

# Operators

## Operators

### Arithmetic Operators

- + (Addition)
- - (Subtraction)
- \* (Multiplication)
  - Can also be used to repeat strings: "A" \* 10 gives "AAAAAAAAAA"
- / (True Division)
  - The normal division, it gives the answer in decimal points. E.g: 25/2 = 12.5
- // (Floor division/Integer division)
  - Gives only the *quotient* part of division. E.g: 25//2 = 12
- % (Modulus)
  - Gives only the remainder part of division. e.g: 25%2 = 1
  - To get last digit of a number, do number%10 e.g: 1234%10 = 4
  - To check whether m is divisible by n, do m%n==0 e.g: 625%5==0 -> True
- \*\* (Power)
  - a\*\*b is equivalent to a<sup>b</sup>. e.g: 2\*\*10 = 2<sup>10</sup> = 1024

### Assignment Operators

- Main assignment operator -> =
  - It is always **variableName** = **value**, not the other way round
- Shorthand assignment operators:
  - a += b is same as a = a + b
  - a -= b is same as a = a - b
  - and so on

### Relational Operators

Used to find the relation between two expressions, whether they are equal, not equal, one is greater than the other, and so on.

- Greater than (>)
- Lesser than (<)
- Greater than or equal to (>=)
- Less than or equal to (<=)
- Not equal to (!=)
- Equal to (==)
  - = operator and == operator are NOT the same.

### Identity Operators

Used to check whether two values refer to the same memory location, that is, if they are located in the same memory location. They are is and is not.

Case 1	Case 2	Case 3
a = 10 b = 10 c = 20 print(a is b) print(a is not c)	a = [1,2,3] b = [1,2,3]  print(a is b) print(a is not b)	a = [1,2,3] b = [1,2,3] print(a is b) a = b print(a is b)
Output: True True	Output: False True	Output: False True

### Membership Operators

Used to determine whether a particular element belongs in a sequence or not. Sequences include strings, lists, sets, tuples and any other collection of values.

The membership operators are in and not in.

E.g:

a = [1, 2, 3]

b = 4

print(b in a, 10 in a)

Output: True False

### Logical Operators

Used to build complex Boolean expressions along with relational operators. They compare one or two Boolean expressions and give a single output.

- Logical AND operator (and)

Returns True only if both sides are True, else returns False. If expression is a and b,

a	b	Result
False	False	False
False	True	False
True	False	False
True	True	True

- Logical OR operator (or)

Returns True if at least one side is True, else returns False. If expression is a or b,

a	b	Result
False	False	False
False	True	True
True	False	True
True	True	True

- Logical NOT operator (not)

Returns True if value is False, or returns False if value is True. If expression is not a,

a	Result
True	False
False	True