# Python - Operadores Lógicos

## Operadores Lógicos Python

Operadores lógicos Python são usados para formar expressões booleanas compostas. Cada operando para esses operadores lógicos é em si uma expressão booleana. Por exemplo,

## Exemplo

```
age > 16 and marks > 80
percentage < 50 or attendance < 75
```

Junto com a palavra-chave False, Python interpreta None, zero numérico de todos os tipos e sequências vazias ( strings , tuples , lists ), dicionários vazios e conjuntos vazios como False. Todos os outros valores são tratados como True.

Existem três operadores lógicos em Python. Eles são " **e** ", " **ou** " e " **não** ". Eles devem estar em letras minúsculas.

## Operador lógico "e"

Para que a expressão booleana composta seja True, ambos os operandos devem ser True. Se algum ou ambos os operandos forem avaliados como False, a expressão retornará False.

## Tabela verdade do operador lógico "e"

A tabela a seguir mostra os cenários.

| a | b | a e b |
|---|---|---|
| F | F | F |
| F | T | F |
| T | F | F |
| T | T | T |

## Operador lógico "ou"

Por outro lado, o operador ou retorna True se algum dos operandos for True. Para que a expressão booleana composta seja False, ambos os operandos devem ser False.

### Tabela verdade do operador lógico "ou"

As tabelas a seguir mostram o resultado do operador "ou" com diferentes condições:

| a | b | a ou B |
|---|---|---|
| F | F | F |
| F | T | T |
| T | F | F |
| T | T | T |

## Operador lógico "não"

Este é um operador unário. O estado do operando booleano a seguir é invertido. Como resultado, não verdadeiro torna-se falso e não falso torna-se verdadeiro.

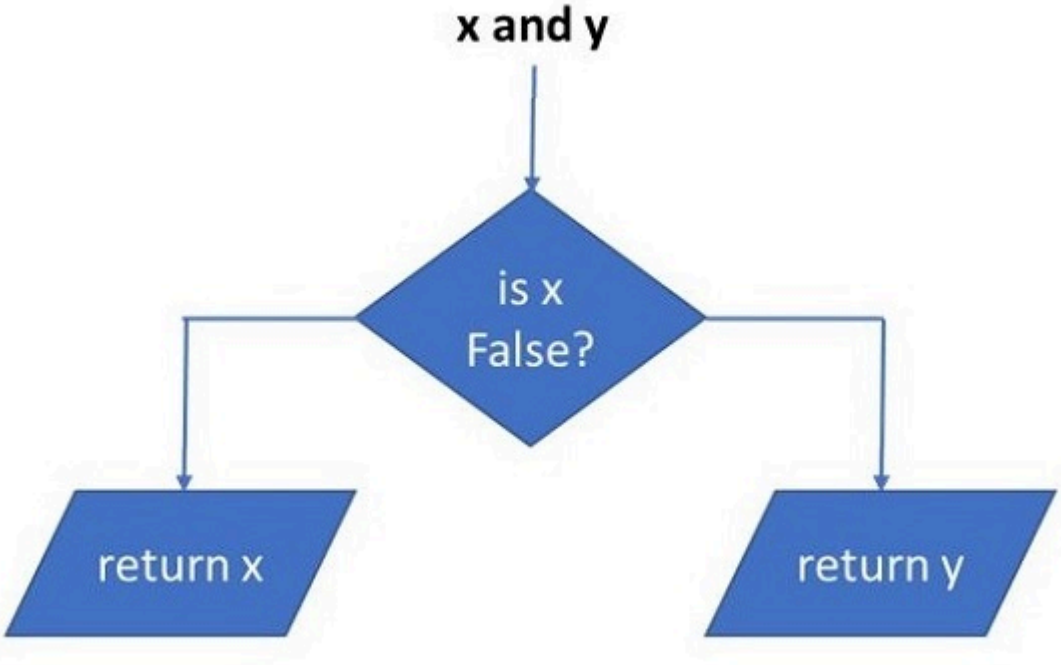
### Tabela Verdade do Operador Lógico "não"

