

Python Programming



**RGM College of Engineering & Technology
(Autonomous)**

Department of Computer Science & Engineering

Academic Year : 2020-2021

FUNCTIONS - 1

UNIT – VI

Functions - Defining Functions, Calling Functions, Types of Arguments
- Keyword Arguments, Default Arguments, Variable-length arguments,
Anonymous Functions, Fruitful functions (Function Returning Values),
Scope of the Variables in a Function - Global and Local Variables.
Recursive functions, Illustrative examples on all the above topics.

Topics Covered:

1. Introduction
2. Types of Functions
3. Types of Parameters
4. Types of Variables
5. Recursive Functions
6. Anonymous Functions
7. Function Aliasing
8. Nested Functions



Guido Van Rossum

Dept. of CSE, RGM CET(Autonomous), Nandyal

Learning Mantra

**If you really strong in the basics, then
remaining things will become so easy.**

Agenda:

1. Introduction

2. Types of Functions

1.INTRODUCTION

Introduction:

- ❑ If a group of statements are repeatedly required then it is not recommended to write these statements every time separately. We have to define these statements as a single unit and we can call that unit any number of times based on our requirement without rewriting. This unit is nothing but **function**.
- ❑ The main advantage of functions is **Code Reusability**.

Note:

- ❑ In other programming languages, functions are known as methods or procedures or subroutines etc.

Eg:

I.

```
a = 20
b = 10
print('The Sum : ', a + b)
print('The Difference : ', a - b)
print('The Product : ', a * b)
```

```
The Sum : 30
The Difference : 10
The Product : 200
```

II.

```
a = 20
b = 10
print('The Sum : ', a + b)
print('The Difference : ', a - b)
print('The Product : ', a * b)
a = 200
b = 100
print('The Sum : ', a + b)
print('The Difference : ', a - b)
print('The Product : ', a * b)
```

```
The Sum : 30
The Difference : 10
The Product : 200
The Sum : 300
The Difference : 100
The Product : 20000
```

III.

a = 20

b = 10

print('The Sum : ', a + b)

print('The Difference : ', a - b)

print('The Product : ', a * b)

a = 200

b = 100

print('The Sum : ', a + b)

print('The Difference : ', a - b)

print('The Product : ', a * b)

a = 2000

b = 1000

print('The Sum : ', a + b)

print('The Difference : ', a - b)

print('The Product : ', a * b)

The Sum : 30

The Difference : 10

The Product : 200

The Sum : 300

The Difference : 100

The Product : 20000

The Sum : 3000

The Difference : 1000

The Product : 2000000

Here, we written 15 lines of code. **What is the problem in this code?**

- ❑ Have you observed that the same code (3 lines) is repeating thrice in the code.

Generally we never recommended to write the group of statements repeatedly in the program.

Problems of writing the same code repeatedly in the program are as follows:

1. Length of the program increases.
2. Readability of the program decreases.
3. No Code Reusability.

How can we resolve this problem?

- ❑ We have to define these statements as a single unit and we can call that unit any number of times based on our requirement without rewriting. This unit is nothing but function.

How can you solve this problem by defining function for the above example?

I.

```
def calculate( a, b):
```

```
    print('The Sum : ', a + b)
```

```
    print('The Difference : ', a - b)
```

```
    print('The Product : ', a * b)
```

```
a = 20
```

```
b = 10
```

```
calculate( a, b)
```

```
The Sum : 30
```

```
The Difference : 10
```

```
The Product : 200
```

II.

```
def calculate( a, b):
```

```
    print('The Sum : ', a + b)
```

```
    print('The Difference : ', a - b)
```

```
    print('The Product : ', a * b)
```

```
a = 20
```

```
b = 10
```

```
calculate( a, b)
```

```
a = 200
```

```
b = 100
```

```
calculate( a, b)
```

```
The Sum : 30
```

```
The Difference : 10
```

```
The Product : 200
```

```
The Sum : 300
```

```
The Difference : 100
```

```
The Product : 20000
```

III.

```
def calculate( a, b):  
    print('The Sum : ', a + b)  
    print('The Difference : ', a - b) # Function 'calculate()' executes 3 times  
    print('The Product : ', a * b)  
  
a = 20  
b = 10  
calculate( a, b)    # Function call  
  
a = 200  
b = 100  
calculate( a,b)  
  
a = 2000  
b = 1000  
calculate( a,b)
```

The Sum : 30
The Difference : 10
The Product : 200
The Sum : 300
The Difference : 100
The Product : 20000
The Sum : 3000
The Difference : 1000
The Product : 2000000

IV.

```
def calculate( a, b):  
    print('The Sum : ', a + b)  
    print('The Difference : ', a - b)    # Function 'calculate()' executes 3 times  
    print('The Product : ', a * b)  
calculate(20,10)                        # Function call  
calculate(200,100)  
calculate(2000,1000)                   # Concise code resulted because of code reusability
```

```
The Sum : 30  
The Difference : 10  
The Product : 200  
The Sum : 300  
The Difference : 100  
The Product : 20000  
The Sum : 3000  
The Difference : 1000  
The Product : 2000000
```

Note:

We are writing the function once and calling that function 'n' times.

2.TYPES OF FUNCTIONS

Python supports 2 types of functions:

1. Built in Functions

2. User Defined Functions

1. Built in Functions

- ❑ The functions which are coming along with Python software automatically are called built in functions (or) pre-defined functions.

Eg:

`id()`

`type()`

`input()`

`eval()` etc..

2. User Defined Functions:

- ❑ The functions which are developed by programmer explicitly according to his/her business requirements are called user defined functions.

Syntax to create user defined functions:

```
def function_name(parameters) :
```

```
    Stmt 1
```

```
    Stmt 2
```

```
    ---
```

```
    Stmt n
```

```
return value
```

Note:

While creating functions we can use 2 keywords:

1. def (mandatory)

2. return (optional)

Eg 1: Write a function to print Hello message.

Program:

```
def wish():  
    print("Hello Good Morning")  
  
wish()  
wish()  
wish()
```

Output:

Hello Good Morning

Hello Good Morning

Hello Good Morning

Any question?



If you try to practice programs yourself, then you will learn many things automatically

Spend few minutes and then enjoy the study

Thank You