Python Operators

Operators in Python are special symbols or keywords used to perform operations on variables and values. Python supports a variety of operators, each with specific functionality. Below is a categorized explanation with real-time use cases for each type.

## Arithmetic Operators

Used to perform mathematical operations like addition, subtraction, etc.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Operator | Meaning | Example Code | Real-time Use Case | | + | Addition | a + b | Calculating total price of items in a cart | | - | Subtraction | a - b | Finding the remaining balance after withdrawal | | \* | Multiplication | a \* b | Calculating area of a rectangle | | / | Division | a / b | Computing average marks | | % | Modulus | a % b | Finding if a number is even or odd | | \*\* | Exponentiation | a \*\* b | Calculating power values like 2^3 | | // | Floor Division | a // b | Distributing items evenly among people | |

## Comparison Operators

Used to compare values. The result is either True or False.

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Meaning | Example Code | Real-time Use Case |
| == | Equal to | a == b | Checking if user input matches stored password |
| != | Not equal to | a != b | Checking if two items in cart are different |
| > | Greater than | a > b | Checking if marks are above pass threshold |
| < | Less than | a < b | Checking if age is under voting limit |
| >= | Greater than or equal to | a >= b | Eligibility for senior citizen benefits |
| <= | Less than or equal to | a <= b | Child ticket eligibility in transport |

## Logical Operators

Used to combine conditional statements.

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Meaning | Example Code | Real-time Use Case |
| and | Logical AND | a > 10 and b < 20 | Validating multiple form fields |
| or | Logical OR | a > 10 or b < 20 | Checking if at least one field is filled |
| not | Logical NOT | not(a > 10) | Inverting a permission or condition |

## Assignment Operators

Used to assign values to variables.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Operator | Meaning | Example Code | Real-time Use Case | | = | Assign | a = b | Storing user input | | += | Add and assign | a += 1 | Incrementing page view count | | -= | Subtract and assign | a -= 1 | Decrementing item stock | | \*= | Multiply and assign | a \*= 2 | Doubling investment | | /= | Divide and assign | a /= 2 | Splitting bill amount | | %= | Modulus and assign | a %= 2 | Toggling between two states | | \*\*= | Exponent and assign | a \*\*= 2 | Squaring a number in-place | | //= | Floor divide and assign | a //= 2 | Updating item count per box | |

## Bitwise Operators

Used to perform bit-level operations.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Operator | Meaning | Example Code | Real-time Use Case | | & | Bitwise AND | a & b | Setting specific permission bits | | | | Bitwise OR | a | b | Combining feature flags | | ^ | Bitwise XOR | a ^ b | Toggling bits | | ~ | Bitwise NOT | ~a | Inverting all bits in access control | | << | Left Shift | a << 1 | Multiplying by powers of 2 | | >> | Right Shift | a >> 1 | Dividing by powers of 2 | |

## Membership Operators

Used to test if a value is in a sequence.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Operator | Meaning | Example Code | Real-time Use Case | | in | Value found | 'apple' in fruits | Checking if user selected item exists in cart | | not in | Value not found | 'banana' not in cart | Validating blacklist/ban list | |

## Identity Operators

Used to compare memory locations.

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Meaning | Example Code | Real-time Use Case |
| is | Same object | a is b | Checking if cached item is reused |
| is not | Different object | a is not b | Checking for re-initialization of data |