Functions - 1

Built-In Functions

Introduction

- In python programming language when we wrote large programs sequentially using all the constructs of a language.
- As a result, the program may become too complex and large, this leads to difficulties in debugging, testing and even understanding .
- The best way to develop a large program is to construct it from smaller pieces or modules .
- A module can be a single function or group of related functions carrying out a specific task, then we can divide large program into a series of individual modules that related to each other.
- Finally it is easer to break down a difficult task into a series of smaller tasks and then solve those subtasks individually and later combined into a single unit.
- These subtasks are made user-defined functions and in python programing language such modules are called as functions.
- The functions are much easier to understand ,debug and test.

Why Functions

- A function is a piece of code in a program. The function performs a specific task. The advantages of using functions are:
- Reducing duplication of code
- Decomposing complex problems into simpler pieces
- Improving clarity of the code
- Reuse of code
- Information hiding

Functions in Python are first-class citizens. It means that functions have equal status with other objects in Python. Functions can be assigned to variables, stored in collections or passed as arguments. This brings additional flexibility to the language.

Function Types

- There are two basic types of functions.
  - Built-in Functions
    - Always available for usage
      - Ex : dir(), len(), type(), input() … etc
    - Those contained in external modules
  - User defined Functions
    - Programmer has to defined

Built-in functions (Always available)

- Functions that come built into the Python language itself are called built-in functions and are readily available to us.
  - Example functions *print(), input(), int(), float()*, *range(), and type()*

Some of the Built-in Functions in Python

Built-in Function	Description
abs()	Return the absolute value of a number.
len()	Return the length (the number of items) of an object.
max()	Return the largest item in an iterable.
min()	Return the smallest item in an iterable
pow()	Return power raised to a number.
range()	Return an iterable sequence.
round()	Return the rounded floating point value.
sorted()	Return a new sorted list.
sum()	Sums the items of an iterable from left to right and returns total.
type()	Return the type of an object.
open()	Open file and return a corresponding file object.

Built-in functions (using modules)

- Functions that come built into the Python language itself are called built-in functions and are not readily available to us but need to import from modules
  - To view modules

```
>>> help("modules")
```

  - To view the what are the functions available in that modules

```
>>> import math  # import module first
>>> help(math)
```

Module `math` Functions

- Python provides a collection of standard code stored in libraries called *modules*.
  - Contains function definitions and other elements
    - All of which are related in some way
  - Calling a function
    - ***`functionName ( argument1, argument2 )`***
  - The **`import`** keyword is used to include a module
  - Invoking functions from a module
    - Use the module name followed by the dot operator (.)
    - **`moduleName.functionName( argument )`**
  - Example : $f(x)=2x+3$ we can compute $f(5)$=13 and $f(0)$=3.

Importing Issues

- Import one or more specific functions and variables
  - Allows a programmer to import only a specific part of a module
    - `from math import sqrt, log`
  - Import everything the module has to offer:
    - `from math import *`
- The **`import/as`** statement
  - Creates a reference that is used to call the module
    - **`from math import sqrt as squareRoot`**
    - **`squareRoot( 9.0 )`**
- Importing One or More Modules
  - Use the keyword **`import`** followed by the desired module
    - `import math`
      - Use the module name followed by the dot operator (.)
      - **y = math.sqrt(x)**
  - Several imports may be made over several lines
  - One **`import`** can exist with several coma separated modules
    - `import math, random`

Function : sqrt()

- In Python, a function is a named block of code that performs a specific task.
- Example mathematical square root function.
  - Python has a function in its standard library named sqrt
  - accepts one numeric value and produces a floating-point result
  - The interpreter is not automatically aware of the sqrt function, like type, int, str, and range
  - from math import sqrt
  - sqrt(4)   # legal, a function calling
  - sqrt(“16”)   # Illegal, a string is not a number