

## Assignment :- 2. Python

① [30/3]

Q. What are data types in python? Explain.

A: Every value in python has a datatype. Since everything is an object in python programming, data types are actually classes and variables are instances (object) of these classes.

There are various data types in python.

### \* python numbers

integer, floating point numbers and complex numbers fall under python numbers category. The ones defined as int, float, and complex classes in python

we can use the type() function to know which class a variable or a value belongs to similarly, the isinstance() function is used to check if an object belongs to a particular class

a = 5

```
print(a, "is a type", type(a))
```

a = 2.0

```
print(a, "is a type", type(a))
```

a = 1+2j

```
print(a, "is a complex number?", isinstance(a, complex))
```

## \* python list

⑨ 30/07

List is an ordered sequence of items. It is one of the most used datatype in python and is very flexible. All the items in a list do not need to be of the same type.

Declaring a list is pretty straight forward. Items separated by comma are enclosed within brackets [ ].

```
a = [1, 2, 3, 'python']
```

We can use the slicing operator [ ] to extract an item or a range of items from a list. The index starts from 0 in python.

```
a = [5, 10, 15, 20, 25, 30, 35, 40]
```

```
# a[2] = 15
```

```
print("a[2] = ", a[2])
```

```
# a[0:3] = [5, 10, 15]
```

```
print("a[0:3] = ", a[0:3])
```

```
# a[5:] = [30, 35, 40]
```

```
print("a[5:] = ", a[5:])
```

### Output

a[2] = 15

a[0:3] = [5, 10, 15]

a[5:] = [30, 35, 40]

List are mutable, meaning, the value of elements of a list can be altered.

## \* python Tuple

Tuple is an ordered sequence of items. Same as a list. The only difference is that tuples are

immutable. Tuples once created cannot be modified.

Tuples are used to write-protected data and are usually faster than lists as they can't change dynamically.

It is defined within parentheses () where items are separated by commas

$$t = (5, 'program', 1+3j)$$

We can use the slicing operator [] to extract items but we cannot change its value.

$$t = (5, 'program', 1+3j)$$

$$\# t[1] = 'program'$$

$$\text{print}(t[1])$$

$$\# t[0:3] = (5, 'program', (1+3j))$$

$$\text{print}(t[0:3])$$

# generates error

# tuples are immutable

$$t[0] = 10$$

## \* ) Python String

String is sequence of unicode characters. We can use single quotes or double quotes to represent string. multi-line string can be denoted using triple quotes [''' ] or [""" ]

$$s = "This is a string"$$

`print(s)`

$$s = ''' A multiline$$

string'''

`print(s)`

## \* Python Set

Set is an unordered collection of unique items. set is defined by values separated by comma inside braces {}, items in a set are not ordered.

```
a = { 5, 2, 3, 1, 4 }
```

# printing set variable

```
print("a = ", a)
```

# data type of variable a

```
print(type(a))
```

## \* Python Dictionary

Dictionary is an unordered collection of key-value.

It is generally used when we have a huge amount of data. Dictionaries are optimized for retrieving data. We must know the key to retrieve the value.

```
d = { 1: "value", 'key': 2 }
```

```
print(type(d))
```

```
print("d[1] = ", d[1]),
```

```
print("d['key'] = ", d['key']),
```

# Generates error

```
print("d[2] = ", d[2]))
```

## ⑧ Briefly explain history of python?

3048

- \* Python laid its foundation in the late 1980s.
- \* The implementation of Python was started in the December 1989 by Guido van Rossum at CWI in Netherland.
- \* In February 1991, van Rossum published the code (labeled version 0.9.0) to alt.sources.
- \* In 1994, Python 1.0 was released with new features like.. lambda, map, filter and reduce.
- \* Python 2.0 added new features like, list comprehensions, garbage collection system.
- \* On December 3, 2008, Python 3.0 (also called "Py3k") was released. It was designed to unify fundamental flaws of the language.
- \* ABC programming language is said to be the predecessor of Python language which was capable of exception handling and interfacing with Amoeba operating system.
- \* Python is influenced by following programming languages:
  - \* ABC Language
  - \* modula-2

Q) Explain all the operators in Python

7

2018

A: \* Arithmetic operators

\* Relational operators

\* Logical operators

\* Bitwise operators

\* Assignment operators

\* Special operators

\* Membership operators

① Arithmetic operators: Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication and division.

Operator	Description	Syntax
+	Addition: adds two operands	$x+y$
-	Subtraction: subtracts two operands	$x-y$
*	multiplication: multiplies two operands	$x*y$
/	Division (float): divides the first operand by the second	$x/y$
//	Division (floor): divides the first operand by the second	$x//y$
%	modulus: returns the remainder when first operand is divided with second	$x \% y$

②) Relational operators:- Relational operators compare the values. It either returns True (or) False according to the condition.

