**--------Module and Packages--------**

**🡪 Module:**

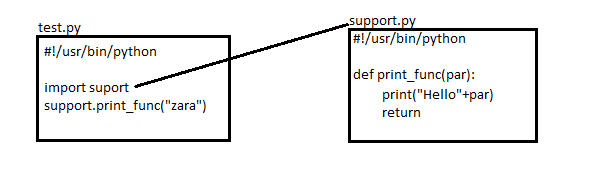
- A module allows you to logically orgranise your python code.

- Grouping related code into a module makes the code eaiser to understand and use.

Note:

A module is a python object with arbitarrily name attributes that you ca bind and referance.

Example:

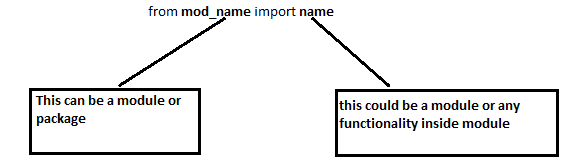


O/P-

Hello zara

The **from … import** statement :

Syntax:



Example:

**from fib import fibonaci**

This statement doesn’t import the entire module fib into the current namespace, it just introduces the

Item **fibonaci** from the module **fib** into **global symbol table** of the importing module.

The **from … import \*** statement :

Syntax:

from **mod\_name** import **\***

**-**It is also possible all names from a module into a current namespace by using above statement.

- In this way we can import all items from a module into current namespace.

**Locating Module:**

When we import a module , the python interpreter searches for the module in the following sequences:-

1. The current directory
2. If the module is not found , python then searches each directory in the shell variable **PYTHONPATH**
3. if all above fails , python checks the default path.

On unix , this default path is normally /usr/local/bin/python.

The module search path is stored in the system module sys as the sys.path variable.

The **sys.path** variable contain the current directory, PYTHONPATH and installation-dependent default.

Example:

Suppose your modules are present @ **D:\loops\module** (support.py)

& your script file which you want to execute is present @ **c:\user\Desktop\Python\_practice** (loop1.py)

-Now if you execute **loop1.py** by **import support** it will give you error.

-But if you add the path **D:\loops\module** to **PYTHONPATH** environment variable then, it will be work now if you **import support** in loop1.py

Open python interactive mode:

>>> import sys

>>>print(sys.path)

[……;….;D:\loops\module;…]

Note:

-If we are storing our modules in a different folder we can also add to the **sys.path** otherwise that module will not work.

-We can do that by adding that path to **PYTHONPATH** environment variable.

-If we are running everything from single folder then do as below..

from loops.module import support

**🡪 Package:**

* A package is a hirarchical file directory structure that defines a single python application environment that consists of modules & subpackages & subpackages & so on..
* Consider a file **pots.py** available in **Phone** directory.
* Pots.py

|  |
| --- |
| #!/usr/bin/python  Def Pots():  print(“I am a pots phone”) |

Simillar way, we have another two files having different functions with the same name as above..

EX: Phone/Isdn.py file having function Isdn()

Phone/G3.py file having function G3()

Now creat one more file “\_\_init\_\_.py” in **Phone** directory

Phone/\_\_init\_\_.py

* To make all of your functions available when you have all of these classes available when you

Import the Phone package

|  |
| --- |
| #!/usr/bin/python  #now import your phone package  Import Phone  Phone.Pots()  Phone.Isdn()  Phone.G3() |

O/P-

I am pots phone

I am 3G phone

I am ISDN Phone