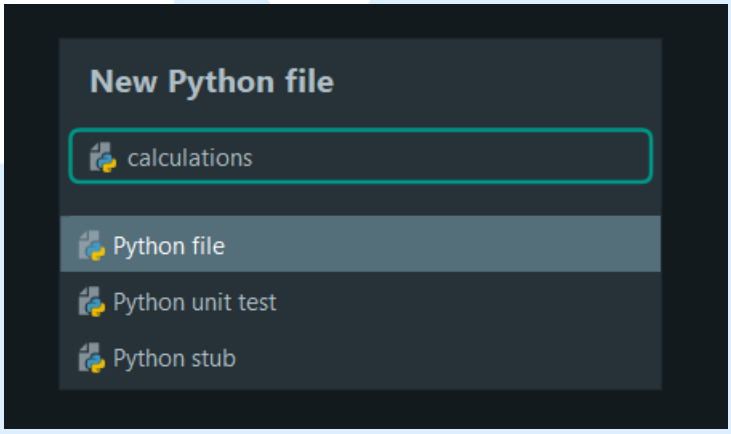
Modules in Python

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.

Let us create our own module.

- Create a new Python file named ***calculations.py***.



- Write 4 functions namely ***add***, ***sub, multiply, division***.
- Write the logic as shown below.

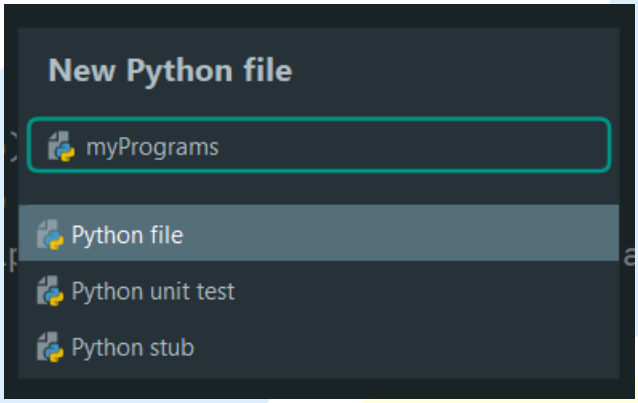
```
def add(a, b):
    total = a + b
    print(f"Addition of {a} and {b} is {total}")


def sub(a, b):
    total = a - b
    print(f"Subtraction of {a} and {b} is {total}")


def multiply(a, b):
    total = a * b
    print(f"Multiplication of {a} and {b} is {total}")


def division(a, b):
    total = a / b
    print(f"Division of {a} and {b} is {total}")
```

Importing a module

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 **calculations.py**, we need to put the following command at the **top** of our python file.

Create another file name as per your choice. For this example, we will create, ***myProgram.py.***



Let us start by importing all the functions we defined in ***calculations.py***. To do so, we write **import <filename>.**

```
import calculations
```

Now we will be using all the functions defined in ***calculations.py***.

```
import calculations

calculations.add(5, 10)
calculations.sub(5, 10)
calculations.multiply(5, 10)
calculations.division(5, 10)
```

Output

```
Addition of 5 and 10 is 15
Subtraction of 5 and 10 is -5
Multiplication of 5 and 10 is 50
Division of 5 and 10 is 0.5
```

By writing **import,** all the functions defined in that file will be imported.

Importing only specific functions using FROM-IMPORT

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

```
from calculations import add, sub

add(5, 10)
sub(5, 10)

# Below code will give us error
# Because we did not import multiply and division function
# multiply(5, 10)
# division(5, 10)
```

Output

```
Addition of 5 and 10 is 15
Subtraction of 5 and 10 is -5
```

Importing everything using *

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

```
from calculations import *

add(5, 10)
sub(5, 10)

# Below code will now work
multiply(5, 10)
division(5, 10)
```

Output

```
Addition of 5 and 10 is 15
Subtraction of 5 and 10 is -5
Multiplication of 5 and 10 is 50
Division of 5 and 10 is 0.5
```

Some In-Built modules in Python

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

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

# using square root(sqrt) function contained
# in math module
print(math.sqrt(36))

# 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
```

Output

```
6.0
3.141592653589793
114.59155902616465
1.0471975511965976
0.9092974268256817
0.8775825618903728
0.23414336235146527
24
1
0.2743492462273365
52.63954872295263
1
1667234694.9992154
1970-01-06
```

CODE & DEBUG
YOUR FUTURE IS OUR PRIORITY

Contact: +91-9712928220 | Mail: info@codeanddebug.in
Website: codeanddebug.in