04-09-25

Operators:

* Operators are used to perform operation on variables and values

Eg: print(10+15) → Operator
10 + 15
Operands

Types of Operators:- [7 types]

1) Arithmetic Operators:

[+, -, *, /, %, **, //]

% → Modulus [x%5]
** → Exponentiation x**y
// → Floor division [x//y]

2) Assignment Operators:

* It is used to assign values to the variables

[=, +=, -=, *=, /=, %=, //=, **=, &=, |=, ^=, ^=, >>=, <<=, :=]

3) Comparison Operators:-

It is used to compare two values

== → Equal
!= ⇒ Not Equal
> ⇒ Greater Than
< ⇒ Less than
>= ⇒ Greater than or equal to
<= ⇒ Less than or equal to

4) LOGICAL OPERATORS:-

It is Used to Combine the Conditional Statement

Oper

and → Returns true if both Condition Statement are true

or => Return true if One condition is true

not => Reverse the result, returns false if the result is true

5) Identity Operator:-

*It is Used to compare the objects, not if they are equal but if they are actually the same object with the same memory location

Oper

i) is: Return true if the bothe variables are the same object

Eg: x is y

ii) is not: Returns true if both variables are Not the same object

Eg:. x is not y

6) Membership Operator:

It is used to test if a sequence is present in an object.

Opes

i) In: Returns true if a sequence with the specified value present in the object

Eg: x in y

ii) not in: Returns true if a sequence with the specified value is not present in the object

7) Bituise Operator:-

* It is used to compare the (binary) numbers.

Opes

i) & (AND) ⇒ Set each bit to 1 if both bits are '1'

ii) | (OR) ⇒ Set each bit to 1 if one of the two bits is '1'

Truth Table

		And	OR
1	1	1	1
1	0	0	1
0	0	0	0
0	1	0	1

iii) ^ [XOR]: Set each bit to 1 if only one of two bits is 1

iv) ~ [NOT]: Inverse all the bits

v) << → Left shift

vi) >> → Right shift

Operator

* ** → Exponentiation

* () → Parentheses

* +x, -x, ~x → Unary plus, Unary minus, and bitwise NOT