Logical Operators

16 June 2025 12:22

Logical operators in Python work based on the **Boolean values** of the operands. They evaluate expressions and return either True or False, or in some cases, return one of the operands.

Types of Logical Operators:

1. **Logical AND (and)**
2. **Logical OR (or)**
3. **Logical NOT (not)**

1.Logical AND(and):

Syntax: Operand1 and Operand2

Truth table of AND:

Operand1	Operand2	Output
0	0	0
0	1	0
1	0	0
1	1	1

Rules

- and checks the **truth** of both operands.
- If the **first operand is False**, it returns the **first operand**.
- If the **first operand is True**, it returns the **second operand**.

If operand1==False
Output =Operand1

If Operand1==True
Output=Operand2

Note:

- Any non-zero number, non-empty string/list/set is considered **True**
- Zero, empty string (''), empty list ([]), empty tuple (()), empty set (set()), and None are considered **False**

Examples:-

```
10 and 34        # Output: 34 (both are truth)
10.5 and 0.0     # Output: 0.0 (second is false, returned)
0 and 0.0        # Output: 0 (first is false, returned)

True and False   # Output: False
True and True    # Output: True
False and False  # Output: False

'tea' and 'coffee'   # Output: 'coffee' (both non-empty strings)
'milk' and []        # Output: [] (second is empty list = False)
[] and ()            # Output: [] (first is false)

(1,2) and [3,4]      # Output: [3, 4]
{1,2,3} and {1:2}    # Output: {1: 2}
{1:2} and {1:2}      # Output: {1: 2}
```

2.Logical OR (or):-

Syntax: Operand1 or Operand2

Truth table of OR:

Operand1	Operand2	Output
0	0	0
0	1	1
1	0	1
1	1	1

Rules:

- Returns **first operand** if it is **True (truthy)**.
- Otherwise, returns the **second operand**.

If operand1==False
Output =Operand2

If Operand1==True
Output=Operand1

Examples:-

```
10 or 34          #Output:- 10
0 or 20           # Output:- 20
False or True     # Output:- True
True or False     #Output:-  True
'' or 'hello'     # Output:- 'hello'
[] or [1,2]       # Output:- [1, 2]
() or (3,4)       # Output:- (3, 4)
{1:2} or {}       # Output:- {1: 2}
```

3.Logical NOT(not):-

Syntax:- not(operand)

Truth Table of NOT:

Operand1	Operand2
0	1
0	0

Rules:

- Returns the **opposite** Boolean value of the operand.
- If operand is True, returns False.
- If operand is False, returns True

If Operand==False
Output =True

If Operand==True
Output=False

Examples:-

```
not (True)       # False
not (False)      # True
not (10)         # False (10 is truth)
not (0)          # True (0 is false)
not ''           # True
not 'hello'      # False
not []           # True
not [1, 2]       # False
not None         # True
```

Key points:

Operator	Behaviour
and	Returns second if both are True; else returns first
or	Returns first if True; else returns second
not	Inverts the Boolean value

These operators are widely used in **conditions**, **loops**, in Python.