

1. What are the datatypes in python? explain

= Data types are the classification or categorization of data items. Data types represent a kind of value which determines what operations can be performed on that data.

Python has following standard or builtin data types

Numeric:

A numeric value is any representation of data which has a numeric value.

Integer, float and complex comes under numeric.

Boolean:

Data with one of two built-in values True or False.

Sequence Type:

A sequence is an ordered collection of similar or dissimilar datatypes.

String: A string value is a collection of characters.

List: A list object is an ordered collection of one or more data items put in square bracket.

Tuple: A Tuple object is an ordered collection of one or more data items put in parentheses.

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.

eg: {1: "Steve", 2: "Shanu", 3: "shan", 4: "piyu"}

2 Briefly explain history of python.

= In the late 1980s, history was about to be written. It was that time when working of python started. Soon after that, Guido Van Rossum began doing its application based work in Decm of 1989 by at Centrum Wiskunde and informatica which is situated in Netherland.

The inspiration of the name came from BBC's TV show 'Monty python's' Flying circus' he was big fan of that show hence he named it as python.

The language was released in 1991, when it was released it used a lot fewer codes to express the concept, when we compare it with java, c++ and c. Its design philosophy was quite good too. Its main objective is to provide reliability and advanced developer productivity.

3. Explain all the operators in python

= Operators are special symbols that represent computations like addition and multiplication.

* The operators +, -, *, /, and ** perform addition, subtraction, multiplication, division and exponentiation

Python has the following groups

- * Arithmetic operators
- * Assignment operators
- * Logical operators
- * Comparison operators
- * Identity operators
- * Membership operators
- * Bitwise operators

Assignment operators:

operator	example
=	$x = 5$
+=	$x += 3$
-=	$x -= 3$
*=	$x *= 3$
/=	$x /= 3$
%=	$x \% = 3$
**=	$x ** = 3$
&=	$x \& = 3$
=	$x = 3$

Comparison operators:

operator	example
==	$x == y$
!=	$x != y$
>	$x > y$
<	$x < y$
>=	$x >= y$
<=	$x <= y$

Logical operators:

operator and	example
	$x < 5 \text{ and } x < 10$
or	$x < 5 \text{ or } x < 4$
not	$\text{not}(x < 5 \text{ and } x < 10)$

Python identity operators:

operator	example
<code>is</code>	<code>x is y</code>
<code>is not</code>	<code>x is not y</code>

membership operators

operator	example
<code>in</code>	<code>x in y</code>
<code>not in</code>	<code>x not in y</code>

Bitwise operators:

operator	example
<code>&</code>	<code>x & y</code>
<code> </code>	<code>x y</code>
<code>^</code>	<code>x ^ y</code>
<code>~</code>	<code>~x</code>
<code><<</code>	<code>x << 1</code>
<code>>></code>	<code>x >> 1</code>

4. What are the features of python

= * Simple

- * Easy to learn
- * Free and open source
- * High level language
- * Python is Beginner's language
- * Portable
- * Interactive
- * Interpreted
- * Object oriented
- * Extensible
- * Embeddable
- * Extensive Libraries
- * Databases
- * GUI programming
- * 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 a language which is 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.

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 the next line. This is particularly advantages in the debugging process in interactive mode of operation. Python is used in similar way as Unix command line.

Interactive Python is very much helpful for the debugging purpose. It returns the prompt on the corresponding output of the statement if appropriate and returns error for incorrect statements.