

1) What are the data types in Python? Explain.

Every value in Python has a data-type. Since everything is an object in Python Programming, data types are usually classes and variables are instance of these classes.

There are various data types in Python. Five of the important datatypes are listed below.

a) Numeric:

A numeric value is any representation of data which has a numeric value. Python identifies 3 types of numbers.

- i) integer: It includes Positive or Negative whole numbers.
- ii) Float: It includes any real number with a floating point representation in which a fractional component.
- iii) complex number: It includes the combination of a real number and imaginary component represented as $x + jy$, where x and y are real numbers and value of j is -1 .

b) Boolean:

Data with one of two built-in values True or False. In this 'T' and 'F' are of upper case. true and false are not valid and Python will throw an error for them.

c) sequence Type:

A sequence is an ordered collection of similar or different data types. Python has the following built-in sequence data types.

- i) string: A string value is a collection of one or more characters put in single, double or triple quotes.
- ii) list: A list object is an ordered collection of one or more data items, not necessarily of the same type, put in square brackets.
- iii) tuple: A tuple object is an ordered collection of one or more data items, not necessarily of the same type, put in parenthesis.

d) Dictionary

A dictionary object is an unordered collection of data in a key-value pair form. A collection of such pairs is enclosed in curly brackets.

EX: {1: "steve", 2: "Bill", 3: "Ram", 4: "Farah"}.

2) Briefly explain "History of Python".

A Python is a interpreted, high level, general purpose programming language. Created by "Guido van Rossum" and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant white space. Its language constructs and object oriented approach aim to help programmers write clear, logical code for small and large scale projects.

Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including structured, object oriented and functional programming.

Python is often described as a "batteries included" language due to its comprehensive standard library. Python was conceived in the late 1980's as a successor to the ABC language.

Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting.

Python 3.0, released in 2008, was a major version of the language which is not completely backward-compatible and many of the Python 2 codes doesn't even unmodified on Python 3.

The Python 2 language was officially discontinued in 2020 and Python 2.7.18 is the last Python 2.7 released and therefore, the last Python 2 released.

No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.5.x and later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation.

A non-profit organization, the Python software foundation, manages and directs the resources for Python and CPython developments.

②

3) Explain all the operators in Python.

operators are special symbols in Python that carry out arithmetic or logical computations. The value that operator operates on is called operand.

a) Arithmetic operators:

Arithmetic operators are used to perform mathematical operations like +, -, *, /, etc.

operator	meaning	Example
+	add 2 operands (or) unary Plus	$x+y$
-	subtracts right operand from the left (or) unary minus	$x-y$
*	multiplies 2 operands	$x*y$
/	divides left operand by the right one	x/y
% (modulus)	gives the remainder of the division of left operand by right	$x\%y$
// (floor division)	Division that results into whole number adjusted to the left in number line	$x//y$
**	left operand raised to the power of right operand	$x**y$

b) comparison operators :
comparison operators are used to compare values . It returns either True or false according to the condition .

operator	meaning	Example
$>$	greater than - True if left operand is greater than the right	$x > y$
$<$	less than - True if left operand is less than the right	$x < y$
$==$ (Equal to)	True if both operands are equal	$x == y$
$!=$ (Not equal to)	True if operands are not equal	$x != y$
$>=$ (greater than or equal to)	True if left operand is greater than or equal to the right	$x >= y$
$<=$ (less than or equal to)	True if left operand is less than or equal to the right	$x <= y$

c) logical operators :

operator

meaning

Example

AND

True if both the operands are true

x and y

OR
AND

True if either of the
operands is
true

x or y

(3)

NOT

True if operand
is false

not x

d) Bit-wise operators:

Bit wise operators act on operands as if they were strings of binary digits. They operate bit by bit, hence the name. For example, 2 is 10 in binary and 7 is 111.

operators	meaning	Example
$\&$	Bitwise AND	$x \& y = 0$
$ $	Bitwise OR	$x y = 14$
\sim	Bitwise NOT	$\sim x = -1$
\wedge	Bitwise XOR	$x \wedge y = 14$
\gg	Bitwise right shift	$x \gg 2 = 2$
\ll	Bitwise left shift	$x \ll 2 = 40$

e) Assignment operators:

Assignment operators are used in Python to assign values to variables. $a = 5$ is a simple assignment operator that assigns value 5 on the right to the variable on the left. There are various compound operators in Python like $a += 5$ that adds to the variable and later assigns the same. It is equivalent to $a = a + 5$.

operator

Example

Equivalent to

$=$
 $x +=$

$x = 5$
 $x += 5$

$x = 5$
 $x = x + 5$

$- =$

$x -= 5$

$x = x - 5$

$* =$

$x /= 5$

$x = x / 5$

$/ =$

% =	x% = 5	x = x%5
// =	x// = 5	x = x//5
** =	x** = 5	x = x**5
+ =	x+ = 5	x = x+5
- =	x- = 5	x = x-5
* =	x* = 5	x = x*5
>> =	x>> = 5	x = x>>5
<< =	x<< = 5	x = x<<5

f) special operators :

~~is and is not are the~~
Python language offers some special types of operators like the identity operators or the membership operators. They are special operators.

g) identity operators :

~~is and is not are the~~
identity operators in Python. They are located at the same part of the memory. Two variables that are equal does not imply that they are identical.

operator	meaning	example
is	True if the operands are identical	x is true
is not	True if the operands are not identical	x is not true

h) membership operators :
'in' and 'not in' are the membership operators in Python. They are used to test whether a value or variable is found in a sequence.

operator	meaning	example
in	True if value/variable is found in the sequence	5 in x

4) Explain features of Python.

Python provides lots of features that are listed below.

1) Easy to learn and use
Python is high level programming language. It is easy to learn and use. It is developer friendly and high level programming language.

2) Expressive language
Python language is more expressive that means it is more understandable and readable.

3) Interpreted language:
Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

4) Cross Platform language.
Python can run equally on different platforms such as windows, LINUX, UNIX and Macintosh etc. so we can say that Python is a portable language.

5) Free and open source:
Python language is freely available at official web address. The source code is also available. Therefore it is open source.

6) Object oriented language:
Python supports object oriented language and concepts of classes, objects and encapsulation etc. come into existence.

7) Extensible
Python is an extensible language. It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code.

8) Large standard library:
Python has a large and broad library and provides rich set of module and functions for rapid application development.

9) GUI programming support
Graphical user interfaces can be developed using Python.

10) Integrated:
It can be easily integrated with the languages like C, C++ and JAVA etc.,

5) Justify why Python is interactive interpreted language.
unlike C/C++ etc., Python is an interpreted object oriented language. By interpreted it is meant that each time a Program is run, the interpreter checks through the code for errors and then interprets the instructions into machine readable byte code.

An interpreter is a translator in computer language which translates the given code line by line in machine readable byte codes. And if any error is encountered it stops the translation until the error is fixed.

unlike C language, which is a compiled Programming language the compiler translates the whole code in one-go rather than line by line. This is the reason why in C language all the errors are listed during compilation only. When a Python statement is entered, and is followed by the return key, if appropriate, the result will be printed on the screen immediately in the next line. This is particularly advantageous in the debugging process.

In interactive mode of operation, Python is used in a similar way as the UNIX command line or the terminal. Interactive Python is very much helpful for the debugging purpose.