Operators in Python

- Arithmetic Operators
- Assignment Operators
- Relational Operators
- Logical Operators
- Unary operators
- Bitwise Operators

Defination :- Operators are used to perform on variables and values.

Arithmetic Operator :- Arithmetic operators are used to with numeric values to perform common mathematical operations. $x = 3;\ y = 8$

- Addition (+) Eg :- x + y // 13
- subtraction (-) Eg :- x - y // -3
- Multiplication (*) Eg :- x * y // 40
- Division (/) Eg :- x / y // 0.625
- Modulus (%) Eg :- x % y // 5
- Exponential (**) Eg : x ** y // 390625
- Flour division (//) Eg : x // y // 0

Assignment operators :-

Assignment operators is used to assign values to variables.

Operators.	Examples	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3
:=	~~x := 3~~	~~x = x~~ x = 3

print (x := 3)

print (x)

Relational Operators.

Relational operators are used to compare two values.

Operator	Name	Example.
==	Equal	x == y
!=	Not Equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than / Equal to	x >= y
<=	less than / Equal to	x <= y.

Logical Operators.

Logical Operators are used to combine conditional statements.

Operator	Description	Example.
and	Returns "True", if both statements are true	x < 5 and x < 10.

or — Returns "True", if one of the statement is true — $x < 5$ or $x < 4$

not — Reverse the result, returns False if the result is true. — not ($x < 5$ and x

Truth Tables

AND

x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	xy
0	0	0
0	1	1
1	0	1
1	1	1

NOT

x	x'
0	1
1	0

Unary Operators.

Unary operators in python are operators that operate on a single ~~command~~ operand.

They are typically used to alter the value.

- Unary plus (+) => returns the +ve value
- Unary minus (-) => returns the -ve value.

Bitwise Operators:-

Bitwise Operators are used to compare (binary) numbers.

Operator	Name	Description.
&	And	Set each bit to 1, if both bits are 1
|	Or	Sets each bit to 1, if one of two bits is 1
^	XOR	Sets each bit to 1, if only one of two bits is 1
~	NOT	Inverts all the bits.

<<	left shift	shift left by pushing zeros in from the right and let the leftmost bits fall off.
>>	Right shift	shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off.

Number system in python

- A number system refers to how numbers and stored in different bases.

- Types of number system.

System	Base	Digits used	Example
Decimal	10	0-9	25
Binary	2	0, 1	0b1101
Octal	8	0-7	0o31
Hexadecimal	16	0-9, A-F	0x1F

prefix = . Decimal has no prefix

. Binary prefix is "0b or 0B"

. Octal prefix is "0o/0O"

. Hexadecimal prefix is "0x/0X".

Swap Variables in Python:

```
a = 5
b = 6
```

// method (1) for swaping using "temp"

```
temp = a
a = b
b = temp
```

~~output~~ :

```
print (a)
print (b)
```

o/p:- 6
5

// method (2) of swaping using operators.

```
a = 5
b = 6

a = a + b
b = a - b
a = a - b
```

```
print (a)
print (b)
```

o/p:- 6
5

// method ③ of swaping using "XOR"

```
a = a ^ b
b = a ^ b
a = a ^ b.
```

// method ④ of swaping using with no extra variable / operators. (simple method)

```
a, b = b, a.
```