



Урок 6. Python: Логические значения и сложные условия

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 6 из 33 · 45 минут · Модуль «Ветвления. Логические переменные. Сложные условия»

Уровень: начинающий программист; Python 3.

Программа: <https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

Число должно быть не меньше 10 и не больше 20. Одно неверное слово — `or` вместо `and` — пропускает почти всё. Проверим правило таблицей.

Результат урока: Составить и проверить условие с `and`, `or`, `not`, выделить границы и объяснить короткое вычисление.

01. Проблемная ситуация

0-5 минут. Рассмотрите два правила: `x >= 10 and x <= 20` и `x >= 10 or x <= 20`. Подставьте `x=5,15,25`.

Какое правило задаёт отрезок? Запишите контрпример неправильному. Числа — учебные координаты, не оценки учеников.

02. Понятия и разобранный пример

5-14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

И, или, не: смысл прежде синтаксиса

Сравнение даёт `True` или `False`. Для логических операндов `A` и `B` истинно, когда истинны оба; `A or B` — когда истинно хотя бы одно; `not A` меняет значение на противоположное. В задачах урока операндами служат именно сравнения или `bool`, хотя Python допускает для `and/or` и другие объекты.

Отрезок `10...20` включает обе границы: `x >= 10 and x <= 20`. За его пределами `x < 10 or x > 20`. Скобки полезны для группировки. Не смешивайте альтернативу «хотя бы одно» с требованием «ровно одно»: обычное `or` допускает истинность обоих.

Python вычисляет `and` и `or` слева направо с остановкой, когда результат уже определён. В `d!=0 and n%d==0` остаток не вычисляется при `d=0`. Поменять части местами нельзя: защита должна сработать до опасной операции. Это свойство порядка вычисления, а не случайное везение теста.

Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

Для 9 первое сравнение `False`, поэтому ответ `False`. Для 10 оба `True`, граница входит. Для 20 второе тоже `True`. Для 21 второе `False`. Вывод программы: `9 False;10 True;15 True;20 True;21 False` — каждая пара на своей строке.

```
def inside(x):  
    """Вернуть bool: целое x находится в отрезке 10...20."""  
    return x >= 10 and x <= 20
```

```
for x in [9, 10, 15, 20, 21]:  
    print(x, inside(x))
```

A=False, B=False	and=False; or=False
A=False, B=True	and=False; or=True
A=True, B=False	and=False; or=True
A=True, B=True	and=True; or=True

Отрезок `10...20` включает 10 и 20; 9 и 21 не входят. Условие: `x >= 10 and x <= 20`.



03. Практика 1. Проверяем материал

Таблица четырёх сочетаний

14–25 минут. Учебная программа видит два логических признака: есть основной файл и есть резервная копия. Получите `available` (хотя бы одна) и `missing` (нет обеих).

has_file, has_backup	Доступна хотя бы одна копия
False,False	False
False,True	True
True,False	True
True,True	True

2 минуты — заполните таблицу; 4 — напишите функцию и драйвер; 3 — все четыре комбинации; 2 — объясните случай `True, True`. Продукт: таблица с двумя результатами и исходник. Критерии: `or` для наличия, `not` для отсутствия, все сочетания, словесное объяснение.

04. Практика 2. Перенос способа

Защита деления коротким условием

25–33 минуты. Напишите `divisible(n,d)` для целых `n,d` от 0 до 100: при `d=0` вернуть `False`, иначе проверить делимость `n` на `d`. За 2 минуты прогноз `(12,3),(12,5),(0,4),(12,0)`, за 4 код, за 2 проверка порядка частей `and`. Критерии: нет деления на ноль, четыре верных результата, объяснение нулевого числителя.

05. Обсуждение и границы вывода

33–40 минут. Одни предлагают соединять все проверки словом `or`, чтобы «подошёл хоть один тест». Другие сначала переводят условие на обычный язык. На примерах отрезка и резервной копии объясните, почему один оператор не подходит всем задачам. Приведите контрпример.

06. Индивидуальная проверка

40–45 минут. Вычислите `not(False or True)`; составьте условие для целого `x` вне отрезка `3...7`; проверьте `x=3` и `x=8`. Четыре результата — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

available: False, True, True, True; missing: True, False, False, False в порядке таблицы. Одновременное наличие двух копий не запрещает доступ. Возврат пары в Python здесь удобен для печати двух логических результатов.

```
def copies(has_file, has_backup):
    available = has_file or has_backup
    missing = not available
    return available, missing

for a, b in [(False, False), (False, True),
            (True, False), (True, True)]:
    print(a, b, copies(a, b))
```

Практика 2: разбор и критерии

True, False, True, False. Ноль делится на положительное d без остатка. При $d=0$ условие сразу False, выражение $n\%d$ не вычисляется. Это отдельное определение поведения для нулевого делителя, а не утверждение, что математическое деление на ноль допустимо.

```
def divisible(n, d):
    """Целые n, d в 0...100; при d=0 вернуть False."""
    return d != 0 and n % d == 0

for n, d in [(12, 3), (12, 5), (0, 4), (12, 0)]:
    print(divisible(n, d))
```

Обсуждение: возможный вывод

Для отрезка нужны обе границы одновременно, поэтому and. Для копий достаточно любой, поэтому or. При $x=25$ ошибочное or пропустит число из-за $x \geq 10$. Назначение правила определяет соединение, а не желание получить больше True.

Индивидуальная проверка: ответ и критерии

False; $x < 3$ or $x > 7$; для 3 False, для 8 True. 4 балла. Строгие знаки исключают границы из множества «вне».

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, редактор, папку с правом записи, распечатку ученической части PDF, бумагу и ручку. Python 3.11+ и IDLE или редактор; запуск `python lesson.py` (либо `python3 lesson.py`). Внешние библиотеки не нужны. Установку и проверку запуска учитель выполняет заранее. Все программы, данные и ожидаемые ответы даны; регистрация детей и интернет на занятии не нужны.

Уровень — начинающий программист, 8 класс. Предпосылки: сравнения, if, функции и логические значения. В паре автор прогнозирует и вводит код, проверяющий ведёт журнал, затем роли меняются. Без компьютера выполните трассировку полного кода на бумаге и сопоставьте с готовыми результатами; учитель демонстрирует подготовленный запуск. Бумажный вариант не считается фактическим запуском ученика. Перед практикой учитель выдаёт готовый файл разобранного примера, чтобы дети тратили время на изменение и проверку, а не длинное перепечатывание. Матрица программы предназначена учителю.

Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ



07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Ветвления. Логические переменные. Сложные условия	Разбирает пример и создаёт две подпрограммы	Исходники, прогнозы и протокол проверки
Составление и отладка программ	Сравнивает фактический результат с контрактом	Объяснение границ и самостоятельный ответ

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

Python: логические операции и порядок вычисления

<https://docs.python.org/3/reference/expressions.html#boolean-operations>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 18.10.2026.