Python Modules and Packages:

1. Folders which contains classes and functions with the extension '.py'. Also called as scripts(modules).
2. Modules can be of Predefined and User-Defined.
   Ex: re, abc, time, threading, tkinter, os → examples of Predefined modules.
3. If the developer creates his/her own module then such module is called 'UserDefined Module'.
4. To get a module onto a python program developer has to use keyword 'import' to get that module.
5. Importing module is of several ways.
   a. import module-name
   b. import module-name as temp-name
   c. from module-name import class(or)function-name
   d. from module-name import class(or)function-name, fun-name2, so..on..
   e. from module import *
6. Generally, modules names are written in small case.

NOTE: If two or modules have same functions, and developer imports both the modules then which module's function will be executed?

➔ Last called (Most recent imported) module's function will be executed…

Modules and Classes in Python:

```
#create in classpak1 module

class First:
   def fun_first(self):
      print("classpak1.First.fun_first()")
      pass
   pass
-----------------------------------------
#create in classpak2 module

class Second:
   def fun_second(self):
      print("classpak2.Second.fun_second()")
      pass
   pass
-----------------------------------------------------
```

#import and exe in different file

```
import classpak1
import classpak2

obj1 = classpak1.First()
obj1.fun_first()

obj2 = classpak2.Second()
obj2.fun_second()
```

→Above prog can be done in the following manner also

```
from classpak1 import First
from classpak2 import Second

obj1 = First()  #not need to mention module name
obj2 = Second()

obj1.fun_first()
obj2.fun_second()
```

Package (or) Folder in Python:

1. Multiple no of functions can be stored in 'modules'.
2. Multiple of modules can be stored in 'packages'.
3. In python packages or folders are created manually but not programmatically.
4. A package can contain sub package(s). But this case is not applicable to modules.

NOTE: How to check the no of classes in a module?

import <module>
print(dir(module-name))

- ➔ **dir(module-name)** will give all the no of classes of a module.
- ➔ If no classes are present in the module then module names are shown.

5. Predefined module named 'sys' contains an attribute called 'path' which in turn contains a function called 'append()'.
6. This function 'append()' is used to append a 'package' path to the current python file, so that developer can import one or more modules from the package.
7. In 'append()' , provide the physical path location of the modules.

Ex:

```
import sys
sys.path.append("C:\Users\pendima\Desktop\Pyhton r\pak1")

#from A import FirstClass
#from B import SecondClass

import A, B

obj1 = A.FirstClass()
obj2 = B.SecondClass()

obj1.firstMethod()
obj2.secondMethod()
```

---

8. When we have sub-packages and want to use super and sub package, then developer has to append both the packages separately using ‘sys’ module.

Ex:

```
#pak1 folder /A.py module
class FirstClass:
   def firstMethod(self):
      print("FirstClass.firstMethod() from A.py module of 'pak1' package")
      pass
   pass

#pak1/B.py module
class SecondClass:
   def secondMethod(self):
      print("SecondClass.secondMethod() from B.py module of 'pak1' package")
      pass
   pass

#pak1/subpak1/C.py module
class ThirdClass:
   def thirdMethod(self):
      print("ThirdClass.thirdMethod() from C.py module of 'pak1/subpak1' package")
      pass
   pass
```

```
#out of all modules
import sys
sys.path.append("C:\Users\pendima\Desktop\Pyhton r\pak1")

#from A import FirstClass
#from B import SecondClass

import A, B

obj1 = A.FirstClass()
obj2 = B.SecondClass()

obj1.firstMethod()
obj2.secondMethod()

sys.path.append("C:\Users\pendima\Desktop\Pyhton r\pak1\subpak1")
import C
obj3 = C.ThirdClass()
obj3.thirdMethod()
```

One more Example on packages and modules in python:

1. Create 3 packages ‘pak2’,’pak3’ and ‘user’.
2. In pak2 create module employee and class Employee with a constructor and a function.
3. In pak3 create module student and class Student with a constructor and a function.
4. In user package create class ‘UseEmpStu’ class and access the data from Employee and Student classes.

```
#package --> pak2, module --> employee

class Employee:
   def __init__(self,eid,ename,esal):
      self.eid = eid
      self.ename = ename
      self.esal = esal
      pass

   def displayEmpInfo(self):
      print("Emp id={} Emp name={} and Emp salary={}".format(self.eid,self.ename,self.esal))
      pass
```

```
#package --> pak3
#module --> student

class Student:
   def __init__(self, sid, sname):
      self.sid = sid
      self.sname = sname
      pass

   def displayStudentInfo(self):
      print("Student id:{} Stu name:{}".format(self.sid, self.sname))


#package --> user
#module or Class --> UseEmpStu.py

import sys
sys.path.append("C:\Users\pendima\Desktop\Pyhton r\pak2")
sys.path.append("C:\Users\pendima\Desktop\Pyhton r\pak3")

from Employee import Employee
from Student import Student

emp =  Employee(121,'abcd',25000)
print('Employee ID: {}' .format(emp.eid))
print('Employee Name: {}' .format(emp.ename))
print('Employee Salary: {}' .format(emp.esal))
print('\n-------------------------\n')

stu =  Student(100,'xyz')
print('Student ID: {}' .format(stu.sid))
print('Student Name: {}' .format(stu.sname))
```

Generally used Pre-defined modules of Python:

1. Module 'math'
   import math

```
info = dir(math)
for i in info:
   print(i)
   pass
```

2. Ex – 2

```
import math
print(math.ceil(30.3))
print(math.fabs(10))    #converts its argument to float
print(math.factorial(5))
print(math.floor(30.9))
print(math.pow(3,4))
print(math.sqrt(4))
print(math.sin(90))
print(math.cos(0))
print(math.pi)
print(math.e)   #exponential value
print(math.exp(1))
```

---

module ‘os’

```
import os
info = dir(os)
for i in info:
   print(i)
   pass
```

Ex:

```
import os

os.rename('oldfile','newfile')
os.remove('filename')
os.mkdir('new dir name')
os.chdir('new dir name')
print(os.getcwd()) #prints current working dir
os.rmdir('dir name') #removes dir
```

---

module ‘random’:

```
import random
c = dir(random)
for i in c:
   print(i)
   pass
```

```
import random

print(random.randint(1,100))
print(random.choice(['red', 'black', 'green']))

myList = [2,10.5, False, "abcd", "xyz"]
print(random.choice(myList))
```

---

module ‘time’

```
import time

t = dir(time)
for tt in t:
   print(tt)
   pass
```

Ex-1:

```
import time

print("Welcome to time module")
time.sleep(2)
print(time.strftime('%X %x %Z'))    #returns current formatted time
print("Have a good day")
```

Ex-2:

```
from datetime import datetime

now = datetime.now() # current date and time

year = now.strftime("%Y")
print("year:", year)

month = now.strftime("%m")
print("month:", month)

day = now.strftime("%d")
print("day:", day)

time = now.strftime("%H:%M:%S")
print("time:", time)

date_time = now.strftime("%m/%d/%Y, %H:%M:%S")
print("date and time:",date_time)
```

---

module ‘sys’:

```
import sys

s = dir(sys)
for ss in s:
    print(ss)
    pass
```