NEP NEW SYLLABUS

PYTHON PROGRAMMING

For F.Y.B.Sc. Cyber and Digital Science : Semester - II

[Course Code CDS - 153 - MJ : Credits - 2]

CBCS Pattern under NEP

w.e.f. Academic Year 2024

Dr. Ms. Manisha Bharambe

M.Sc. (Comp. Sci.), M.Phil. Ph.D. (Comp. Sci.)

Vice Principal, Associate Professor, Department of Computer Science

MES's Abasaheb Garware College, Karve Road, Pune

Dr. Ranjit Patil

M.Sc. (Comp. Sci.), M.Phil. NET, Ph.D.

Principal, Dr. D. Y. Patil Arts, Commerce and Science College

Pimpri

Price ₹ 360.00

NIRALI PRAKASHAN

ADVANCEMENT OF KNOWLEDGE

N8333

Python Programming **ISBN 978-93-6109-268-8**

First Edition : February 2025

© : Authors

The text of this publication, or any part thereof, should not be reproduced or transmitted in any form or stored in any computer storage system or device for distribution including photocopy, recording, taping or information retrieval system or reproduced on any disc, tape, perforated media or other information storage device etc., without the written permission of Authors with whom the rights are reserved. Breach of this condition is liable for legal action.

Every effort has been made to avoid errors or omissions in this publication. In spite of this, errors may have crept in. Any mistake, error or discrepancy so noted and shall be brought to our notice shall be taken care of in the next edition. It is notified that neither the publisher nor the authors or seller shall be responsible for any damage or loss of action to any one, of any kind, in any manner, therefrom. The reader must cross check all the facts and contents with original Government notification or publications.

Published By :
NIRALI PRAKASHAN
Abhyudaya Pragati, 1312, Shivaji Nagar,
Off J.M. Road, Pune – 411005
Tel - (020) 25512336/37/39
Email : niralipune@pragationline.com

Polyplate

Printed By :
YOGIRAJ PRINTERS AND BINDERS
Survey No. 10/1A, Ghule Industrial Estate
Nanded Gaon Road
Nanded, Pune - 411041

DISTRIBUTION CENTRES

PUNE

Nirali Prakashan
(For orders outside Pune)
S. No. 28/27, Dhayari Narhe Road, Near Asian College
Pune 411041, Maharashtra
Tel : (020) 24690204; Mobile : 9049672225
Email : bookorder@pragationline.com

Nirali Prakashan
(For orders within Pune)
119, Budhwar Peth, Jogeshwari Mandir Lane
Pune 411002, Maharashtra
Mobile : 9657703145, 9890997937
Email : niralilocal@pragationline.com

MUMBAI

Nirali Prakashan
Rasdhara Co-op. Hsg. Society Ltd., 'D' Wing Ground Floor, 385 S.V.P. Road
Girgaum, Mumbai 400004, Maharashtra
Mobile : 7045821020, Tel : (022) 2385 6339 / 2386 9976
Email : niralimumbai@pragationline.com

DISTRIBUTION BRANCHES

DELHI

Nirali Prakashan
Room No. 2 Ground Floor
4575/15 Omkar Tower, Agarwal Road
Darya Ganj, New Delhi 110002
Mobile : 9818561840
Email : delhi@niralibooks.com

BENGALURU

Nirali Prakashan
Maitri Ground Floor, Jaya Apartments,
No. 99, 6th Cross, 6th Main,
Malleswaram, Bengaluru 560003
Karnataka; Mob : 9686821074
Email : bengaluru@niralibooks.com

NAGPUR

Nirali Prakashan
Above Maratha Mandir, Shop No. 3,
First Floor, Rani Jhanshi Square,
Sitabuldi Nagpur 440012 (MAH)
Tel : (0712) 254 7129
Email : nagpur@niralibooks.com

KOLHAPUR

Nirali Prakashan
438/2, Bhosale Plaza, Ground Floor
Khasbag, Opp. Balgopal Talim
Kolhapur 416 012, Maharashtra
Mob : 9850046155
Email : kolhapur@niralibooks.com

JALGAON

Nirali Prakashan
34, V. V. Golani Market, Navi Peth,
Jalgaon 425001, Maharashtra
Tel : (0257) 222 0395
Mob : 94234 91860
Email : jalgaon@niralibooks.com

SOLAPUR

Nirali Prakashan
R-158/2, Avanti Nagar, Near Golden
Gate, Pune Naka Chowk
Solapur 413001, Maharashtra
Mobile 9890918687
Email : solapur@niralibooks.com

marketing@pragationline.com | www.niralibooks.com

Also find us on www.facebook.com/niralibooks

Preface ...

We take an opportunity to present this Text Book on **"Python Programming"** to the students of First Year B.Sc. (Cyber and Digital Science) Semester-II as per the New CBCS Syllabus under NEP (National Education Policy), effective from June 2024.

The book has its own unique features. It brings out the subject in a very simple and lucid manner for easy and comprehensive understanding of the basic concepts. The book covers Fundamentals of Python Programming, Built-In Data Structures in Python, Strings and Arrays, Functions and Objects Oriented Concepts and Introduction to Python Modules and Libraries.

A special word of thank to Shri. Dineshbhai Furia, and Mr. Jignesh Furia for showing full faith in us to write this text book. We also thank to Ms. Madhuri Gabhale and Mr. Akbar Shaikh of M/s Nirali Prakashan for their excellent co-operation.

We also thank Ms. Chaitali Takle, Mr. Ravindra Walodare, Mr. Sachin Shinde, Mr. Ashok Bodke, Mr. Moshin Sayyed and Mr. Nitin Thorat.

Although every care has been taken to check mistakes and misprints, any errors, omission and suggestions from teachers and students for the improvement of this text book shall be most welcome.

Authors

■■■

Syllabus ...

