

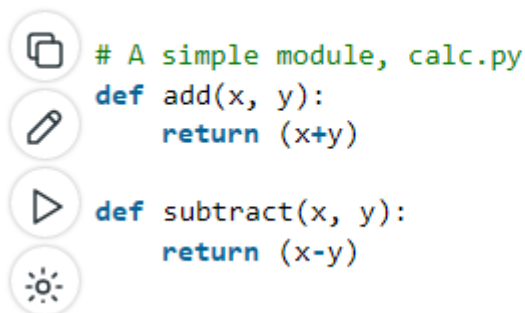
Python Module

A Python module is a file containing Python definitions and statements. A module can define functions, classes, and variables. A module can also include runnable code. Grouping related code into a module makes the code easier to understand and use. It also makes the code logically organized.

Create a simple Python module

Let's create a simple `calc.py` in which we define two functions, one **add** and another **subtract**.

Python3



```
# A simple module, calc.py
def add(x, y):
    return (x+y)

def subtract(x, y):
    return (x-y)
```

Import Module in Python

We can import the functions, and classes defined in a module to another module using the [import statement](#) in some other Python source file.

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 for importing a module. For example, to import the module `calc.py`, we need to put the following command at the top of the script.

Syntax of Python Import

`import module`

Importing modules in Python

Now, we are importing the **calc** that we created earlier to perform add operation.

- Python3

```
# importing module calc.py
```

```
import calc
```

```
print(calc.add(10, 2))
```

Output:

12

The from-import Statement in Python

Python's *from* statement lets you import specific attributes from a module without importing the module as a whole.

Importing specific attributes from the module

Here, we are importing specific `sqrt` and `factorial` attributes from the `math` module.

- Python3

```
# importing sqrt() and factorial from the
```

```
# module math
```

```
from math import sqrt, factorial
```

```
# if we simply do "import math", then
```

```
# math.sqrt(16) and math.factorial()
```

```
# are required.
```

```
print(sqrt(16))
```

```
print(factorial(6))
```

Output:

4.0

720

Import all Names

The * symbol used with the from import statement is used to import all the names from a module to a current namespace.

Syntax:

```
from module_name import *
```

From import * Statement

The use of * has its advantages and disadvantages. If you know exactly what you will be needing from the module, it is not recommended to use *, else do so.

- Python3

```
# importing sqrt() and factorial from the
```

```
# module math
```

```
from math import *
```

```
# if we simply do "import math", then
```

```
# math.sqrt(16) and math.factorial()
```

```
# are required.
```

```
print(sqrt(16))
```

```
print(factorial(6))
```

Output

```
4.0
```

```
720
```

Locating Python Modules

Whenever a module is imported in Python the interpreter looks for several locations. First, it will check for the built-in module, if not found then it looks for a list of directories defined in the [sys.path](#). Python interpreter searches for the module in the following manner –

- First, it searches for the module in the current directory.
- If the module isn't found in the current directory, Python then searches each directory in the shell variable [PYTHONPATH](#). The PYTHONPATH is an environment variable, consisting of a list of directories.
- If that also fails python checks the installation-dependent list of directories configured at the time Python is installed.

Directories List for Modules

Here, sys.path is a built-in variable within the sys module. It contains a list of directories that the interpreter will search for the required module.

- Python3

```
# importing sys module
```

```
import sys
```

```
# importing sys.path
```

```
print(sys.path)
```

Output:

```
['/home/nikhil/Desktop/gfg', '/usr/lib/python38.zip', '/usr/lib/python3.8',  
'/usr/lib/python3.8/lib-dynload', '', '/home/nikhil/.local/lib/python3.8/site-  
packages', '/usr/local/lib/python3.8/dist-packages', '/usr/lib/python3/dist-  
packages', '/usr/local/lib/python3.8/dist-packages/IPython/extensions',  
'/home/nikhil/.ipython']
```

Renaming the Python module

We can rename the module while importing it using the keyword.

Syntax: *Import Module_name as Alias_name*

- Python3

```
# importing sqrt() and factorial from the
```

```
# module math
```

```
import math as mt
```

```
# if we simply do "import math", then
```

```
# math.sqrt(16) and math.factorial()
```

```
# are required.
```

```
print(mt.sqrt(16))
```

```
print(mt.factorial(6))
```

Output

```
4.0
```

```
720
```

Python built-in modules

There are several built-in modules in Python, which you can import whenever you like.

- Python3

```
# importing built-in module math
```

```
import math
```

```
# using square root(sqrt) function contained
```

```
# in math module
```

```
print(math.sqrt(25))
```

```
# using pi function contained in math module
```

```
print(math.pi)
```

```
# 2 radians = 114.59 degrees
```

```
print(math.degrees(2))
```

```
# 60 degrees = 1.04 radians
```

```
print(math.radians(60))
```

```
# Sine of 2 radians
```

```
print(math.sin(2))
```

```
# Cosine of 0.5 radians
```

```
print(math.cos(0.5))
```

```
# Tangent of 0.23 radians
```

```
print(math.tan(0.23))
```

```
# 1 * 2 * 3 * 4 = 24
```

```
print(math.factorial(4))
```

```
# importing built in module random
```

```
import random
```

```
# printing random integer between 0 and 5
```

```
print(random.randint(0, 5))
```

```
# print random floating point number between 0 and 1
```

```
print(random.random())
```

```
# random number between 0 and 100
```

```
print(random.random() * 100)
```

```
List = [1, 4, True, 800, "python", 27, "hello"]
```

```
# using choice function in random module for choosing
```

```
# a random element from a set such as a list
```

```
print(random.choice(List))
```

```
# importing built in module datetime
```

```
import datetime
```

```
from datetime import date

import time

# Returns the number of seconds since the

# Unix Epoch, January 1st 1970

print(time.time())

# Converts a number of seconds to a date object

print(date.fromtimestamp(454554))
```

Output:

```
5.0
3.14159265359
114.591559026
1.0471975512
0.909297426826
0.87758256189
0.234143362351
24
3
0.401533172951
88.4917616788
True
1461425771.87
```