

Q) What are the data types in python explain?

A. (i) Numbers -

Number data types store numeric values. Number objects are created when you assign a value to them. Ex - $a = 10$, $b = 11.05$

(ii) Strings -

Strings are identified as a contiguous set of characters represented in the quotation marks. It allows either a pair of single or double quotes.

Ex - $x = \text{"orange"}$.

(iii) Lists -

Lists are the most versatile of python's compound datatypes. A list contains items separated by commas, enclosed with square bracket. Ex - $y = (1, 2, 3, 4)$

(iv) Tuples -

A tuple is another sequence datatype is similar to list. A tuple consists of number of values separated by commas, enclosed within parenthesis.

Ex - $z = (1, 2, 3, 4, 5, 6, 7, 8)$

(v) Dictionaries -

python's dictionaries are kind of hash table type. They work like associate array or hashes found in perl. It can be any arbitrary python object. These are enclosed within curly braces.

Ex - $\{ \text{"name": "Tanu", age: 18} \}$

Q2) Explain briefly history of python?

Python was conceived in the late 1980s

by Guido van Rossum at CWI in Netherlands.

- python was named for the BBC TV show Monty Python's flying circus.

- python 2.0 was released on October 16, 2000 with new features including a cycle detecting garbage and collector for memory management support for unicode.

python 3.0, backward incompatible release, released on December 2, 2008 after a long period of testing.

Many of major features have been backported and unsupported, python 2.6 and 2.7

- With its release it used a lot fewer codes to express the concepts, when we compares it with java, c, c++.

5) Explain all the operators in python -

A operators are the symbols that performs mathematical operations between two operands.

(i) Arithmetic operators -

<u>operator</u>	<u>Description</u>	<u>syntax</u>
+	Addition	$a+b$
-	Subtraction	$a-b$
*	multiplication	$a*b$
/	Division	a/b
//	floor division	$a//b$
%	modulus	$a\%b$
**	power	$a**b$

(ii) Relational operators -

<u>operator</u>	<u>Descriptions</u>	<u>syntax</u>
>	Greater than	$a > b$
<	less than	$a < b$
>=	Greater than / equal to	$a >= b$
<=	less than / equal to	$a <= b$
=	equal to	$a = b$
!=	Not equal to	$a != b$

(iii) logical operator -

<u>operator</u>	<u>Description</u>	<u>syntax</u>
and	logical AND OR NOT	$a \text{ and } b$
or		$a \text{ or } b$
not		$\text{not } a$

④ Bitwise operators

operator	Description	Syntax
$\&$	Bitwise AND	$a \& b$
$ $	Bitwise OR	$a b$
\sim	Bitwise NOT	$\sim a$
\wedge	Bitwise XOR	$a \wedge b$
\gg	Bitwise right shift	$a \gg$
\ll	Bitwise left shift	$a \ll$

⑤ Special operators

is - True if operands are identical
is not - True if operands are not identical

⑥ Membership operator -

in - True if value is found in sequence

not in - True if value is not found in sequence

⑦ Explicit features of python

A. (i) Easy to learn and use.

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

(ii) Expressive language

python is expressive language since it is more understandable and readable.

(iii) Interpreted language

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

(iv) Cross-platform language

python can run equally on different platforms like windows, linux, unix etc. so we can say that it is portable language.

(v) Free and open source

This language is freely available at official web addresses. The source-code is also available so, it is open source.

(vi) Object oriented language.

This language supports the concept of classes and objects come into existence.

(vii) Extensible

It implies that other language such as c/c++ can be used to compile the code. Thus it can be further used in our python code.

(viii) Large standard library -

python has a large and broad library and provides rich set of module and functions for rapid application development.

(ix) GUI programming support -

Graphical user interface can be developed using python.

(x) Interpreted

It can be easily integrated with language like c/c++, java.

⑤ Justify why python is interactive interpreted language?

A. python is interpreted language since interpreter executes the code line by line at a time. This makes debugging easier. And converts machine readable bytecodes. And if any error is encountered it stops. The translation until the error is fixed.

python is interactive. When a python statement is entered and is followed by the return key, if appropriate the result will be printed on screen immediately in next line. It is used in a similar way as the unix command line or the terminal and is helpful for the debugging purpose.