1. **Fundamentals of Python Programming** **(6 Hours)**
   - Introduction to Python.
   - Features and Applications of Python.
   - Comments, identifiers and reserved words in Python Data types in Python, Data type Conversion.
   - Python Print Function and Input Function.
   - Python Operators (Arithmetic, Comparison, Assignment, Bitwise, Logical, Membership, Identity), Operator Precedence.
   - Indentation in Python.
   - Conditional Statements, Loop Statements, Control Statements (break, continue, pass).
2. **Built-in Data Structures in Python** **(8 Hours)**
   - Python List - Concept, Declaration, Inserting, Updating, Deleting and Accessing Elements, Built-in Operators and Functions, Indexing and Slicing Elements.
   - Python Tuple - Concept, Creating and Accessing Elements, Tuple Operators and Built-in Tuple Functions.
   - Python Set - Concept, Declaration, Inserting, Updating, Deleting and Accessing Elements, Set Operations.
   - Python Dictionary - Concept, Declaration, Inserting, Updating, Deleting Elements and different ways of accessing Dictionary Elements, Built-in Functions, Dictionary Properties Python Data Structure Conversion.
3. **Strings and Arrays** **(6 Hours)**
   - Concept of String.
   - Types of String (Single Quotes, Double Quotes, Triple Quotes).
   - Creating and Accessing String.
   - String Operators.
   - Python Standard String Handling Functions Concept of Array.
   - Creating and Accessing Array Elements.
   - Array Operations (Traverse, Insertion, Deletion, Search and Update).
   - Built-in Array Methods.
4. **Functions and Object Oriented Concepts** **(6 Hours)**
   - Defining and Calling Function.
   - Function Arguments - Required Arguments, Default Arguments, Keyword Arguments, Variable - Length Arguments.
   - Scope of Variable - Basic Rules.
   - Order of Arguments (positional and keyword) Void Function and Lambda Functions Recursion.
   - Object Oriented Programming Concept.
   - Python Classes and Objects, Accessing Members.
   - Python Constructor.
   - Data Hiding.
   - Class Variables, Instance Variables, Class Methods and Static Methods.
5. **Introduction to Python Modules and Libraries** **(4 Hours)**
   - Introduction to Built-in Modules in Python(OS, Random, Math, Datetime, Calendar, Sys, Collections, Statistics).
   - Introduction to Python Libraries (NumPy, Pandas, Matplotlib).

■■■

Contents ...

1. **Fundamentals of Python Programming** **1.1 – 1.94**

2. **Built-In Data Structures in Python** **2.1 – 2.62**

3. **Strings and Arrays** **3.1 – 3.30**

4. **Functions and Objects Oriented Concepts** **4.1 – 4.61**

5. **Introduction to Modules and Libraries** **5.1 – 5.26**

■■■

CHAPTER 1

Fundamentals of Python Programming

Objectives...

- To learn Basic Concepts in Python Programming
- To study Working of Python Programming
- To understand Input, Output and Imports Functions in Python Programming
- To learn Variables, Identifiers, Operators etc. in Python Programming

1.1 INTRODUCTION

- Today, Python is the trendiest programming language programming.
- There are several reasons for why Python programming language is the preferable choice of the programmers/developers over other popular programming languages like C++, Java and so on.
- Python is popular programming language because of it provides more reliability of code, clean syntax of code, advanced language features, scalability of code, portability of code, support object oriented programming, broad standard library, easy to learn and read, support GUI mode, interactive, versatile and interpreted, interfaces to all major commercial databases and so on.
- There are two major Python versions namely, Python 2 and Python 3. Python 3.0 was developed with the same philosophy as in prior versions.
- Python 3.11.4 is the newest/latest major release of the Python programming language (released on June 6, 2023). In this book we will use Python 3.9.6 release of the Python programming language. Python scripts normally have the file extension .py.

1.2 PYTHON PROGRAMMING LANGUAGE

- Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language.

- Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

Why Python?

1. Python can be treated in a procedural way, an object-oriented way or a functional way.
2. Python's simple, easy to learn syntax emphasizes readability of code.
3. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
4. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
5. Python has powerful development libraries include AI (Artificial Intelligence), ML (Machine learning) etc.

- Today, there are many free and commercial editors available for Python. Following table lists Python editors:

Table 1.1

Sr. No.	Editor	Description	Icon/Logo
1.	IDLE	• IDLE is a popular Integrated Development Environment (IDE) written in Python and it has been integrated with the default language. • Mainly used by the beginner level developers who want to practice on Python development.	
2.	PyCharm	• PyCharm is one of the widely used Python IDE which was created by Jet Brains. • With PyCharm, the developers can write a neat and maintainable code. It helps to be more productive and gives smart assistance to the developers. • It takes care of the routine tasks by saving time and thereby increasing profit accordingly.	
3.	Spyder	• It was mainly developed for scientists and engineers to provide a powerful scientific environment for Python. • It offers an advanced level of edit, debug, and data exploration feature. • It is very extensible and has a good plugin system and API.	

contd. …

4.	PyDev	• PyDev is an outside plugin for Eclipse. It is basically an IDE that is used for Python development. • It is linear in size. It mainly focuses on the refactoring of python code, debugging in the graphical pattern, analysis of code etc. It is a strong python interpreter.	
5.	Jupyter Notebook	• The Jupyter Notebook is a browser-based graphical interface to the IPython shell. • Allows us to create and share documents that contain live code, equations, visualizations and narrative text.	

- In this text book, IDLE is used for Python programming. IDLE (Integrated Development and Learning Environment) is an Integrated Development Environment (IDE) for Python.
- To start IDLE interactive shell, search for the IDLE icon in the start menu and double click on it and we will get the following window, (See Fig. 1.1).
- In Python IDLE shell not only we can execute commands one by one like in Python Command Prompt but also can create .py files and see execution of those files.

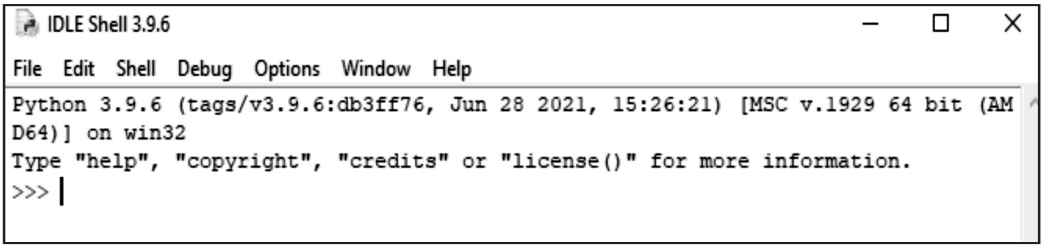


Fig. 1.1: Python Shell

1.2.1 History

- Python laid its foundation in the late 1980s. Python was developed by Guido Van Rossum at National Research Institute for Mathematics and Computer Science in Netherlands in 1990.
- Inspired by Monty Python's Flying Circus, a BBC comedy series, he named the language Python.
- Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.
- ABC programming language is said to be the predecessor of Python language which was capable of Exception Handling and interfacing with Amoeba Operating System.
- Like Perl, Python source code is now available under the GNU General Public License (GPL). In February 1991, Guido Van Rossum published Python 0.9.0 (first release). In addition to exception handling, Python included classes, lists and strings.

