🐍Python Operators (Deep Dive)

Operators are **symbols** that perform operations on variables and values.
Example: `2 + 3` → here `+` is an operator.

---

1. Arithmetic Operators

Used for basic mathematical operations.

Operator	Meaning	Example	Output
`+`	Addition	`10 + 5`	`15`
`-`	Subtraction	`10 - 5`	`5`
`*`	Multiplication	`10 * 5`	`50`
`/`	Division (float result)	`10 / 3`	`3.333...`
`//`	Floor division	`10 // 3`	`3`
`%`	Modulus (remainder)	`10 % 3`	`1`
`**`	Exponent (power)	`2 ** 3`	`8`

Example:

```
a, b = 10, 3
print(a + b)  # 13
print(a // b) # 3
print(a % b)  # 1
```

---

2. Comparison / Relational Operators

Used to compare two values → returns `True` or `False`.

Operator	Meaning	Example	Output
`==`	Equal to	`5 == 5`	`True`
`!=`	Not equal	`5 != 3`	`True`
`>`	Greater than	`10 > 3`	`True`

Operator	Meaning	Example	Output
`<`	Less than	`10 < 3`	`False`
`>=`	Greater than or equal to	`5 >= 5`	`True`
`<=`	Less than or equal to	`3 <= 5`	`True`

Example:

```
x, y = 10, 20
print(x > y)   # False
print(x != y)  # True
```

---

3. Logical Operators

Used to combine conditional statements.

Operator	Meaning	Example	Output
`and`	True if both are true	`(5 > 3 and 10 > 5)`	`True`
`or`	True if at least one is true	`(5 > 3 or 10 < 5)`	`True`
`not`	Negation	`not(5 > 3)`	`False`

Example:

```
a, b = True, False
print(a and b) # False
print(a or b)  # True
print(not a)   # False
```

---

4. Assignment Operators

Used to assign values to variables.

Operator	Meaning	Example	Equivalent
`=`	Assign	`x = 5`	`x = 5`
`+=`	Add and assign	`x += 3`	`x = x + 3`

Operator	Meaning	Example	Equivalent
`-=`	Subtract and assign	`x -= 3`	`x = x - 3`
`*=`	Multiply and assign	`x *= 2`	`x = x * 2`
`/=`	Divide and assign	`x /= 2`	`x = x / 2`
`//=`	Floor divide and assign	`x //= 2`	`x = x // 2`
`%=`	Modulus and assign	`x %= 2`	`x = x % 2`
`**=`	Power and assign	`x **= 3`	`x = x ** 3`

Example:

```
x = 10
x += 5  # x = 15
x *= 2  # x = 30
```

---

5. Membership Operators

Used to test if a value exists in a sequence (string, list, tuple, set, dict).

Operator	Meaning	Example	Output
`in`	True if value present	`"a" in "apple"`	`True`
`not in`	True if value not present	`"z" not in "apple"`	`True`

Example:

```
fruits = ["apple", "banana", "mango"]
print("apple" in fruits)     # True
print("grape" not in fruits) # True
```

---

6. Identity Operators

Used to compare **memory location** of objects.

Operator	Meaning	Example	Output
`is`	True if both refer to same object	`x is y`	True/False

Operator	Meaning	Example	Output
`is not`	True if not same object	`x is not y`	True/False

Example:

```
a = [1, 2, 3]
b = a
c = [1, 2, 3]

print(a is b)      # True (same object reference)
print(a is c)      # False (different objects with same content)
print(a == c)      # True (values are same)
```