Python Packages

We organize a large number of files in different folders and sub-folders based on some criteria, so that we can find and manage them easily. In the same way, a package in Python takes the concept of the modular approach to next logical level. As you know, a module can contain multiple objects, such as classes, functions, etc. A package can contain one or more relevant modules. Physically, a package is actually a folder containing one or more module files.

Let's create a package named mypackage, using the following steps:

- Create a new folder named `\MyApp`.
- Inside `MyApp`, create a subfolder with the name 'mypackage'.
- Create an empty `__init__.py` file in the mypackage folder.
- Using a Python-aware editor like IDLE, create modules greet.py and functions.py with the following code:

greet.py

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

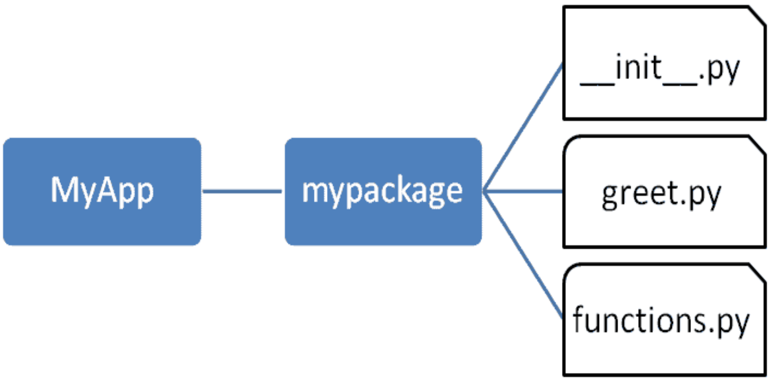
functions.py

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

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

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

That's it. We have created our package called mypackage. The following is a folder structure:



Importing a Module from a Package

Now, to test our package, navigate to the `MyApp` folder and make a test file called test.py and put the next code in it:

```
from mypackage.greet import SayHello
from mypackage.functions import power, sum

SayHello("omar")
print(power(3, 2))
print(sum(15, 30))r
```

Output :

```
Hello  omar
9
45
```

__init__.py

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` exposes specified resources from its modules to be imported.

An empty `__init__.py` file makes all functions from the above modules available when this package is imported. Note that `__init__.py` is essential for the folder to be recognized by Python as a package. You can optionally define functions from individual modules to be made available.

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. Modify `__init__.py` as below:

```
from .functions import average, power
from .greet import SayHello
```

The specified functions can now be imported in the interpreter session or another executable script.

Put next code in test.py

```
from mypackage import power, average, SayHello
SayHello()
x=power(3,2)
print("power(3,2) : ", x)
```

Output:

```
Hello omar
Power(3,2): 9
```

Note that functions `power()` and `SayHello()` are imported from the package and not from their respective modules, as done earlier.