

14. PYTHON – MODULE

Table of Contents

1. Modules	2
2. import module.....	3
3. Renaming or aliasing a module.....	4
4. from and import keywords	5
5. import * (star symbol).....	6
6. Member aliasing.....	6

14. PYTHON – MODULE

1. Modules

- ✓ In python a module means, a saved python file.
- ✓ This file can contain a group of classes, methods, functions and variables.
- ✓ Every Python file mean **.py** or **.python** extension file is called as a module.

```
Program      module program
Name         additionmultiplication.py

x = 10

def addition(a, b):
    print("Sum of two values: ", (a+b))

def multiplication(a, b):
    print("Multiplication of two values: ", (a*b))
```

- ✓ Now **additionmultiplication.py** file is a module.
- ✓ **additionmultiplication.py** module contains one variable and two functions.

2. import module

- ✓ **import** is a keyword in python.
- ✓ By using import keyword we can import (get) modules in our program.
- ✓ Once we imported the module then we can use members (variables, functions & etc) of module.

Program Name importing **additionmultiplication** module and calling members
demo1.py

```
import additionmultiplication

print(additionmultiplication.x)
additionmultiplication.addition(1, 2)
additionmultiplication.multiplication(2, 3)
```

output

```
10
Sum of two values: 3
Multiplication of two values: 6
```

Make a note:

- ✓ Whenever we are using a module in our program, for that module compiled file will be generated and stored in the hard disk permanently.

3. Renaming or aliasing a module.

- ✓ **as** is a keyword in python
- ✓ By using **as** keyword we can rename/alias existing module.

Syntax

```
import additionmultiplication as admul
```

- ✓ Here **additionmultiplication** is module name and alias name is **admul**
- ✓ We can access members by using alias name **admul**

Program Name

importing module
demo2.py

```
import additionmultiplication as admul
```

```
print(admul.x)  
admul.addition(1,2)  
admul.multiplication(3, 4)
```

output

```
10  
Sum of two values: 3  
Multiplication of two values: 12
```

4. from and import keywords

- ✓ **from** is keyword in python
- ✓ We can import some specific members of module by using **from** keyword.
- ✓ The main advantage of **from** keyword is we can access members directly without using module name.

Program Name	from and import keywords demo3.py
	<pre>from additionmultiplication import x, addition print(x) addition(10,20)</pre>
output	10 Sum of two values: 30

Program Name	NameError: name 'multiplication' is not defined demo4.py
	<pre>from additionmultiplication import x, addition print(x) multiplication(10,20)</pre>
Error	10 NameError: name 'multiplication' is not defined

5. import * (star symbol)

- ✓ We can use * symbol to import all members of a module.
- ✓ We can import all members of a module as by using import * (symbol)

Program importing by using *
Name demo5.py

```
from additionmultiplication import *
```

```
print(x)  
addition(10, 20)  
multiplication(10, 20)
```

output

```
10  
Sum of two values: 30  
Multiplication of two values: 200
```

6. Member aliasing

- ✓ We can give alias name to the members of a module

Program member aliasing
Name demo6.py

```
from additionmultiplication import x as y, addition as add
```

```
print(y)  
add(10, 20)
```

output

```
10  
Sum of two values: 30
```