


ХОЧУ ПОМОЧЬ
ПРОЕКТУ

Приемы работы со строками в Python



[Справочник по языку Python3.](#) / Приемы работы со строками в Python

Использование текстовых строк str в Python

Этот раздел просвещен исследованию свойств строки в Python на примерах и практическому применению [операции с текстовыми строками](#).

Раздел представляет собой короткие примеры с решением конкретных задач, а так же разбираются различные случаи [манипулирования и преобразования, выполняемые со строками](#).

Смотрите также:

- [Тип str - текстовые строки в Python.](#)
- [Общие операции с последовательностями list, tuple, str в Python.](#)

[Преобразование строки в число в Python](#)

В Python нельзя смешивать строковые и числовые типы в таких операторах, как '+', но можно вручную выполнить необходимые преобразования перед выполнением.

[Строку в список, кортеж или множество символов в Python](#)

Для преобразования строки в список символов необходимо исходную строку передать в класс list(). Это полезно для каких то преобразований в цикле с отдельными символами в строке (удаление/добавление/изменение) или если нужно отобразить строку наоборот (развернуть строку).

[Одинарные, двойные и тройные кавычки в строке Python](#)

Текстовые строки могут быть заключены в одинарные кавычки ('...') или двойные кавычки ("..."), что в результате будет одно и то же. Знак обратной косой черты '\' может использоваться для экранирования кавычек.

[Вывод специальных символов в строке Python "как есть"](#)

Вывод специальных символов в строке Python "как есть". Использование необработанных (сырых) строк Python.

[Объединение и повторение строк в Python](#)

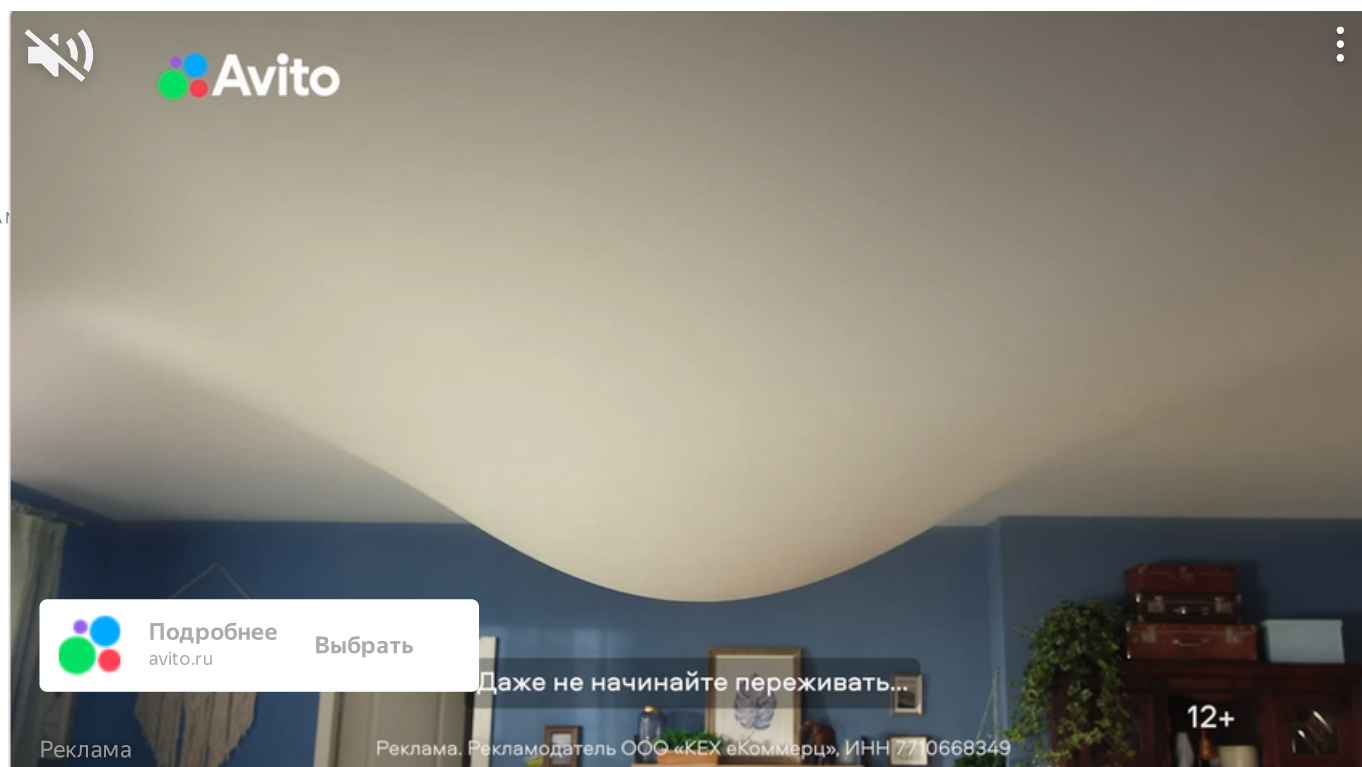
Строки могут быть объединены (склеены) оператором '+' и повторены с помощью '*'. Два или более строковых литерала, те, которые заключены в кавычки рядом друг с другом, автоматически объединяются.

[Индексирование строк в Python](#)

Строки могут быть индексированы, причем первый символ имеет индекс 0. В Python не существует отдельного типа "символ" (char), символ - это просто строка размером один

[Использование срезов строк в Python](#)

Использование срезов позволяет получить подстроку. Пропущенный первый индекс по умолчанию равен нулю, пропущенный второй индекс по умолчанию равен размеру/длине разрезаемой строки.



