Operators in Python

Operators are **symbols or keywords** that perform operations on **variables and values**.

```
x = 5
y = 2
print(x + y)   # → 7
```

Here, + is an **operator**, and x & y are **operands**.

Types of Operators in Python

No.	Type	Description
1.	**Arithmetic Operators**	Perform mathematical operations
2.	**Comparison (Relational) Operators**	Compare two values
3.	**Assignment Operators**	Assign values to variables
4.	**Logical Operators**	Combine conditional statements
5.	**Bitwise Operators**	Work on binary numbers (bits)
6.	**Membership Operators**	Test for sequence membership (in, not in)
7.	**Identity Operators**	Compare memory locations (is, is not)

1. Arithmetic Operators

These perform **basic math operations**.

Operator	Description	Example	Output
+	Addition	5 + 3	8
-	Subtraction	5 - 2	3
*	Multiplication	4 * 3	12
/	Division (float)	5 / 2	2.5
//	Floor Division (integer result)	5 // 2	2
%	Modulus (remainder)	5 % 2	1
**	Exponentiation (power)	2 ** 3	8

Example:

```
a = 10
b = 3

print(a + b)   # 13
```

Operators in Python

```
print(a - b)   # 7
print(a * b)   # 30
print(a / b)   # 3.3333
print(a // b)  # 3
print(a % b)   # 1
print(a ** b)  # 1000
```

2. Comparison (Relational) Operators

Used to **compare two values** — always return True or False.

Operator	Description	Example	Outpu
==	Equal to	5 == 5	True
!=	Not equal to	5 != 3	True
>	Greater than	5 > 3	True
<	Less than	5 < 3	False
>=	Greater than or equal to	5 >= 5	True
<=	Less than or equal to	4 <= 5	True

Example:

```
x = 10
y = 20

print(x == y)  # False
print(x != y)  # True
print(x > y)   # False
print(x < y)   # True
print(x >= 10) # True
print(x <= 15) # True
```

3. Assignment Operators

Used to **assign or update values** in variables.

Operator	Example	Same As
=	x = 5	Assign 5 to x
+=	x += 3	x = x + 3
-=	x -= 2	x = x - 2
*=	x *= 4	x = x * 4
/=	x /= 2	x = x / 2
//=	x //= 3	x = x // 3
%=	x %= 2	x = x % 2

Operators in Python

Operator	Example	Same As
**=	x **= 2	x = x ** 2

Example:

```
x = 10
x += 5
print(x)  # 15

x -= 3
print(x)  # 12

x *= 2
print(x)  # 24

x //= 5
print(x)  # 4
```

4. Logical Operators

Used to **combine multiple conditions.**

Operator	Description	Example	Result
And	True if both conditions are true	(x > 5 and x < 15)	True
or	True if at least one condition is true	(x > 15 or x == 10)	True
not	Reverses the result	not(x > 5)	False

Example:

```
x = 10

print(x > 5 and x < 15)   # True
print(x > 15 or x == 10)  # True
print(not(x > 5))         # False
```

5. Bitwise Operators

Operate on **binary numbers (bits)** — 0s and 1s.

Operators in Python

Operator	Description	Example	Binary Example	Output
&	AND	5 & 3	0101 & 0011	1
`	`	OR	`5	3`
^	XOR	5 ^ 3	0101 ^ 0011	6
~	NOT	~5	~0101	-6
<<	Left shift	5 << 1	0101 → 1010	10
>>	Right shift	5 >> 1	0101 → 0010	2

Example:

```
a = 5   # 0101
b = 3   # 0011

print(a & b)  # 1
print(a | b)  # 7
print(a ^ b)  # 6
print(~a)     # -6
print(a << 1) # 10
print(a >> 1) # 2
```

6. Membership Operators

Used to check **if a value exists** in a sequence (string, list, tuple, etc.)

Operator	Description	Example	Output
in	Returns True if value found	'a' in 'apple'	True
not in	Returns True if value not found	'x' not in 'apple'	True

Example:

```
fruits = ["apple", "banana", "cherry"]

print("apple" in fruits)      # True
print("mango" not in fruits)  # True
print("mango" in fruits)      # False
```

7. Identity Operators

Used to **compare memory locations** of two objects.

Operators in Python

Operator	Description	Example	Output
is	True if both refer to the same object	x is y	True / False
is not	True if they are different objects	x is not y	True / False

Example:

```
x = [1, 2, 3]
y = [1, 2, 3]
z = x

print(x is z)      # True  (same object)
print(x is y)      # False (different objects, same value)
print(x == y)      # True  (same content)
print(x is not y)  # True
```

Explanation:

- is → compares **object identity** (memory address)
- == → compares **values**

Summary Table

Type	Example	Description
Arithmetic	+ - * / % ** //	Basic math operations
Comparison	== != > < >= <=	Compare values
Assignment	= += -= *= /= //= %= **=	Assign values
Logical	and or not	Combine conditions
Bitwise	`&	^ ~ << >>`
Membership	in, not in	Test membership in sequence
Identity	is, is not	Compare object identities

Solved Example: All in One

```
a = 10
b = 5
list1 = [1, 2, 3, 4, 5]

print(a + b)       # Arithmetic → 15
print(a > b)       # Comparison → True
a += b             # Assignment
```

Operators in Python

```
print(a)              # → 15
print(a > 5 and b < 10)  # Logical → True
print(a & b)          # Bitwise → 0
print(3 in list1)     # Membership → True
print(a is b)         # Identity → False
```

Output:

```
15
True
15
True
0
True
False
```

Assignment

Ques 1: Write a Python program to input two numbers and perform all arithmetic operations.

Sample Output:

```
a = 15
b = 4
Addition: 19
Subtraction: 11
Multiplication: 60
Division: 3.75
Floor Division: 3
Modulus: 3
Exponentiation: 50625
```

Ques 2: Check if a number entered by the user is greater than, equal to, or less than 10 using comparison operator.

```
num = 8

print(num == 10) - False
print(num != 10) - True
print(num > 10) - False
print(num < 10) - True
print(num >= 10) - False
print(num <= 10) - True
```