

Assignment - 2

1. What the data types of python? Explain

⇒ The Standard type of python:

- * Numeric
- * Sequence type
- * Boolean
- * Set
- * dictionary

→ Numeric :- In python numeric datatype represent the data which has numeric value. Numeric value can be integer, floating number (or) even complex numbers.

These are divided into

- Integers
- Float
- Complex numbers

* Integer :- It is represented by int class. It contains positive (or) negative whole numbers.

* Float :- It is represented by float class. It is a real number with floating point representation. It is specified by decimal point.

* Complex numbers :- Complex number is represented by complex class. It is specified as (real part) + (imaginary part) i.

→ Sequence type :- Sequence is the ordered collection of similar (or) different data types. Sequences allows to store multiple values in an organized and efficient fashion.

There are several sequence types in python :-

- String
- List
- Tuple.

⇒ String :- Strings are arrays of bytes representing unicode characters. It is represented by str class.

⇒ List :- Lists are just like the arrays declared in other languages. It is represented by list class.

⇒ Tuple :- Tuples are created by placing sequence of values separated by 'comma' with (or) without the use of parentheses for grouping of data sequence. It is a bit tricky. There must be a 'comma' to make it tuple.

⇒ Boolean :- Data type with one of the two built-in values, True (or) False. In python True and False should be capital 'T' and 'F'. Otherwise it shows error, it is terminated as bool.

⇒ Set :- Set is an unordered collection of datatype that is iterable, mutable and has no duplicate elements. The major advantage of using a Set is as opposed to a list, is that it has highly optimized method for checking whether specific element is contained in the set.

⇒ Dictionary :- Dictionary can be created by placing a sequence of element within curly {} braces, separated by 'comma'. Dictionary holds a pair of values, one being the key and the other corresponding pair element being key : value. Whereas keys can't be repeated and must be immutable.

2. Briefly Explain history of python.

Python was conceived in the late 1980's by GUIDO VAN ROSSUM at centrum wiskunde & informatica (CWI) in the Netherlands as to the ABC language. (itself inspired by SETL) capable of exception handling and interfacing with the Amoeba operating system

→ Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting.

→ Python interpreters are available for many operating systems. A global community of programmers develops and maintain cpython, an open source reference implementation.

→ python is a multi-paradigm programming language object oriented programming and structured programming.

3. Explain all the operators in Python.

* Arithmetic operator : It is used to perform mathematical operations like addition, subtraction multiplication and division.

<u>operator</u>	<u>meaning</u>	<u>example</u>
+	adds 2 operands	$x+y$
-	subtracts 2 operands	$x-y$
*	multiplies 2 operands	$x*y$
/	divides 1st operand by second [float]	x/y
//	floor division	$x//y$
**	left operand raised to the power of right	$x**y$

Comparison operators : Comparison operators are used to compare values. It returns either 'True' (or) 'False' according to the condition.

<u>operator</u>	<u>meaning</u>	<u>Example</u>
>	Greater than	$X > Y$
<	Less than	$X < Y$
==	Equal to	$X == Y$
!=	Not Equal to	$X != Y$
>=	Greater than or Equal to	$X >= Y$
<=	Less than or Equal to	$X <= Y$

Logical operators : Logical operators are the and, or, not operators.

<u>operator</u>	<u>meaning</u>	<u>Example</u>
and	True if both the operands are true	$X \text{ and } Y$
or	True if either of the operands is true	$X \text{ or } Y$
not	True if operand is false (complements the operand)	not X

Bitwise operators : Bitwise operators act on operands as if they were strings of binary digits. They operate bit by bit.

<u>operator</u>	<u>meaning</u>	<u>Example</u>
&	Bitwise AND	$X \& Y$
	Bitwise OR	$X Y$
~	Bitwise Not	$\sim X$

\wedge	Bitwise XOR	$X \wedge Y$
\gg	Bitwise right Shift	$X \gg$
\ll	Bitwise Left Shift	$X \ll$

Assignment operators: Assignment operators are used in python to assign values to variable.

<u>operator</u>	<u>meaning</u>	<u>Example</u>
$+=$	Add AND	$X += Y$
$-=$	Subtract AND	$X = X + Y$ $X -= Y$
$*=$	multiply AND	$X = X - Y$ $X * = Y$
$/=$	divison AND	$X = X * Y$ $X /= Y$
$\%=$	modulus AND	$X = X / Y$ $X \% = Y$
$\ =$	floor AND	$X = X \% Y$ $X \ = Y$
$**=$	Exponent AND	$X \ = Y$ $X ** = Y$
$ =$	Bitwise OR	$X ** = Y$ $X = Y$
$\wedge=$	Bitwise XOR	$X = X Y$ $X \wedge = Y$ $X = X \wedge Y$

Special operators: is and is not are the identity operators in python. They are used to check if two values are located on the same part of the memory.

<u>operator</u>	<u>meaning</u>	<u>Example</u>
is	True if the operands are identical	$X \text{ is true}$

is not

True if the operands
are not identical

x is not
true

membership operator:

in and not in are the membership operators in Python.

<u>operator</u>	<u>meaning</u>	<u>Example</u>
in	True if value is found in the sequence	5 in x
not in	True if value is not found in sequence	5 not in x

4. Explain the features of python.

- * Easy to code
- * Free and open source
- * Object oriented language
- * Extensible
- * Large standard library
- * GUI programming support
- * Integrated and Interpreted language
- * Portable language
- * High level language
- * Dynamically Typed language.

5. Justify why python is interactive interpreted language.

- * python program runs directly from the source code.

* Python converts source code written by the programmer into intermediate language which is again translated into the native language/machine language that is executed so python is interpreted language.

* python processed at runtime by the interpreter. Program need to be compiled before its execution.