

11.5 Методы строк: split(), join()

Шаг 1

Тема урока: строковые методы

1. Метод `split()`

2. Метод `join()`

Аннотация. Строковые методы `split()` и `join()`.

В предыдущем модуле мы детально изучили основные строковые методы, однако обошли стороной два важных: `split()` и `join()`, имеющих отношение к спискам. Они как бы противоположны по смыслу: метод `split()` разбивает строку по произвольному разделителю на список слов, а метод `join()` собирает строку из списка слов через заданный разделитель.

Метод `split()`

Метод `split()` разбивает строку на слова, используя в качестве разделителя последовательность пробельных символов, и возвращает список из этих слов.

Приведённый ниже код:

```
s = 'Python is the most powerful language'  
words = s.split()  
print(words)
```

выводит:

```
['Python', 'is', 'the', 'most', 'powerful', 'language']
```

Строка `s`

`s.split()`

Список `words`

'Python is the most powerful language'

['Python', 'is', 'the', 'most', 'powerful', 'language']

Таким образом, вызов метода `split()` разбивает **строку** на слова и возвращает **список**, содержащий все слова.

Рассмотрим следующий программный код:

```
numbers = input().split()
```

Если при запуске этой программы ввести строку `1 2 3 4 5`, то список `numbers` будет следующим `['1', '2', '3', '4', '5']`. Обратите внимание, что список будет состоять из строк (тип `str`), а не из чисел (тип `int`). Если требуется получить именно список чисел, то затем нужно элементы списка по одному преобразовать в числа с помощью команды `int()`.

Приведённый ниже код:

```
numbers = input().split()  
for i in range(len(numbers)):  
    numbers[i] = int(numbers[i])  
print(numbers)
```

выводит (если на вход программе была подана строка `1 2 3 4 5`):

```
[1, 2, 3, 4, 5]
```

Необязательный параметр

У метода `split()` есть необязательный параметр, который определяет, какой набор символов будет использоваться в качестве разделителя между элементами списка. Например, вызов метода `split('.)')` вернёт список, полученный разделением исходной строки по символу `'.'`.

Приведённый ниже код:

```
ip = '192.168.1.24'  
numbers = ip.split('.') # указываем явно разделитель  
print(numbers)
```

выводит:

```
['192', '168', '1', '24']
```



Метод `join()`

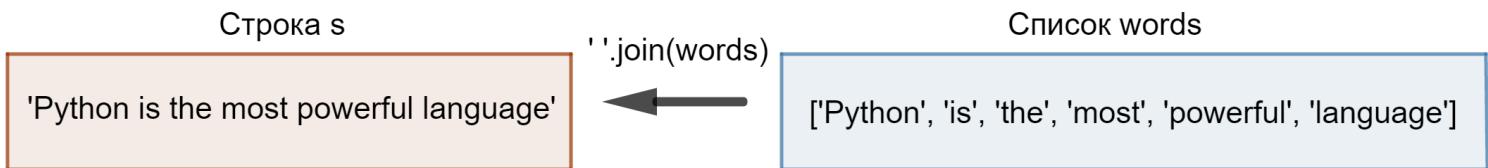
Метод `join()` собирает строку из элементов списка, используя в качестве разделителя строку, к которой применяется метод.

Приведённый ниже код:

```
words = ['Python', 'is', 'the', 'most', 'powerful', 'language']  
s = ' '.join(words)  
print(s)
```

выводит:

```
Python is the most powerful language
```



Обратите внимание: все слова разделены одним пробелом, поскольку метод `join()` вызывался на строке, состоящей из одного символа пробела `' '`.

Рассмотрим ещё пару примеров:

```
words = ['Мы', 'учим', 'язык', 'Python']  
print('*'.join(words))  
print('-'.join(words))  
print('?'.join(words))  
print('!'.join(words))  
print('*****'.join(words))  
print('abc'.join(words))  
print('123'.join(words))
```

Приведённый выше код выводит:

```
Мы*учим*язык*Python  
Мы-учим-язык-Python  
Мы?учим?язык?Python  
Мы!учим!язык!Python  
Мы*****учим*****язык*****Python  
МыавсучимавсязыкавсPython  
Мы123учим123язык123Python
```



Запомни: строковый метод `split()` служит для преобразования строки в список, а метод `join()` – для преобразования списка в строку.

Примечания

Примечание 1. Существует большая разница между результатами вызова методов `s.split()` и `s.split(' ')`. Разница в поведении проявляется, когда строка содержит несколько пробелов между словами.

Приведённый ниже код:

```
s = 'I love Python'  
words1 = s.split()  
words2 = s.split(' ')  
print(words1)  
print(words2)
```

выводит:

```
['I', 'love', 'Python']  
['I', 'love', '', 'Python']
```

Если в строке встречаются последовательные разделители, то они будут считаться как отдельные разделители, и между ними будет создана пустая строка (`' '`).

