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 → __add__(a, b)

various operators in python (in alphabetical order)

Addition	a + b
Concatenation	a + b
Containment test	a in obj
Division (True Division)	a / b
Division (Floor Division)	a // b
Bitwise AND	a & b
Bitwise Exclusive OR	a ^ b
Bitwise NOT	~ 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] = values`
Slice Deletion	`del 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 <= b`
ordering (greater than or equal to)	`a >= b`
equality	`a == b`
not equal	`a != b`

Operators

- Arithmetic Operators **, /, // *, %, +, -
- Relational Operators >, <, >=, <=, ==, !=
- Logical Operators not, and, or
- Bitwise Operators & | ^ ~ >> <<
- Assignment Operators = += -= /= //= %= **= *= &= |= ^= >>= <<=
- Identity operators is, is not
- Membership Operators in, not in

no ++, -- operators in Python

Arithmetic Operators

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

$2**3 = 2^3 = 2\times2\times2 = 8$

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

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

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

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

```
  3 | 11 | 3
      9
      --
      2
```

3 % 4 → 3

```
  4 | 3 | 0
      0
      --
      3
```

> x % y → 0
>
> x is completely divisible by y

x/5 → 2.0

x/10 →

243/10 → 24.3

x//10 → x without last digit

243//10 → 24

x%10 → last digit

243%10 → 3

5.5 % 3 → 2.5

/ 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

Logical Operators

not
and
or

logical operators must be written in lowercase only.

not True → False
not False → True

True and True → True
True and False → False
False and X → False

False or False → False
False or True → True
True or X → True

Every non zero value → True

Zero → False

Non empty string → True

Empty string → False

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

Bitwise Operators

& | ^ ~ >> <<

0 & 0 → 0
0 & 1 → 0
1 & 0 → 0
1 & 1 → 1

0 | 0 → 0
0 | 1 → 1
1 | 0 → 1
1 | 1 → 1

x = 25 & 37

25 = 0 1 1 0 0 1
37 = 1 0 0 1 0 1
1 = 0 0 0 0 0 1

x = 44 | 71

44 = 0 1 0 1 1 0 0
71 = 1 0 0 0 1 1 1
111 = 1 1 0 1 1 1 1

$0 \wedge 0 \rightarrow 0$

$0 \wedge 1 \rightarrow 1$

$1 \wedge 0 \rightarrow 1$

$1 \wedge 1 \rightarrow 0$

$x = 56 \wedge 29$

$56 = 111000$

$29 = 011101$

$37 = 100101$

$\sim 0 \rightarrow 1$

$\sim 1 \rightarrow 0$

$x = \sim 5$ → 0 = +ve, 1 = -ve

$5 =$ 0 0000 0101

$\sim 5 =$ 1 1111 1010 $= -k = -6$

2's → 0000 0110 $= 6$

$k = b1$ ⟲ 2's

$-k = b2$

$k = 1101$

1's 0010

+ 1

2's 0011

$-k =$

35 >> 2

35 = 00100011

1000 = 8

12 << 3

12 = 1100000 = 96

Assignment Operator

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

>>= <<= %=

$x = 4$

↑ must be a variable

$3 = 4$ Error

$x = x + 3$

↑ Container ↑ content

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

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

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

$x \hat{}= 10 \rightarrow x = x \hat{} 10$

$x++$ 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
```

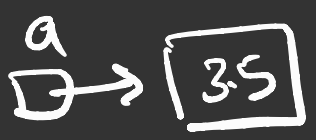
Identity operater

- is
- is not

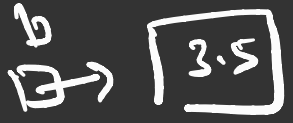
It checks whether the two references refering to the same object or no.

It results in True or false.

a = 3.5

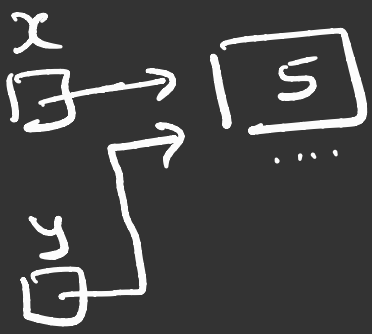


b = 3.5



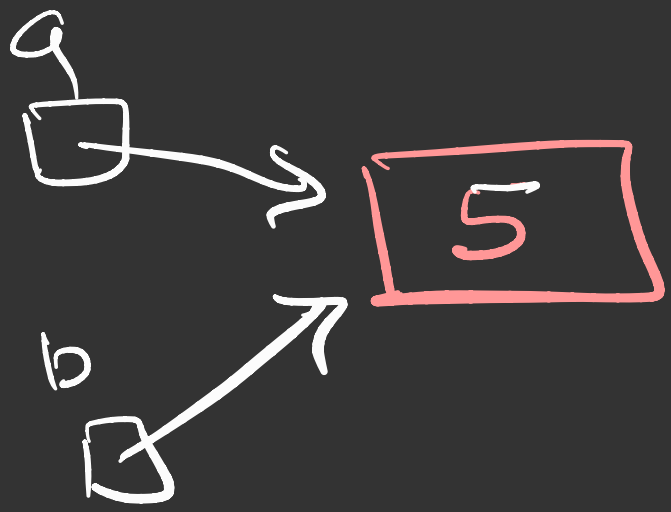
x = 5

y = 5



a = 5

b = 5



reference_count

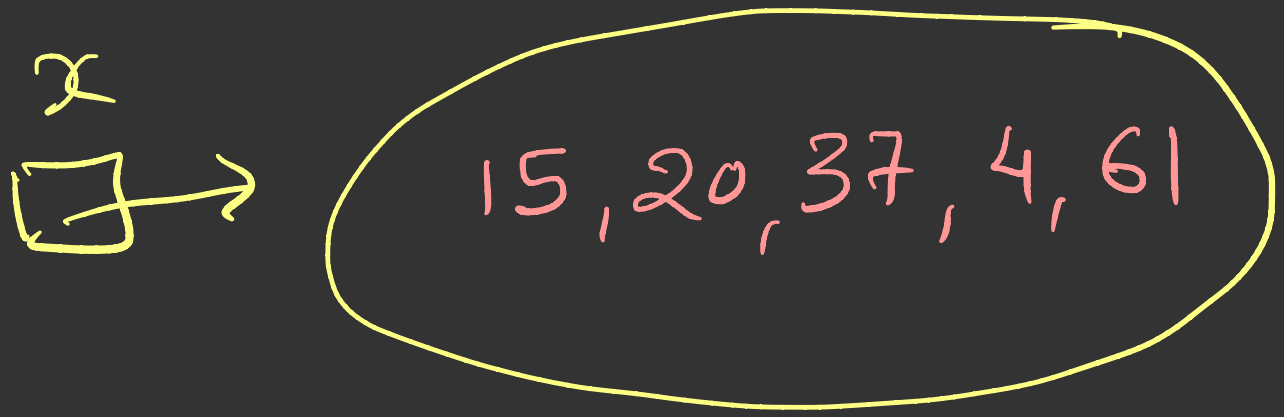
2

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



15 in x True

25 in x False

int, float, complex, bool are not iterable

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