Package
in Python

What is Package?

- They are simply directories, but with a twist. 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.

Difference between Package and module

- 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.

How to create package

- create a folder with the name 'mypackage'.
- create an empty __init__.py file in the mypackage folder.
- Using a Python editor like IDLE, create modules a.py and b.py with following code:

a.py

```
def fun(name):
        print("Hello " + name)
        return
```

b.py

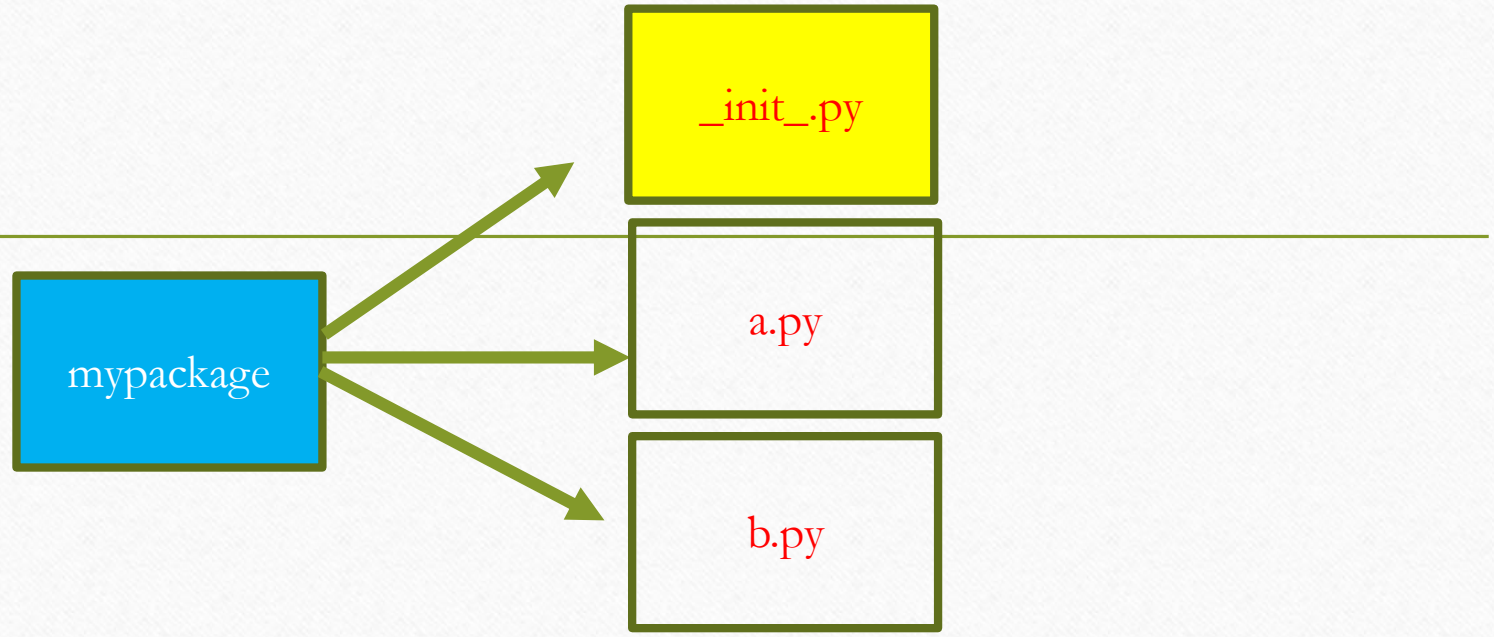
```
def sum(x,y):
        return x+y

def average(x,y):
        return (x+y)/2

def power(x,y):
        return x**y
```

to test package

- >>> from mypackage import b
- >>>b.power(3,2)
- 9
- >>> from mypackage.b import sum
- >>> sum(10,20)
- 30



The package folder contains a special file called __init__.py, which stores the package's content. It serves two purposes:

1.The Python interpreter recognizes a folder as the package if it contains __init__.py file.

2.__init__.py is essential for the folder to be recognized by Python as a package.

The __init__.py file is normally kept empty. However, it can also be used to choose specific functions from modules in the package folder and make them available for import.

init.py

```
from .b import average, power

from .a import fun
```

create test.py in myapp to test mypackage.

```
from mypackage import power, average, fun

fun()

x=power(3,2)

print("power(3,2) : ", x)
```