

# Assignment - 2

## Module - 1

1) What are the data types in python?

Explain.

python has the following data types built-in by default, in these categories:

Text Type : str

Numeric Type : int, float, complex

Sequence Type : list, range, tuple

Mapping Type : dict

Set Type : set, frozenset

Boolean Type : bool

Binary Types : bytes, bytearray, memoryview.

Standard data types:

A variable can hold different types of values. python provides various standard data types that define the storage method on each of them. The data types defined in python are given below.

① Number ② String ③ List ④ Tuple

⑤ Dictionary.

→ Numbers :- Number stores numeric values. python creates number objects when a number is assigned to a variable.

for example, a=3, b=5 #a and b are number objects.

python supports 4 types of numeric data.

1. int (signed integer like 10, 2, 29 etc.)

2. long (long integers used for higher range of values like 908090800L,

-0x1929245656L etc.)

3) Float (Float is used to store floating point numbers like 1.9, 9.902, 15.2 etc.)

4) Complex (Complex numbers like  $2.14j$ ,  $2.0 + 2.3j$ , etc.)

→ String :- The string can be defined as the sequence of characters represented in the quotation marks.

ii) In python, we can use single, double, or triple quotes to define a string.

iii) String handling in python is a straightforward task since there are various inbuilt functions and operators provided.

iv) In case of string handling, the operator  $+$  is used to concatenate two strings as the operation `"hello" + "python"` returns `"hellopython"`.

v) The operator  $*$  is known as repetition operator as the operation `"python" * 2` returns `"python python"`.

→ List

i) Lists are similar to array in C. However, the list can contain data of different types. The items stored in the list are separated with a comma (,) and enclosed within square brackets `[]`.

ii) We can use slice `[:]` operators to access the data of the list. The concatenation operator  $+$  and repetition operator  $*$  works with the list in the same way as they were working with the strings.

→ Tuple

i) A tuple is similar to the list in

many ways, like lists, tuples also contain the collection of the items of different data types. The items of the tuple are separated with comma (,) and enclosed in parentheses ( )

ii) A tuple is a read-only data structure as we can't modify the size and value of the items of a tuple.

→ Dictionary

Dictionary is an ordered set of a key-value pair of items. It is like an associative array or a hash table where each key stores a specific value. Key can hold any primitive data type whereas value is an arbitrary python object.

The items in the dictionary are separated with the comma and enclosed in the

Curly braces { }.

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 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.

→ Python is a widely used general-purpose, high-level programming language.

→ It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

③ Explain all the Operators in python.

① Arithmetic Operators: Used to perform mathematical operations like addition, subtraction, multiplication and division.

<u>Operator</u>	<u>Description</u>	<u>Syntax</u>
+	adds two Operands	$x + y$
-	Subtracts 2 Operands	$x - y$
*	Multiplies 2 Operands	$x * y$
/	divides 1 <sup>st</sup> Operand by the second [float]	$x / y$
//	divides 1 <sup>st</sup> Operand by the second [floor]	$x // y$
%	Modulus: returns the remainder when 1 <sup>st</sup> Operand is divided by the second	$x \% y$
**	power: Returns first raised to power second	$x ** y$

② Relational Operators: It compares the values. It either returns True or false according to the condition.

<u>Operator</u>	<u>Description</u>	<u>Syntax</u>
>	Greater than	$x > y$
==	Equal to	$x == y$
!=	Not equal	$x != y$
>=	Greater than or equal to	$x >= y$

(3) Logical Operators: It performs logical AND, logical OR and logical NOT Operations.

<u>Operator</u>	<u>Description</u>	<u>Syntax</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	$\text{not } x$

(4) Bitwise Operators: It acts on bits and performs bit by bit operation.

<u>Operator</u>	<u>Description</u>	<u>Syntax</u>
$\&$	Bitwise AND	$x \& y$
$ $	Bitwise OR	$x   y$
$\sim$	Bitwise NOT	$\sim x$
$\wedge$	Bitwise XOR	$x \wedge y$
$>>$	Bitwise right shift	$x >>$
	Left shift	$x <<$

(5) Assignment Operators: Used to assign values to the variables.

<u>Operator</u>	<u>Description</u>	<u>Syntax</u>
$=$	Assign value	$x = y$
$+=$	Add AND: Add right side Operand with left side Operand and then assign to left Operand	$a += b$ $a = a + b$
$-=$	Subtract AND	$a -= b$ $a = a - b$

$*$	Multiply AND	$\Rightarrow a * b$ $a = a * b$
$/$	DIVISION AND	$\Rightarrow a / b$ $a = a / b$
$\%$	Modulus AND	$\Rightarrow a \% b$ $a = a \% b$
$//$	Divide (floor) AND	$\Rightarrow a // b$ $a = a // b$
$**$	Exponent AND	$\Rightarrow a ** b$ $a = a ** b$
$\&$	AND on	$\Rightarrow a \& b$ $a = a \& b$
$ $	perform Bitwise OR	$\Rightarrow a   b$ $a = a   b$
$\wedge$	perform Bitwise XOR	$\Rightarrow a \wedge b$ $a = a \wedge b$
$>>$	perform Bitwise right shift	$\Rightarrow a >> b$ $a = a >> b$

## ⑥ Special Operators:

o Identity Operators - is and is not are the identity Operators both are used to check; two values are located on the same part of the memory. Two Variables that are equal does not imply that they are identical.

is  $\rightarrow$  True if the Operands are identical  
is not  $\Rightarrow$  True if the Operands are not identical.

Membership Operators : in and not in are the membership operators; used to test whether a value or variable is in a sequence.  
in : True if value is found in the sequence  
not in : True if value is not found in the sequence.

4) Explain the features of python

- python is a dynamic, high level, free open source and interpreted programming language.
- It supports Object-Oriented programming as well as procedural oriented programming.
- In python, we don't need to declare the type of variable because it is a dynamic typed language.

Features in python.

- 1) Easy to code.
- 2) free and open source.
- 3) Object-oriented programming language.
- 4) GUI programming Support (Graphical User interfaces)
- 5) High-Level programming language.
- 6) Extensible features
- 7) python is portable language.
- 8) python is Integrated language
- 9) Interpreted Language
- 10) Large standard library.
- 11) Dynamically Typed Language

5. Justify Why python is interactive interpreted language.

- Unlike c/c++ etc, python is an interpreted Object-oriented programming language.
- python program runs directly from the source code.
- Each time python programs are executed Code is required.
- python converts source code written by the programmer into intermediate language which is again translated into a native language / machine language that is executed. so python is an Interpreted language.
- It is processed at runtime by the interpreter.
- The program need not be compiled before its execution.
- It is similar to PERL and PHP.