Тема урока: логические операторы

Как быть в ситуации, когда у нас есть несколько условий? В Python есть три логических оператора, которые позволяют создавать сложные условия:

- `and` — логическое умножение;
- `or` — логическое сложение;
- `not` — логическое отрицание.

Оператор and

Предположим, мы написали программу для учеников от двенадцати лет, которые учатся по крайней мере в 7 классе. Доступ к ней тем, кто младше, надо запретить. Следующий код решает поставленную задачу:

```
age = int(input('Сколько вам лет?: '))
grade = int(input('В каком классе вы учитесь?: '))
if age >= 12 and grade >= 7:
    print('Доступ разрешен.')
else:
    print('Доступ запрещен.')
```

Мы объединили два условия при помощи оператора `and`. Оно означает, что в этом ветвлении блок кода выполняется только при выполнении **обоих условий одновременно**!
Оператор `and` может объединять произвольное количество условий:

```
age = int(input('Сколько вам лет?: '))
grade = int(input('В каком классе вы учитесь?: '))
city = input('В каком городе вы живете?: ')
if age >= 12 and grade >= 7 and city == 'Москва':
    print('Доступ разрешен.')
else:
    print('Доступ запрещен.')
```

Оператор or

Оператор `or` также применяется для объединения условий. Однако, в отличие от `and`, для выполнения блока кода достаточно выполнения **хотя бы одного из условий.**

```
city = input('В каком городе вы живете?: ')
if city == 'Москва' or city == 'Санкт-Петербург' or city ==
'Екатеринбург':
    print('Доступ разрешен.')
else:
    print('Доступ запрещен.')
```

Доступ будет разрешен в случае, если хотя бы одно из условий выполнится.
Логическое выражение `X and Y` истинно, если **оба** значения X и Y истинны.
Логическое выражение `X or Y` истинно, если **хотя бы одно** из значений X и Y истинно.

Мы можем использовать оба логических оператора одновременно:

```
age = int(input('Сколько вам лет?: '))
grade = int(input('В каком классе вы учитесь?: '))
city = input('В каком городе вы живете?: ')
if age >= 12 and grade >= 7 and (city == 'Москва' or city ==
'Санкт-Петербург'):
    print('Доступ разрешен.')
else:
    print('Доступ запрещен.')
```

Такой код проверяет, что возраст учеников от двенадцати лет и учатся они по крайней мере в 7 классе и живут в Москве или Санкт-Петербурге.

Оператор not

Оператор `not` позволяет инвертировать (т.е. заменить на противоположный) результат логического выражения. Например, следующий код:

```
age = int(input('Сколько вам лет?: '))
if not (age < 12):
    print('Доступ разрешен.')
else:
    print('Доступ запрещен.')
```

полностью эквивалентен коду:

```
age = int(input('Сколько вам лет?: '))
if age >= 12:
    print('Доступ разрешен.')
else:
    print('Доступ запрещен.')
```

В первом примере мы поместили выражение `age < 12` в скобки для того, чтобы было чётко видно, что мы применяем оператор `not` к значению выражения `age < 12`, а не только к переменной `age`.

Приоритеты логических операторов

Логические операторы, подобно арифметическим операторам (+, -, *, /), имеют приоритет выполнения. Приоритет выполнения следующий:

- в первую очередь выполняется логическое отрицание `not`;
- далее выполняется логическое умножение `and`;
- далее выполняется логическое сложение `or`.

Для **явного указания порядка** выполнения условных операторов **используют скобки**.

Решение задач

Задача 1.

Напишите программу, которая определяет, является ли заданное натуральное число трёхзначным числом, делящимся на 7. Программа должна вывести «YES», если число является трёхзначным числом, делящимся на 7, или «NO» в противном случае.

Формат входных данных

На вход программе подаётся целое число x.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

Входные данные	Выходные данные
105	YES
1045	NO
994	YES

Задача 2.

Напишите программу, которая проверяет, что все три цифры натурального трёхзначного числа различны.

Формат входных данных

На вход в программе подаётся натуральное трехзначное число x.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

Входные данные	Выходные данные
123	Цифры различны
788	Цифры не различны
597	Цифры различны

Задача 3. Координатная плоскость

Напишите программу, которая по координатам точки, не лежащей на осях координат, определяет номер координатной четверти, в которой она находится.

Формат входных данных

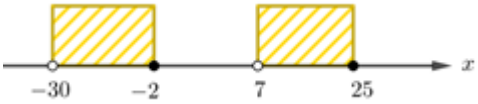
На вход в программе подаётся два натуральных трехзначных числа x и y – координаты точки.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

Входные данные	Выходные данные
1 2	1 четверть
-1 2	2 четверть
-2 -5	3 четверть

Задача 4. Принадлежность



Напишите программу, которая принимает целое число *x* и определяет, принадлежит ли данное число указанным промежуткам.

Формат входных данных

На вход программе подаётся целое число *x*.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

Примечание. Если точка выколотая, то граница не включается, если точка закрашенная, то граница включается.

Входные данные	Выходные данные
-332	Не принадлежит
-30	Не принадлежит
-21	Принадлежит

Задача 5. Красивое число

Назовем число красивым, если оно является четырехзначным и делится нацело на 7 или на 17. Напишите программу, определяющую, является ли введённое число красивым. Программа должна вывести «YES», если число является красивым, или «NO» в противном случае.

Формат входных данных

На вход программе подаётся натуральное число.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

Входные данные	Выходные данные
1043	YES
1045	NO
2751	YES

Задача 6. Високосный год

Напишите программу, которая определяет, является ли год с данным номером високосным. Если год является високосным, то выведите «YES», иначе выведите «NO».

Год является високосным, если его номер кратен 4, но не кратен 100, или если он кратен 400.

Формат входных данных

На вход программе подаётся натуральное число.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

Входные данные	Выходные данные
2020	YES
2012	YES
2009	NO