🧮 Python Operators with Examples

🔢 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
```

✍️ Assignment Operators

These operators assign values to variables.

```
a = 10
b = 20

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

🔄 Comparison Operators

Used to compare two values.

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

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

⚖️ Equality Operators

Checks if two values are equal or not.

```
a = 10
b = 20

print(a == b)  # Equal to
print(b != a)  # Not equal to
```

🧠 Logical Operators

Used to combine conditional statements.

```
print(True and True)
print(True and False)
print(False and True)
print(False and False)

print(True or True)
print(True or False)
print(False or True)
print(False or False)

print(not True)
print(not False)
```

🔄 Compound Assignment Operators

Performs operations and assigns the result back to the variable.

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

🔍 Membership Operators

Checks if a value exists in a sequence.

```python
name = "Python"

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

🖼️ **Illustration:**  Membership Operators

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)
```