③ 3048

### operator

### Description

### Syntax

>

Greater-than: True if left operand is greater than the right

$x > y$

= =

equal to: True if both operand are not equal to 0  
False if

$x = y$

!=

Operands are not equal  
greater than or equal to:  
0: True if left

$x \neq y$

$\geq$

operand is greater than  
or equal to the right  
less than or equal to:  
0: True if left operand is  
less than or equal to  
the right

$x \geq y$

④ Logical operators:- Logical operators perform logical AND, Logical OR and logical NOT

operator	Description	Syntax
and	logical AND: True if both the operands are true	$x \text{ and } y$
or	logical OR: True if either of the operands is true	$x \text{ or } y$
not	logical NOT: True if operand is false	$\text{not } z$

④) Bitwise operators :- Bitwise operator acts on bits  
and performs bit by bit operations

operator	Description	Syntax
&	Bitwise AND	$x \& y$
	Bitwise OR	$x   y$
~	Bitwise Not	$\sim x$
^	Bitwise XOR	$x ^ y$
>>	Bitwise right shift	$x >> y$
	Bitwise left shift	$x$

⑤) Assignment operators :- Assignment operators are  
used to assign values to the variable.

operator	Description	Syntax
=	Assign value of right side of expression to left side operand odd AND: odd right side operand	$x = y + 2$

$+=$

left side operand &

3048

$$a + b$$

$$a = a + b$$

then assign to left  
operand

Subtract AND:

Subtract right

$- =$

operand from left

$$a - b$$

operand and then  
assign to left  
operand

Multiply AND:

Multiply right

$* =$

operand with left  
operand and then  
assign to left  
operand.

$$a * b$$

$$a = a * b$$

$/ =$

Divide AND: Divide  
left operand with  
right operand and  
then assign to left  
operand

$$a / b$$

$$a = a / b$$

Modulus AND: Take

modulus using left  
and right operand

$\% =$

and assign result to  
left operand

$$a \% b$$

$$a = a \% b$$

// = Divide left operand  
with right operand  
and then assign the  
value (floor) to left operand

$$a//b$$

$$a = a//b$$

Exponent AND:

calculate  
exponent (raise power)

$$a^{**} = b$$

$$a = a^{**} - b$$

\*\* = value using operands  
and assign value to left  
operand perform  
Bitwise

AND on operands

$$a \& = b$$

$$a = a \& b$$

& = and assign value to  
left operand

perform Bitwise OR

$$a | = b$$

| = on operands and assign  
value to left operand

$$a = a | b$$

Perform Bitwise

XOR on operands

& = and assign value to  
left operand

$$a \& = b$$

$$a = a \& b$$

Perform Bitwise

Right shift on operands

$$a >> = b$$

>> = and assign value to left  
operand perform

$$a = a >> b$$

⑥ special operators : There are some special

type of operator like

(ii)

\* identity operators

3048

\* membership operators

\* identity operators :- is and is not are the identity operators both are used to check if two values are located on the same part of the memory. Two variables that are equal does not imply that they are identical.

is : True if the operands are

is not : True if the operands are

\* 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

not in : True if value is not found

Q) Explain the Features of python?

(12)

A: There are many features in python.

3048

- \* Easy to code.
- \* free and open source.
- \* object-oriented language.
- \* GUI programming support.
- \* High level language.
- \* extensible feature.
- \* python is portable language.
- \* interpreted language.
- \* Large standard library.
- \* Dynamically typed language.

⇒ easy to code:-

python is high level programming language, python is very easy to learn language as compared to other language like c, c++, java script, java etc, it is very easy to code in python language and anybody can learn python basic in few hours or days. it is also developer friendly language.

⇒ Free and open source:-

python language is freely available at official website and you can download it from the given download link below click on the download python keyword.

Download python

Since it is open-source, this means that source code is also available to the public.

③ object-oriented language 3048 ⑬  
one of the key features of python is object-oriented programming. python supports object-oriented language and concepts of classes, objects encapsulation etc.

#### ④ GUI programming support

Graphical user interfaces can be made using a module such as pygtk, PyQt, wxPython or Tk in python. PyGTK is the most popular option for creating graphical apps with python.

#### ⑤ High-level language

python is a high-level language. when we write programs in python, we do not need to remember the system architecture, nor do we need [compile that code in c/c++ language.]

#### ⑥ extensible Feature

python is a extensible language. we can write our some python code into c or c++ language and also we can compile that code in c/c++ language.

#### ⑦ python is portable language.

python language is also a portable language. for ex. if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it we can run this code on any platform.

## ⑤ Python is integrated language.

(14) 3048

python is also an integrated language because we can easily integrate python with other language like C, C++ etc.

## ⑥ Interpreted language

python is an interpreted language. because python code is executed line by line at a time. like other language C, C++, Java etc. there is no need to compile python code - the make it easier to debug our code, the source code of python is converted into an immediate form called byte code.

## ⑦ Large standard library

python has a large standard library which provides such set of modules and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers.

## ⑧ Dynamically typed language

python is a dynamically-typed language that means the type for a variable is decided at run time not in advance. because of this feature we don't need to specify the type of variable.

30/08

Q) justify why python is an interactive interpreted language.

Ans: unlike C/C++ etc, Python is an interpreted object-oriented 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 byte code.

An interpreter is a translator in computer language which translates the given code line by line in machine readable byte code. And if any error is encountered it stops the translation until the error is fixed.

Unlike C language, which is a 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.

When a Python statement is entered, and is followed by the return key, if appropriate, the result will be printed on the screen immediately in the next line. This is particularly advantageous in the debugging process. In interactive mode of operation, Python is used in a similar way as the UNIX command line or the terminal. Interactive Python is very much helpful for the debugging purpose.

It simply returns the ">>>>" prompt on the corresponding output of the statement if appropriate and return error for return key. If appropriate the result

2018]

will be printed on the screen immediately in the next line. This is particularly advantageous in this way if you have any doubt like whether the syntax is correct or the module you are importing exists or anything like that. You can be sure within seconds. Python interactive mode