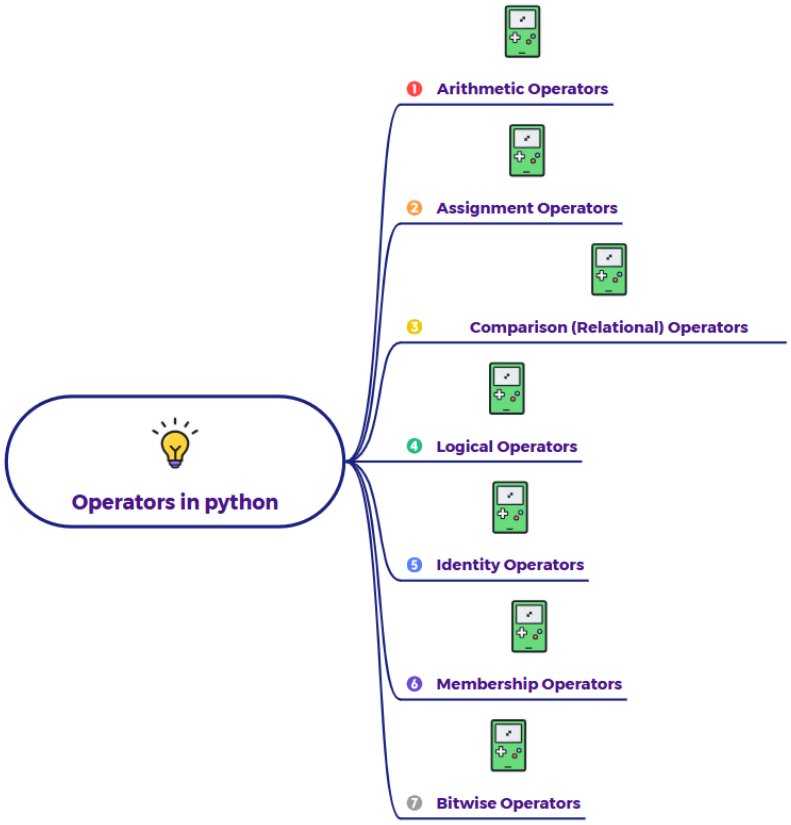
Different operators in python?



In Python, **operators** are special symbols or keywords that perform operations on values and variables (called **operands**).

Here are the different types of operators in Python:

1. **Arithmetic Operators:**
    - Used to perform common mathematical operations.
    - + (Addition): 5 + 3 gives 8
    - - (Subtraction): 5 - 3 gives 2
    - * (Multiplication): 5 * 3 gives 15
    - / (Division): 5 / 2 gives 2.5 (always returns a float)
    - % (Modulo): 5 % 2 gives 1 (remainder of division)
    - ** (Exponentiation): 5 ** 2 gives 25 (5 raised to the power of 2)

- // (Floor Division): 5 // 2 gives 2 (division that rounds down to the nearest whole number)

2. **Assignment Operators:**
    - Used to assign values to variables.
    - = (Assign): x = 10
    - += (Add and assign): x += 5 (same as x = x + 5)
    - -= (Subtract and assign): x -= 5 (same as x = x - 5)
    - *= (Multiply and assign): x *= 5 (same as x = x * 5)
    - /= (Divide and assign): x /= 5 (same as x = x / 5)
    - %= (Modulo and assign): x %= 5 (same as x = x % 5)
    - **= (Exponentiate and assign): x **= 2 (same as x = x ** 2)
    - //= (Floor divide and assign): x //= 2 (same as x = x // 2)
3. **Comparison (Relational) Operators:**
    - Used to compare two values and return a Boolean result (True or False).
    - == (Equal to): 5 == 5 is True, 5 == 3 is False
    - != (Not equal to): 5 != 3 is True
    - > (Greater than): 5 > 3 is True
    - < (Less than): 5 < 3 is False
    - >= (Greater than or equal to): 5 >= 5 is True
    - <= (Less than or equal to): 5 <= 3 is False
4. **Logical Operators:**
    - Used to combine conditional statements.
    - and: Returns True if both statements are true. (True and False is False)
    - or: Returns True if at least one of the statements is true. (True or False is True)

- not: Reverses the result; returns False if the result is true, and vice versa. (not True is False)

5. **Identity Operators:**
    - Used to compare the memory locations of two objects.
    - is: Returns True if both variables point to the *same object* in memory.
    - is not: Returns True if both variables point to *different objects* in memory.
6. **Membership Operators:**
    - Used to test if a sequence (like a string, list, or tuple) contains a specific element.
    - in: Returns True if a value is found in the sequence.
    - not in: Returns True if a value is *not* found in the sequence.
7. **Bitwise Operators:**
    - Used to perform operations on individual bits of integers. These are typically used in lower-level programming, data compression, encryption, etc.
    - & (AND): Sets each bit to 1 if both bits are 1.
    - | (OR): Sets each bit to 1 if at least one of the bits is 1.
    - ^ (XOR): Sets each bit to 1 if only one of the bits is 1.
    - ~ (NOT): Inverts all the bits.
    - << (Left shift): Shifts bits to the left, adding zeros on the right.
    - >> (Right shift): Shifts bits to the right, adding zeros on the left.