- In 1994, Python 1.0 was released with new features like lambda, map, filter, and reduce which aligned it heavily in relation to functional programming.
- Python 2.0 added new features like list comprehensions, garbage collection system and it supported Unicode.
- On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify fundamental flaw of the language. In Python 3.0 the print statement has been replaced with a print() function.
- Python widely used in both industry and academia because of its simple, concise and extensive support of libraries.
- Python is available for almost all operating systems such as Windows, Mac, Linux/Unix etc. Python can be downloading from http://www.python.org/downloads.

1.2.2 Features

- Features of Python programming language are given below:
  1. **Simple and Easy-to-Learn:** Python is a simple language with few keywords, simple structure and its syntax is also clearly defined. This makes Python a beginner's language. In Python programming programs are easy to write and execute as it omits some cumbersome, poorly understandable and confusing features of other programming language such as C++ and Java.
  2. **Interpreted and Interactive:** Python is processed at runtime by the interpreter. We need not compile the Program before executing it. The Python prompt interact with the interpreter to interpret the programs that we have written. Python has an option namely interactive mode which allows interactive testing and debugging of code.
  3. **Object-Oriented:** Python supports Object Oriented Programming (OOP) concepts that encapsulate code within objects. All concepts in OOPs like data hiding, Operator Overloading, inheritance etc. can be well written in Python. It supports functional as well as structured Programming.
  4. **Platform Independence/Cross Platform Language/Portable:** Python can run on a wide variety of hardware and software platforms and has the same interface on all platforms, such as Windows, Unix, Linux and Macintosh. Fig. 1.2 shows execution of Python code by interpreter.

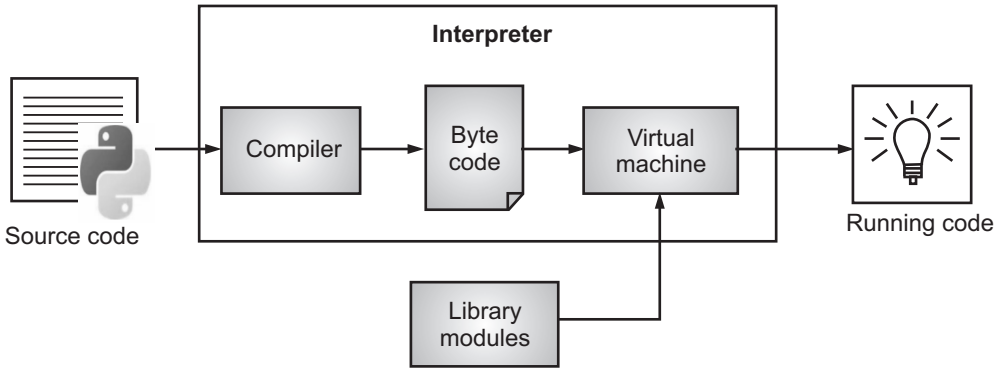


Fig. 1.2: Execution of Python Code

Python source code goes through Compiler which compiles the source code into a format known as byte code. Byte code is a lower level, platform independent, efficient and intermediate representation of the source code. As soon as source code gets converted to byte code, it is fed into PVM (Python Virtual Machine). The PVM is the runtime engine of Python; it's always present as part of the Python system, and is the component that truly runs the scripts. Technically, it's just the last step of what is called the Python interpreter.

5. **Scalable:** Python provides a better structure and support for large Programs than shell scripting. It can be used as a scripting language or can be compiled to byte code {intermediate code that is platform independent} for building large applications.
6. **Extendable:** You can add low-level modules to the Python interpreter. These modules enable Programmers to add to or customize their tools to be more efficient. It can be easily integrated with C, COM, ActiveX, CORBA, and Java.
7. **Dynamic:** Python provides very high-level dynamic data types and supports dynamic type checking. It also supports automatic garbage collection.
8. **GUI Programming and Databases:** Python supports GUI applications that can be created and ported to many libraries and windows systems, such as Windows Microsoft Foundation Classes (MFC), Macintosh, and the X Window system of Unix Python also provides interlaces to all major commercial databases.
9. **Broad Standard Library:** Python's library is portable and cross platform compatible on UNIX, Linux, Windows and Macintosh. This helps in the support and development of a wide range of applications from simple text processing to browsers and complex games.
10. **Free and Open Source:** Python programming language is developed under an OSI approved open source license making it freely available at official web address. The source code is also available for use. The Python software can be freely distributed and any one can use and read its source code make changes/ modifications to it and use the pieces in new free programs.

1.2.3 Applications

- Python is known for its general purpose nature that makes it applicable in almost each domain of software development.
- Here, we are specifying applications areas where python can be applied:
  1. **Web Applications:** We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML, XML and JSON, etc. It also provides Frameworks such as Django, Pyramid, Flask etc. to design and develop web based applications. Some important developments are PythonWikiEngines, PythonBlogSoftware etc.

2. **GUI Based Desktop Applications:** Python has simple syntax, modular architecture, rich text processing tools and the ability to work on multiple operating systems which make it a desirable choice for developing desktop-based applications. There are various GUI toolkits like wxPython, PyQt or PyGtk available which help developers create highly functional Graphical User Interface (GUI).
3. **Scientific and Numeric Applications:** Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, Ipython, Numeric Python etc. SciPy is group of packages of engineering, science and mathematics.
4. **Software Development:** Python is helpful for software development process. It works as a support language and can be used for build control and management, testing etc.
5. **Business Applications:** With features that include special libraries, extensibility, scalability and easily readable syntax, Python is a suitable coding language for customizing larger applications. Python is used to build Business applications like ERP and e-commerce systems. Tryton is a high level application platform.
6. **Console Based Application:** We can use Python to develop console based applications. For example, IPython.
7. **Audio or Video based Applications:** Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of real applications are TimPlayer, cplay etc.
8. **3D CAD Applications:** To create CAD application Fandango is a real application which provides full features of CAD.
9. **Enterprise Applications:** Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications are OpenErp, Tryton, Picalo etc.
10. **Applications for Images:** Using Python several application can be developed for image. Applications developed are VPython, Gogh, imgSeek etc.
11. **Education:** Python is a superb language for teaching programming, both at the introductory level and in more advanced courses.

1.2.4 Installing Python

- Python distribution is available for a wide variety of platforms such as Unix, Linux, Macintosh and Windows. We need to download only the binary code applicable for the platform and install Python.

- The most up-to-date and current source code, binaries, documentation, news, etc. is available on the official website of Python https://www.python.org/.

Installing Python in Windows:

Step 1 : Open any internet browser then type http:///www.python.org/downloads/ in address bar and Enter. The Home page will appear, (See Fig. 1.3).