[Срез строки Python от заданного символа/знака до конца строки](#)

Можно воспользоваться `str.partition()` разбивает строку на три части: все что слева от разделителя, сам разделитель, то что справа от разделителя. Если разделитель не найден, то возвращается исходная строка и две пустых строки, т. е. в любом случае возвращается кортеж из 3 элементов.

[Создание макета для печати базовыми методами строк в Python](#)

Когда вам нужно быстрое отображение некоторых переменных для отладки, вы можете преобразовать любое значение в строку с помощью функций `repr()` или `str()`

[Способы форматирования текстовых строк в Python](#)

Форматирование строковыми литералами, форматирование методом `str.format()`, форматирование оператором `'%'` в стиле языка C.

[Подсчет повторений слов в тексте, деление строки на список слов](#)

Деление строки Python на список слов или подстрок. Разделить строку на список слов можно несколькими способами, разберем 2 самых популярных.

[Удаление слова из строки Python по фрагменту или шаблону](#)

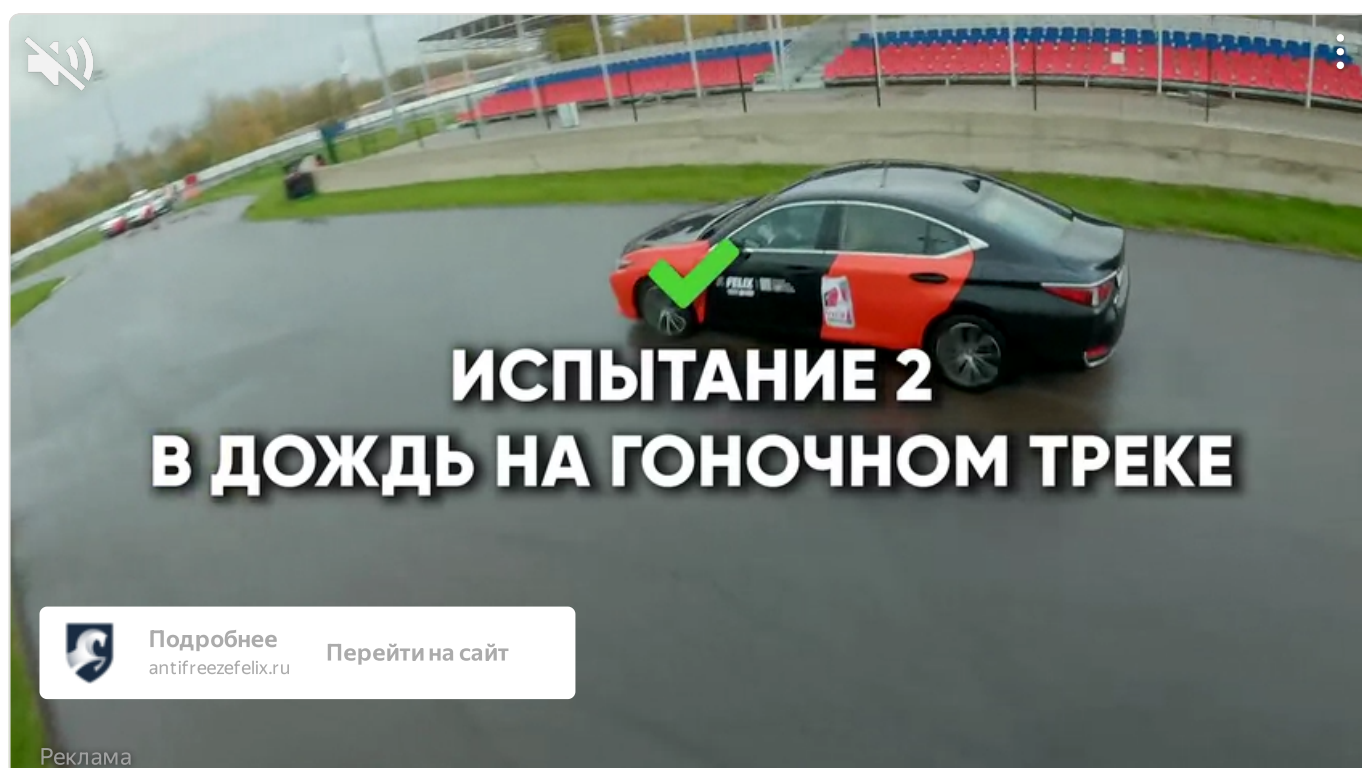
Удаление слова из строки Python по фрагменту или шаблону.

[Удаление пунктуации из строки в Python](#)

Для удаления пунктуации из строки Python воспользуемся методом строки `str.replace()`. А именно, в цикле пройдемся по всем символам пунктуации, и если они есть, то просто заменим его на пустую строку.

[Деление строки по заглавным буквам в Python](#)

Для деления текста или строки по заглавным буквам воспользуемся позитивной опережающей проверкой регулярных выражений и функцией `re.split()` модуля `re`.



[Создание отчетов и писем по шаблонам в Python](#)

Вверх

Довольно часто возникают ситуации, когда нужно генерировать определенные отчеты или манипулировать строками. Общим для всех этих ситуаций является то, что создание отчета или обработка строк следует определенному шаблону. Обычно шаблоны очень похожи и мы их используем повторно.

РЕКЛАМА

DOCS-Python.ru™, 2023 г.

⋮

(Внимание! При копировании материала ссылка на источник обязательна)

@docs_python_ru