

## Modules

A group of functions, variables and classes saved to a file, which is nothing but module.

Every Python file (.py) acts as a module.

Eg: krishna.py

```
1) x=888
2)
3) def add(a,b):
4)     print("The Sum:",a+b)
5)
6) def product(a,b):
7)     print("The Product:",a*b)
```

krishna module contains one variable and 2 functions.

If we want to use members of module in our program then we should import that module.

import modulename

We can access members by using module name.

modulename.variable

modulename.function()

test.py:

```
1) import krishna
2) print(krishna.x)
3) krishna.add(10,20)
4) krishna.product(10,20)
5)
6) Output
7) 888
8) The Sum: 30
9) The Product: 200
```

### 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.

Renaming a module at the time of import (module aliasing):

Eg:

import krishna as m

here krishna is original module name and m is alias name.

We can access members by using alias name m

test.py:

```
1) import krishna as m
2) print(m.x)
3) m.add(10,20)
4) m.product(10,20)
from ... import:
```

We can import particular members of module by using from ... import .

The main advantage of this is we can access members directly without using module name.

Eg:

```
from krishna import x,add
print(x)
```

```
add(10,20)
product(10,20)==> NameError: name 'product' is not defined
We can import all members of a module as follows
from krishna import *
test.py:
1) from krishna import *
2) print(x)
3) add(10,20)
4) product(10,20)
```

Various possibilities of import:

```
import modulename
import module1,module2,module3
import module1 as m
import module1 as m1,module2 as m2,module3
from module import member
from module import member1,member2,memebr3
from module import memeber1 as x
from module import *
```