Chapter 1 – Introduction to Python

Introduction

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by Python Software Foundation. It was Designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is called general-purpose, high level, object- oriented programming language as it is used in almost every domain we can think of as mentioned below:

- Web Development
- Game Development
- Software Development
- AI &ML
- Data Analytics Applications

Features of Python

1. ***Open Source:*** We can use Python software without any license and it is freeware. Its source code is open, so that we can we can customize based on our requirement.

2. ***Simple and easy to learn:*** Python is a simple programming language. When we read Python program, we can feel like reading English statements. The syntaxes are very simple and only 30+ keywords are available. When compared with other languages, we can write programs with very a smaller number of lines. Hence more readability and simplicity. We can reduce development and cost of the project.

3. ***High Level Programming language:*** Python is high level programming language and hence it is programmer friendly language. Being a programmer, we are not required to concentrate low level activities like memory management and security etc.

4. ***Platform dependency:*** Once we write a Python program, it can run on any platform without rewriting once again. Internally PVM is responsible to convert into machine understandable form.

5. ***Dynamically Typed:*** In Python we are not required to declare type for variables. Whenever we are assigning the value, based on value, type will be allocated automatically. Hence Python is considered as dynamically typed language

6. ***Extensible:*** We can use other language programs in Python.

7. ***Embedded:*** We can use Python programs in any other language programs. i.e we can embed Python programs anywhere.

8. ***Interpreter:*** We are not required to compile Python programs explicitly. Internally Python interpreter will take care that compilation.

Identifiers

A name in Python program is called identifier. It can be class name or function name or module name or variable name.

Ex: a=10

Rules to define identifiers in Python:

- Alphabet Symbols (Either Upper case OR Lower case)
- If Identifiers start with Underscore (_) then it indicates it is private.
- Identifier should not start with Digits.
- Identifiers are case sensitive.
- We cannot use reserved words as identifiers.
- There is no length limit for Python identifiers. But not recommended to use too lengthy identifiers.
- Dollar ($) Symbol is not allowed in Python

Keyword or Reserved Words

In Python some words are reserved to represent some meaning or functionality. Such type of words is called Reserved words.

Keywords in Python				
False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Note:

- All Reserved words in Python contain only alphabet symbols.
- Except the following 3 reserved words, all words contain only lower-case alphabet symbols. True, False & None

Data Types

Data Type represent the type of data present inside a variable. In Python we are not required to specify the type explicitly. Based on value provided, the type will be assigned automatically. Hence Python is also called Dynamically Typed Language.

Python contains the following inbuilt data types

- ***Int data type:***
  We can use int data type to represent whole numbers (integral values) Eg: a=10
- ***Float data type:***
  We can use float data type to represent floating point values (decimal values) Eg: f=1.23
- ***Complex data Type:***
  A complex number is of the form a+bj – where, a is the real part and b is the imaginary part. a and b contain integers or floating-point values. Complex data type has some inbuilt attributes to retrieve the real part and imaginary part
- ***Str data type:***
  str represents String data type. A String is a sequence of characters enclosed within single quotes or double quotes. (s1='durga', s1="durga")

Operators

Operator is a symbol that performs certain operations.

Python provides the following set of operators

1. ***Arithmetic Operators:***
   - + ➔ Addition
   - - ➔ Subtraction
   - * ➔ Multiplication
   - / ➔ Division operator
   - % ➔ Modulus operator
   - // ➔ Floor Division operator
   - ** ➔ Exponent operator or power operator

2. ***Relational Operators:***

   >, >= ,< ,<=

3. ***Logical Operators:***

   and, or, not

   We can apply logical operator for all data types.

4. ***Bitwise Operators:***

   These operators are applicable only for int and Boolean types.

   - & ➔ If both bits are 1 then only result is 1 otherwise result is 0
   - | ➔ If at least one bit is 1 then result is 1 otherwise result is 0
   - ^ ➔ If bits are different then only result is 1 otherwise result is 0
   - ~ ➔ Bitwise complement operator
   - <<➔ Bitwise Left shift
   - >>➔ Bitwise Right Shift

5. ***Assignment Operators:***

   We can use assignment operator to assign value to the variable.

6. ***Ternary Operator:***

   Syntax: x = first Value if condition else second Value

   If condition is True then first Value will be considered else second Value will be considered

7. ***Special operators:***

   Python defines the following 2 special operators (i.e., Identity Operators, Membership operators)

   - **Identity Operators is** and **is not** are the identity operators both are used to check if two values are located on the same part of the memory. Two variables that are equal do not imply that they are identical.
   - **Membership Operators in** and **not in** are the membership operators; used to test whether a value or variable is in a sequence.

8. **Operator Precedence:**

   If multiple operators present then which operator will be evaluated first is decided by operator precedence.

   - ( ) ➔ Parenthesis
   - ** ➔ Exponential
   - * ➔ Multiplication
   - / ➔ Division
   - + ➔ Addition
   - - ➔ Subtraction

Type Casting

We can convert one type value to another type. This conversion is called Typecasting.

The following are various inbuilt functions for type casting.

1. **int():** We can use this function to convert values from other types to int
2. **float():** We can use float() function to convert other type values to float type
3. **complex():** We can use this method to convert other type values to complex type
4. **bool():** We can use this function to convert other type values to bool type.
5. **str():** We can use this method to convert other type values to str type.