LAB # 11

MODULES AND PACKAGES

OBJECTIVE

Getting familiar with the envoirnment for using modules and packages.

THEORY

Modules
A module allows you to logically organize your Python code. A file containing a set of functions you want to include in your application.

Create a Module
To create a module just save the code you want in a file with the file extension .py:

Example
Save this code in a file named ***mymodule.py***

```
def greeting(name):
  print("Hello, " ,name)
```

Use a Module
Now use different ways to call the module we just created, by using the ***import*** statement:

i) import module1[, module2[,... moduleN]
ii) import module as m
iii) from module import functionName

Example
Import the module named ***mymodule***, and call the greeting function:

```
import mymodule
mymodule.greeting("XYZ")
```

Built-in Modules
There are several built-in modules in Python, which you can import, here is the example of math and sys module;

Python - math Module
Math module provide the usage of mathematical functions.

Example:

```
from math import pi
r=int(input("Enter:"))
print("Area:" , pi*(r**2))
```

Output:

```
>>> %Run task1.py
enter:3
Area: 28.27
```

Functions in Python Math Module

Functions	Description
ceil(x)	Returns the smallest integer greater than or equal to x.
factorial(x)	Returns the factorial of x
floor(x)	Returns the largest integer less than or equal to x
exp(x)	Returns e**x
cosh(x)	Returns the hyperbolic cosine of x
sinh(x)	Returns the hyperbolic cosine of x
tanh(x)	Returns the hyperbolic tangent of x
pow(x, y)	Returns x raised to the power y
sqrt(x)	Returns the square root of x
pi	Mathematical constant, the ratio of circumference of a circle to it's diameter (3.14159...)
e	mathematical constant e (2.71828...)

Python - sys Module

The sys module provides functions and variables used to manipulate different parts of the Python runtime environment

Example:

```
import sys
print(sys.version) #version number of the current Python interpreter
```

Output:

```
>>> %Run task2.py
3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 14 2019, 23:09:19) [MSC v.1916 32 bit (Intel)]
```

Packages

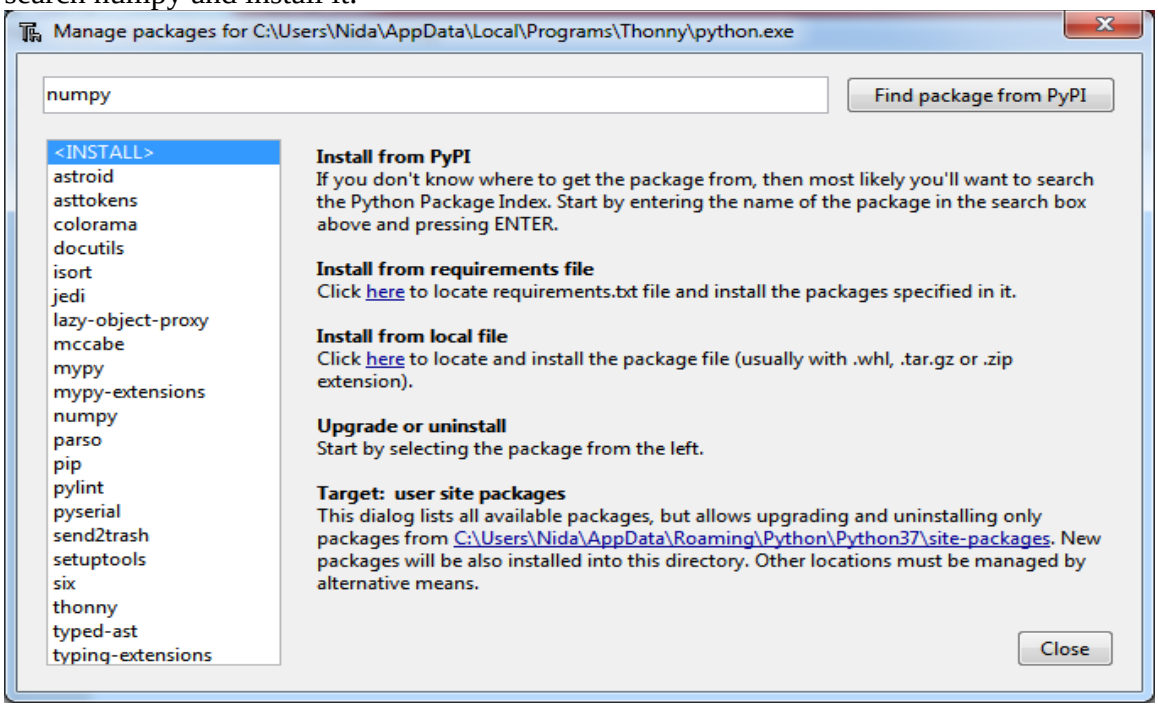
A package is a collection of python modules, i.e., a package is a directory of python modules containing an additional __init__.py file. Each package in Python is a directory which must contain a special file called __init__.py. This file can be empty, and it indicates that the directory it contains is a Python package, so it can be imported the same way a module can be imported. There are 130k + packages and still growing , numpy is one of the most running and useful python's package

NumPy

NumPy is a purposely an array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with python.

Installation:

In Thonny, there is a menu and select ***Tools*** option and then select ***Manage packages,*** search numpy and install it.



Example: Create 1D,2D,3D array

```
import numpy as np
#1D array
print("1D:",np.arange(2,6).reshape(4))
#2D array
print("2D:",np.arange(2,10).reshape(2,4))
#3D array
print("3D:",np.arange(24).reshape(4,3,2))
```

EXERCISE

A. Point out the errors, if any, and paste the output also in the following Python programs.

1. Code:

```
import sys as s
print(sys.executable)
print(sys.getwindowsversion())
```

Output:

2. Code:

```
import datetime
from datetime import date
import times
# Returns the number of seconds
print(time.time())
# Converts a number of seconds to a date object
print(datetime.datetime.now())
```

Output:

3. Code:

```
From math import math
# using square root(sqrt) function contained
print(Math.sqrt(25) )
print(Math.pi)
# 2 radians = 114.59 degreees
print(Math.degrees(2))
```

Output:

B. What would be the output of the following programs:

1. Code:

```
import calendar
yy = 2017
mm = 11
# display the calendar
print(calendar.month(yy, mm))
```

Output:

2. Code:

```
import sys
print(sys.argv)
for i in range(len(sys.argv)):
   if i==0:
       print("The function is",sys.argv[0])
   else:   print("Argument:",sys.argv[i])
```

Output:

3. Code:

```
import numpy as np
# Creating array object
arr = np.array( [[ 1, 2, 3],
                 [ 4, 2, 5]] )

# Printing array dimensions (axes)
print("No. of dimensions: ", arr.ndim)

# Printing shape of array
print("Shape of array: ", arr.shape)

# Printing size (total number of elements) of array
print("Size of array: ", arr.size)
```

Output:

C. Write Python programs for the following:

1. Write a NumPy program to create an 1D array of 10 zeros, 10 ones, 10 fives

2. Write a NumPy program to create a 3x3 matrix with values ranging from 2 to 10.