**Chapter 1 🡪 Introduction to Python**

**a) What is Python :-**

Sol :-

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

* web development (server-side),
* software development,
* mathematics,
* system scripting.

**b) History of Python :-**

Sol :-

[Python](https://www.geeksforgeeks.org/python-programming-language/) is a widely used general-purpose, high-level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

**4. Who, When, Where story behind the name?**

Sol :-

In the late 1980s, history was about to be written. It was that time when working on Python started. Soon after that, Guido Van Rossum began doing its application based work in December of 1989 by at Centrum Wiskunde & Informatica (CWI) which is situated in Netherlands. It was started firstly as a hobby project because he was looking for an interesting project to keep him occupied during Christmas. The programming language which Python is said to have succeeded is ABC Programming Language, which had the interfacing with the Amoeba Operating System and had the feature of exception handling. He had already helped to create ABC earlier in his career and he had seen some issues with ABC but liked most of the features. After that what he did as really very clever. He had taken the syntax of ABC, and some of its good features. It came with a lot of complaints too, so he fixed those issues completely and had created a good scripting language which had removed all the flaws. The inspiration for the name came from BBC’s TV Show – ‘Monty Python’s Flying Circus’, as he was a big fan of the TV show and also he wanted a short, unique and slightly mysterious name for his invention and hence he named it Python! He was the “Benevolent dictator for life” (BDFL) until he stepped down from the position as the leader on 12th July 2018. For quite some time he used to work for Google, but currently, he is working at Dropbox.   
The language was finally released in 1991. When it was released, it used a lot fewer codes to express the concepts, when we compare it with Java, C++ & C. Its design philosophy was quite good too. Its main objective is to provide code readability and advanced developer productivity. When it was released it had more than enough capability to provide classes with inheritance, several core data types exception handling and functions.

**5. Features of Python**

Sol :-

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming.  
In Python, we don’t need to declare the type of variable because it is a dynamically typed language.

**Features in Python**

There are many features in Python, some of which are discussed below –

**1. Easy to code:**

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developer-friendly language.

**2. Free and Open Source:**

Python language is freely available at the official website and you can download it from the given download link below click on the **Download Python** keyword.  
[Download Python](https://www.python.org/downloads/)  
Since it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

**3. Object-Oriented Language:**

One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation, etc.

**4. GUI Programming Support:**

Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python.  
PyQt5 is the most popular option for creating graphical apps with Python.

**5. High-Level Language:**

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

**6. Extensible feature:**

Python is a **Extensible** language. We can write us some Python code into C or C++ language and also we can compile that code in C/C++ language.

**7. Python is Portable language:**

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

**8. Python is Integrated language:**

Python is also an Integrated language because we can easily integrated python with other languages like c, c++, etc.

**9. Interpreted Language:**

Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called **bytecode**.

**10. Large Standard Library**

Python has a large standard library which provides a rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers, etc.

**11. Dynamically Typed Language:**

Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don’t need to specify the type of variable.

**6. Where all we use python**

Sol :-

Python is commonly used for developing websites and software, task automation, data analysis, and data visualization. Since it’s relatively easy to learn, Python has been adopted by many non-programmers such as accountants and scientists, for a variety of everyday tasks, like organizing finances.

7. IDE's of Python

Sol :-

**An IDE (Integrated Development Environment) is a software application used by developers for creating programs.** IDEs are meant to make the developer’s job easier by combining tools that are necessary during software development. Your typical IDE will contain tools such as:

* a text editor;
* a compiler and/or interpreter;
* a debugger and code profiler;
* version control integration;
* a number of supporting utilities to interface with external tooling (Docker, cloud deployments, etc.)

…all combined into a single user interface.

Example :- Spyder , Pycharm , IDLE , VS Code

**8. Installation of python**

Sol :-

Basic steps for installation of python are :-

Step 1: Select the version of Python to install.

Step 2: Download Python executable installer.

Step 3: Run Executable installer.

Step 4: Verify python was installed on Windows.

Link :- <https://phoenixnap.com/kb/how-to-install-python-3-windows>

In the above link more details every step is described .

**Python fundamentals** :-

**1.Tokens :-**

A token is the smallest individual unit in a python program. All statements and instructions in a program are built with tokens.

