**Python training Notes:**

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

**Day 4: 14 Sep 2017**

**Expectation Setting : ASL (Assisted Self-Learning) 3Hrs session daily**

**And then do self-study and hands on assignments form below learning course link:**

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

**Micro Learning Online link :** <https://persistentuniversity.persistent.co.in/microlearning/Course/837>

Welcome to Micro Learning!!

You can now complete a course by learning small units at a time, i.e. learning Micro Nuggets. Take out 15 minutes time daily and complete one micro nugget and a short quiz. Click on Complete button to ensure the completion and to get access to next Micro Nugget on next day.

Use the link given above to visit the Micro Learning Site to view the next micro nugget.

Every Micro Nugget is of short period 10-15mins approximately.

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

<https://persistentuniversity.persistent.co.in/CollaborativeLearningGroup/View.aspx?SkillID=8451>

***My Python recording youtube as part of Smart India Hackathon***

<https://www.youtube.com/watch?v=D2SkCdNZW08>

**Topics:**

**File Handling**

**Modules Packages**

**\*\*\*\*\*To Do for Day4:**

Nugget 1 : Introduction to Python & Python Fundamentals

Nugget 2 : Python Basics

Nugget 3 : Python Control Structures

663

Nugget 4 : Functions & Modules

Nugget 5 : I/O & Exception handling

1. Complete reading above Nuggets

2. Please excute all codes in these Nuggets

3. Start solving assignment at the end of Nuggets

4. Solve Subjective Assignment for Nugget 1 to 3 : Only for self Practice

**Try Below Codes:**

**File Handling**

**1\_fileopen.py**

f = open("myfile.txt", "w") #file object created in write mode

print dir(f)

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

f.write("AAAAAAAA\n")

f.close();

**2\_fileread.py**

myfile = open('data.txt', "r")

f = open("target.txt", "w")

while True:

str = myfile.readline()

if str:

print str

f.write(str)

else:

print "Nothing in file"

break

myfile.close()

f.close()

#print myfile

#print myfile.tell() #error

**3\_file\_read\_inloop.py**

f = open('data.txt')

while True:

char = f.read(1)

if not char: break

print char

f.close()

#as opposed to

#for char in open('test.txt').read():

# print char

**4\_FileRead1\_readlines.py**

fin=open('data.txt','r')

data =fin.readlines() #list

#print data

for line in data:

print line

fin.close()

"""

list1 =[1,55,99]

print list1[-2]

print list1[0]

"""

**5\_FileWritre1.py**

f=open('target.txt','w')

print f.tell()

f.write('PSL')

print f.tell()

f.close()

**6\_file\_seek.py**

f = open('target.txt', "w")

f.write("hello World\n")

f.close()

f = open("myfile.txt")

while True:

str = f.readline()

if str:

print str

else:

break

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

f.seek(5) #absolute positioning with context to start of your file

print f.read()

f.close()

**Modules**

**mymath.py**

"""mymath - our example math module"""

pi = 3.14159

def area(r):

"""area(r): return the area of a circle with radius r."""

global pi

return(pi \* r \* r)

# write Fibonacci series up to n

def fib(n):

a, b = 0, 1

while b < n:

print b

a, b = b, a+b

if \_\_name\_\_ == "\_\_main\_\_":

#import sys

#fib(int(sys.argv[1]))

print "Learning Modules in Python......"

fib(35) #calling a function

print "END!!!!!"

**mymath2\_test.py**

#from mymath import pi

#from mymath import pi, area

from mymath import \*

print pi

print area(5)

fib(25)

**mymath2\_test2.py**

import mymath

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

print mymath.pi

mymath.pi+=100

print mymath.pi

print mymath.area(5)

mymath.fib(35)

mymath1.pi

Package :

1. Create Animals folder
2. Create Birds and Mammals module files in Animals folder
3. Create \_\_init\_\_ inside Animals package folder
4. Create test.py testing file outside Animals package folder

class Birds:

def \_\_init\_\_(self): #constructor

''' Constructor for this class. '''

# Create some member animals

self.members = ['Sparrow', 'Robin', 'Duck'] #members is an instance property

def printMembers(self): #instance method

print('Printing members of the Birds class')

for member in self.members:

print('\t%s ' % member)

"""

b1 =Birds()

b1.printMembers()

"""

class Mammals:

def \_\_init\_\_(self):

''' Constructor for this class. '''

# Create some member animals

self.members = ['Tiger', 'Elephant', 'Wild Cat']

def printMembers(self):

print('Printing members of the Mammals class')

for member in self.members:

print('\t%s ' % member)

**\_\_init\_\_.py**

from Mammals import Mammals

from Birds import Birds

#importing Mammals class from Mammals module file

**Test.py**

# Import classes from your brand new package

from Animals import Mammals

from Animals import Birds

# Create an object of Mammals class & call a method of it

myMammal = Mammals()

myMammal.printMembers()

# Create an object of Birds class & call a method of it

myBird = Birds()

myBird.printMembers()

**Assignments to do:**

**Assignment on ----->File Handling : Country.txt file :**

**Supportive documents available at** <https://persistentuniversity.persistent.co.in/CollaborativeLearningGroup/View.aspx?SkillID=8451>

**>Day 4 Assignment folder**

4.1 Read file content of given file “empdata”, and print total sal and all emp deatils.

File data:

1a:ABC:25:25000

2a:XYZ:30:30000

3a:LMN:45:60000

Increment sal by 15000 and print updated data

**4.2**

**Language and Country**

Store the Country data only for Language and its list of countries in a dictionary.

Display the o/p as shown below -

>>>

{'Portuguese': ['Brazil'], 'Franch': ['Cameroon', 'Djibouti', 'Equatorial Guinea', 'France'], 'Chinese': ['China'], 'Vietnamese': ['Vietnam'], 'German': ['Germany'], 'English': ['United Kingdom', 'United States', 'Fiji', 'Canada', 'Ireland'], 'Japanese': ['Japan'], 'Greek': ['Greece'], 'Indian': ['India'], 'Spanish': ['Venezuela', 'Argentina', 'Honduras'], 'Arabic': ['Yemen', 'Bahrain'], 'Hungerian': ['Hungary'], 'Italian': ['Italy']}