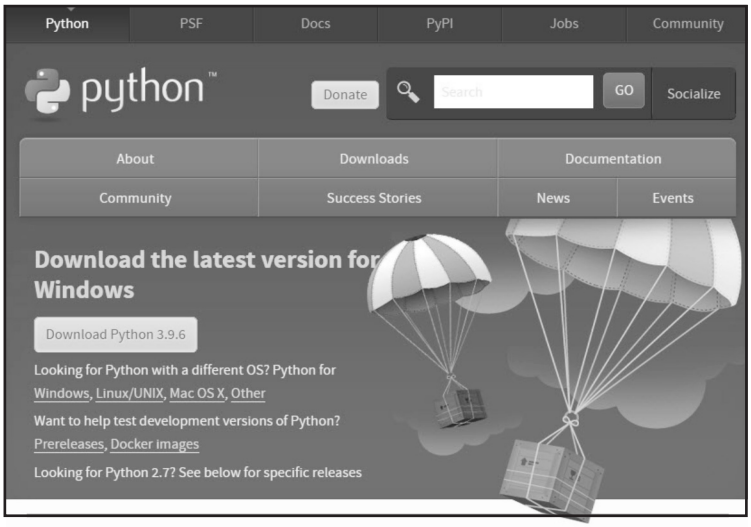


Fig. 1.3: Home Page

Step 2 : Click on download to get the latest version of Python.



Fig. 1.4: Python Release Versions

Step 3 : Once Python version is downloaded in this case 3.9.6. Open the Python 3.9.6 version pack and double click on "Install Now", option for to start installation (See Fig. 1.5).

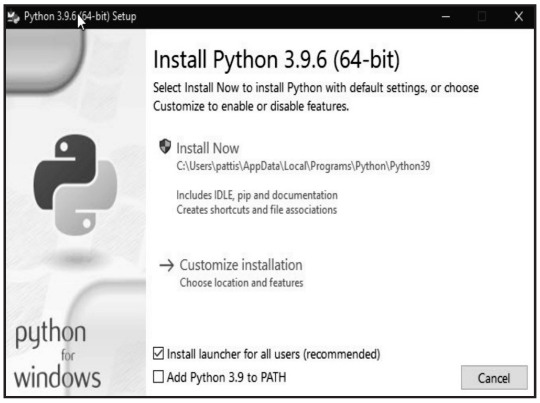


Fig. 1.5

Step 4 : When installation of starts it shows Setup Progress as shown in Fig. 1.6.

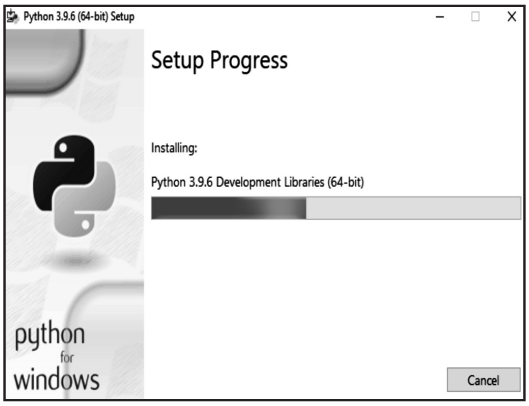


Fig. 1.6

Step 5 : After complete the installation click close button to finish installation.

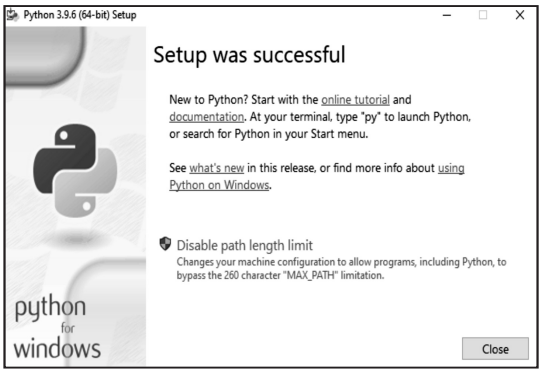


Fig. 1.7

Starting Python in different Modes:

1. **Starting Python (Command Line):**

- A Python script can be executed at command line also. This can be done by invoking the interpreter on the application.

- In command line mode, we type the Python programming program on the Python shell and the interpreter prints the result. The steps are given below:

Step 1 : Press Start button, (See Fig. 1.8).

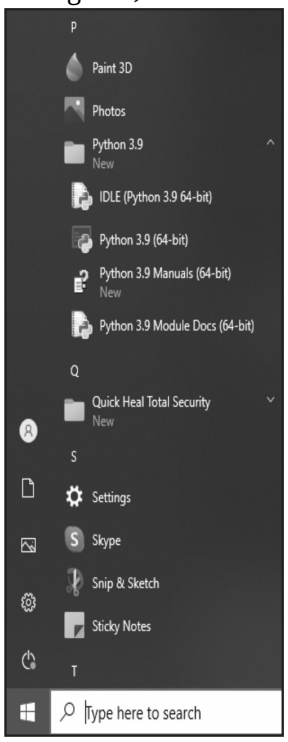


Fig. 1.8

Step 2 : Click on All Programs and then click on Python 3.9 (64 bit) as shown in Fig. 1.9. We will see the Python interactive prompt in Python command line.

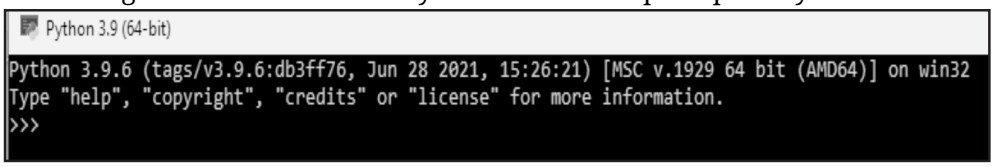


Fig. 1.9

Python command prompt contains an opening message >>> called command prompt. The cursor at command prompt waits for to enter Python command. A complete command is called a statement. For example check first command to print message, in Fig. 1.10.

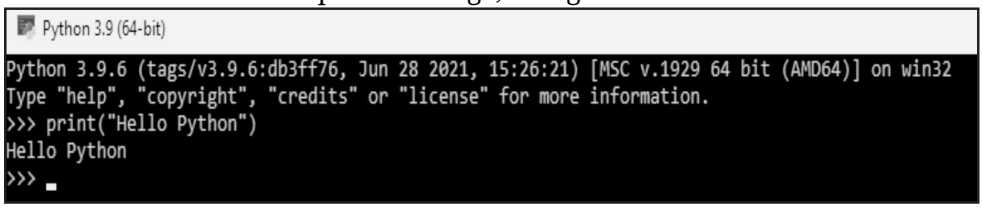


Fig. 1.10

Step 3 : To exit from the command line of Python, use Ctrl+z or quit() followed by Enter.

