



Логика boolean, NoneType, сравнения, логическая алгебра и условная инструкция `if-elif-else`

Тип `bool` может иметь два значения: `True` (правда) и `False` (ложь).

```
true_bool = True
print(type(true_bool))

>> <class 'bool'>

false_bool = True
print(type(false_bool))

>> <class 'bool'>
```

Тип `NoneType` может иметь одно значения: `None` (ничто).

```
none_type = None
print(type(none_type))

>> <class 'NoneType'>
```

Сравнения

$x > y$

Выдает `True` если `x` больше чем `y`. В противном случае, выдает `False`.

```
#
x = 5
y = 4
print(x > y)

>> True
#
x = 4
y = 4
print(x > y)

>> False
#
x = 4
y = 5
print(x > y)

>> False
```

$x \geq y$

Выдает **True** если **x** больше или равен **y**. В противном случае, выдает **False**.

```
#
x = 5
y = 4
print(x >= y)

>> True
#
x = 4
y = 4
print(x >= y)

>> True
#
x = 4
y = 5
print(x >= y)

>> False
```

$x < y$

Выдает **True** если **x** меньше чем **y**. В противном случае, выдает **False**.

```
#
x = 4
y = 5
print(x < y)

>> True
#
x = 4
y = 4
print(x < y)

>> False
#
x = 4
y = 5
print(x < y)

>> False
```

$x \leq y$

Выдает **True** если **x** меньше или равен **y**. В противном случае, выдает **False**.

```
#
x = 4
y = 5
print(x <= y)

>> True
#
x = 4
y = 4
print(x <= y)

>> True
#
x = 5
y = 4
print(x <= y)
```

```
>> False
```

$x == y$

Выдает **True** если **x** равен **y**. В противном случае, выдает **False**.

```
#
x = 4
y = 4
print(x == y)

>> True
#
x = 5
y = 4
print(x == y)

>> False
```

$x != y$

Выдает **True** если **x** не равен **y**. В противном случае, выдает **False**.

```
#
x = 4
y = 5
print(x != y)

>> True
#
x = 4
y = 4
print(x != y)

>> False
```

Логическая алгебра

Логическая алгебра это алгебра в котором операции производятся не на числах, а на логике (правда или ложь). В Питоне логическая алгебра производится при помощи двух ключевыми словами `and` (и) и `or` (или).

`x and y` выдает `True` только если и `x` и `y` является `True`. В противном случае выдается `False`.

```
x = True
y = True
print(x and y)

>> True

x = False
y = True
print(x and y)

>> False

x = True
y = False
print(x and y)

>> False

x = False
y = False
print(x and y)

>> False
```

`x or y` выдает `True`, если или `x` или `y` или оба значения являются `True`. В противном случае выдается `False`.

```
#
x = True
y = True
print(x or y)

>> True
#
x = True
```

```
y = False
print(x or y)

>> True
#
x = False
y = True
print(x or y)

>> True

x = False
y = False
print(x or y)

>> False
```

Например:

Допустим, что мы хотим проверить, находится ли число `5` между числами `3` и `7`. Математически это можно выразить `3 < 5 < 7`. В Питоне можно использовать то же обозначение:

```
x = 5
print(3 < x < 7)

>> True
```

Также, для этого примера, можно использовать логическую алгебру при помощи ключевого слова `and`. Для того чтобы число `z` было между числами `x` и `y`, число `z` должно быть больше чем `x`, и меньше чем число `y`. Поэтому верхнее неравенство можно записать следующим способом.

```
x = 5
print(z > 3 and z < 7)

>> True
```

Условная инструкция `if-elif-else`

При помощи ключевого слова `if`, можно задать условия для запуска кода.

```
x = 5
y = 4

if x > y:
    print('x is bigger than y')

>> x is bigger than y
```

При помощи ключевого слова `else`, можно запустить код в случае если условия `if` не соблюдены.

```
x = 4
y = 5

if x > y:
    print('x is bigger than y')
else:
    print('x is smaller than y')

>> x is smaller than y
```

При помощи ключевого слова `elif`, можно задать дополнительные условия для запуска кода.

```
x = 4
y = 4

if x > y:
    print('x is bigger than y')
elif x == y:
    print('x is equal to y')
else:
    print('x is smaller than y')
```

```
>> x is equal to y
```