

# Operators

Operators are special symbols in Python that carry out arithmetic or logical computation. The value that the operator operates on is called the operand.

## Operator Types

1. Arithmetic operators
2. Comparison (Relational) operators
3. Logical (Boolean) operators
4. Bitwise operators
5. Assignment operators
6. Special operators

## Arithmetic Operators

Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication etc.

`+, -, *, /, %, //, **` are arithmetic operators

Example:

```
In [3]: x, y = 10, 20

#addition
print(x + y)

#subtraction(-)

#multiplication(*)

#division(/)

#modulo division (%)

#Floor Division (//)

#Exponent (**)
```

30

## Comparision Operators

Comparison operators are used to compare values. It either returns True or False according to the condition.

`>, <, ==, !=, >=, <=` are comparision operators

```
In [4]: a, b = 10, 20

print(a < b)  #check a is less than b

#check a is greater than b

#check a is equal to b

#check a is not equal to b (!=)

#check a greater than or equal to b

#check a less than or equal to b

True
```

## Logical Operators

Logical operators are **and**, **or**, **not** operators.

```
In [5]: a, b = True, False

#print a and b
print(a and b)

#print a or b

#print not b

False
```

## Bitwise operators

Bitwise operators act on operands as if they were string of binary digits. It operates bit by bit

`&, |, ~, ^, >>, <<` are Bitwise operators

```
In [7]: a, b = 10, 4

#Bitwise AND
print(a & b)

#Bitwise OR

#Bitwise NOT

#Bitwise XOR

#Bitwise rightshift

#Bitwise Leftshift

0
```

## Assignment operators

Assignment operators are used in Python to assign values to variables.

`a = 5` is a simple assignment operator that assigns the value 5 on the right to the variable `a` on the left.

`=, +=, -=, *=, /=, %=, //=, **=, &=, |=, ^=, >>=, <<=` are Assignment operators

```
In [8]: a = 10

a += 10      #add AND
print(a)

#subtract AND (-=)

#Multiply AND (*=)

#Divide AND (/=)

#Modulus AND (%=)

#Floor Division (//=)

#Exponent AND (**=)

20
```

## Special Operators

# Identity Operators

**is** and **is not** are the identity operators in Python.

They are used to check if two values (or variables) are located on the same part of the memory.

```
In [9]: a = 5
        b = 5
        print(a is b)      #5 is object created once both a and b points to same object

        #check is not

True
```

```
In [10]: l1 = [1, 2, 3]
         l2 = [1, 2, 3]
         print(l1 is l2)

False
```

```
In [12]: s1 = "Satish"
         s2 = "Satish"
         print(s1 is not s2)

False
```

# MemberShip Operators

**in** and **not in** are the membership operators in Python.

They are used to test whether a value or variable is found in a sequence (string, list, tuple, set and dictionary).

```
In [13]: lst = [1, 2, 3, 4]
         print(1 in lst)      #check 1 is present in a given list or not

         #check 5 is present in a given list

True
```

```
In [16]: d = {1: "a", 2: "b"}
         print(1 in d)

True
```