



Урок 17. Python: Символы, индексы и перебор

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 17 из 33 · 45 минут · Модуль «Символьные строки»

Уровень: начинающий программист; Python 3.

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

В строке АБА буква А встречается дважды, но индексы начинаются с 0. Как перечислить символы и не обратиться к несуществующему элементу пустой строки?

Результат урока: Безопасно перебрать строку по индексам, посчитать заданный символ и проверить пустой и односимвольный случаи.

01. Проблемная ситуация

0–5 минут. На бумаге подпишите индексы строки АБА. Ученик пишет `range(len(s)+1)`. Какой индекс окажется лишним? Что произойдёт с пустой строкой? Сначала объясните, затем сравните с правилом `0 ≤ i < len(s)`.

02. Понятия и разобранный пример

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

Длина не является последним индексом

Строка Python — неизменяемая последовательность кодовых точек Unicode. Для простых букв в наших примерах один элемент соответствует одной букве. Видимый символ сложного текста может состоять из нескольких кодовых точек; этот урок не считает пользовательские символы сложных эмодзи или составных знаков.

При длине n допустимы неотрицательные индексы $0 \dots n-1$. `range(n)` выдаёт именно эти индексы и ничего не выдаёт для $n=0$. Прямой перебор `for ch in s` удобнее, если позиция не нужна; перебор индексов нужен для сопоставления позиции и символа.

Сравнение чувствительно к регистру и алфавиту. Кириллическая А и латинская А — разные элементы. Примеры здесь используют кириллическую А, а пробел тоже занимает позицию. Не удаляйте пробелы молча: это изменяет исходную строку и индексы.

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

В строке "А БА" длина 4: 0→А, 1→пробел, 2→Б, 3→А. Счётчик А изменится на индексах 0 и 3, итог 2. Для пустой строки цикл не выполняется, итог 0; обращений `s[0]` нет.

```
def count_a(s):
    total = 0
    for i in range(len(s)):
        if s[i] == "А":
            total += 1
    return total
```

```
for s in ["", "А", "АБА", "А БА", "аА", "ААА"]:
    print(repr(s), count_a(s))
```

Индекс	0 1 2 3
Строка «А БА»	А пробел Б А
Сравнение с А	да нет нет да
Счётчик после позиции	1 1 1 2

Трассировка счётчика кириллической А в строке из четырёх элементов; пробел имеет индекс 1.



03. Практика 1. Проверяем материал

Буква, регистр и пустая строка

14–25 минут. Заполните для каждой строки журнал «длина, допустимые индексы, ответ». Для А БА выпишите каждый шаг.

Строка (кавычки не входят)	Длина / число кириллических А
""	0 / 0
"А";"АБА"	1 / 1;3 / 2
"А БА"	4 / 2
"аА"	2 / 1
"ААА" (латиница)	3 / 0

2 минуты — прогноз;4 — запуск;3 — замените `range(len(s))` на `range(len(s)+1)`, зафиксируйте `IndexError` на пустой строке и исправьте;2 — объясните различие алфавитов. Критерии: шесть ответов, границы индексов, пустая строка, точное сравнение.

04. Практика 2. Перенос способа

Соседние пары с перекрытием

25–33 минуты. Посчитайте соседние пары "аа" в кириллической строке, разрешая перекрытие. Например, в "ааа" две пары: индексы(0,1) и(1,2). За 2 минуты предскажите "", "а", "аа", "ааа", "аабаа" (последняя а латинская), за 4 напишите функцию, за 2 проверьте. Критерии: соседство, перекрытие, короткие строки, правильная верхняя граница.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты сравните перебор символов и индексов, за 3 объясните выбор для одиночного счётчика и соседних пар, за 2 обсудите, почему одинаково выглядящие А и а могут считаться по-разному.

06. Индивидуальная проверка

40–45 минут. Для строки "БАА" назовите длину, последний неотрицательный индекс, результат `count_a` и число соседних пар "АА" при аналогичном регистрозависимом счётчике. По баллу за пункт.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Ответы 0,1,2,2,1,0. При ошибочном+1 даже пустая строка получает попытку s[0]; правильный цикл не делает ни шага. Индексы А БА:0,1,2,3. Исправленный полный код находится в примере.

Практика 2: разбор и критерии

Ответы 0,0,1,2,1. Перебираем i от 0 до len(s)-2; s[i+1] тогда существует. Для длины 0 или 1 range(len(s)-1) пуст. Замена на range(len(s)) вызывает выход за границу.

```
def adjacent_aa(s):
    total = 0
    for i in range(len(s) - 1):
        if s[i] == "a" and s[i+1] == "a":
            total += 1
    return total

for s in ["", "a", "aa", "aaa", "aaбаа"]:
    print(repr(s), adjacent_aa(s))
```

Обсуждение: возможный вывод

Для одиночного символа достаточно for ch in s, для пары нужны две соседние позиции либо иной способ хранить предыдущий символ. Unicode различает алфавиты; визуальное сходство не означает равенства. Контракт должен говорить, какие символы считаются.

Индивидуальная проверка: ответ и критерии

3;2;2;1. Это заглавные АА: функция adjacent_aa из второй практики считает строчные и на "БАА" вернёт 0; для заглавных необходимо изменить сравниваемые буквы.

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: Python: for, range, len, счётчик и сравнение. 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
Символьные строки	Безопасно перебрать строку по индексам, посчитать заданный символ и проверить пустой и односимвольный случаи.	Исходник, таблица прогнозов и фактических результатов
Символьные строки, обращение к символам, операции и преобразования	Соседние пары с перекрытием	Самостоятельное решение и индивидуальная проверка по четырём критериям

Источники и права на изображения

Исходная программа курса

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

Python: строки Unicode и последовательности

<https://docs.python.org/3/library/stdtypes.html#text-sequence-type-str>

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.