**2.KeyWords :-**

Keywords are words that have some special meaning or significance in a programming language. They can’t be used as variable names, function names, or any other random purpose. They are used for their special features. In Python we have 33 keywords some of them are: try, False, True, class, break, continue, and, as, assert, while, for, in, raise, except, or, not, if, elif, print, import, etc.

**3.DataTypes and types :-**

In programming, data type is an important concept.

Variables can store data of different types, and different types can do different things.

Python has the following data types built-in by default, in these categories:

|  |  |
| --- | --- |
| Text Type: | Str |
| Numeric Types: | int, float, complex |
| Sequence Types: | list, tuple, range |
| Mapping Type: | Dict |
| Set Types: | set, frozenset |
| Boolean Type: | Bool |
| Binary Types: | bytes, bytearray, memoryview |

While the **primitive data types** include **Integers, Float, Strings and Boolean**, the non-**primitive data types** are **Array, List, Tuples, Dictionary, Sets and Files**. Some of these **non-primitive data types**, such as **List, Tuples, Dictionaries and Sets**, are in-built in Python. There is another category of data structures in Python that is **user-defined**; that is, users define them. These include **Stack, Queue, Linked List, Tree, Graph and HashMap.**

**4.Variables :-**

Variables are also known as **Literals** nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable. The equal sign (=) 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 value stored in the variable.

counter = 100 # An integer assignment

miles = 1000.0 # A floating point

name = "Soumo" # A string

print counter

print miles

print name

**5. Comments :-**

A comment in Python starts with the hash character, #, and extends to the end of the physical line. A hash character within a string value is not seen as a comment, though. To be precise, a comment can be written in three ways - entirely on its own line, next to a statement of code, and as a multi-line comment block.In the following sections I'll describe each type of comment :-

**Single-Line Comments**

Such a comment starts with a hash character (#), and is followed by text that contains further explanations.

Example :- name = input(“Enter your name : ”) # Code to take name as input from user.

**Multi-Line Comments**

As already mentioned above, an entire comment block is also understood by Python. These comments serve as in-line documentation for others reading your code, and explain things in more detail, usually.

**6.Indentations**

Indentation refers to the spaces at the beginning of a code line.

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

Python uses indentation to indicate a block of code.

Example :-

if 5 > 2:  
 print("Five is greater than two!")

**7.Operators**

Operators are used to perform operations on variables and values.

Python divides the operators in the following groups:

* Arithmetic operators(+ , - , \* , / , % , \*\* , //)
* Assignment operators(= , += , -= , \*= , /= , %= , //= , \*\*= , &= , |=, ^= , >>= , <<=)
* Comparison operators(== , != , > , < , >= , <=)
* Logical operators(and , or , not)
* Identity operators(is , is not)
* Membership operators(in , not in)
* Bitwise operators(& , | , ^ , ~ , << , >>)

**8. Input and Output statements**

**How to Take Input from User in Python**

Sometimes a developer might want to take user input at some point in the program. To do this Python provides an input() function.

