

# Operators Operands

## Operators and Operands in Python

In Python, **operators** are symbols used to perform operations on variables and values (known as **operands**). Python supports various types of operators, including **arithmetic**, **comparison**, **logical**, **assignment**, **boolean**, and **membership** operators.

### 1. Arithmetic Operators

- `+` : Addition
- `-` : Subtraction
- `*` : Multiplication
- `/` : Division (float)
- `//` : Integer Division (floor division)
- `%` : Modulus (remainder)
- `**` : Exponentiation (power)

```
x = 10
y = 3

addition = x + y      # 13
subtraction = x - y   # 7
multiplication = x * y # 30
division = x / y      # 3.333...
integer_division = x // y # 3
modulus = x % y       # 1
power = x ** y        # 1000
```

## 2. Comparison Operators

- == : Equal to
  - != : Not equal to
  - > : Greater than
  - < : Less than
  - >= : Greater than or equal to
  - <= : Less than or equal to

```
x = 10
y = 5

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

### 3. Logical Operators

- and : Logical AND
  - or : Logical OR
  - not : Logical NOT

```
x = 10
y = 5

print(x > 3 and y < 10)    # True, both
conditions are True
print(x > 10 or y < 10)   # True, one
condition is True
print(not(x > 5))         # False, since x > 5
is True, 'not' makes it False
```

## 4. Assignment Operators

- `=` : Assign value
- `+=` : Add and assign
- `-=` : Subtract and assign
- `*=` : Multiply and assign
- `/=` : Divide and assign
- `//=` : Floor divide and assign
- `%=` : Modulus and assign
- `**=` : Exponent and assign

```
x = 10

x += 5  # Equivalent to: x = x + 5 (x becomes 15)
x -= 3  # Equivalent to: x = x - 3 (x becomes 12)
x *= 2  # Equivalent to: x = x * 2 (x becomes 24)
x /= 4  # Equivalent to: x = x / 4 (x becomes 6.0)
```

## 5. Boolean Operators

- `True` : Boolean True value
- `False` : Boolean False value

```
is_sunny = True
is_raining = False

print(is_sunny)    # Output: True
print(is_raining)  # Output: False
```

## 6. Membership Operators

- `in` : Returns True if a value is found in a sequence
- `not in` : Returns True if a value is not found in a sequence

```
fruit = "apple"
print("a" in fruit)      # True, 'a' is in "apple"
print("b" not in fruit)  # True, 'b' is not in
                        "apple"
```