

Full stack web development using python

Operators



Saurabh Shukla (MySirG)

Agenda

- ① Various operators in Python
- ② Arithmetic Operators
- ③ Relational Operators
- ④ Logical Operators
- ⑤ Bitwise Operators
- ⑥ Assignment operators
- ⑦ Identity operators
- ⑧ Membership operators

Operators

operators are functions in python.

$a+b \rightarrow \text{--add--}(a, b)$

various operators in python (in alphabetical order)

Addition

$a+b$

Concatenation

$a+b$

Containment test

$a \in \text{obj}$

Division (True Division)

a/b

Division (Floor Division)

$a//b$

Bitwise AND

$a \& b$

Bitwise Exclusive OR

$a \wedge b$

Bitwise NOT

$\sim a$

Bitwise OR

$a | b$

Exponentiation

$a ** b$

Identity

a is b

Identity

a is not b

Indexed Assignment

obj[i] = a

Indexed Deletion

del obj[i]

Indexing

obj[i]

Left Shift

a << b

Logical AND

a and b

Logical OR

a or b

Modulo

a % b

Multiplication or Repetition

a * b

Negation

- a

Negation (logical)

not a

positive

+ a

Right Shift

a >> b

Slice Assignment

$s[i:j] = \text{values}$

Slice Deletion

~~$s[i:j]$~~

Slicing

$s[i:j]$

String formatting

$s \% s1$

Subtraction

$a - b$

ordering (less than)

$a < b$

ordering (greater than)

$a > b$

ordering (less than or equal to)

$a \leq b$

ordering (greater than or equal to)

$a \geq b$

equality

$a == b$

not equal

$a != b$

Operators

- Arithmetic Operators $\ast\ast, /, //, \ast, \%, +, -$
- Relational Operators $>, <, >=, <=, ==, !=$
- Logical Operators $\text{not}, \text{and}, \text{or}$
- Bitwise Operators $\& | \wedge \sim >> <<$
- Assignment Operators $= += -= /= //= \%= \ast\ast=$
 $*-= \&-= |= \wedge= >>= <<=$
- Identity Operators $\text{is}, \text{is not}$
- Membership Operators $\text{in}, \text{not in}$

no $++, --$ operators in Python

Arithmetic Operators

*, /, //, *, +, -, %

$$2^{**} 3 = 2^3 = 2 \times 2 \times 2 = 8$$

$$-2^{**} 2 = -2^2 = -(2)^2 = -4$$

$$(-2)^{**} 2 = (-2)^2 = 4$$

$$2^{**} -2 = 2^{-2} = \frac{1}{2^2} = \frac{1}{4} = 0.25$$

$$3 * 4 = 12$$

5/2 2.5

10/5 2.0

Floor value

2.3 → 2

2.9 → 2

2.1 → 2

2.0 → 2

7.3 → 7

$\%$, modulo

$$11 \%_3 2 \quad 3 \overline{)11} \quad \begin{array}{r} 9 \\ \hline 2 \end{array}$$

$$3 \%_4 3 \quad 4 \overline{)3} \quad \begin{array}{r} 0 \\ \hline 3 \end{array}$$

$$x \% y \rightarrow 0$$

x is completely
divisible by y

$$x/5 \rightarrow 2.0$$

$$x/10 \rightarrow$$

$$243/10 \rightarrow 24.3$$

$$x/110 \rightarrow x \text{ without last digit}$$

$$243/110 \rightarrow 24$$

$$x \% 10 \rightarrow \text{last digit}$$

$$243 \% 10 \rightarrow 3$$

$$5.5 \% 3 \rightarrow 2.5$$

--> modulo operator can't perform on float in other P.L

/ always return float result

// always return floor value , int type
or float type depending on operands.

+,* can be used with str type
values also

- + is addition operator when operands are numbers (int, float, complex, bool)
 - + is concatenation operator when operands are str
 - + operation is invalid when applied between a number and a string.
-
- * is used to multiply two numbers
 - * is repetition operator when applied between a str and an int

Relational Operators

<, >, <=, >= → inequality operators

==, != → equality operators



Never gives error

- Relational operators always give result in True or False.
- When truth value is converted to int, it becomes 1 for True and 0 for False.
- Relational operators can also be used to compare two strings
- Only == and != operators can be used between two complex type values.
- == and != never yield error

not
and
or

Logical Operators

logical operators must be written in lowercase only.

not True \rightarrow False
not False \rightarrow True

True and True \rightarrow True
True and False \rightarrow False
False and X \rightarrow False

False or False \rightarrow False
False or True \rightarrow True
True or X \rightarrow True

