

## Assignment - 2

Q : what are the data types in python? Explain

Ans There are 5 types of data types in python.

1. Numbers
2. Strings
3. Lists
4. Tuples
5. Dictionary

1. Number : This data type support all numeric values, i.e floating point numbers, integers and complex numbers. They are defined as 'int', 'float' and 'complex' classes in Python.

Ex :- integer type      floating point number      complex number  
 $x=5$                            $x=5.1$                            $x=2+5j$

here these numbers can be type casted from one form to other form but complex numbers cannot be converted to any other form.

2. Strings : It is a collection of characters enclosed one within (or) intended with single or double quotes, by which Python interpreter assumes the closing quote as the end of a string. String comes under sequence type.

Ex :-  $a = 'Hi'$

$A = "Hi"$

`Print("a,A" a,A)`

O/P :- Hi Hi

3. List :- This is the most versatile data type of Python. It is the ordered collection of one or more than one <sup>values</sup> items in it which is not necessarily of the same type. These items are enclosed by ~~by~~ with square brackets. Values in a list are separated by comma.

Ex:- my\_list = [1, 2, 3, 4].

4. Tuples : Similar to lists tuples are a type

4. Tuples: These are similar to lists. All the elements in tuples are separated by comma but enclosed by parenthesis. The major difference between tuple and lists is that the values are specified in tuple they cannot be changed.

Ex:-

abc = (1, 2, 3)

Print(abc)

Op: (1, 2, 3)

5. Dictionaries : Dictionary (or) dict in Python are <sup>in</sup> way of storing elements just like Python list. But rather than accessing elements using index, you can assign a fixed key to it and access the element using the key.

Ex:- a = { 'apple': 'fruit', 'beenroot': 'vegetable', 'cake': 'dessert'}

a[ 'chips' ] = 'snack'

print(a[ 'apple' ])

print(a[ 'chips' ])

Op:

fruit

snack.

Every value in Python has a data type. Since everything is object oriented, data types are actually classes and variables are instances of these classes.

(2) Briefly explain history of Python

Ans Python is an interpreted, high level, general purpose programming language. Developed by "Guido van Rossum" and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant white space. It is a language constructed <sup>and</sup> object-oriented programming which makes it easy for the programmer to understand the logic steps in large scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms. Python is often dynamically typed described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980's as a successor to the ABC language. Python 3.0 was introduced in 2008 with enhanced features like list comprehension & a garbage collection system with reference counting.

Python 3.0 was released in 2008. It was a major version of language which is not completely backward-compatible and many of the Python 2 codes doesn't run unmodified on Python 3.

The Python 2 language was officially discontinued in 2020 and Python 2.7.18 is the last Python 2.7

and therefore, the last Python 2 released. Python interpreters are available for many operating systems. A non profit organization, the Python Software Foundation, manages and directs the resources for Python and CPython developments.

(3) Explain the operators in Python?

Any operators are special symbols in Python that carry out arithmetic or logical computations. The value that operator operates on is called Operand.

(a) Arithmetic Operators

Arithmetic Operators are used to perform mathematical operations.

Operator	Meaning	Example
+	Add 2 operands or unary plus	$x+y=2$
-	Subtract 2 operands (or) unary minus	$x-y=x-2$
*	Multiple 2 operands	$x*y$
/	divide left operand by the right operand	$x/y$
%	Modulus remainder of the division of left operand by right	$x \% y$
//	Floor division: int value of the division	$x//y$
**	Exponent: left operand raised to the power of right	$x^{**} 2$ ( $x$ to the power of 2)

## comparison Operators

comparison operators are used to compare values. It returns either true True or False according to the condition.

operator	Meaning	Example
$>$	greater than - True if left operand is greater than the right	$x > y$
$<$	less than - True if right operand is greater than left	$x < y$
$= =$	Equal to - True if both operands are equal	$x == y$
$!=$	Not equal	$x != y$
$\geq$	greater than or equal to True if left operand is greater than or equal to the right	$x \geq y$
$\leq$	less than or equal to True if left operand is less than or equal to the right	$x \leq y$

## Logical Operator

logical operators are the 'and', 'or', 'not' operators.

operator	meaning	examples
and	True if both the operands are true.	x & y
or	True if either of the operands is true	x   y
not	True if operand is false (complements the operand)	not x

### Bitwise Operator

Bitwise Operator act on operands as if they were strings of binary digits. They operate bit by bit, hence the name.

operator	meaning	example
&	Bitwise AND	x & y
	Bitwise OR	x   y
~	Bitwise not	~x
^	Bitwise XOR	x ^ y
>>	Bitwise Right shift	x >> 2
<<	Bitwise left shift	x << 2

## Assignment Operators

These are used to assign the value to the variable in Python.

operator	example	equivalent to
=	$n = 5$	$n = 5$
+=	$n += 5$	$n = n + 5$
-=	$n -= 5$	$n = n - 5$
*=	$n *= 5$	$n = n * 5$
/=	$n /= 5$	$n = n / 5$
%=	$n %= 5$	$n = n \% 5$
//=	$n //= 5$	$n = n // 5$
**=	$n **= 5$	$n = n \text{**} 5$
<=	$n <= 5$	$n = n < 5$
!=	$n != 5$	$n = n \neq 5$
>>=	$n >>= 5$	$n = n >> 5$

## Special Operators

Python has a few special operators like membership operators and identity operators.

### (i) identity operators

'is' & 'is not' are identity operators. They are used to check if two values (or variables) are located on the same part of memory.

Operator	Meaning	Example
is	True if the Operands are identical.	n is true
is not	True if the Operands are not identical.	n is not true

Membership Operators: 'in' and 'not in' are the membership operators. They are used to test whether a value is found in a sequence.

Operator	Meaning	Example
in	True if value is found in sequence	5 in x.
is not in	True if value is not found in sequence	5 not in x

Q) Explain the features of Python.

Ans: Python provides lots of features that are listed below.

- easy to learn and use

Python is high level programming language. It is easy learn and use. It is developed friendly and high level programming language.

- Expressive language

Python language is more expressive that means it is more understandable and readable.

- interpreted language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

- free and open source

Python is freely available at official web address. The source code is also available. Therefore it is open source & can be modified or added to as per needs.

- Object oriented programming

Python supports OOPS concepts classes, objects & encapsulation. It comes into instance.

- Extensible

Python is extensible because it implies that other languages such as C/C++ can be used to compile like code and thus it can be used further in one standard Python code.

- Standard library

Python has large based library & provides rich set of module and functions for rapid application development.

- GUI programming support

Graphical user Interface can be developed using Python.

- Integrated

It can be easily integrated with the languages like C, C++ & JAVA etc.

5. Justify why Python is interactive interpreted language  
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's language which translates the given code line-by-line in machine readable byte codes. And if any error is encountered it stops the translation until the error is fixed. 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 advantage in the debugging process. It simply returns the ">>>" prompt on the corresponding output of the statement if appropriate and return error for incorrect statements.