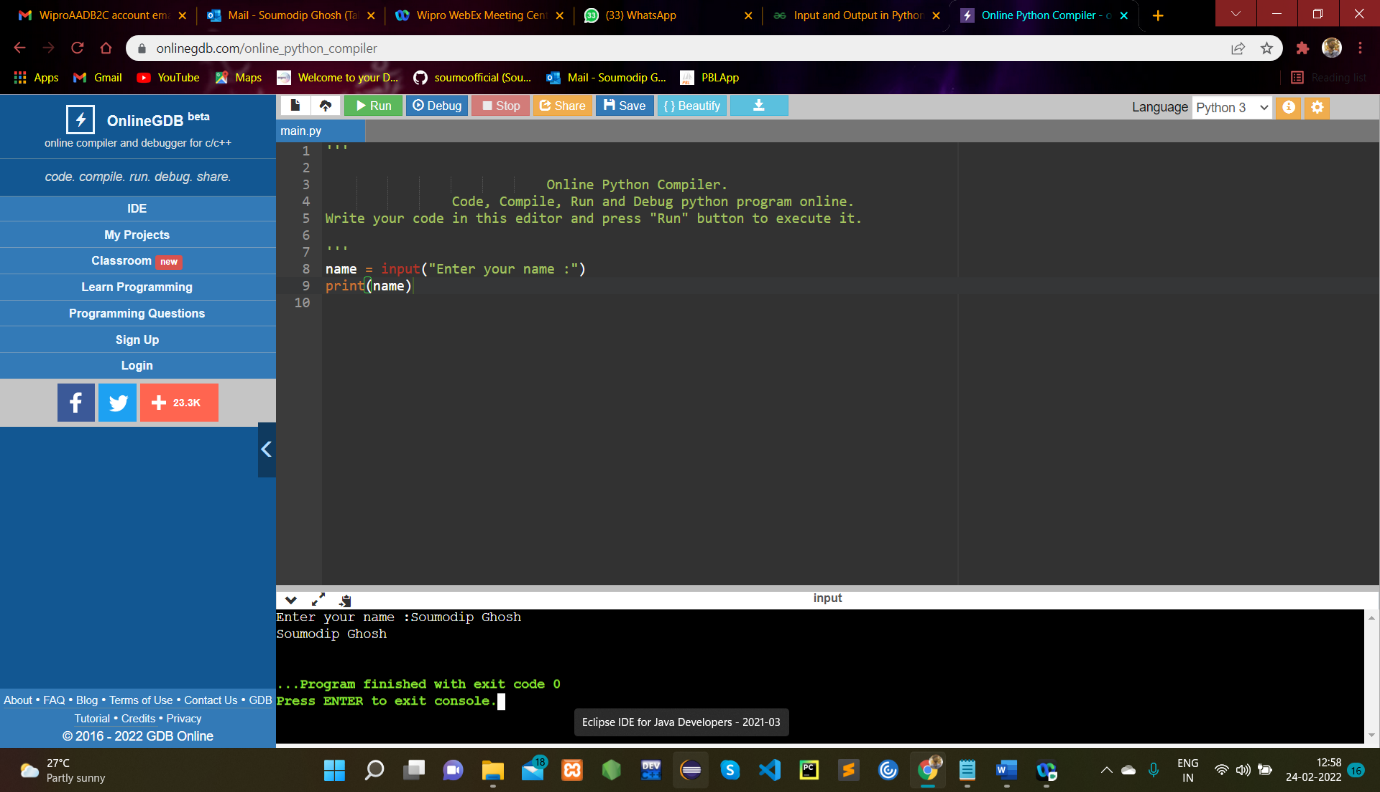
**Syntax:**

Example :-

input('prompt')

where prompt is an optional string that is displayed on the string at the time of taking input.

**Input & Output :-**



**9.Type Casting**

There may be times when you want to specify a type on to a variable. This can be done with casting. Python is an object-orientated language, and as such it uses classes to define data types, including its primitive types.

Casting in python is therefore done using constructor functions:

* int() - constructs an integer number from an integer literal, a float literal (by removing all decimals), or a string literal (providing the string represents a whole number)
* float() - constructs a float number from an integer literal, a float literal or a string literal (providing the string represents a float or an integer)
* str() - constructs a string from a wide variety of data types, including strings, integer literals and float literals

**10. Condition statements**

## **Introduction**

Decision-making is as important in any programming language as it is in life. Decision-making in a programming language is automated using conditional statements, in which Python evaluates the code to see if it meets the specified conditions.

The conditions are evaluated and processed as true or false. If this is found to be true, the program is run as needed. If the condition is found to be false, the statement following the If condition is executed.

### Python has six conditional statements that are used in decision-making:-

1. If the statement

2. If else statement

3. Nested if statement

4. If…Elif ladder

5. Short Hand if statement

6. Short Hand if-else statement

**11. Looping statements**

Python programming language provides following types of loops to handle looping requirements.

|  |  |
| --- | --- |
| **Sr.No.** | **Loop Type & Description** |
| 1 | [while loop](https://www.tutorialspoint.com/python/python_while_loop.htm)  Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body. |
| 2 | [for loop](https://www.tutorialspoint.com/python/python_for_loop.htm)  Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable. |
| 3 | [nested loops](https://www.tutorialspoint.com/python/python_nested_loops.htm)  You can use one or more loop inside any another while, for or do..while loop |

**Chapter 2 🡪 Data Structures**

Some Inbuilt Functions :

1. len() : Returns the length of an object.

Example :

Input :

list1 = [1, 2 , 3 , 4]

x = len(list1)

Output :

4

2. max() : Returns the largest item in a iterable.

Example :

Input :

print(max(1,2))

Output :

2

3. min() : Returns the smallest item in a iterable.

