Functions

- Functions are common to all programming languages, and it can be defined as a block of re-usable code to perform specific tasks.
- But defining functions in Python means knowing both types first: built-in and user-defined.
- Built-in functions are usually a part of Python packages and libraries, whereas user-defined functions are written by the developers to meet certain requirements.
- In Python, all functions are treated as objects, so it is more flexible compared to other high-level languages.

Importance of user-defined functions in Python

- In general, developers can write user-defined functions or it can be borrowed as a third-party library.
- This also means your own user-defined functions can also be a third-party library for other users.
- User-defined functions have certain advantages depending when and how they are used.
- User-defined functions are reusable code blocks; they only need to be written once, then they can be used multiple times. They can even be used in other applications, too.
- These functions are very useful, from writing common utilities to specific business logic. These functions can also be modified per requirement.
- The code is usually well organized, easy to maintain, and developer-friendly. Which means it can support the modular design approach.
- As user-defined functions can be written independently, the tasks of a project can be distributed for rapid application development.
- A well-defined and thoughtfully written user-defined function can ease the application development process.

Defining a Function :

Function in Python is defined by the **"def "** statement followed by the function name and parentheses (())

Consider below program

```
def fun():
        print("Inside fun")
```

Calling a function :

To call a function we have to specify name of the function and pass the parameters in parenthesis.

```
fun()
```

Parameters for the functions

As python is dynamically typed language we can specify just name of variable to accept the input from caller.
If function wants to return anything then we can use return keyword.

```
def Add(no1, no2):
        ans = no1 + no2
        return and

Ret = Add(10,11)
print("Addition is ", Ret)
```