2. Starting Python IDLE:

- When we install Python 3, we also get IDLE (Integrated Development Environment).
- IDLE includes a color syntax-highlighting editor, a debugger, the Python Shell, and a complete copy of Python 3's online documentation set.
- The steps are given below:

Step 1 : Press Start button and click on IDLE (Python 3.9, 64-bit) options.

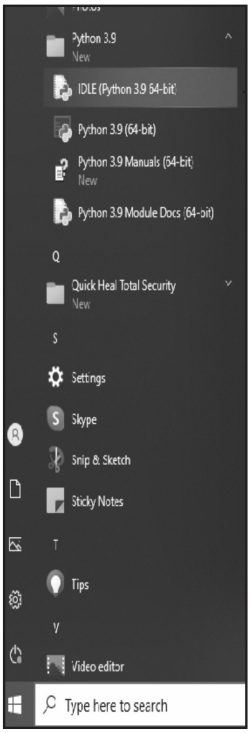


Fig. 1.11

Step 2 : We will see the Python interactive prompt i.e. interactive shell.

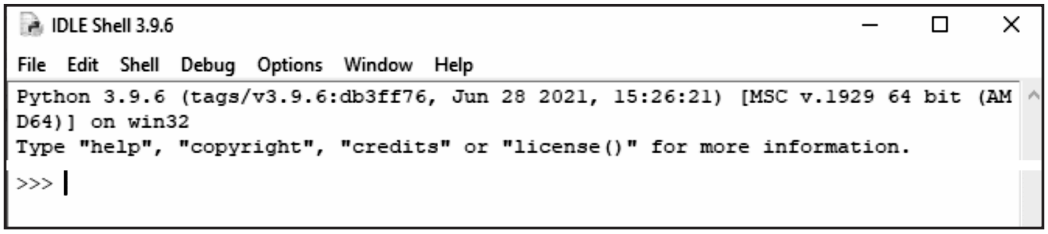


Fig. 1.12

Python interactive shell prompt contains opening message >>>, called shell prompt. A cursor is waiting for the command. A complete command is called a statement. When we write a command and press enter, the python interpreter will immediately display the result.

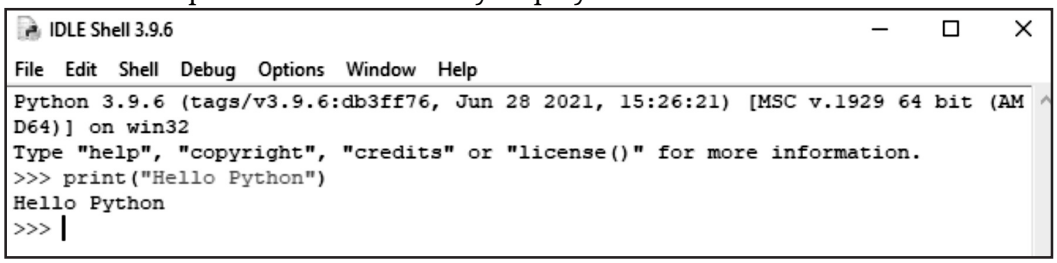


Fig. 1.13

Executing Python Programs Scripts:

- In Python IDLEs shell window, click on File, and select the New File or press Ctrl+N.

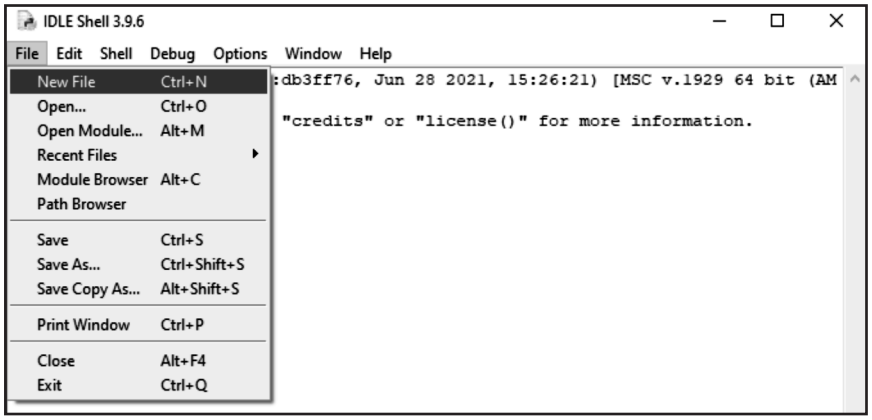


Fig. 1.14

- As soon as we click on New File, the window appears as shown in Fig. 1.15.

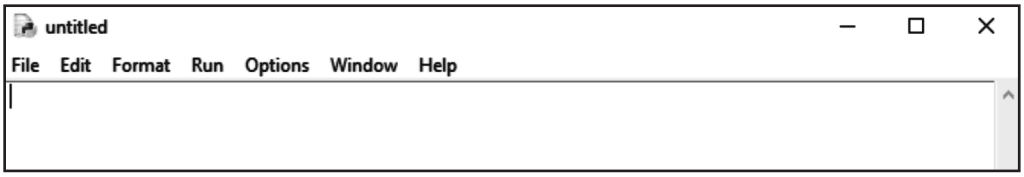


Fig. 1.15

- Write Python program in script mode, (See Fig. 1.16).

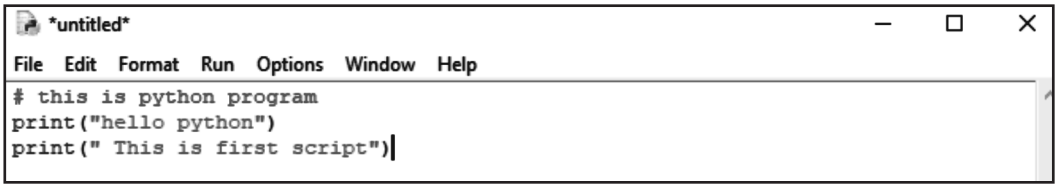


Fig. 1.16

- Save the code shown in Fig. 1.16 with filename. By default, python interpreter will save it using the filename.py. Here, we save the script with file name test.py.

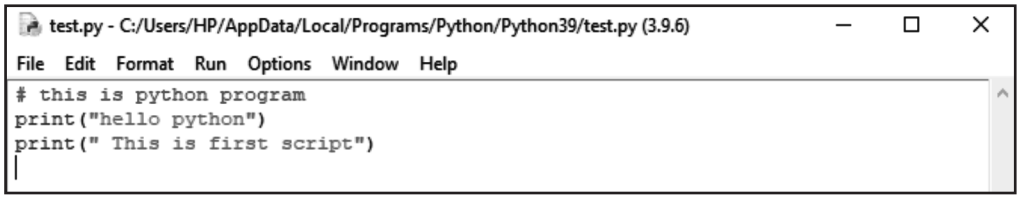


Fig. 1.17

- To run the Python program, click on Run and then Run Module option or we can press Ctrl+F5.

