Operators in Python

Operations:

It is an in-built task performed on operands(values) based on the operator used.

Operand:

They are the values (or variable holding value) involved in an operation

Operator:

They are special symbols or keywords which has got in-built functionalities

There are totally **7 types** of operators:

1. Arithmetic operator
2. Comparison operator
3. Logical operator
4. Bitwise operator
5. Assignment operator
6. Identity operator
7. Membership Operator

Let’s see the operators individually

1. Arithmetic Operators

Arithmetic operators perform mathematical operations on numeric values.

Operator	Description	Example	Output
+	Addition	10 + 5	15
-	Subtraction	10 - 5	5
*	Multiplication	10 * 5	50
//	Floor Division (Quotient without decimals)	11 // 2	5
/	Float Division (Quotient with decimals)	11 / 2	5.5
%	Modulus (Remainder)	11 % 2	1
**	Exponentiation (Power)	3 ** 2	9

2. Comparison (Relational) Operators

Comparison operators return a Boolean value (True or False).

Operator	Description	Example	Output
>	Greater than	10 > 5	True
<	Less than	10 < 5	False
>=	Greater than or equal to	10 >= 10	True
<=	Less than or equal to	5 <= 3	False
==	Equal to	10 == 10	True
!=	Not equal to	10 != 5	True

3. Logical Operators

Logical operators combine multiple boolean expressions.

Operator	Description	Example	Output
and	Returns True if both conditions are True	(10 > 5) and (5 > 3)	True
or	Returns True if at least one condition is True	(10 > 5) or (5 < 3)	True
not	Reverses boolean value	not(10 > 5)	False

Truth Table for Logical Operators:

Logical AND (and)

Condition 1	Condition 2	Output
True	True	True
True	False	False
False	True	False
False	False	False

Logical OR (or)

Condition 1	Condition 2	Output
True	True	True
True	False	True
False	True	True
False	False	False

Logical NOT (not)

Input	Output
True	False
False	True

4. Assignment Operators

Assignment operators assign values to variables.

Operator	Description	Example	Equivalent To
=	Assign	a = 10	a = 10
+=	Add and assign	a += 5	a = a + 5
-=	Subtract and assign	a -= 5	a = a - 5
*=	Multiply and assign	a *= 5	a = a * 5
/=	Divide and assign	a /= 5	a = a / 5
//=	Floor divide and assign	a //= 5	a = a // 5
%=	Modulus and assign	a %= 5	a = a % 5
**=	Exponentiate and assign	a **= 5	a = a ** 5

Bitwise Assignment Operators

Operator	Example	Equivalent To
&=	a &= 5	a = a & 5
`	=`	`a
^=	a ^= 5	a = a ^ 5
<<=	a <<= 5	a = a << 5
>>=	a >>= 5	a = a >> 5

5. Bitwise Operators

Bitwise operators work at the binary level.

Operator	Name	Example	Binary Equivalent
&	Bitwise AND	5 & 3	0101 & 0011 = 0001 (1)
`	`	Bitwise OR	`5
^	Bitwise XOR	5 ^ 3	0101 ^ 0011 = 0110 (6)
~	Bitwise NOT	~5	~0101 = 1010 (-6 in two's complement)
<<	Left Shift	5 << 1	0101 << 1 = 1010 (10)
>>	Right Shift	5 >> 1	0101 >> 1 = 0010 (2)

6. Identity Operators

Identity operators compare **memory locations** of objects.

Operator	Description	Example	Output
is	Returns True if both variables refer to the same object	a = [1,2,3] b = a a is b	True
is not	Returns True if they refer to different objects	a = [1,2,3] b = [1,2,3] a is not b	True

7. Membership Operators

Membership operators check if a value is present in an **iterable** (string, list, tuple, dictionary)

Operator	Description	Example	Output
in	Returns True if the value exists in the iterable	"a" in "apple"	True
not in	Returns True if the value does not exist in the iterable	"z" not in "apple"	True