

- Ques 2
- Q) what are the datatypes in python? Explain
- A) The data types defined in the python are;
1. Numbers
  2. string
  3. List
  4. Tuple
  5. Dictionary

Numbers: Number store numeric value python supports

4. types of numeric data

1. int (signed integers like 10, 2, 29 etc)
2. long (long integers used for a higher range of values like 19080908002 etc.)
3. float (. it is used to store floating point numbers like 1.9, 9.9002 etc)
4. complex (complex numbers like 2+4j)

string: The string can be defined as the sequence of characters represented in the quotation marks. In python we use single, double or triple quotes to define a string.

Ex: "hello world"

List: List are similar to arrays in c. However, the list contain data of different types. The items

Items stored in the list are separated with a comma and enclosed with in the square brackets [ ].

we can use slice[i] operators to access the data of the list

Ex: `l = [1, "hi", "Python", 2]`

`Print(l[3:])`

O/P [ ]  
Tuple: A tuple is similar to the list in many ways like lists, Tuple also contain the collection of the items of different data types. The items of tuples are separated with a comma(,) and enclosed in the parenthesis( )

Ex: `t = ("hi", "Python", 2)`

`Print(t[1:])`

O/P: ('Python', 2)

Dictionary: Dictionary is an ordered set of a key-value pair of items. It is like an associative array. Key can hold any primitive data type whereas value is an arbitrary python object.

Ex: `d = {1: "jimmy", 2: "Alex", 3: "John"}`

`Print("1st name is" + d[1])`

O/P: 1st name is jimmy

2) Briefly explain the history of python!

A) Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989; Van Rossum, shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's Benevolent Dictator for Life, a title the Python community bestowed upon him to reflect his long-term commitment to the project's chief decision-maker. He now shares his leadership as a member of a five person steering council.

In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member "steering council" to lead the project.

Python 2.0 was released on 16 October 2000 with many major new features, including a cycle detecting garbage collector and support for Unicode.

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its

major features were backported to Python 2.6.x and 2.7.x version series. Releases of Python 3 include the utility, which automates the translation of Python 2 code of Python 3.

Python 2.7's end-of-life date was initially set at 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-portable to Python 3.

③ Explain the operators in Python? or list

(i) Arithmetic operators:

These are used to perform arithmetic operations with two operands. It includes addition (+), subtraction (-), multiplication (\*), divide (/), remainder (%), floor division (//) and exponent (\*\*).

(ii) Comparison operator:

These are used to compare the value of the two operands and returns boolean True or False accordingly.

The comparison operators are:

= =, !=, <=, >=, >, <

(iii) Assignment operators:

These are used to assign the value of the right expression to the left operand.

Set of Assignment operators:

=, +=, -=, \*=, /=, \*\*=, %=

iv) Bitwise operators:  
The bitwise operators perform 'bit by bit' operation on the values of two operands.

Binary and (&)      Binary xor (^)      Left shift (ll)  
Binary or (|)      Negation (~)      Right shift (gg)

v) Logical operators:  
These are used primarily in the expression evaluation to make a decision. Python supports and, or, not logical operators.

vi) Membership operators:

They are used to check the membership of value inside a python. If the value is present in data structure, then the resulting value is true otherwise it returns false.

\* in and not in are membership operators

vii) Identity operators:

is - It is evaluated to be true if the reference present at both side point to the same object.

is not - It is evaluated to be true if the reference present at both side do not point to the same object.

v) Explain the features of python

1) Easy to learn and use.

Python is easy to learn and use. It is developer friendly and high-level programming language.

2) Expressive language.

It means that is more understandable and

readable in programming. Read more about

3) Interpreted language.

Interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

4) Cross-platform language.

It can run equally on different platforms such as windows, linux, unix etc. We can say Python is a portable language.

5) Free and open language.

It is freely available at official web address. Source code is also available. It is open source.

6) Object-oriented language.

It supports object-oriented language and concepts of classes and objects come into existence.

7) Extensible  
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(good) standard library  
Python has 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 languages like c,c++,java etc.

⑤ Justify why python is interactive interpreted language?

Python is an interacted interpreted language because

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 byte code. We can integrate it.

We can easily integrate

Python with other languages like C, C++ 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 byte code.

Bytecode is a low-level language

which is used by Python interpreter to execute the program

Bytecode is a low-level language which is used by Python interpreter to execute the program

Bytecode is a low-level language which is used by Python interpreter to execute the program

Bytecode is a low-level language which is used by Python interpreter to execute the program

Bytecode is a low-level language which is used by Python interpreter to execute the program

Bytecode is a low-level language which is used by Python interpreter to execute the program

Bytecode is a low-level language which is used by Python interpreter to execute the program

Bytecode is a low-level language which is used by Python interpreter to execute the program

Bytecode is a low-level language which is used by Python interpreter to execute the program

Bytecode is a low-level language which is used by Python interpreter to execute the program

Bytecode is a low-level language which is used by Python interpreter to execute the program