

1. What the data types in python? Explain.

Every value in python has a datatype. Since everything is an object in python programming, data types are actually classes and variables are object of these classes.

They are defined as int, float and complex classes in python. We can use the `type()` function to know which class a variable or a value belongs to. Similarly, the `isinstance()` function is used to check if an object belongs to a particular class.

```
a=5
```

```
print(a, " is of type", type(a))
```

```
a=2.0
```

```
print(a, " is of type", type(a))
```

```
a=1+2j
```

```
print(a, " is complex number?", isinstance(1+2j, complex))
```

O/p

5 is of type <class 'int'>

2.0 is of type <class 'float'>

(1+2j) is complex number? True.

Integers can be of any length, it is only limited by the memory available.

A floating-point number is accurate up to 15 decimal places.

Integer and floating points are separated by decimal points. 1 is an integer, 1.0 is a floating point number.

Complex numbers are written in the form,  $x+yj$ , where  $x$  is the real part and  $y$  is the imaginary part.



a = 123456

a

123456

b = 0.1234567890123456789

b

0.12345678901234568

c = 1+2j

c

(1+2j)

2) Briefly explain history of python.

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 Netherland. 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 Amecoba 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.



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 reability and advanced developer productivity. When it was released it had more than enough capability to provide classes with inheritance, several core datatypes exception handling and functions.

### 3) Explain all the operators in python.

Arithmetic operators: These are used to perform mathematical operations like addition, subtraction, multiplication and division.

operator	Description	Syntax
+	adds two operands	$x+y$
-	subtracts two operands	$x-y$
*	multiplies two operands	$x*y$
/	(float) divides the first operand by the second	$x/y$
//	divides the first operand by second (floor)	$x//y$
%	Modulus: returns the remainder when first operand is divided by second	$x\%y$
**	power: Returns first raised to power second	$x**y$



2. Relational operators: Relational operator compares the values. It either returns True or false according to the condition.

<u>Operator</u>	<u>Description</u>	<u>Syntax</u>
>	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 right	$x >= y$
<=	Less than or equal to: True if left operand is less than or equal to the right.	$x <= y$

3. Logical operators: They perform logical AND, logical OR and Logical NOT operations.



operatorDescriptionSyntax

and

Logical AND: True if both  
the operands are true. $x$  and  $y$ 

or

Logical OR: True if either  
of the operands is true $x$  or  $y$ 

not

Logical NOT: True if operand  
is falsenot  $x$ 

4. Bitwise operators: It acts on bits and performs bit by bit operation.

operatorDescriptionSyntax

&amp;

Bitwise AND

 $x \& y$ 

|

Bitwise OR

 $x | y$ 

~

Bitwise NOT

~ $x$ 

^

Bitwise XOR

 $x \wedge y$ 

&gt;&gt;

Bitwise right shift

 $x >>$ 

&lt;&lt;

Bitwise left shift

 $x << x$ 

5. Special operators: There are some special type of operators like -

Identity operator - 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 does not imply that they are identical.

is True if the operands are identical

is not True if the operands are not identical.



4. Explain the features of python.

Simple - It is simple to operate in programming.

Easy to learn - It is easy compared to C, C++, JavaScript, Java etc.

Free and open source - Freely available at official website.

High-level language - we don't need to remember system architecture.

Python is a beginner's language.

Portable - Can be able to run code on other platforms.

Interactive - we can integrate python with C, C++ etc.

Interpreted - source code converted into immediate form called bytecode.

Object Oriented - It supports object oriented & classes, objects encapsulation.

Extensible - can also compile that code in C/C++ language.

Embeddable - It is embedded one on another code.

Extensive libraries - There are many libraries like regular expressions.

Databases - It consists of databases in python.

GUI Programming - Using a module like PyQt5, PyQt4, wxPython or Tk.

Scalable

5) Justify why python is interactive interpreted language.

Unlike C/C++ etc, Python is an interpreted object oriented programming 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 bytecode.



An interpreter is a translator in computer's language which translates the given code line-by-line in machine readable bytecodes. 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.