Types of operators in Python

- ✓ Arithemaic operators
- ✓ Comparision / Relational operators
- Logical operators
- Assignment operators
- Identity operators
- Membership Operators
- Binary operators

Logical Operators in Python

Logical operators are used to combine 2 or more conditions

Operator	Meaning
and	It will return true when both conditions are true
or	It will returns true when at-least one of the condition is true
not	If the condition is true, logical NOT operator makes it false

Let us see some examples

a = 40
b = 20
c = 50

a > b and a > c

False

a = 40
b = 20
c = 50

a > b and c > a

True

Assignement operator

Python assignment operator is used to assign values to declared values.

(=) most commonly used in python

Eg

```
a = 10
b = True
x, y, z = 10, 20, 30
```

Compound Assignment Operator

Python allows us to combine arithematic and bitwise operator with assignment operator

eg.

$x = x + 5$

can be written as

$x += 5 \quad \Rightarrow x = x + 5$

Examples of using Compound assignment

Operator	Example	Meaning
+=	x+=5	x=x+5
-=	x-=5	x=x-5
*=	x*=5	x=x*5
/=	x/=5	x=x/5
%=	x%=5	x=x%5
//=	x//=5	x=x//5
**=	x**=5	x=x**5
&=	x&=5	x=x&5
!=	x!=5	x=x!5
^=	x^=5	x=x^5
>>=	x>>=5	x=x>>5
<<=	x<<=5	x=x<<5

★ Python does not have any increment operator like ++ or --

++10 ⇒ +(+10) ⇒ 10

but `a++` is an error as it doesn't make any sense.

Identity Operator

Identity operators in Python are **is** and **is not**.

They serve 2 purposes -

1. To verify if the reference point to the same memory location or not

2. To determine if the value is of certain class or type.

Behaviour of is and is not

is :- is returns True if the operands are identical else false

is not :- is not returns True if operands are not identical else false

Example: id( )

```
a=2
b=3

c= a is b
print(c)
```

False

```
a=2
b=2
c= a is b

print(c)
```

True

a is int ??

type(a) is int ?

Difference between is & ==

Equality operator (==) is used to compare the values.

Identity operator (is) is used to compare the IDs.