



Урок 9. Python: условие и цикл с предусловием

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 9 из 34 · 45 минут · Модуль «Циклы»

Уровень: начинающий программист; Python 3.

Программа: <https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

Копилка пополняется по 3 условные единицы, пока не достигнет цели. Как сделать повторение конечным и не выполнить лишний шаг?

Результат урока: Написать `while` с проверкой до тела, проследить изменение переменной и объяснить завершение, включая нулевое число повторений.

01. Проблемная ситуация

Псевдокод «пока сумма меньше цели: напечатать сумму» не изменяет сумму. При сумме 0 и цели 10 он не завершится. На бумаге добавьте действие, которое приближает завершение, и предскажите, хватит ли трёх прибавлений по 3.

02. Понятия и разобранный пример

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

Проверка до выполнения

Условие — выражение с результатом `True` или `False`. `if` выполняет блок не более одного раза; `while` проверяет условие перед каждым повторением. После двоеточия тело записывают с отступом в 4 пробела. Строки без отступа после тела выполняются уже после цикла.

В программе `total` начинается с 0, шаг равен 3. Пока `total < target`, увеличиваем `total` и счётчик `steps`. Для `target=10` значения после тела: 3,6,9,12; шагов 4. Остановиться на 9 нельзя, ведь цель ещё не достигнута.

Завершение обосновано: положительный шаг увеличивает `total`, а конечная цель фиксирована. Если шаг 0 или отрицательный, это рассуждение не работает. В условии ниже шаг всегда 3, вход `target` — неотрицательное целое. При `target=0` тело не выполнится ни разу.

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 `target=10` перед последней итерацией `total=9 < 10`. После неё `total=12`, `steps=4`, новая проверка `12 < 10` ложна. Две финальные строки сообщают число шагов и итог; это не печать на каждом шаге.

```
target = int(input())
total = 0
steps = 0
while total < target:
    total = total + 3
    steps = steps + 1
print(steps)
print(total)
```



Python: условие и цикл с предусловием

До цикла	total=0; steps=0
Проверка	total < 10?
Тело: да	total += 3; steps += 1
Выход: нет	total=12; steps=4

Трассировка while при цели 10: 0 → 3 → 6 → 9 → 12; условие проверяется до каждого шага.

03. Практика 1. Проверяем материал

Накапливаем взносы модели

Это учебная модель, не финансовый совет. Работайте с неотрицательными целыми; не запускайте намеренно бесконечный вариант.

Ввод target	Шаги; итог
10	4; 12
9	3; 9
0	0; 0

Запишите таблицу total до проверки, результат условия, total после тела для target=10. Запустите полный код на трёх входах. Затем объясните, почему `<=` вместо `<` даёт лишнее пополнение при target=9. Результат — трассировка и журнал тестов. Критерии: обновление total, счётчик, верный тест 9, нулевой тест.

04. Практика 2. Перенос способа

Когда повторений не будет

Новая задача: робот начинает в точке `x=5` и делает шаг вправо на 2, пока `x<limit`. `limit` — любое неотрицательное целое. Напишите `robot.py`, выводящий число шагов и финальный `x` одной строкой. Тесты: 10 → 3 11; 5 → 0 5; 0 → 0 5. Объясните, почему начальный `x` теперь не равен нулю.

05. Обсуждение и границы вывода

«Цикл всегда должен выполняться хотя бы один раз». Согласны ли вы? Используйте тесты `target=0` и `limit=5`. Альтернатива — сначала выполнить действие, а потом проверить условие. Почему это другая логика?

06. Индивидуальная проверка

Старт `v=1`, шаг `+4`, повторение пока `v<10`. Запишите все значения после тела, число шагов и финальную ложную проверку. Назовите действие, удаление которого сделает цикл бесконечным. Критерии: трасса, завершение, причина.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

При `target=10` проверки: `0<10, 3<10, 6<10, 9<10` истинны, `12<10` ложна. Для 9 нужно 3 шага; `<=` дало бы 4 и итог 12. При 0 цикл пропускается. Полный правильный код — разобранный пример, изменений для практики 1 не требуется. По 1 баллу за четыре критерия.

Практика 2: разбор и критерии

Начальная позиция — часть условия. Для предела 10 позиции после шагов 7,9,11; при 5 и 0 условие сразу ложно. Критерии: начало 5, шаг 2, проверка до тела, все тесты.

```
limit = int(input())
x = 5
steps = 0
while x < limit:
    x = x + 2
    steps = steps + 1
print(steps, x)
```

Обсуждение: возможный вывод

`while` с предусловием может выполнить тело ноль раз. Принудительный первый шаг изменил бы результат в двух граничных тестах. Выбор конструкции определяется условием задачи, не привычкой автора.

Индивидуальная проверка: ответ и критерии

После тела 5,9,13; 3 шага; `13<10` ложно. Без увеличения `v` значение 1 остаётся меньше 10 постоянно. 3 балла.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, локальную папку с правом записи, Python/IDLE, распечатку ученической части PDF, бумагу и ручку. Внешние библиотеки не нужны: только стандартный Python 3.11 или новее. Все программы, входные данные, таблицы и ключи находятся в статье и PDF. Детские аккаунты, сеть и регистрация не нужны.

Уровень — начинающий, 7 класс. Предпосылки: целые числа, сравнения и присваивание. Работа в паре: автор прогнозирует и вводит код, проверяющий ведёт журнал; затем роли меняются. На бумаге трассируйте готовый код и входные данные; для графики используйте координатную сетку и схему. Учитель может показать заранее полученный вывод/PNG. Бумажный вариант проверяет понимание, но не выдаётся за реальный запуск учеником. Чтение, ввод и проверка включены в план. Матрица программы — материал учителя.

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
Цикл с предусловием. Бесконечные циклы	Обновляет переменную и доказывает завершение	Конечная трассировка и нулевой тест
Условный оператор	Различает if и while	Одно выполнение против повторений

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

Официальное руководство Python 3.11: управление выполнением

<https://docs.python.org/3.11/tutorial/controlflow.html>

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: 17.10.2026.