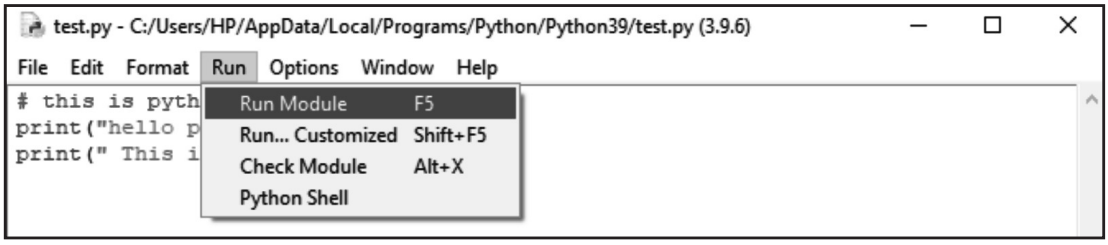


Fig. 1.18

- After clicking Run Module, we will get the output of program on Python shell.

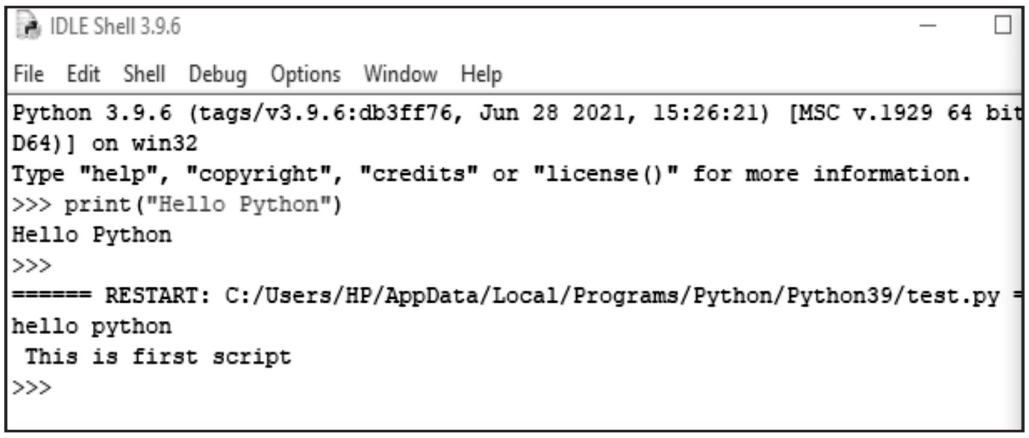


Fig. 1.19

1.2.5 Running Simple Python Program

- Python program is set of instructions. Python programs are typically organized with one statement per line. Before to write and run a Python program we understand structure of a Python program.
- Fig. 1.20 shows a typical program structure of Python programming.
- Python programming programs are structured as a sequence of statements. A Python statement is smallest program unit.
- Statements are the instructions that are written in a program to perform a specific task. A Python statement is a complete instruction executed by the Python interpreter.
- By default, the Python interpreter executes all statements sequentially, but we can change order of execution using control statements.

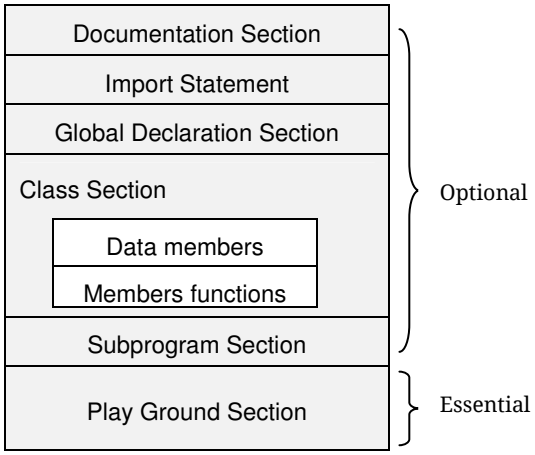


Fig. 1.20: Typical Program Structure of Python Programming

- Program structure of Python programming contains following sections:
  1. **Documentation Section** includes the comments that specify the purpose of the program. A comments that is a non-executable statement which is ignored by the compiler while program execution. Python comments are written anywhere in the program.
  2. **Import Section** is includes different built in or user defined modules.
  3. **Global Declaration Section** is used to define the global variables for the programs.
  4. **Class Section** describes the information about the user defined classes in the Python program. A class is a collection of data members and member functions called method that operate on data members.
  5. **Sub Program Section** includes user defined functions. The functions include the set of statements that need to be executed when the function is called form anywhere.
  6. **Pay Ground Section** is the main section of Python program and the main section starts where the function calling.
- Following simple Python program shows structure of Python program which calculates area and circumference of circle using class:

Program 1.1:

```
# Documentation section
# This program calculates the area and circumference of a circle using a class structure.
import math                    # Import statement
# Global declaration section
radius = 5
```

```
# Class section
class Circle:
 def __init__(self, radius):
    self.radius = radius       # Using instance variables for better
                               encapsulation
 def get_area(self):
     """Calculate and return the area of the circle."""
     return math.pi * self.radius * self.radius
 def get_circumference(self):
     """Calculate and return the circumference of the circle."""
     return 2 * math.pi * self.radius
# Subprogram section
def show_radius():
   """Display the global radius."""
print("Radius =", radius)
# Playground section
show_radius()  # Display the global radius
# Create an instance of Circle
circle = Circle(radius)
# Calculate and display area and circumference
print("Area =", circle.get_area())
print("Circumference =", circle.get_circumference())
```

Output:

```
Radius = 5
Area = 78.53981633974483
Circumference = 31.41592653589793
```

- **Comments:** Comments in python is start with a '#' symbol and are ignored by the interpreter.
- **Import Statements:** There are any number of import statement in the program which are used to import modules or libraries into the program.
- **Variables:** In Python, variables do not need to be declared with a specific type. The data type of variable is decided at run time by the interpreter. Python supports several built-in data types including integers, floats, strings, booleans, and lists.
- **Operators:** Operators are used to perform operations on variables and data. Python supports arithmetic, relational, and logical operators.

