1. What are the data types in python ? Explain

Data types are classification or categorization of data.

a. Numeric

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

→ Integer : Positive or negative whole numbers

→ Float : Any real numbers with a floating point representation.

→ Complex number : A number with a real and imaginary component.
for example: 2+3i

b. Boolean

Data with one of 2 built in values True or False.

c. String

A string value is a collection of one or more characters put in single, double quotes.

d. List

A list object is an ordered collection of one or more data items which are mutable, not necessarily same type, but should be put in square brackets.

e. Tuple

A Tuple object is an ordered collection of one or

more data items, not neccesarily same type, put in parenthesis.
Data items are immutable.

f. Dictionary:
A dictionary object is unordered collection of data in a key: value pair form.
A collection of such pairs is enclosed in cusly brackets.
For Eg: {1: "steve" 2: "Bill"}

2. Briefly Explain history of Python.

Ans → Python laid its foundation in late 1980s
→ The implementation of Python was started in December 1989 by Guido Van Rossum at CWI in Netherland.
→ In February 1991, van Rossum published the code (labeled version 0.9.0) to alt.source.
→ In 1994, Python 1.0 was released with new features like: lambda, map, filter, and reduce.
→ Python 2.0 added new features like: list comprehensions, garbage collection system.
→ On December 3, 2008, Python 3.0 was released. It was designed to rectify fundamental flaw of language.
→ Python is influenced by following programming language:
- ABC language
- Modula-3

3. Explain all the Operators in python.

Ans Types of Operators

a. Arithematic Operators

→ Addition (+)

Eg: 3+4 # Addition btw 2 data members

Output - 7

→ Subtraction (-) - Subtracts the value on right from one on left.

Eg: 5-4

output - 1

→ Multiplication (*)

Multiplies the values on either side of operator

Eg >>> 3*4

Output - 12

→ Division (/)

Divides the value of left by one on right.

Results in floating point value

Eg 3/4

Output - 0.75

→ Modulus (%)

Divides the left & right operand giving remainder

Eg 4%2

Output → 0

→ Exponential (**)

Raises first member to power of second

Eg 2**4

Output → 16

→ floor division (//)

Divides and returns the integer value of quotient. It dumps digits ~~of dum~~ after the decimal.

Eg 10//3

Output → 3

b. Relational Operators

→ Less than (<)

This operator checks if the value on left of operator is lesser than the one on right.

Eg: 3<4

Output: True

→ Greater than (>)

It checks if value on left is greater than that one on the right of operator.

Eg: 4>5

Output: False

→ Less than or equal to (<=)

It checks if the value of left of operator is less than or equal to that on right.

Eg: 5 <= 10

Output: True

→ Greater than or equal to (>=)

It checks if the value of left of operator is greater than or equal to that on right.

Eg: 5 >= 10

Output: False

→ Equal (==)

This checks if value on left side of operator is equal to one on right.

Eg: 3 == 3

Output - True

→ Not Equal (!=)

This checks if value on left side of operator is not equal to one on right.

Eg: 3 != 4

Output - True

C. Assignment Operator

→ Assign (=)

Assigns value to expression on left.

Eg a = 7 # Here 7 is value assigned to a.

→ Add & Assign (+=)

Adds value on either side and assigns it is to the expression on left.

a += 10 is same as a = a+10

Eg: a += 2.

→ Subtract & Assign (-=)

Subtracts value on either side and assigns it to expression on left

Eg: a -= 10

→ Multiply & Assign (*=)

Multiply value on either side & assigns it to the expression on left

Eg: a *= 2

→ Divide & Assign (/=)

Divides value on left by the one on right. Then it assignes it to expression on left.

Eg: a /= 7

→ Modulus and Assign (%=)

Perform modulus on values on either side and assignes it to left.

Eg: a %= 3

→ Exponent and Assign (**=)

Perform exponentiation on the values on either side and then assigns it to expression on left.

Eg: a **= 5

→ Floor Divide and Assign (//=)

Perform floor division on values on either side. Then assigns it to expression on left.

Eg: a //= 3

d. Logical Operators

→ and

If the conditions on both the sides of operator are true, then the whole expression is true.

Eg: a=777 and 2>-1

Output : False

→ or

The expression is false only if both statement around the operator are false. Otherwise, it is true

Eg: a = 777 or 2>-1

Output: True

→ not

converts True to false and false to True.

Eg: a=not(0)

print(a)

Output : True

d. Bitwise Operator

Operator	Symbol	
Binary AND	&	bit by bit AND Operation on 2 values Eg: 2&3 Output : 2
Binary OR	|	2|3 Output: 3
Binary XOR (^)	^	2^3 Output : 1
Binary One's Complement (~)	~	~-3 Output: 2
Binary Left Shift	<<	2<<2 Output: 8
Binary Right Shift	>>	3>>2 Output: 1

4. Explain the features of Python.

Ans Features

a. Easy to code: Python is high level programming language. It is very easy to code in python language and anybody can learn python.

b. Free and Open Source: Python is freely available at official website. Since it is open source, this means that source code is also available to public.

c. Object-Oriented Language:
Python supports object oriented language and concepts of classes, object encapsulation etc.

d. GUI Programming Support:
Graphical Users Interface can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python

e. High-Level language:
Python is a high-level language, we do not need to remember the system architecture, nor do we need to manage memory.

f. Extensible: We can write our python code C or C++ language and we can compile that code in C or C++ language

g. Portable: It is write once run anywhere program, only any operating system and code need not be changed.

5. Justify why python is interactive interpreted language.

Ans Python is an interpreted language because python code is executed line by line at a time unlike other language 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. It is also an integrated language because we can easily integrated python with other languages like c, c++.