



Python Operators with Examples

12 What are Operators?

Operators are symbols that perform operations on operands.

```
● ● ●  
# Example:  
a = 10  
b = 20 # a and b are operands  
  
print(a + b) # '+' is an operator
```

Arithmetic Assignment Operators

These operators assign values to variables.

```
a = 10
b = 20

print(a + b) # Addition ✓
print(a - b) # Subtraction ✓
print(a / b) # Division ✓ 0.5
print(a * b) # Multiplication ✓
print(a ** b) # Exponentiation
print(a // b) # Floor division 0
print(b % a) # Modulus → 0
```

Handwritten notes:

- 0.5
- $2 \overline{) 10} \text{ (5.0)}$
- $\rightarrow \frac{10}{20}$
- $2/10$
- a^b
- $10 \quad 2$
- Remainder

Comparison Operators

Used to compare two values.

```
a = 10
b = 20
c = 10

print(a < b)      # Less than → True
print(a > b)      # Greater than → False
print(a <= b)     # Less than or equal to → False
print(a >= b)     # Greater than or equal to → False
print(a <= c)     # Less than or equal to → True
print(a >= c)     # Greater than or equal to → True
```

Equality Operators

Checks if two values are equal or not.

```
• • •  
a = 10  
b = 20  
  
print(a == b) # Equal to F  
print(b != a) # Not equal to T
```

Assignment operator

Logical Operators

Used to combine conditional statements.

```
print(True and True) T
print(True and False) F
print(False and True) F
print(False and False) F
```

```
print(True or True) T
print(True or False) T
print(False or True) T
print(False or False) F
```

```
print(not True) F
print(not False) T
```

Compound Assignment Operators

Performs operations and assigns the result back to the variable.



```
print("Before incrementation")  
a = 10  
print(a)  
a += 1 → a = a + 1  
print("After incrementation")  
print(a)
```

a = 10

Membership Operators

Checks if a value exists in a sequence.

```
name = "Python"

print('P' in name) → # True
print('z' not in name) # True
print('p' in name) # False (case-sensitive)
```

 Illustration:  Membership Operators

ID Identity Operators

→ Checks if two variables refer to the same object in memory.

```
a = 10
b = 20

print(a is 10) # True
print(a is not b) # True
print(b is 20) # True
```


Swap Two Variables Without Temp Variable

Python allows swapping values directly.

```
print("Before swapping")
a = 10
b = 20
print("a =", a)
print("b =", b)

a, b = b, a # Swap without temp variable

print("After swapping")
print("a =", a)
print("b =", b)
```

