Modules and Packages

What is Python Module?

- A python module can be defined as a python program file which contains a python code including python functions, class, or variables.
- Modules in Python provides us the flexibility to organize the code in a logical way.
- To use the functionality of one module into another, we must have to import the specific module.
- A module is a file consisting of Python code.
- A module can define functions , classes and variables

Advantage of modules

- **Reusability**:-Module can be used in some other python code. Hence, it provides facility of code reusability.
- **Categorization**:-Similar type of modules can be placed in one module.

Importing a module

- There are different ways by which we can import modules:
- **The import statement:-**
- The import statement is used to import all the functionality of one module into another.
- Here, we must notice that we can use the functionality of any python source file by importing that file as the module into another python source file.
- **The from-import statement:-**Instead of importing the whole module into the namespace, python provides the flexibility to import only the specific attributes of a module.
- This can be done by using from? import statement.
- The from...import statement is always better to use if we know the attributes to be imported from the module in advance. It doesn't let our code to be heavier. We can also import all the attributes from a module by using *.

- Create a file.py

- ```
  # displayMsg prints a message to the name being passed.
  def displayMsg(name):
     print("Hi " ,name)
  ```

- Create another python file as mymodule.py

- ```
  import file
  name = input("Enter the name?")
  file.displayMsg(name)
  ```

Calculator.py

```
def sum(a,b):
   return a+b
def mul(a,b):
   return a*b
def div(a,b):
   return a/b

def sub(a,b):
   return a-b
```

- **Test_mdule.py**

- ```
  from calculator import sum
  #it will import only the sum() from calculator.py
  a = int(input("Enter the first number"))
  b = int(input("Enter the second number"))
  print("Sum = ",sum(a,b))
  ```

Python Packages

- The packages in python facilitate the developer with the application development environment by providing a hierarchical directory structure where a package contains sub-packages, modules, and sub-modules.

- The packages are used to categorize the application level code efficiently.
  ```

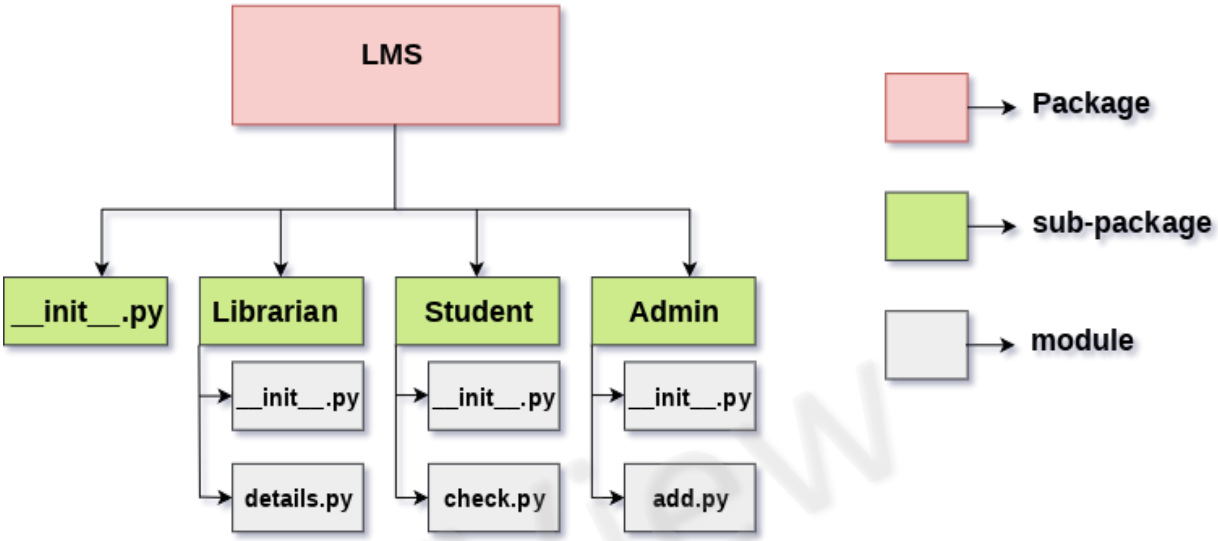
- Let's create a package named Employees in your home directory. Consider the following steps.
- 1. Create a directory with name Employees on path /**home**.
- 2. Create a python source file with name ITEmployees.py on the path /**home**/**Employee**

ITEmployees.py

- **def** getITNames():
- **return** List
- 3. Similarly, create one more python file with name BPOEmployees.py and create a function getBPONames().
- 4. Now, the directory Employees which we have created in the first step contains two python modules.
- To make this directory a package, we need to include one more file here, that is __init__.py which contains the import statements of the modules defined in this directory.
- **__init__.py**
- **from** ITEmployees **import** getITNames
- **from** BPOEmployees **import** getBPONames

  5. Now, the directory **Employees** has become the package containing two python modules. Here we must notice that we must have to create __init__.py inside a directory to convert this directory to a package.
- 6. To use the modules defined inside the package Employees, we must have to import this in our python source file. Let's create a simple python source file at our home directory (/home) which uses the modules defined in this package.
- **Test.py**
- **import** Employees
- **print**(Employees.getNames())
- **Output:**
- ['John', 'David', 'Nick', 'Martin']
- We can have sub-packages inside the packages. We can nest the packages up to any level depending upon the application requirements.

- The following image shows the directory structure of an application Library management system which contains three sub-packages as Admin, Librarian, and Student. The sub-packages contain the python modules.



Task to be done

- Create a directory **Info**
- Place a different modules inside the directory ,We are placing 3modules

msg1.py,msg2.py and msg3.py respectively and place corresponding codes in respective modules .

Let us place msg1()in msg1.py, msg2()in msg2.py, msg3()in msg3.py

Create a file __init__.py which specifies attributes in each module

Import the packages and use the attributes using package.

math module

- Python math module is defined as the most popular mathematical functions, which includes trigonometric functions, representation functions, logarithmic functions, etc
- math.sqrt()
- This function returns the square root of any given number.

- **Example**
- **import** math

  x = 20

  y = 14

  z = 17.8995

  print('sqrt of 20 is ', math.sqrt(x))

  print('sqrt of 14 is ', math.sqrt(y))

  print('sqrt of 17.8995 is ', math.sqrt(z))

  Output:

  sqrt of 20 is 4.47213595499958 sqrt of 14 is 3.7416573867739413 sqrt of 17.8995 is 4.230780069916185

Python Random module

- The Python random module functions depend on a pseudo-random number generator function random(), which generates the float number between 0.0 and 1.0.
- **import** random
  b=random.random()
  print(b)
  a=random.randint(1,90)
  print(a)