



Урок 18. Python: Поиск, вставка и замена

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 18 из 33 · 45 минут · Модуль «Символьные строки»

Уровень: начинающий программист; Python 3.

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

В учебной заметке надо вставить метку перед первым разделителем |. Если его нет, текст должен остаться прежним. Почему результат find нельзя сразу использовать как индекс?

Результат урока: Обработать отсутствующий фрагмент и собрать новую строку срезами, различая первую и массовую замену.

01. Проблемная ситуация

0–5 минут. Для "ab|cd", "abcd", "|ab" предскажите индекс первого |. Значение –1 означает «не найдено». Что сделает ошибочная вставка через `s[:–1]+"!" + s[–1:]` с текстом "abcd"?

02. Понятия и разобранный пример

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

find возвращает позицию или признак отсутствия

`s.find("|")` возвращает индекс первого вхождения либо –1. Ноль — успешный результат для разделителя в начале. Поэтому if i: неверно различает успех и неудачу: 0 ложно, а –1 истинно. Проверяйте `i == –1` явно.

Строку нельзя изменить присваиванием `s[i]="!"`. Вставка создаёт новую строку: `s[:i]+"!" + s[i:]`. Левый срез не включает позицию i, правый включает её. Исходная строка остаётся прежней, если результат не присвоить переменной.

Для замены используем `replace(old,new,count)`: без count заменяются все неперекрывающиеся вхождения, с count=1 — первое. Это не повторение замены до устойчивого состояния. Условие урока допускает строки длиной 0...30 из простых букв и знаков; фрагменты поиска фиксированы и непусты.

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

"ab|cd": i=2, s[:2]="ab", s[2:]="cd"; итог "ab!|cd". "abcd": i=–1, возвращаем исходное. Ошибочная вставка без проверки дала бы "abc!d". Для "|ab" i=0 и вставка допустима: "!|ab".

```
def mark_first(s):
    i = s.find("|")
    if i == -1:
        return s
    return s[:i] + "!" + s[i:]
```

```
for s in ["ab|cd", "abcd", "|ab", "ab|", "a|b|c", ""]:
    print(repr(s), repr(mark_first(s)))
```

s = ab cd	find(" ") = 2
s[:2]	ab
Новая вставка + s[2:]	! + cd
Результат	ab! cd; исходный s не изменён

Срезы при вставке перед первым разделителем: позиция 2 принадлежит правой части, поэтому | сохраняется.



03. Практика 1. Проверяем материал

Вставка перед первым разделителем

14–25 минут. Проверьте шесть строк таблицы и запишите найденный индекс для каждой.

s	mark_first(s)
"ab cd";"abcd"	"ab! cd";"abcd"
" ab";"ab "	"! ab";"ab! "
"a b c"	"a! b c"
""	""

2 минуты — прогноз; 4 — запуски; 3 — уберите проверку—1 и покажите ошибку на "abcd", затем замените проверку на `if not i` и объясните ошибку на "|ab"; 2 — восстановите. Критерии: индексы, шесть результатов, два контрпримера, сохранение исходного текста при отсутствии.

04. Практика 2. Перенос способа

Замена всех неперекрывающихся вхождений

25–33 минуты. Одним вызовом замените все неперекрывающиеся пары "--" на "-". Не повторяйте операцию до исчезновения пар. За 2 минуты предскажите "a--b", "----", "---", "ab", ""; за 4 напишите `shorten`; за 2 сравните с `replace(...,1)`. Критерии: один проход, неперекрывание, сохранённые одиночные знаки, различие первой и полной замены.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты сравните фразы «заменить пары один раз» и «сжать любую серию до одного знака», за 3 предъявите различающий вход, за 2 запишите уточнение контракта. Объясните, почему повторный запуск не всегда сохраняет результат.

06. Индивидуальная проверка

40–45 минут. Для "|x|" запишите `find("|")`, `mark_first`, результат замены всех | на / и результат такой замены с `count=1`. Четыре пункта — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Индексы 2, -1, 0, 2, 1, -1. Без проверки -1 строка `abcd` превратится в `abc!d`. Проверка `if not i` ошибочно считает индекс 0 отсутствием. Правильный файл дан выше; обе мутации выполняются отдельно.

Практика 2: разбор и критерии

Один вызов даёт `"a-b", "--", "--", "ab", ""`. Для `"----"` две неперекрывающиеся пары становятся двумя одиночными дефисами; повторный вызов дал бы `"-"`, но это другая операция. С `count=1` результат `"--"`.

```
def shorten(s):
    return s.replace("--", "-")

for s in ["a-b", "----", "---", "ab", ""]:
    print(repr(s), repr(shorten(s)),
          repr(s.replace("--", "-", 1)))
```

Обсуждение: возможный вывод

Для `"----"` один вызов `replace` даёт `--`, сжатие серии `----` — `-`. Операция `shorten` не идемпотентна на этом входе: второй вызов меняет результат. Необходимо явно задать один вызов, все вхождения или повторение до устойчивости.

Индивидуальная проверка: ответ и критерии

`0; "!|x|"; "/x/"; "/x|"`. Кавычки обозначают границы строки и в результат не входят.

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: строки, `len`, индексы и условные операторы. 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	Обработать отсутствующий фрагмент и собрать новую строку срезами, различая первую и массовую замену.	Исходник, таблица прогнозов и фактических результатов
Символьные строки, обращение к символам, операции и преобразования	Замена всех неперекрывающихся вхождений	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: find, replace и операции со строками

<https://docs.python.org/3/library/stdtypes.html#str.find>

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.