Every non zero value → True

→ False

Zero

Non empty string

→ True

→ False

Empty string

when operands are non-bool then
result will also be non-bool

Bitwise Operators

& | ^ ~ >> <<

$$0 \& 0 \rightarrow 0$$

$$0 \& 1 \rightarrow 0$$

$$1 \& 0 \rightarrow 0$$

$$1 \& 1 \rightarrow 1$$

$$0 | 0 \rightarrow 0$$

$$0 | 1 \rightarrow 1$$

$$1 | 0 \rightarrow 1$$

$$1 | 1 \rightarrow 1$$

$$| x = 25 \& 37$$

$$25 = 011001$$

$$37 = \frac{100101}{000001}$$

$$x = 44 | 71$$

$$44 = 0101100$$

$$71 = 1000111$$

$$111 = \underline{110111}$$

$$0 \wedge 0 \rightarrow 0$$

$$0 \wedge 1 \rightarrow 0$$

$$1 \wedge 0 \rightarrow 0$$

$$1 \wedge 1 \rightarrow 1$$

$$x = 56 \wedge 29$$

$$56 = 111000$$

$$29 = \underline{011101}$$

$$37 = \underline{100101}$$

$\sim 0 \rightarrow 1$

$\sim 1 \rightarrow 0$

$$\boxed{K = b_1 \leftarrow 2's}$$

$$-K = b_2$$

$$K = 1101$$

1's 0010
 $\frac{+ 1}{0011}$

$2's =$

$x = \sim 5 \rightarrow 0 = +ve$
 $1 = -ve$

$5 = \boxed{0}0000101$

$\sim 5 = \boxed{1}111010 = -5 = -6$

$2's \rightarrow 000000110 = 6$

35 >> 2

35 = , 001000,11
→

$$1000 = 8$$

12 << 3



12 = 1100000 ≈ 96

Assignment Operator

= += -= *= /= //= &= |= ^= **=

>>= <<= %=

$x = 4$

↑
must be a variable

$3 = 4$ Error

$x += 5 \rightarrow x = x + 5$

$x -= 3 \rightarrow x = x - 3$

$x *= 2 \rightarrow x = x * 2$

$x ^= 10 \rightarrow x = x ^ 10$

$x = x + 3$

↑ ↑
Container content

$x++$ Error

$x = x + 1$

$x += 1$

How to assign values to multiple variables
in a single line?

$x=4, y=3, z=2$ Error

$x, y, z = 4, 3, 2$ Correct

$x, y, z = 2, 3$ Error

Identity operator

- is
- is not

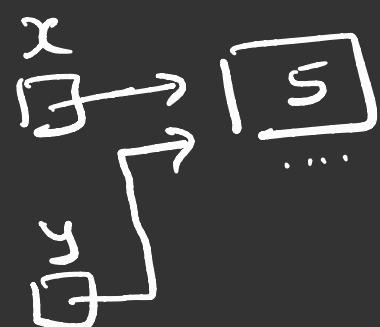
It checks whether the two references referring to the same object or no.

It results in True or False.



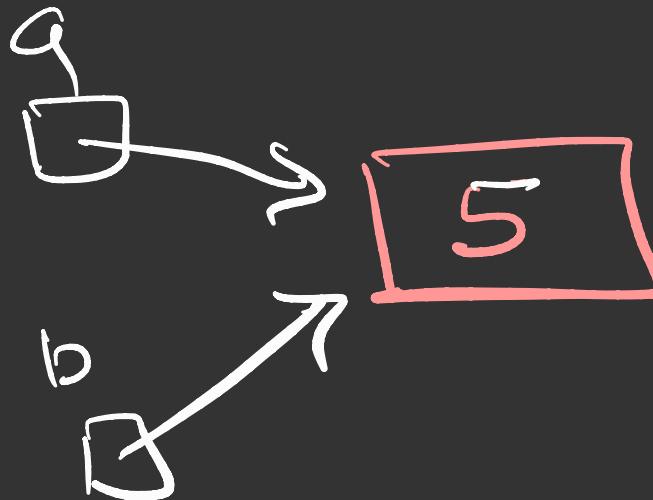
$$x = 5$$

$$y = 5$$



$a = 5$

$b = 5$



reference_count

2

Membership operators

- in
- not in
- These operators are applicable only on containers (iterable)
- They result True or False

Container is a type which can contain multiple values and also iterable

x
→

15, 20, 37, 4, 61

15 in x True

25 in x False

int, float, complex, bool are not iterable

str, range, list, tuple, set, dict are iterable