- **Control Structures:** Control structures are used to control the flow of a program. Python supports if-else statements, for loops, and while loops.
- **Functions:** Functions are used to group a set of related statements together by using def keyword in python.
- **Classes:** Classes are used to define objects that have specific attributes and methods. They are used to create more complex data structures and encapsulate code.
- **Exceptions:** Exceptions are used to handle errors that may occur during the execution of a program.

Note: Proper indentation is important in python program.

For Example:

```
# Program Name or description      # Documentation section
import math                        # import statement
a=5                                # global declaration Section
b=a+10                             # Arithmatic statement
print(b)                           # value of b is the output
if (a<b):                          # if loop
   print(a)
```

Program 1.2: # Program to find area and circumference

```
# Documentation section
import math                                    # import statement
radius=5                                       # global declaration Section
class Circle():                                # class section
   def getArea(self):                          # function definition
      return math.pi*radius*radius
   def getCircumference(self):                 # function definition
      return radius*2*math.pi
c=Circle()                                     # creating object c
print("Area =",c.getArea())
print("Circumference=",c.getCircumference())
```

Output:

```
Area = 78.53981633974483
Circumference = 31.41592653589793
```

- Python has two basic modes namely, normal and interactive.
  1. The **normal script mode** is the mode where the scripted and finished .py files are run in the Python interpreter.

2. The **interactive mode** is a command line shell which gives immediate feedback for each statement, while running previously fed statements in active memory.

- As new lines are fed into the interpreter, the fed program is evaluated both in part and in whole.

1. Interactive Mode:

- Interactive mode is used for quickly and conveniently running single line or blocks of code. Here's an example using the python shell that comes with a basic python installation.
- The ">>>" indicates that the shell is ready to accept interactive commands. For example, if we want to print the statement "Interactive Mode", simply type the appropriate code and hit enter.

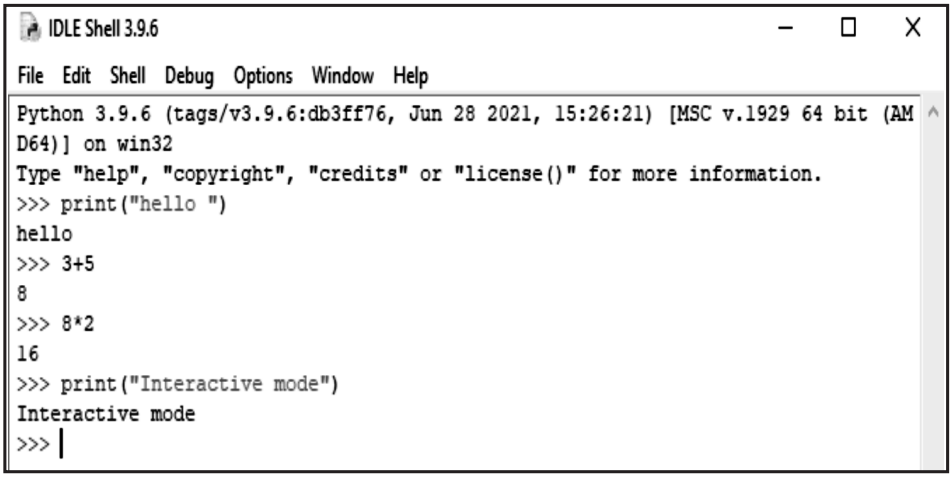


Fig. 1.21

2. Script Mode:

- In the standard Python shell we can go to "File → New File" (or just hit Ctrl + N) to pull up a blank script to write the code. Then save the script with a ".py" extension.
- We can save it anywhere we want for now, though we may want to make a folder somewhere to store the code as we test Python out.
- To run the script, either select "Run → Run Module" or press F5.
- We should see something like the following, (See Fig. 1.22 (a) and 1.22 (b)).

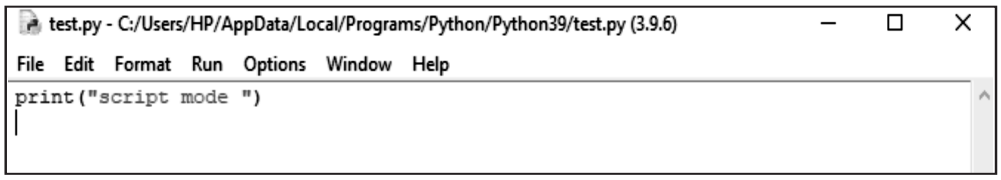


Fig. 1.22 (a)

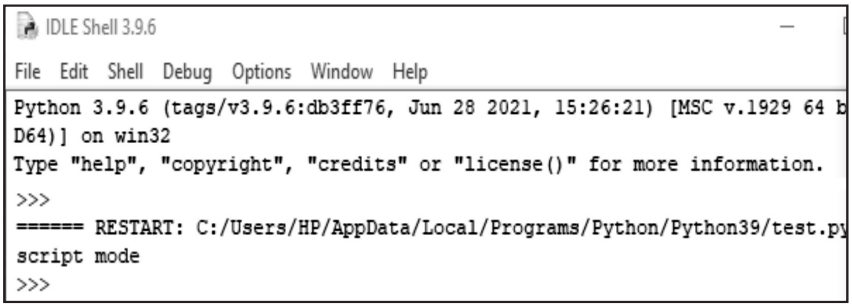


Fig. 1.22 (b)

Running Simple Python Script/Program to Display 'Welcome' Message:

- There are two modes for executing Python program namely Interactive mode programming and Script mode programming.
  - In **interactive mode programming**, interpreter is invoked and the programmer can code statements directly to the interpreter without passing a script file as a parameter.
  - In **script mode programming,** the complete script is written in an editor such as Notepad in Windows and then interpreter is invoked with a script parameter. It begins execution of the script and continues until the script is finished.

1. Interactive Mode Programming:

- Click on All Programs and then click on Python 3.9 (64-bit). We will see the Python interactive prompt in Python command line.
- This method invokes the interpreter without passing a script file and brings up the following prompt, (See Fig. 1.23).

