**Day – 7 Assignment – 3**

1) What is the difference module and package?

A module is a single file (or files) that are imported under one import and used.

e.g.

>>import my\_module

A package is a collection of modules in directories that give a package hierarchy.

>>from my\_package.timing.danger.internets import function\_of\_love

Any Python file is a module, its name being the file's base name without the .py extension. A package is a collection of Python modules: while a module is a single Python file, a package is a directory of Python modules containing an additional \_\_init\_\_.py file, to distinguish a package from a directory that just happens to contain a bunch of Python scripts. Packages can be nested to any depth, provided that the corresponding directories contain their own \_\_init\_\_.py file.

The distinction between module and package seems to hold just at the file system level. When you import a module or a package, the corresponding object created by Python is always of type module. Note, however, when you import a package, only variables/functions/classes in the \_\_init\_\_.py file of that package are directly visible, not sub-packages or modules. As an example, consider the xml package in the Python standard library: its xml directory contains an \_\_init\_\_.py file and four sub-directories; the sub-directory etree contains an \_\_init\_\_.py file and, among others, an ElementTree.py file. See what happens when you try to interactively import package/modules: