda keyword.

Syntax of Lambda Function:

lambda arguments: expression

» Lambda functions can have any number of arguments but only one expression. The expression is evaluated and returned. Lambda functions can be used wherever function objects are required.

```
1  power = lambda x: x ** x
2  print(power(5))
```

```
C:\Users\gsanjeevareddy\Desktop>python functions.py
3125
```

» Here we use lambda function to calculate power of a value which is written in expression.

Functions

Use of lambda function:

» We use lambda function when we require a nameless function for a short period of time.

» Lambda functions are used along with built-in functions like filter(), map(),reduce().

Use with filter:

» filter() function in Python takes in a function and a list as arguments.

» The function is called with all the items in the list and a new list is returned which contains items for which the function evaluates to True.

```
1  # Program to filter out only the items
2
3  A = [8,11,89,10,34,2,9,6,13,99]
4
5  B = list(filter(lambda x: (x>10) , A))
6
7  print(B)
```

```
C:\Users\gsanjeevareddy\Desktop>python functions.py
[11, 89, 34, 13, 99]
```

Functions

Use with map:

» The map() function in Python takes in a function and a list..

» The function is called with all the items in the list and a new list is returned which contains items returned by that function for each item.

```
1  # Program to calculate power  each
2
3  A = [1, 5, 4, 6, 8, 11, 3, 12]
4
5  B = list(map(lambda x: x * x , A))
6
7  print(B)
```

```
C:\Users\gsanjeevareddy\Desktop>python functions.py
[1, 25, 16, 36, 64, 121, 9, 144]
```

Functions

Use with reduce:

» reduce() function in Python takes in a function and a list as arguments.

» At each step, reduce passes the current action, along with the next item from the list, to the passed-in lambda function.

» By default, the first item in the sequence initialized the starting value.

» To use reduce we have to import reduce module from “functools”.

```
1  from functools import reduce
2  print (reduce( (lambda a, b: a ** b), [1, 2, 3, 4] ))
```

```
C:\Users\gsanjeevareddy\Desktop>python functions.py
1
```

Function calling method

Python is nether call by value nor call by reference, instead its call by reference object

Call by value	Call by reference
Copies the value and then passed to the function. That's why update will not affect the actual value	Copies the address and then passed to function. So the update will reflected to the actual value
Eg: use immutable objects (int, string, tuple) and it behaves as call by values	Eg: use mutable objects (list, dict, sets)

https://www.youtube.com/watch?v=hDmnu1qGfkw

Assignment - 5

1. Write a Python function to find the Max of three numbers..

2. Write a Python function to sum all the numbers in a list.
   Sample List : (8, 2, 3, 0, 7)
   Expected Output : 20
3. Write a Python function to multiply all the numbers in a list.
   Sample List : (8, 2, 3, -1, 7)
   Expected Output : -336
4. Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument..

5. Write a Python program to access a function inside a function