**Operators In Python**

Operators in Python are special symbols or keywords that perform operations on variables and values. They are like tools that help us manipulate data in Python.

**Types of Operators in Python:**

Python has several types of operators:

1. Arithmetic Operators
2. Comparison Operators
3. Logical Operators
4. Bitwise Operators
5. Assignment Operators
6. Membership Operators
7. Identity Operators

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + | Addition | x + y |
| - | Subtraction | x - y |
| \* | Multiplication | x \* y |
| / | Division | x / y |
| % | Modulus(Remainder) | x % y |
| \*\* | Exponentiation | x \*\* y |
| // | Floor Division | x // y |

x = 10

y = 3

print(x + y) # Output: 13

print(x % y) # Output: 1 (remainder)

print(x // y) # Output: 3 (quotient)

print(x \*\* y) # Output: 1000

**2. Comparison Operators**

These compare two values and return True or False.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| == | Equal to | x == y |
| != | Not equal to | x != y |
| > | Greater than | x > y |
| < | Less than | x < y |
| >= | Greater than or equal to | x >= y |
| <= | Less than or equal to | x <= y |

x = 10

y = 20

print(x > y) # Output: False

print(x <= y) # Output: True

print(x == 10) # Output: True

**3. Logical Operators**

These are used to combine conditional statements.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| and | Returns True if both conditions are true | (x > 5 and y > 10) |
| or | Returns True if at least one condition is true | (x > 5 or y > 30) |
| not | Reverses the result | not(x > 5) |

x = 10

y = 5

print(x > 5 and y > 3) # Output: True

print(x > 15 or y < 10) # Output: True

print(not(x > 5)) # Output: False

**4. Bitwise Operators**

These work with binary numbers.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| & | AND | x & y |
| | | OR | x | y |
| ^ | XOR | x ^ y |
| ~ | NOT | ~x |
| << | Left Shift | x << 2 |

x = 4

# 0100 in binary

y = 5 # 0101 in binary

print(x & y) # Output: 4 (0100)

print(x | y) # Output: 5 (0101)

print(x << 1) # Output: 8 (shifts left by 1)

**5. Assignment Operators**

These are used to assign values to variables.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Same as** |
| = | x = 3 |  |
| += | x += 3 | x = x + 3 |
| -= | x -= 3 | x = x - 3 |
| %= | x %= 3 | x = x % 3 |
| /= | x /= 3 | x = x / 3 |

x = 5

x += 2

print(x) # Output: 7

x \*= 3

print(x) # Output: 21

**6. Membership Operators**

These check if a value is part of a sequence (like a list or string).

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| in | Returns True if exists | "a" in "apple" |
| not in | Returns False if exists | "b" not in "apple" |

x = [1, 2, 3]

print(2 in x) # Output: True

print(4 not in x) # Output: True

**7. Identity Operators**

These check if two variables refer to the same object.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| is | Returns True if same | x is y |
| is not | Returns True if not | x is not y |

x = [1, 2, 3]

y = x

z = [1, 2, 3]

print(x is y) # Output: True (same object)

print(x is z) # Output: False (different objects)