

```
1 A function is a block of code that performs a specific task.
2
3 Suppose, you need to create a program to perform addition and subtraction of
  numbers. You can create two functions to solve this problem:
4
5 create an addition function
6 create a subtraction function
7
8 Dividing a complex problem into smaller chunks makes our program easy to
  understand and reuse.
```

Types of function

There are two types of function in Python programming:

1. Standard library functions - These are built-in functions in Python that are available to use.
2. User-defined functions - We can create our own functions based on our requirements.

Python Function Declaration

In [1]:

```
1 #Syntax
2
3 def function_name(arguments):
4     # function body
5
6     return
```

```
1 def - keyword used to declare a function
2 function_name - any name given to the function
3 arguments - any value passed to function
4 return (optional) - returns value from a function
```

In []:

```
1
```

Calling a Function in Python

In [9]:

```
1
2 def demol():
3     print("Inside Demo")
4     return
5
6 demol()
7
8 print("We are outside demo..")
```

```
Inside Demo
We are outside demo..
```

Python Function Arguments

In [19]:

```
1 # function with two arguments
2 def add_numbers1(num1=10, num2=30):
3     return num1+num2
4
5
6 s = add_numbers1()
7 print("Sum= ", s)
8
```

```
Sum= 9
```

In []:

```
1
```

In []:

```
1
```

Function Argument with Default Values

In Python, we can provide default values to function arguments.

We use the = operator to provide default values. For example,

In [19]:

```
1 def add_numbers( a = 7, b = 8):  
2     sum = a + b  
3     print('Sum:', sum)  
4  
5  
6 # function call with two arguments  
7 add_numbers(2, 3)  
8  
9 # function call with one argument  
10 add_numbers(a = 2)  
11  
12 # function call with no arguments  
13 add_numbers()
```

Sum: 5

Sum: 10

Sum: 15

Python Keyword Argument

In keyword arguments, arguments are assigned based on the name of arguments. For example,

In [20]:

```
1 def display_info(first_name, last_name):  
2     print('First Name:', first_name)  
3     print('Last Name:', last_name)  
4  
5 display_info(last_name = 'abc', first_name = 'pqr')
```

First Name: pqr

Last Name: abc

The return Statement in Python

A Python function may or may not return a value. If we want our function to return some value to a function call, we use the return statement. For example,

In [12]:

```
1 def add_numbers():  
2     ...  
3     return sum
```

```
1 Note: The return statement also denotes that the function has ended. Any code  
   after return is not executed.
```

In [14]:

```
1  # function definition
2  def sum(num1,num2):
3      sum = num1 + num2
4      return sum
5
6  # function call
7  sum = sum(3,3)
8
9  print( 'Sum:',sum)
```

Sum: 6