Math and Logical Operators in Python

Introduction

In Python, operators are symbols that direct the program to perform specific operations. Similar to how road signs guide drivers, operators help Python execute mathematical calculations, logical decisions, and comparisons.

Math Operators

Math operators in Python are similar to functions in a calculator. They allow for performing basic arithmetic operations:

1. **Addition (`+`)**
    - Combines two numbers.
    - Example: `2 + 3` results in `5`.
2. **Subtraction (`-`)**
    - Subtracts one number from another.
    - Example: `3 - 2` results in `1`.
3. **Division (`/`)**
    - Divides one number by another.
    - Example: `35 / 5` results in `7.0` (returns a float).
4. **Multiplication (`*`)**
    - Multiplies two numbers.
    - Example: `7 * 4` results in `28`.

Logical Operators

Logical operators evaluate conditions and return a Boolean value: `True` or `False`. They are often used in conditional statements to control the program's flow.

1. **AND (`and`)**

- Returns `True` only if **both conditions are true**.
- Example:

```
a = True
b = True
if a and b:
    print("All true")  # Prints "All true"
```

2. **OR (`or`)**

- Returns `True` if **at least one condition is true**.
- Example:

```
a = True
b = False
if a or b:
    print("At least one is true")  # Prints "At least one is true"
```

3. **NOT (`not`)**

- Reverses the logical state of its operand.
- Example:

```
a = True
if not a:
    print("Not true")  # Does not print because `not a` is `False`.
```

Examples

Math Operators

```
print(2 + 2)  # Output: 4
print(3 - 2)  # Output: 1
print(35 / 5) # Output: 7.0 (float)
print(7 * 4)  # Output: 28
```

Logical Operators

```
# AND Operator
a = True
b = True
if a and b:
    print("Both are true")  # Output: Both are true
```

```
# OR Operator
a = True
b = False
if a or b:
    print("At least one is true")  # Output: At least one is true

# NOT Operator
a = True
if not a:
    print("Not true")  # No output since `not a` is False
```

Application: Combining Logical Operators

Logical operators are often used in conditional statements to evaluate complex conditions.
Example:

```
# Checking a restaurant discount condition
is_loyal_customer = True
spent_over_100 = True

if is_loyal_customer and spent_over_100:
    print("Discount applied!")  # Output: Discount applied!
```

Key Points

- Math operators mimic calculator functions for arithmetic calculations.
- Logical operators (`and` , `or` , `not`) allow programs to evaluate conditions and control flow.
- Conditional statements often combine logical operators to check multiple criteria.

Explore more: Refer to the Python documentation for a deeper dive into operators and their use cases.