

Самое важное

Вложенные условия

Python будет строго следовать вашим указаниям, и важно научиться следить за ходом выполнения кода (в этом вам помогут функции `print()`, расставленные в важных местах кода).

Так, если вам нужно проверить какое-либо условие, только если выполняется другое условие, нужно именно это и указать.

```
if <условие_1>:  
    если условие_1 будет выполнено, Python заглянет внутрь условного блока  
    if <условие_2>: — и проверит условие_2.
```

Если же условие_1 выполнено не будет, то условие_2 не будет даже проверяться, так как Python не будет смотреть внутрь условного блока под условием_1.

Оператор `elif` (иногда называют `else-if`)

Мы уже знакомы с операторами `if/else` и знаем, что в `if` мы можем указать какое-то конкретное условие, а в `else` попадут остальные. Но если нам надо проверить 2 (или больше) конкретных условия, то мы обращаемся к оператору `elif`.

Синтаксис прост:

```
if <условие_1>:  
    ...  
elif <условие_2>:  
    ...  
else:  
    ...
```

При этом важно понимать, что ни `elif`, ни `else` не являются обязательными. Мы можем составлять условный блок просто из `if`, можем добавлять `if/elif`, можем `if/else`, а можем собирать полный блок с `if/elif/else`. Можем даже добавлять множество `elif`: `if/elif/elif/elif.../else`. Их количество не ограничено.

Логические операторы `and` и `or`

Бывают ситуации, когда два условия приводят к одному действию.

Нам нужно 1) чтобы оба условия выполнялись вместе, и только после этого мы сможем выполнить какое-то действие, или 2) чтобы одно из условий выполнилось, после чего мы бы могли начать действовать.

Тут нужен оператор `and/or`.

or, также называется ИЛИ, позволяет выбрать одно из условий.

if $x > 0$ or $y < 0$: — если будет выполнено хотя бы одно из этих условий, то Python заглянет внутрь условного блока.

Если и $x > 0$, и $y < 0$ вернут False, то условие выполнено не будет.

and, также называется “И”, позволяет соединить условия вместе, требуя выполнения каждого из них.


if $x > 0$ and $y < 0$: — такое условие сработает, только если $x > 0$ и $y < 0$, то есть внутри Python заглянет, только если получит два ответа True.

Несколько операторов or/and

Как и у операций вычисления, у логических вычислений тоже есть свой приоритет выполнения.

В таблице ниже вы можете увидеть приоритет выполнения некоторых операций: чем выше операция, тем выше приоритет, то есть эти операции будут выполняться पहले.

in, not in, is, is not, <, <=, >, >=, <>, !=, ==	Сравнение, принадлежность, тождественность
not x	Булево НЕ
and	Булево И
or	Булево ИЛИ



Чтобы условия проверялись так, как вы задумали, необходимо использовать скобки.

```
if <условие_1> and <условие_2> or <условие_3>
```

...

Например, при такой записи сперва будет выполнено сравнение условий 1 и 2 при помощи AND. Если оба условия вернули True, то Python заменит их на True и в промежутке получится что-то вроде:

```
if True or <условие_3>,
```

и теперь будут сравниваться уже True и условие_3.

Однако мы можем использовать скобки: if <условие_1> and (<условие_2> or <условие_3>).

Теперь Python будет вынужден первым делом пойти в скобку и выполнить выражение в них: if <условие_1> and (True/False). После того как он получит в скобках ответ True или False, он сможет сравнить первое условие и этот ответ.

Не допускай следующих ошибок!

Если условия не зависят друг от друга и могут выполняться вместе, не помещайте их в один условный блок (или делайте это осторожно).

```
имя_1 = Иван
имя_2 = Сеня
x = 5
y = 1
```

```
if x == 5:
    print("Привет ", имя_1)
elif y == 1:
    print("Привет ", имя_2)
```

Выполнится только один условный блок — тот, который первый выдаст правильный ответ (тот, который стоит выше).

Если же нужно выполнить обе проверки, создавайте разные условные блоки.

```
if x == 5:
    print("Привет ", имя_1)
if y == 1:
    print("Привет ", имя_2)
```

Не забывайте явно прописывать все необходимые логические операции.

Логические операции хоть и похожи на арифметические, но не во всём.

```
if (x or y) > 10:
    ...
```

Внутри скобок будет выбрано только одно значение (первое, которое будет похоже на True), и только после этого выполнится второе условие вне скобок.

Такой код не будет считаться аналогом этому:

```
if (x > 10) or (y > 10):
    ...
```