Example :

Input :

print(min(1,2))

Output :

1

4. del() : The del keyword is used to delete objects. In Python everything is an object, so the del keyword can also be used to delete variables, lists, or parts of a list etc.

Example :

x = ["apple", "banana", "cherry"]  
  
del x[0]  
  
print(x)

Output :

["banana", "cherry"]

**List :**

1. In Python, a list is created by placing elements inside square brackets [], separated by commas.

2. A list can have any number of items and they may be of different types (integer, float, string, etc.).

3. List is changeable(**mutable**)

Example :

list1 = [ ]

my\_list = [1 , 2 , 3 , 4]

Functions that can be applied on a list :

a. insert() : inserts the specified value at the specified position.

Syntax :

list.insert(pos, elmnt)

b. append() : It appends an element to the end of the list.

Syntax :

List.append(ele)

c. pop() : It removes the element at the specified position.

Syntax :

list.pop(pos)

d. remove() : This removes the first occurrence of the element with the specified value.

Syntax :

list.remove(element)

e. index() : Returns the position at the first occurrence of the specified value.

Syntax :

list.index(element)

f. count() : returns the number of elements with the specified value.

Syntax :

list.count(value)

g. sort() : sorts the list ascending by default.

list.sort()

if we set list.sort(reverse = True|False)

reverse = True will sort the list in Descending and default is reverse = False.

h. reverse() : reverses the sorting order of elements.

Syntax :

list.reverse()

i. extend() : adds the specified list elements(or any iterable) to the end of the concurrent list.

Syntax :

list.extend(iterable)

j. range() : returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

Syntax :

range(start, stop ,step)

**Tuple :**

1. Tuples are written with round brackets.

2. Tuples are used to store multiple items in a single variable.

3. Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are [List](https://www.w3schools.com/python/python_lists.asp), [Set](https://www.w3schools.com/python/python_sets.asp), and [Dictionary](https://www.w3schools.com/python/python_dictionaries.asp), all with different qualities and usage.

4. A tuple is a collection which is ordered and **unchangeable**.

Example :

thistuple = ("apple", "banana", "cherry")

**Dictionary :**

1. Dictionaries are used to store data values in key:value pairs.

2. A dictionary is a collection which is ordered\*, changeable and do not allow duplicates.

3. Dictionaries are written with curly brackets, and have keys and values:

Functions that can be applied on a Dictionary :

a. update() : method inserts the specified items to the dictionary.

The specified items can be a dictionary, or an iterable object with key value pairs.

Syntax :

dictionary.update(iterable)

b. clear() : Removes all the elements from the dictionary

Syntax :

car = {  
  "brand": "Ford",  
  "model": "Mustang",  
  "year": 1964  
}  
  
car.clear()

c. copy() : Returns a copy of the dictionary

Syntax :

car = {  
  "brand": "Ford",  
  "model": "Mustang",  
  "year": 1964  
}  
  
x = car.copy()

d. keys() : Returns a list containing the dictionary's keys

Syntax :

x = car.keys()

e. values() : Returns a list of all the values in the dictionary

Syntax :

x = car.values()

**Set :**

1. A set is created by placing all the items (elements) inside curly braces {}, separated by comma, or by using the built-in set() function.

2. It can have any number of items and they may be of different types (integer, float, tuple, string etc.). But a set cannot have mutable elements like [lists](https://www.programiz.com/python-programming/list), sets or [dictionaries](https://www.programiz.com/python-programming/dictionary) as its elements.

my\_set = {1.0, "Hello", (1, 2, 3)}

**String :**

Strings in python are surrounded by either single quotation marks, or double quotation marks.

Functions :

a. isAlpha() : The isalpha() method returns True if all the characters are alphabet letters (a-z).

Syntax :

string.isalpha()

b. isDigit() : method returns True if all the characters are digits, otherwise False

Syntax :

string.isdigit()

c. upper() : method returns a string where all characters are in upper case.

Syntax :

String.upper()

d. lower() : method returns a string where all characters are in lower case.

Syntax :

String.lower()

e. split() : method splits a string into a list.

Syntax :

string.split(separator, maxsplit)

f. strip() : method removes any leading (spaces at the beginning) and trailing (spaces at the end) characters (space is the default leading character to remove)

Syntax :

string.strip(characters)