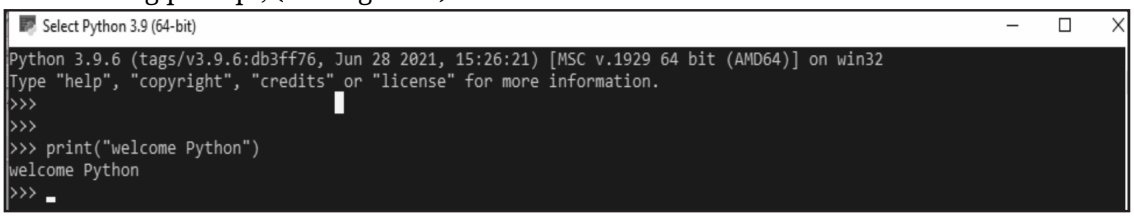


Fig. 1.23

2. Script Mode Programming:

- Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.
- Let us write a simple Python program in a script. Python files have extension .py. Type the following source code in a test.py file:

```
print"Welcome, Python!"
```

- We assume that we have Python interpreter set in PATH variable. Now, try to run this program as follows:

```
$ python test.py
```

Output:

```
Welcome, Python!
```

- On Linux OS to execute a Python script modified test.py file:

```
#!/usr/bin/python
print"Hello, Python!"
```

- We assume that we have Python interpreter available in /usr/bin directory. Now, try to run this program as follows:

```
$ chmod +x test.py     # This is to make file executable
$./test.py
Hello, Python!
```

1.3 BASICS OF PYTHON

- In this section we will study various basic concepts if Python programming such as variables, identifiers, comments and so on.

1.3.1 Identifiers

- A Python identifier is a name given to a function, class, variable, module or other objects that is used in Python program.
- All identifiers must obey the following rules:
  1. An identifier can be a combination of uppercase letters, lowercase letters, underscores, and digits (0-9). Examples Name, myClass, Emp_Salary, var_1, _Address and a1 are valid identifiers.
  2. We can use underscores to separate multiple words in the identifier. For example, Emp_Salary.
  3. An identifier starts with a letter which can be alphabet (either lowercase or uppercase), underscore (_).
  4. Identifiers can be of any length.
  5. Identifiers cannot start with digit and must not contain any space or tabs. Example 2x, 10ID are invalid.
  6. We cannot use Python keywords as identifiers.
  7. Special characters such as %, @, and $ are not allowed within identifiers. Example $Money, @salary are invalid.
  8. Python is a case-sensitive language and this behavior extends to identifiers. Thus, identifier Age and age are two distinct identifiers in Python.
- Example of valid identifiers are: `Circle_Area, EmpName, Student, Sum, Salary2018, _PhoneNo.`
- Example of invalid identifiers includes: `!count, 4marks, %Loan, 2018Salary.`

1.3.2 Reserved Words

- Python keywords are reserved words with that have special meaning and functions. The keywords are predefined words with specific meaning in the Python programs.

- Keywords should not be used as variable name, constant, function name, or identifier in the program code.
- In Python keywords are case sensitive. Keywords are used to define the syntax and structure of the programming language.
- Following table lists keywords in Python programming:

Table 1.2

and	as	assert	break	class	continue
def	del	else	elif	except	exec
false	finally	for	from	global	if
import	in	is	lambda	none	not
or	pass	print	raise	return	true
try	while	with	yield		

1.3.3 Variables

- A variable is like a container that stores values that we can access or change. It is a way of pointing to a memory location used by a program.
- We can use variables to instruct the computer to save or retrieve data to and from this memory location.
- A variable is a name given to a location in the computer's memory location, where the value can be stored that can be used in the program.
- When we create a variable, some space in the memory is reserved or allocated for that variable to store a data value in it.
- The size of the memory reserved by the variable depends on the type of data it is going to hold. The period of time that a variable exists is called its lifetime.
- The variable is so called because its value may vary during the time of execution, but at a given instance only one value can be stored in it.

Variable Declaration:

- A variable is an identifier that holds a value. In programming, we say that we assign a value to a variable. Technically speaking, a variable is a reference to a computer memory, where the value is stored.
- Basic rules to declare variables in python programming language:
  1. Variables in Python can be created from alphanumeric characters and underscore (_) character.
  2. A variable cannot begin with a number.
  3. The variables are case sensitive. Means Amar is differ the 'AMAR' are two separate variables.
  4. Variable names should not be reserved word or keyword.
  5. No special characters are used except underscore (_) in variable declaration.
  6. Variables can be of unlimited length.

- Python variables do not have to be explicitly declared to reserve memory space. The variable is declared automatically when the variable is initialized, i.e., when we assign a value to the variable first time it is declared with the data type of the value assigned to it.
- This means we do not need to declare the variables. This is handled automatically according to the type of value assigned to the variable. The equal sign (=) i.e., the assignment operator is used to assign values to variables.
- The operand to the left of the = operator is the name of the variable and the operand to the right of the = operator is the literal value or any other variable value that is stored in the variable.

  Syntax: `variable=value`

  For example: For variable.

```
>>> a=10
>>> a
10
>>>
```

- Python language allows assigning a single value to several variables simultaneously.

  For example: `a=b=c=1`

  All above three variables are assigned to same memory location, when integer object is created with value 1.

- Multiple objects can also have assigned to multiple variables:

  For example: `a, b, c = 10, 5.4, "hello"`

  In above example Integer object a assigned with value 10, float object b assigned with value 5.4 and string object c assigned with value "hello".

  For Example: If x, y, z are defined as three variable in a program, then x = 10 will store the value 10 in the memory location named as x, y = 5 will store the value 5 in the memory location named as y and x + y will store the value 15 in the memory location named as z (as a result after computation of x + y).

```
>>> x = 10
>>> y = 5
>>> name = "Python"
>>> z = x + y
>>> print(x); print(y); print(name); Print (z)
10
5
Python
15
```

1.3.4 Constants

- A constant is a type of variable whose value cannot be changed during the execution of program.
- In Python, constants are usually declared and assigned in a module. For declaring and assigning value to a constant consider following example:
  - Create a constant.py:

```
PI = 3.14
GRAVITY = 9.8
```

  - Create a main.py:

```
import constant
print(constant.PI)
print(constant.GRAVITY)
```

```
3.14
9.8
```

Literals in Python:

- Literals are also constant values in Python, which are fixed. A literal refers to the fixed value that directly appears in the program. Literals can be defined as, a data that is given in a variable or constant.
- Literals are numbers or strings or characters that appear directly in a program. Python support the following literals:
  - **String Literals:** "hello" , '12345'
  - **Int Literals:** 0, 1, 2, −1, −2
  - **Long Literals:** 89675L
  - **Float Literals:** 3.14
  - **Complex Literals:** 12j
  - **Boolean Literals:** True or False
  - **Special Literals:** None
  - **Unicode Literals:** u"hello"
  - **List Literals:** [], [5, 6, 7]
  - **Tuple Literals:** (), (9), (8, 9, 0)
  - **Dict Literals:** {}, {'x':1}
  - **Set Literals:** {8, 9, 10}

1. String Literals:

- String literals can be formed by enclosing a text in the quotes. We can use both single quote (‘ … ’) as well as double quotes for (“ … ”) a string.
- In simple words, a string literal is a collection of consecutive characters enclosed within a pair of single or double quotes.