

21/06/2020

Assignment 1

Y. Purnu Huthika
321810301009

1. What are the datatypes in python? Explain.

A. Every value in python has a datatype. Since everything is an object in python programming, datatypes are actually classes & variables are instance of these classes.

* There are various datatypes in python. Five of important datatypes are listed below.

a) Numeric:

A Numeric value is any representation of data which has a numeric value. Python identifies 3 types of Numbers.

i. INTEGER: It includes positive & negative whole numbers.

ii. FLOAT: It includes any real number with a floating point representation in which a fractional component.

iii. COMPLEX: It includes the combination of real no; & Imaginary component represented as $x + jy$, where x & y are real numbers & value of j is -1 .

b) Boolean:

• Data with one of two built-in values True & False. In this 'T' & 'F' are of uppercase. true & false are not valid & Python will throw an error for them.

c) Sequence Type:

• A Sequence is an ordered collection of similar & different data types. Python has the following built-in sequence data types:

i. String: A string value is a collection of one or more characters put in single, double or triple quotes.

ii. List: A list object is an ordered collection of one or more data items, not necessarily of the same type, put in square brackets.

iii. Tuple: A Tuple object is an ordered collection of one or more data items, not necessarily of the same type, put in parentheses.

iv. Dictionary:

A dictionary object is an unordered collection of data in a key: value pair form. A collection of such pairs is enclosed in curly brackets.

Ex: {1: "Steve", 2: "Bill", 3: "Ram", 4: "Farah"}

2. Briefly explain history of Python.

A: Python is an interpreted, high level, general purpose programming language. Created by "Guido van Rossum" released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant white space.

→ Its language constructs & object oriented approach aim to help programmers write clear, logical, code for small & large-scale projects.

- Python is dynamically typed & garbage-collected. It supports multiple programming paradigms, including structured object oriented & functional programming.
- Python is often 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 2.0, released in 2000, introduced features like list comprehensions & a garbage collection system with reference counting.
- Python 3.0, released in 2008, was a major revision of the language which is not completely backward-compatible & many of the python2 codes does not run unmodified on python 3.
- The python 2 language was officially discontinued in 2020 & python 2.7.18 is the last Python 2.7 released & therefore last Python 2 released.
- No more security patches & other improvements will be released for it. With python 2's end of life, only python 3.5.x & later are supported.

3. Explain all the operators in python?

A. operators are special symbols in python that carry out arithmetic or logical computation. The value that operator operates on is called operand.

a) Arithmetic operators:

- Arithmetic operators are used to perform mathematical operations like $+$, $-$, $*$, $/$, etc.

operator	Meaning	Example
$+$	add 2 operands, unary plus	$x+y$
$-$	Subtracts right operand from the left, unary minus	$x-y$
$*$	multiplies 2 operands	$x*y$
$/$	divides left operand by the right one	x/y
$//$ (floor division)	Division that results into whole number adjusted to the left on number line	$x//y$
$\%$ (modulus)	gives the remainder of the division of left operand by right	$x\%y$
$**$	left operand raised to the power of right operand	$x**y$

b) 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 is greater than right operand	$x > y$
$<$	less than - True if left operand is less than right	$x < y$
$==$ (Equal to)	True if both operands are Equal	$x == y$
$!=$ (Not equal to)	True if operands are not Equal.	$x != y$
$>=$	True if left operand is greater than or Equal to the right.	$x >= y$
$<=$	True if left operand is less than or equal to the right	$x <= y$

c) Logical operators:

- Logical operators are the and, or, not operators.

operators	Meaning	Example
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	not x

dy Bitwise operators:

- Bitwise operators act on operands as if they were strings of binary digits. They operate bit by bit, hence the name. For ex, 2 is 10 in binary.

Operators	Meaning	Example
$\&$	Bitwise AND	$x \& y = 0$
$ $	Bitwise OR	$x y = 14$
$-$	Bitwise NOT	$-x = -11$
\wedge	Bitwise XOR	$x \wedge y = 14$
$>>$	Bitwise right shift	$x >> 2 = 2$
$<<$	Bitwise left shift	$x << 2 = 40$

ex Assignment operators:

- Assignment operators are used in python to assign values to variables. $a=5$ is a simple assignment operator that assign value 5 on the right to the variable a on the left. There are various compound operators in python like $a+=5$ that adds the variable a , later assign the same. It is equivalent to $a=a+5$.

operator	Example	Equivalent to
$=$	$x=5$	$x=5$
$+=$	$x+=5$	$x=x+5$
$-=$	$x-=5$	$x=x-5$
$*=$	$x*=5$	$x=x*5$
$/=$	$x/=5$	$x=x/5$
$\%=$	$x\%=5$	$x=x\%5$
$//=$	$x//=5$	$x=x//5$
$**=$	$x**=5$	$x=x**5$
$\&=$	$x\&=5$	$x=x\&5$
$ =$	$x =5$	$x=x 5$
$\wedge=$	$x\wedge=5$	$x=x\wedge 5$
$>>=$	$x>>=5$	$x=x>>5$
$<<=$	$x<<=5$	$x=x<<5$

ex Special operators

- is - True if the operand are identical.
- $is not$ - True if the operands are not identical.

9) Membership operators:

in - True if the operands are identical.
not in - True if value is not found in sequence.

4) Explain the features of Python?

1) Easy to learn and use

- Python is easy to learn & use. It is developer friendly & high level programming language.

2) Expressive language

- Python is expressive language since it is more understandable & readable.

3) Interpreted language

- The interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

4) Cross-platform language

- Python can run equally on different platforms like windows, linux, unix etc. So we can say that it is portable language.

5) free and open source

- This language is freely available at official web address. The source code is also available, so, it is open source.

6) Object oriented language

- This language supports the concepts of classes & objects come into existence.

7) Extensible

- It implies that other language such as C++/C can be used to compile the code & thus it can be further used in our Python code.

8) Large standard library

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

9) GUI programming support

- Graphical user interface can be developed using Python.

10) Integrated

- It can be easily integrated with languages like C, C++, Java etc.

5) Justify why Python is interactive interpreted language?

Ans. Python is interpreted language since interpreted executes the code line by line at a time. This makes debugging easier. And converts machine readable bytecodes. And if any error is encountered it stops the translation until the error is fixed.

- Python is interactive. when a python statement is entered & is followed by the return key, if appropriate, the result will be printed on screen immediately in next line. It is used in a similar way as the Unix command line or the terminal And is helpful for the debugging purpose.