- **Packages**
- **Introduction to packages**

Packages are a way of structuring many packages and modules which helps in a well-organized hierarchy of data set, making the directories and modules easy to access. Just like there are different drives and folders in an OS to help us store files, similarly packages help us in storing other sub-packages and modules, so that it can be used by the user when necessary.

- **_init_.py file**

Python that a particular directory is a package, we create a file named __init__.py inside it and then it is considered as a package and we may create other modules and sub-packages within it. This __init__.py file can be left blank or can be coded with the initialization code for the package.

- **Defining packages**

In order for Python to make use of a directory as package, the directory must have a name that is a valid Python identifier and contain a special module named `__init__`. Valid Python names are composed of letters, digits and underscores.

To create a package in Python, we need to follow these three simple steps:

- First, we create a directory and give it a package name, preferably related to its operation.
- Then we put the classes and the required functions in it.
- Finally we create an __init__.py file inside the directory, to let Python know that the directory is a package.

- **Importing from packages**

Example:

Bmw.py

```
class Bmw:
    # First we create a constructor for this class
    # and add members to it, here models
    def __init__(self):
        self.models = ['i8', 'x1', 'x5', 'x6']

    # A normal print function
    def outModels(self):
        print('These are the available models for BMW')
        for model in self.models:
            print('\t%s ' % model)
```

Audi.py

```
class Audi:
```

```
    # First we create a constructor for this class
    # and add members to it, here models
    def __init__(self):
        self.models = ['q7', 'a6', 'a8', 'a3']

    # A normal print function
    def outModels(self):
        print('These are the available models for Audi')
        for model in self.models:
            print('\t%s ' % model)
```

__init__.py

Sample.py

```
from Cars.Bmw import Bmw
from Cars import Audi

# Create an object of Bmw class & call its method
ModBMW = Bmw()
ModBMW.outModels()

# Create an object of Audi class & call its method
ad = Audi()
ad.outModels()
```

- **Defining sub packages**

a directory can contain **sub**-directories and files, a **Python package** can have **sub-packages** and **modules**. A directory must contain a file named __init__.py in order for **Python** to consider it as a **package**. This file can be left empty but we generally place the initialization code for that **package** in this file.

- **Importing from sub packages**

```
# Here, science is a package name and physics is a sub package and display is method name
>>> import science.physics.display
>>> display.display()
This is physics.
```