



Урок 12. Python: цикл со счётчиком и шагом

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 12 из 34 · 45 минут · Модуль «Циклы»

Уровень: начинающий программист; Python 3.

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

Нужно вывести 2,4,6,8,10, но программа останавливается на 8. Откуда берётся ошибка границы и как проверить шаг?

Результат урока: Построить перебор `range(start,stop,step)`, сравнить с `while` и проверить положительный, отрицательный и пустой диапазоны.

01. Проблемная ситуация

Прогнозируйте вывод `for i in range(2,10,2)`. Будет ли 10? Сравните с требованием «все чётные от 2 до 10 включительно». На бумаге запишите правильный `stop` и объясните, почему изменение только шага не решает ошибку.

02. Понятия и разобранный пример

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

Правая граница `range` не входит

`for` перебирает элементы последовательности. `range(start,stop,step)` задаёт целые значения от `start` с заданным шагом, не включая `stop`. Для положительного шага нужны значения меньше `stop`; для отрицательного — больше `stop`.

`range(2,11,2)` даёт 2,4,6,8,10. Эквивалентный `while` начинает `i=2`, повторяет при `i<=10` и увеличивает `i` на 2. В `for` обновление счётчика задаёт `range`; в `while` его пишет автор.

`range(10,1,-3)` даёт 10,7,4. `range(2,10,-2)` пуст: направление шага не ведёт от начала к границе. Нулевой шаг недопустим. Не запускайте огромные диапазоны для проверки: короткая трасса лучше показывает логику границ.

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

Программа ниже выводит сначала числа по одному в строке, затем их сумму 30. Для ручной проверки: $2+4+6+8+10=30$. Последовательность из пяти элементов заканчивается до `stop=11`, а не потому, что число 10 записано отдельно.

```
total = 0
for i in range(2, 11, 2):
    print(i)
    total = total + i
print(total)
```

Python: цикл со счётчиком и шагом

Начало	<code>start=2</code>
Шаг	<code>+2: 2 → 4 → 6 → 8 → 10</code>
Правая граница	<code>stop=11</code> не включается
Контроль	5 элементов; сумма 30

`range(2,11,2)`: пять значений 2,4,6,8,10; граница 11 не входит в перебор.



03. Практика 1. Проверяем материал

Сравниваем for и while

Для каждого диапазона выполните отдельный запуск полного примера, заменяя только аргументы range. Сохраняйте разные версии файла.

Диапазон	Значения; сумма
range(2,11,2)	2,4,6,8,10; 30
range(10,1,-3)	10,7,4; 21
range(2,10,-2)	пусто; 0

Запишите значения и сумму до запуска, проверьте три строки таблицы. Затем напишите равносильный while для первой строки. Сравните вывод построчно, включая сумму. Критерии: правая граница исключена, отрицательный шаг понят, пустой диапазон проверен, while равносильен for.

04. Практика 2. Перенос способа

Обратный ход и неверный знак

Новый порядок: отсчёт 9,6,3,0 по одному числу в строке, затем надпись «старт». Самостоятельно напишите countdown.py с for. Предложенные stop: 0 и -1; выберите подходящий и обоснуйте. Дополнительно объясните результат range(0,0,-3) и почему step=0 не является способом «ничего не делать».

05. Обсуждение и границы вывода

Что выбрать для неизвестного количества входных чисел до нуля: for range или while? Сравните с фиксированным отсчётом 9,6,3,0. Не называйте одну конструкцию «всегда лучшей».

06. Индивидуальная проверка

Выпишите range(3,14,4), число элементов и сумму. Измените stop так, чтобы дополнительно вошло 15. Критерии: значения, сумма/количество, новая граница.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Для while нужны начало 2, условие $i \leq 10$ и шаг +2. Ниже полный вариант. Остальные результаты приведены в таблице. При пустом range печатается только итог 0, тело не выполняется. По 1 баллу за четыре критерия.

```
i = 2
total = 0
while i <= 10:
    print(i)
    total = total + i
    i = i + 2
print(total)
```

Практика 2: разбор и критерии

Нужно range(9, -1, -3), чтобы 0 оказался больше stop и вошёл в перебор. range(0, 0, -3) пуст. Нулевой шаг вызывает ValueError. Критерии: все четыре числа, правильная граница, строка после цикла, объяснение двух особых случаев.

```
for i in range(9, -1, -3):
    print(i)
print("старт")
```

Обсуждение: возможный вывод

range удобен для известного арифметического перебора. Поток до маркера заранее не задаёт число элементов, поэтому while естественнее. Оба вида могут решать многие задачи, но должны ясно выражать условие завершения.

Индивидуальная проверка: ответ и критерии

3, 7, 11; 3 элемента, сумма 21. Следующее значение 15 войдёт при stop=16 (также подойдут 17, 18, 19, но 16 — ближайшая целая граница). 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: while, целые числа и накопление. 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
Циклы по переменной. Шаг изменения	Задаёт start, stop, step	Верный прямой и обратный перебор
Цикл с предусловием	Строит равносильный while	Одинаковый вывод и сумма

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.