| a | não (a) |
|---|---|
| F | T |
| T | F |

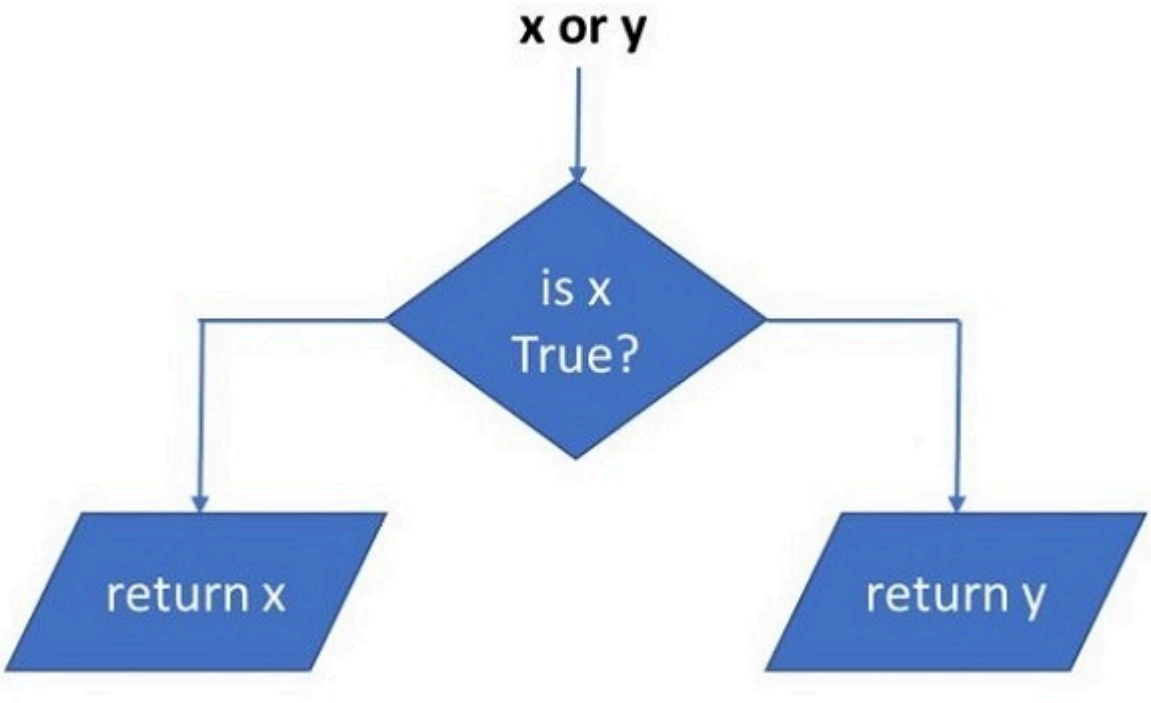


## Como o interpretador Python avalia os operadores lógicos?

A expressão "x e y" avalia primeiro "x". Se “x” for falso, seu valor será retornado; caso contrário, "y" será avaliado e o valor resultante será retornado.




A expressão “x ou y” avalia primeiro “x”; se “x” for verdadeiro, seu valor será retornado; caso contrário, "y" será avaliado e o valor resultante será retornado.






## Exemplos de operadores lógicos Python

Alguns casos de uso de operadores lógicos são fornecidos abaixo -

## Exemplo 1: Operadores Lógicos com Condições Booleanas

```
x = 10
y = 20
print("x > 0 and x < 10:",x > 0 and x < 10)
print("x > 0 and y > 10:",x > 0 and y > 10)
print("x > 10 or y > 10:",x > 10 or y > 10)
print("x%2 == 0 and y%2 == 0:",x%2 == 0 and y%2 == 0)
print ("not (x+y>15):", not (x+y)>15)
```

Ele produzirá a seguinte **saída** -

```
x > 0 and x < 10: False
x > 0 and y > 10: True
x > 10 or y > 10: True
x%2 == 0 and y%2 == 0: True
not (x+y>15): False
```

## Exemplo 2: Operadores Lógicos com Condições Não Booleanas

Podemos usar operandos não booleanos com operadores lógicos. Aqui, não precisamos que quaisquer números diferentes de zero e sequências não vazias sejam avaliadas como True. Conseqüentemente, as mesmas tabelas verdade de operadores lógicos se aplicam.

No exemplo a seguir, operandos numéricos são usados para operadores lógicos. As variáveis "x", "y" são avaliadas como True, "z" é False

```
x = 10
y = 20
z = 0
print("x and y:",x and y)
print("x or y:",x or y)
print("z or x:",z or x)
print("y or z:", y or z)
```

Ele produzirá a seguinte **saída** -

```
x and y: 20
x or y: 10
```

```
z or x: 10
y or z: 20
```

## Exemplo 3: Operadores Lógicos com Strings e Tuplas

A variável string é tratada como True e uma tupla vazia como False no exemplo a seguir -

```
a="Hello"
b=tuple()
print("a and b:",a and b)
print("b or a:",b or a)
```

Ele produzirá a seguinte **saída** -

```
a and b: ()
b or a: Hello
```

## Exemplo 4: Operadores Lógicos para Comparar Sequências (Listas)

Finalmente, dois objetos da lista abaixo não estão vazios. Portanto, x e y retornam o último, e x ou y retornam o primeiro.

```
x=[1,2,3]
y=[10,20,30]
print("x and y:",x and y)
print("x or y:",x or y)
```

Ele produzirá a seguinte **saída** -

```
x and y: [10, 20, 30]
x or y: [1, 2, 3]
```