Function & Modules

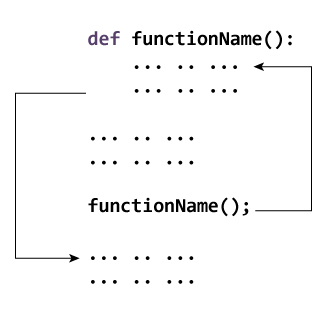
Functions that we define ourselves to do certain specific task are referred as user-defined functions.

If we use functions written by others in the form of library, it can be termed as library functions.

All the other functions that we write on our own fall under user-defined functions. So, our user-defined function could be a library function to someone else.

## Advantages of user-defined functions

1. User-defined functions help to decompose a large program into small segments which makes program easy to understand, maintain and debug.
2. If repeated code occurs in a program. Function can be used to include those codes and execute when needed by calling that function.
3. Programmars working on large project can divide the workload by making different functions.



def add (x,y):

sum = x + y

print (sum)

num1 = 5

num2 = 6

add(num1, num2)

## Loading the module in our python code

We need to load the module in our python code to use its functionality. Python provides two types of statements as defined below.

1. The import statement
2. The from-import statement

## The import statement

The import statement is used to import all the functionality of one module into another. Here, we must notice that we can use the functionality of any python source file by importing that file as the module into another python source file.

We can import multiple modules with a single import statement, but a module is loaded once regardless of the number of times, it has been imported into our file.

The syntax to use the import statement is given below.

**calculation.py:**

1. **def** summation(a,b):
2. **return** a+b
3. **def** multiplication(a,b):
4. **return** a\*b;
5. **def** divide(a,b):
6. **return** a/b;

**Main.py:**

1. **from** calculation **import** summation
2. #it will import only the summation() from calculation.py
3. a = int(input("Enter the first number"))
4. b = int(input("Enter the second number"))
5. **print**("Sum = ",summation(a,b)) #we do not need to specify the module name while acces

## Renaming a module

1. **import** calculation as cal;
2. a = int(input("Enter a?"));
3. b = int(input("Enter b?"));
4. **print**("Sum = ",cal.summation(a,b))