

Modules

Modules refer to a file containing Python statements and definitions.

A file containing Python code, for e.g.: Marvellous.py, is called a module and its module name would be Marvellous.

We use modules to break down large programs into small manageable and organized files.

Furthermore, modules provide reusability of code.

We can define our most used functions in a module and import it, instead of copying their definitions into different programs.

Consider below module named as Marvellous.py

```
def fun():  
    print("Inside fun")
```

```
def gun():  
    print("Inside gun")
```

Consider below program which uses above module as Marvellous

```
print("---- Marvellous Infosystems by Piyush Khairnar----")
```

```
print("Demonstration of Modules")
```

```
import Marvellous
```

```
Marvellous.fun()  
Marvellous.gun()
```

We can import the definitions inside a module to another module or the interactive interpreter in Python.

We use the import keyword to do this.

To import our previously defined module Marvellous we type the following in the Python prompt.

import Marvellous

This does not enter the names of the functions defined in example directly in the current symbol table.

It only enters the module name example there.

Using the module name we can access the function using dot (.) operation.

For example:

```
Marvellous.fun()
```

```
Marvellous.gun()
```

We can also import with renaming as

Import Marvellous as MyModule

According to above syntax we can use MyModule as a name of our module which is alias for Marvellous module.

If want to use all functions from specific module then we can use below syntax

from Marvellous import *

using this syntax we can directly call function from the module without specifying the name of module.

