**Python training Notes:**

**Course Name:** **SCRIPT 307: Basic Python**

**This is Part 1 of the whole training in the duration 17 to 31 July**

**This will be followed with next Part 2 session for Intermediate Python topics in the month of August.**

**Day 6: 24 Jul 2018 - Tuesday (2 Hrs Session)**

**Expectation Setting ASL (Assisted Self-Learning) 2Hrs session daily**

**And then do self-study and hands on assignments from below learning course link and the assignments given below here in this document:**

<https://knowledgecenter.persistent.co.in/ViewCourse/pmoc>

***Please visit the following URL to view the collaborative learning group***

<https://persistentuniversity.persistent.co.in/CollaborativeLearningGroup/view.aspx?SkillId=9144>

**Topics Covered:**

List Comprehension ,

Functional Programming:

map () function , lambda usage in map

filter() reduce() functions, use of lambda in them

complex sorting

operator module

**\*\*\*\*\*To Do for Day 6:**

Nugget 1 : Introduction to Python & Python Fundamentals

Nugget 2 : Python Basics

Nugget 3 : Python Control Structures

Nugget 4 : Functions & Modules

Subjective Assignment for Nugget 1 to 3 : Only for self Practice

522

1. Complete reading these 4 Nuggets from <https://knowledgecenter.persistent.co.in/ViewCourse/pmoc>

2. Please execute all codes in these 4 Nuggets

3. Start solving assignment at the end of Nuggets

\*\*Also read the Python documentation for the topics covered till date.

**Try Below Codes:**

**Functional Programming**

**6\_map\_lambda\_temperature\_ex.py**

Celsius = [39.2, 36.5, 37.3, 37.8]

print "Original temp in Celsius =", Celsius #Original temp in Celsius = [39.2, 36.5, 37.3, 37.8]

F1 = [(float(9)/5)\*x + 32 for x in Celsius] #1) List comprehension

print "Fahrenheit temp list = ", F1

print "------------------------------------------------------------------"

def convert\_to\_Fahrenheit(x): #2) map

return (float(9)/5)\*x + 32

F2 =map(convert\_to\_Fahrenheit, Celsius)

print "Fahrenheit temp list = ", F2

print "--------------------------------------------------------------------"

F2a =map(lambda x : (float(9)/5)\*x +32, Celsius) #3)map and lambda

print "Fahrenheit temp list = ", F2a

print "-------------------------------------------------------------------"

**7\_map\_lambda\_multiple\_list.py**

a = [1,2,3,4]

b = [17,12,11,10,55]

c = [-1,-4,5,9]

def additionFun(x,y):

return x+y

additionlist = map(additionFun, a,b) #additionlist(a[0],b[0])

print "Original list a = ",a

print "Original list b = ",b

print "Addition list = ",additionlist

print "---------------------------------------------"

print map(lambda x,y:x+y, a,b)

#[18, 14, 14, 14]

print map(lambda x,y,z:x+y+z, a,b,c)

#[17, 10, 19, 23]

print map(lambda x,y,z:x+y-z, a,b,c)

#[19, 18, 9, 5]

**2\_filter\_demo.py**

r = [0, 1, 2, 3, 4, 5, 6]

def large(x): return x>3 #user defined function

target\_list = filter(large, r)

print "Target list = ",target\_list #Target list = [4, 5, 6]

print "---------------------------------------------------------"

def lowercase(x): return x.islower()

words =['abc', 'XYZ','lmn','NIL']

upper\_words =filter(lambda x :x.isupper(),words)

print "Words = ",words #Words = ['abc', 'XYZ', 'lmn', 'NIL']

print "Upper Case words = ",upper\_words #Upper Case words = ['XYZ', 'NIL']

**3\_reduce\_demo.py**

r = [0, 1, 2, 3, 4, 5, 6]

def sum(x, y): return x+y

ans = sum(10,20)

print "addition of 10 and 20 =", ans #addition of 10 and 20 = 30

print "-------------------------------------------------"

print reduce(sum, r) #21 sum(0,1) sum(1,2) =3 sum(3,3)

**9\_reduce\_lambda\_ex.py**

print reduce(lambda x,y: x+y, [47,11,42,13]) #113

print "--------------------------------------------------"

#example:get minimum number from a list : reduce and lambda

list1 = [47,11,42,102,13] #greater number =102

f = lambda a,b: a if (a > b) else b

print reduce(f, [47,11,42,102,13]) #102

**ListSorting.py**

unsortedList = ['Aaaa', 'bb', 'cccccccc', 'zzzzzzzzzzzz']

print 'Unsorted list is ', unsortedList

sortedList = sorted(unsortedList, key=len) #

print 'Sorted list is ', sortedList

**5\_lambda\_complex sort.py**

from operator import itemgetter #operator is module file

#sort data by age

student\_tuples = [ ('john', 'A', 25), ('jane', 'B', 32), ('nil', 'B', 50),('dave', 'B', 28) ] #list of tuples

#sorted , lambda

sorted\_students\_byAge = sorted(student\_tuples, key=lambda student: student[2]) # sort by age

print "Student data = ", student\_tuples #Student data = [('john', 'A', 25), ('jane', 'B', 32), ('nil', 'B', 50), ('dave', 'B', 28)]

print "Sorted students by age = ",sorted\_students\_byAge #Sorted students by age = [('john', 'A', 25), ('dave', 'B', 28), ('jane', 'B', 32), ('nil', 'B', 50)]

print "-------------------------------------------------------------------"

sorted\_students\_byAge1 = sorted(student\_tuples, key=itemgetter(2))

print "Student data = ", student\_tuples #Student data = [('john', 'A', 25), ('jane', 'B', 32), ('nil', 'B', 50), ('dave', 'B', 28)]

print "Sorted students by age = ",sorted\_students\_byAge1 #Sorted students by age = [('john', 'A', 25), ('dave', 'B', 28), ('jane', 'B', 32), ('nil', 'B', 50)]

print "-------------------------------------------------------------------"

#multiple levels of sorting

sorted\_students\_byAge\_Grade = sorted(student\_tuples, key=itemgetter(1,2))

print "Sorted students by grade and age = ",sorted\_students\_byAge\_Grade #Sorted students by grade and age = [('john', 'A', 25), ('dave', 'B', 28), ('jane', 'B', 32), ('nil', 'B', 50)]

To do task: Please check what is apply()

**Assignments to do:**

1. Let’s say I give you a list saved in a variable:

a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100].

Write one line of Python that takes this list a and makes a new list b

that has only the even elements of this list in it. (Use filter and lambda)

2. Sort below given list by length of Name field from inner tuple

student\_tuples = [ ('john', 'A', 25), ('jane', 'B', 32), ('nil', 'B', 50),('dave', 'B', 28) ] #list of tuples

Save the solutions in a folder: **Assignments\Day6**

**Assignments\Day6 --🡪**

**Q1.py**

**Q2.py**