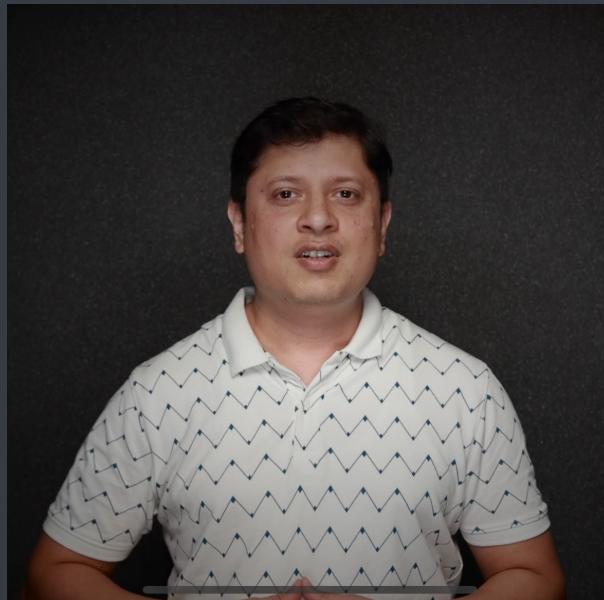


# Full stack web development using python

## 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  $=, +=, -=, /=, //=, \%., **=$   
 $*=, &=, |=, \wedge=, >>=, <<=$
- Identity Operators  $\text{is}, \text{is not}$
- Membership Operators  $\text{in}, \text{not in}$

no  $++, --$  operators in Python

## Arithmetic Operators

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

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

$$-2^2$$

$$-4$$

$$2^{**\underline{3**2}}$$

$$2^{**9}$$

$$2^9 = 512$$

$a / b \rightarrow \text{float}$

$a // b \rightarrow \text{float}$       integer

any of  $a & b$  is float       $\rightarrow \text{float}$

$a & b$  int       $\rightarrow$  int

\*  
+  
-  
%

$5 // 2$       2

$5.0 // 2$       2.0

/ always return float result

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

+,\* can be used with str type  
values also

## 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

& | ^ ~ >> <<

$$0 \& 0 \rightarrow 0$$

$$0 \& 1 \rightarrow 0$$

$$1 \& 0 \rightarrow 0$$

$$1 \& 1 \rightarrow 1$$

$$0 | 0 \rightarrow 0$$

$$0 | 1 \rightarrow 1$$

$$1 | 0 \rightarrow 1$$

$$1 | 1 \rightarrow 1$$

$$5 \& 6$$

$$4$$

$$\begin{aligned} 5 &= 101 \\ 6 &= 110 \\ 4 &= \underline{100} \end{aligned}$$

$$25 \mid 56$$

$$57$$

$$\begin{aligned} 25 &= 011001 \\ 56 &= \underline{111000} \\ 57 &= 111001 \end{aligned}$$

$$0 \wedge 0 \rightarrow 0$$

$$0 \wedge 1 \rightarrow 0$$

$$1 \wedge 0 \rightarrow 0$$

$$1 \wedge 1 \rightarrow 1$$

---

$$25 >> 2$$

6

$$25 << 3$$

200

$$12 \wedge 28$$

$$16$$

$$12 =$$

$$01100$$

$$28 =$$

$$\begin{array}{r} 11100 \\ \hline 10000 \end{array}$$

$$16 =$$

$$\begin{array}{r} \longrightarrow \\ 11001 \end{array}$$



$$\begin{array}{c} 11001000 \\ \downarrow \quad \downarrow \quad \downarrow \\ 128 \quad 64 \quad 32 \end{array}$$

$\sim 0 \rightarrow 1$

$\sim 1 \rightarrow 0$

$$25 = 00011001$$

$$-25 = 11100111$$

$2^s$

$\sim 25$

$\underline{-26}$

$\sim 25$

$$25 = 00011001$$

$$\sim 25 = \boxed{11100110 = -K}$$

$\downarrow 2's$

$$00011001$$

$+ 1$

$$2's \quad \boxed{00011010 = K}$$

$$b_1 = 11001$$

$2s$

$$b_2 = 00111$$

$$1's \quad \begin{array}{r} 00111 \\ 11000 \\ +1 \\ \hline 2's \quad 11001 \end{array}$$

$2's = 1's + 1$

$$1's \quad \begin{array}{r} 00110 \\ +1 \\ \hline 2's \quad 00111 \end{array}$$

# Assignment Operator

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

>>= <<= %=

a = 5



5 = a error

a = a + 2

↑      ↑  
Container content

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

=

a = 5

a + 4

a \* 6

a & 5

a = a + 4

a = a \* 6

a = a & 5

## Identity operator

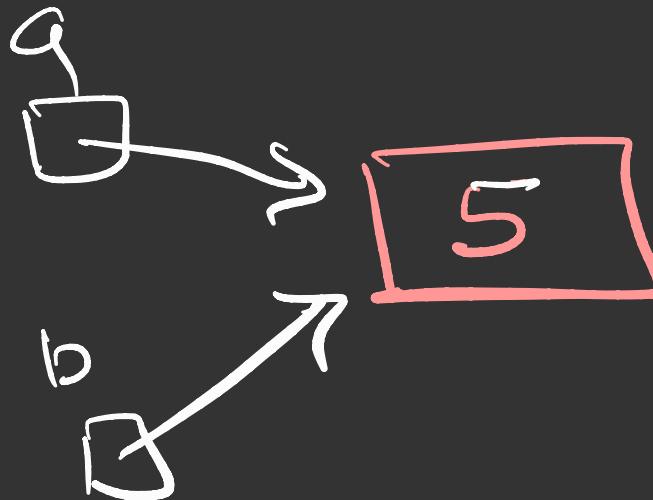
- is
- is not

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

It results in True or False.

$a = 5$

$b = 5$



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