Python Module

A module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code.

Modules are used to categorize Python code into smaller parts. A module is simply a Python file, where classes, functions and variables are defined. Grouping similar code into a single file makes it easy to access. Have a look at below example.

Module Advantage

- **Reusability:** Modules let programmers save source code in files permanently & hence the codes in module files are persistent. Module codes can rerun as many times as required.
- **Categorization:** Similar type of attributes can be placed in one module.

Import Statement

Programmers can use any Python source code as a module by implementing an 'import' statement in another Python program or Python file.

Syntax

```
Import file1,file2..filen
```

When the interpreter encounters an import statement, it imports the module if the module is present in the search path. A search path is a list of directories that the interpreter searches before importing a module.

Example

add.py file

```
def add1(a,b):
   print a+b
```

Save this code in the file by the name add.py. Now import this file in another file or code.

Example	Output

import add
add.add1(10,20)
add.add1(20,30)
	30
	50

Importing Multiple Modules Example

In this example we create three python modules file namely add.py, sub.py, div.py and python file main.py for importing all three files.

add.py file

```
def add1(a,b):
   print a+b
```

sub.py file

```
def sub1(a,b):
    print a-b
```

div.py file

```
def div1(a,b):
    print a/b
```

Now import all three modules file in following example.

main.py file

Example	**Output**

import add,sub,div
add.add1(20,10)
sub.sub1(20,10)
div.div1(20,10)
	30
	10
	2

from.. import statement

from..import statement is used to import particular attribute from a module. In case you do not want whole of the module to be imported then you can use from import statement.

In this example we create one python modules file namely math.py and python file main.py for import attributes from another python file.

math.py file

```
def add1(a,b):
    print a+b

def sub1(a,b):
    print a-b

def div1(a,b):
    print a/b
```

main.py file

Example	Output

from math import add1,sub1,div1
add1(20,10)
sub1(20,10)
div1(20,10)
	30
	10
	2

import whole module

we can import whole of the module using "from main import *"

Example	Output

from math import *
add1(20,10)
sub1(20,10)
div1(20,10)
	30
	10
	2

Packages in Python

A package is a hierarchical file directory structure that defines a single Python application environment that consists of modules and subpackages.

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.

Example

1. First Create directory with name as cal.
2. Then create two modules add.py and sub.py in cal dirct.

add.py file

```
def add(a,b):
    print a+b
```

sub.py file

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

3. Create __init__.py file in cal dirct.

```
from add import add
from sub import sub
```

4. Now import this package and run it

Example	**Output**

import cal
cal.add(20,10)
cal.sub(20,10)
	30
	10