

Assignment-2

Q) What are the data types in Python? Explain?

A. Variables can hold values of different data types. Python is a dynamically typed language hence we need not define the type of the variable while declaring it. The interpreter implicitly binds the value with its type.

Python enables us to check the type of the variable used in the program. Python provides us the type() function which returns the type of the variable passed.

The data types defined in Python are given below.

1. Numbers
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.

Python supports 4 types of numeric data.

1. int (signed integers like 10, -2, 29 etc)
2. long (long integers used for a higher range of values like 908090800L, -0x1929292L 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. In python, we can use single, double, quotes.

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

es list:-

Lists are similar to arrays 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[].

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:-

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 a comma(.) and enclosed in parentheses().

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.

② Briefly explain history of Python.

A. Python is an interpreted, high-level, general purpose programming language. ~~exe~~
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, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vocation" from his responsibilities as Python's Benevolent Dictator for Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decisionmaker. He now shares his leadership as a member of a five-person steering council. In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and van Rossum to a five-member "steering council" to lead the project.

python 2.0 was released on 16 october 2000 with many major new features, including a cycle-detecting garbage collector and support for unicode.

python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to python 2.6x and 2.7x version series. Releases of python 3 include the 2to3 utility, which automates the translation of python 2 code to python 3.

python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-port to python 3.

③ Explain all the operators in python.

a) Arithmetic operators:-

Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication etc.

example

- | | |
|---------------------------------------------------------------|----------|
| + Add two operands or unary plus | $x+y+2$ |
| - Subtract right operand from the left
or unary minus | $x-y-2$ |
| * | $xx*y$ |
| / | x/y |
| % | $x \% y$ |
- (always results into float)
- (remainder of x/y)

11 FLOOR division - division that results into whole number adjusted to the left in the number line.

x/y

** Exponent - left operand raised to the Power of right.

x^{yy}
(x to the power y)

comparison operators:

comparison operators are used to compare values. It returns either TRUE or FALSE according to the condition.

operator meaning

Example

> Greater than - True if left operand ~~x~~ $x > y$ is greater than the right

< Less than - True if left operand is $x < y$ less than the right.

== Equal to - True if both operands $x = y$ are equal

!= Not equal to - True if operands are $x \neq y$ not equal

>= Greater than or equal to - True if left $x \geq y$ operand is greater than or equal to the right

<= Less than or equal to - True if left operand $x \leq y$ is less than or equal to the right

logical operators:

logical operators are the and, or, not operators

operator meaning

and True if both the operands are true.

or True if either of the operands is true.

not True if operand is false (complements the operand)

Bitwise operators:

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

operator	meaning	Example
&	Bitwise AND	$x \& y = 0 \quad 0000\ 0000$
	Bitwise OR	$x y = 14 \quad 0000\ 1110$
~	Bitwise NOT	$\sim x = -11 \quad 1111\ 0101$
^	Bitwise XOR	$x ^ y = 14 \quad 0000\ 1110$
>>	Bitwise rightshift	$x >> 2 = 2 \quad 0000\ 0010$
<<	Bitwise left shift	$x << 2 = 40 \quad 0010\ 1000$

Assignment operators:

Assignment operators are used in python to assign values to variables.

operator	Example	equivalent to
=	$x = 5$	$x = 5$
+=	$x += 5$	$x = x + 5$
-=	$x -= 5$	$x = x - 5$
*=	$x *= 5$	$x = x * 5$
/=	$x /= 5$	$x = x / 5$
%=	$x \% = 5$	$x = x \% 5$
//=	$x // = 5$	$x = x // 5$
**=	$x ** = 5$	$x = x ** 5$
&=	$x \& = 5$	$x = x \& 5$
=	$x = 5$	$x = x 5$
^=	$x ^ = 5$	$x = x ^ 5$
>>=	$x >> = 5$	$x = x >> 5$

special operators

Python language offers some special type of operators like the identity operator or the membership operator. They are described below with examples.

identity 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. Two variables that are equal does not imply that they are identical.

operator meaning example.

is True if the operands are identical. $x \text{ is } x$ is TRUE.

isnot True if the operands are not identical. $x \text{ is not } x$ is not TRUE.

membership operators

in and notin are the membership operators in Python. They are used to test whether a value or variable is found in a sequence. In a dictionary we can only test for presence of key, not the value.

operator meaning example

in True if value/variable is found in the sequence

notin True if value/variable is not found in the sequence 5 not in x

Q) Explain the features of python.

1. 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.

2. Free and open source.

Python language is freely available at official website and you can download it from the given link.

Since, it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

3. object oriented language.

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 PyQt5, PyQt4, wxPython OR TK in python.

PyQt5 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 to manage the memory.

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 example, if we have Python code for windows and if we want to run this code on other platforms such as Linux, Unix and Mac then we do not need to change it, we can run this code on my platform.

8. Python is integrated language:

Python is also an Integrated Language because we can easily integrated Python with other language like C, C++ etc.

9. 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 this makes it easier to debug our code. The source code of Python is converted into an immediate form called bytecode.

10. Large standard library.

Python has a large standard library which provides rich set of module and functions so you donot 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 etc.

11. Dynamically typed Language.

Python is 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.

⑤ Justify why Python is interactive interpreted language.

Python is a high-level interpreted, interactive and object oriented scripting language python is designed to be highly readable. It uses English keywords frequently where as other language use punctuation and it has fewer syntactical constructions than other language.

> Python is interpreted - python is processed at runtime by the interpreter. You donot need to compile your program before executing it. This is similar to PERL and PHP

> Python is interactive - you can actually sit at a python prompt and interact with the interpreter directly to write your program.