# Functions

## Functions

***Function is a sub program which consists of set of instructions used to perform a specific task. A large program is divided into basic building blocks called function.***

- A function is a block of code which only runs when it is called.
- You can pass data, known as parameters, into a function.
- A function can return data as a result.

- Functions can be classified into two categories:

i) user defined function

ii) Built in function

## i) Built in functions

Built in functions are the functions that are already created and stored in python.

These built in functions are always available for usage and accessed by a programmer. It cannot be modified.

Example:

max(10,30,9,20)

min(10,30,9,20)

## ii)User Defined Functions:

- User defined functions are the functions that programmers create for their requirement and use.

Syntax:

```
def fun_name(Parameter1,Parameter2…Parameter n):
      statement1
      statement2…
      statement n
      return[expression]
```

## Creating a Function

In Python a function is defined using the def keyword:

### Example

```
def my_function():
  print("Hello from a function")
```

## Calling a Function

To call a function, use the function name followed by parenthesis:

```
def my_function():
  print("Hello from a function")

my_function()
```

## Functions

- A *function* is like a mini-program within a program.

Example:

```
def hello():
    print('Howdy!’)
    print('Howdy!!!’)
    print('Hello there.')
hello()
hello()
hello()
```

## Function with Arguments

Arguments

- Information can be passed into functions as arguments

Example :

```
def hello(name):
    print('Hello ' + name)

hello('Alice’)
hello('Bob')
```

## Function Argument with Default Values

```
def add_numbers( a = 7,  b = 8):
    sum = a + b
    print('Sum:', sum)

add_numbers(2, 3)

add_numbers(a = 2)

add_numbers()
```

## Return Values and return Statements

- We return a value from the function using the return statement.

```
# function definition
def find_square(num):
    result = num * num
    return result



# function call
square = find_square(3)
print('Square:', square)
```

## Number of Arguments

```
def my_function(fname, lname):
        print(fname + " " + lname)

my_function(“Alice", "Refsnes")
```

## Keyword Arguments

- You can also send arguments with the *key* = *value* syntax.
- This way the order of the arguments does not matter.

```
def my_function(child3, child2, child1):
  print("The youngest child is " + child3)

my_function(child1 = "Emil", child2 = "Tobias", child3 = "Linus")
```

## Default Parameter Value

- The following example shows how to use a default parameter value.
- If we call the function without argument, it uses the default value:

Example

```
def my_function(country = "Norway"):
  print("I am from " + country)

my_function("Sweden")
my_function("India")
my_function()
my_function("Brazil")
```

## Passing a List as an Argument

- You can send any data types of argument to a function (string, number, list, dictionary etc.), and it will be treated as the same data type inside the function.
- E.g. if you send a List as an argument, it will still be a List when it reaches the function:

Example:

```
def my_function(food):
  for x in food:
    print(x)

fruits = ["apple", "banana", "cherry"]

my_function(fruits)
```

## Practise Programs:

- Develop a Python program where a student scores marks in two subjects:
  i) 85 marks in Math
  ii) 78 marks in Science
  Find which subject the student scored higher using comparison operators.

- A shop offers two discounts on a product worth $500:
  i) 10% off
  ii) $40 off

Write a Python program to find which discount gives a better deal using comparison operators