Types of Operators in Python (Simple Explanation)

Python has different types of operators to perform various operations. Here's a complete list:

1. Arithmetic Operators (For math calculations)

Operator | Meaning | Example (a = 10, b = 3) | Output

+ | Addition | a + b | 13

- | Subtraction | a - b | 7

* | Multiplication | a * b | 30

/ | Division (gives decimal result) | a / b | 3.333

// | Floor Division (gives whole number) | a // b | 3

% | Modulus (gives remainder) | a % b | 1

** | Exponentiation (power) | a ** b | 1000

2. Assignment Operators (For assigning values)

Operator | Meaning | Example (x = 5) | Output

= | Assign value | x = 10 | x = 10

+= | Add and assign | x += 3 (same as x = x + 3) | 8

-= | Subtract and assign | x -= 2 | 3

*= | Multiply and assign | x *= 2 | 10

/= | Divide and assign | x /= 2 | 2.5

//= | Floor divide and assign | x //= 2 | 2

%= | Modulus and assign | x %= 2 | 1

**= | Power and assign | x **= 3 | 125

3. Comparison Operators (For checking conditions)

Operator | Meaning | Example (a = 10, b = 5) | Output

== | Equal to | a == b | False

!= | Not equal to | a != b | True

> | Greater than | a > b | True

< | Less than | a < b | False

>= | Greater than or equal to | a >= b | True

<= | Less than or equal to | a <= b | False

4. Logical Operators (For combining conditions)

Operator | Meaning | Example (x = True, y = False) | Output

and | True if both are True | x and y | False

or | True if at least one is True | x or y | True

not | Reverses condition | not x | False

5. Bitwise Operators (For working with binary numbers)

Operator | Meaning | Example (a = 5 (0101), b = 3 (0011)) | Output

& | AND (1 if both bits are 1) | a & b | 1 (0001)

| | OR (1 if at least one bit is 1) | a | b | 7 (0111)

^ | XOR (1 if bits are different) | a ^ b | 6 (0110)

~ | NOT (Flips bits) | ~a | -6

<< | Left Shift (Shifts bits left) | a << 1 | 10 (1010)

>> | Right Shift (Shifts bits right) | a >> 1 | 2 (0010)

6. Membership Operators (For checking if a value is in a list, string, etc.)

Operator | Meaning | Example (x = [1, 2, 3]) | Output

in | True if value exists | 2 in x | True

not in | True if value doesn't exist | 5 not in x | True

7. Identity Operators (For checking if two variables refer to the same object)

Operator | Meaning | Example (a = [1, 2], b = a, c = [1, 2]) | Output

is | True if both refer to the same object | a is b | True

is not | True if they are different objects | a is not c | True

Summary Table

Type | Operators | Purpose

Arithmetic | + - * / // % ** | Math calculations

Assignment | = += -= *= /= //= %= **= | Assign values

Comparison | == != > < >= <= | Compare values

Logical | and or not | Combine conditions

Bitwise | & | ^ ~ << >> | Work with binary numbers

Membership | in, not in | Check if value exists

Identity | is, is not | Check if objects are the same