

## Assignment - 2

Amrutha kr

321910306057

1. What are the data types of Python? Explain.

A: The standard types of Python:

1] Numeric

2] Sequence type

3] Boolean

4] Set

5] Dictionary

1] Numeric: The Python numeric data type represents the data which has numeric value. Numeric value can be integer, floating number (or) even complex numbers. These are divided into

\* Integer

\* 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) by Complex class. It is specified as (real part) + (imaginary part)j.

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

There are several sequence types of 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 parenthesis for grouping of data sequence. It is a bit tricky. There must be 'comma' to make it tuple.

→ Boolean: Data types with one of the two built-in values, True (or) False. In Python, True and False should be Capital.

'r' 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 1980s by GUIDO VAN ROSSUM at Centrum Wiskunde & Informatica (CWI) in the Netherlands as to the ABC Language (itself) 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 a Python an open source reference implementation.

→ Python is a multi-paradigm programming language object oriented programming and Structured programming.

9] 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 1 <sup>st</sup> operand by second [float]	$x / y$
//	float 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	$x = y$
$\neq$	not equal to	$x \neq y$
$\geq$	greater than (or) equal to	$x \geq y$
$\leq$	less than (or) equal to	$x \leq y$

### 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	$x$ and $y$
or	True if Either of the operands is true	$x$ 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	$\sim x$
^	Bitwise XOR	$x \wedge y$
>>	Bitwise right shift	$x >>$
<<	Bitwise left shift	$x <<$

Assignment operators: Assignment operators are used in python to assign values to Variable

<u>operator</u>	<u>meaning</u>	<u>Example</u>
+	add AND	$x += y$ $x = x + y$
-	subtract AND	$x -= y$ $x = x - y$
*	multiply AND	$x *= y$ , $x = x * y$
/	division AND	$x /= y$ , $x = x / y$
%	modulus AND	$x \% = y$ $x = x \% y$
//	floor AND	$x // = y$ , $x // y$
**	Exponent AND	$x ** = y$ , $x = x ** y$
	Bitwise OR	$x  = y$ , $x = x   y$
^	Bitwise XOR	$x ^= y$ , $x = x \wedge y$

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	$x$ is true

is not True if the operands are not identical \* is not true

### membership operator:

in and not in are the membership operators in Python.

operator

meaning

Example

in

True if value is found in the sequence

5 in x

not in

True if value is not found in

5 not in x

sequence

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 an interpreted language.

- \* Python is processed at runtime by the interpreter. Program need to be compiled before its execution.