=========================================

Packages in Python

=========================================

Index

-----------

=>Purpose of Package

=>Definition of Package

=>Development of Package

=>Number of Approaches to re-use Packages

a) By Using sys.path.append()

b) By using PYTHONPATH Environmental Variable

=>Programming Examples

-------------------------------------------------------------------------------------------------------------------

=============================================

Package in Python

=============================================

=>The Function concept is used for Performing some operation and provides code re-usability within the same program and unable to provide code re-usability across programs.

=>The Modules concept is a collection of Variables, Functions and classes and we can re-use the code across the Programs provided Module name and main program present in same folder but unable to provide code re-usability across the folders / drives / enviroments.

=>The Package Concept is a collection of Modules.

=>The purpose of Packages is that to provide code re-usability across the folders / drives / enviroments.

=>To deal with the package, we need to the learn the following.

a) create a package

b) re-use the package

--------------------------------------------------------------------------------------------------

a) create a package:

----------------------------

=>To create a package, we use the following steps.

i) create a Folder

ii) place / write an empty python file called \_\_init\_\_.py

iii) place / write the module(s) in the folder where is it considered as Package Name

Example:

--------------

bank <-----Package Name

-----------

\_\_init\_\_.py <----Empty Python File

simpleint.py <--- Module Name

aop.py-----Module Name

icici1.py---Module Name

welcome.py <--- Module Name

========================================================

b) re-use the package

---------------------------------

=>To the re-use the modules of the packages across the folders/drives /enviroments, we have to two approaches. They are

i) By using sys module

ii) by using PYTHONPATH Environmental Variable Name

------------------------------------------------------------------------------------------

i) By using sys module:

-------------------------------------

Syntax:

----------- sys.path.append("Absolute Path of Package")

=>sys is pre-defined module

=>path is a pre-defined object / variable present in sys module

=>append () is pre-defined function present in path and is used for locating the package name of python( specify the absolute path)

Example:

sys.path.append("E:\\KVR-PYTHON-6pM\\ACKAGES\\BANK")

(or)

sys.path.append("E:\KVR-PYTHON-6PM\ACKAGES\BANK")

(or)

sys.path.append("E:\KVR-PYTHON-6PM/ACKAGES/BANK")

-----------------------------------------------------------------------------------------------

ii) by using PYTHONPATH Enviromental Variables:

-----------------------------------------------------------------------

=>PYTHONPATH is one of the Enviromental Variable

=>Search for Enviromental Variable

Steps for setting:

------------------------------

Var name: PYTHONPATH

Var Value: E:\KVR-PYTHON-7AM\PACKAGES\BANK

The overall path

PYTHONPATH= E:\KVR-PYTHON-11AM\PACKAGES\BANK

--------------------------------------------------------------------------------------

#kvrmath.py---File name and Module name

def kvrsqrt(n):

print("sqrt({})={}".format(n,n\*\*0.5))

#MathCal.py---File Name and Module Name

def fact(n):

if(n<0):

print("{} Not posssible to calculate Factorial".format(n))

else:

f=1

for i in range(1,n+1):

f=f\*i

else:

print("fact({})={}".format(n,f))

#LetterCount.py----Module Name

def LetterCount():

line=input("Enter a Line of text:")#aaabbzzzzeeee

d={} # Empty dict

for ch in line:

if ch in d:

d[ch]=d[ch]+1 # OR d[ch]=d.get(ch)+1

else:

d[ch]=1

else:

print("-"\*50)

for let,noc in d.items():

print("\t{}-->{}".format(let,noc))

print("-" \* 50)

#d={'a':3,'b':2,'z':4,'e':4}