Примечание 2. Методы `split()` и `join()` являются строковыми методами, они не могут применяться к спискам!

Приведённый ниже код:

```
print([1, 2].split())
```

приводит к возникновению ошибки:

```
AttributeError: 'list' object has no attribute 'split'
```

Примечание 3. Строковый метод `join()` не работает для списков, у которых элементы не строкового типа данных.

Приведённый ниже код:

```
numbers = [1, 2, 3, 4] # список чисел  
s = '*'.join(numbers)  
print(s)
```

приводит к возникновению ошибки:

```
TypeError: sequence item 0: expected str instance, int found
```

Примечание 4. На самом деле, метод `join()` в качестве аргумента может принимать не только список, но и строку.

Приведённый ниже код:

```
s = '+'.join('pygen')  
print(s)
```

выводит:

Шаг 2

Заполните пропуски.

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/2>

Шаг 3

Что покажет приведённый ниже код?

```
s = 'a      b c'  
print(s.split())
```

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/3>

Шаг 4

Что покажет приведённый ниже код?

```
s = 'a      b c'  
print(s.split(' '))
```

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/4>

Шаг 5

Что покажет приведённый ниже код?

```
print('-'.join(['pen', 'pineapple', 'apple', 'pen']))
```

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/5>

Шаг 6

Что покажет приведённый ниже код?

```
letters = ['B', ' ', 'T', ' ', 'S']  
print(letters.split())
```

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/6>

Шаг 7

Что покажет приведённый ниже код?

```
print('-'.join('DNA'))
```

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/7>

Шаг 8

Что покажет приведённый ниже код?

```
symbols = ['I', 'D', 'O', 'L']
print(symbols.join('-'))
```

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/8>

Шаг 9

С помощью функции `list()` можно из строки получить список ее символов, а с помощью функции `join()` можно склеить элементы списка, вставляя между ними разделитель.

Что покажет приведённый ниже код?

```
s = 'BEEGEEK'
chars = list(s)
s = '**'.join(chars)
print(s)
```

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/9>

Шаг 10

Построчный вывод

На вход программе подаётся строка текста. Напишите программу, которая выводит слова введённой строки в столбик.

Формат входных данных

На вход программе подаётся строка текста.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

► Тестовые данные

Sample Input:

```
У лукоморья дуб зеленый златая цепь на дубе том
```

Sample Output:

```
У  
лукоморья  
дуб  
зеленый  
златая  
цепь  
на  
дубе  
тому
```

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/10>

Шаг 11

Инициалы

На вход программе подаётся строка текста, содержащая имя, отчество и фамилию человека. Напишите программу, которая выводит инициалы человека.

Формат входных данных

На вход программе подаётся строка текста, содержащая имя, отчество и фамилию человека.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

► Тестовые данные

Sample Input 1:

```
Владимир Семенович Высоцкий
```

Sample Output 1:

```
В.С.В.
```

Sample Input 2:

```
Гуев Тимур Ахсарбекович
```

Sample Output 2:

```
Г.Т.А.
```

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/11>

Шаг 12

Windows OS

В операционной системе Windows полное имя файла состоит из буквы диска, после которого ставится двоеточие и символ `\`, затем через такой же символ перечисляются подкаталоги (папки), в которых находится файл, в конце пишется имя файла (`C:\Windows\System32\calc.exe`).

На вход программе подаётся одна строка с корректным именем файла в операционной системе Windows. Напишите программу, которая разбирает строку на части, разделенные символом `\`. Каждую часть вывести в отдельной строке.

Формат входных данных

На вход программе подаётся одна строка.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

Примечание. В Python символ `\` обычно используется для создания специальных символьных последовательностей, которые представляют собой управляющие символы или экранированные последовательности. Например, `\n` представляет символ новой строки, `\t` представляет символ табуляции и так далее. Однако если символ `\` используется как часть строки, его следует экранировать, то есть использовать два обратных слэша `\\"`.

► Тестовые данные

Sample Input:

```
C:\Windows\System32\calc.exe
```

Sample Output:

```
C:  
Windows  
System32  
calc.exe
```

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/12>

Шаг 13

Диаграмма

На вход программе подаётся строка текста, содержащая целые числа. Напишите программу, которая по заданным числам строит столбчатую диаграмму.

Формат входных данных

На вход программе подаётся строка текста, содержащая целые числа, разделённые символом пробела.

Формат выходных данных

Программа должна вывести столбчатую диаграмму.

► Тестовые данные

Sample Input 1:

```
1 2 3 4 5
```

Sample Output 1:

```
+  
++  
+++  
++++  
+++++
```

Sample Input 2:

```
5 3 1 7 10 2
```

Sample Output 2:

```
+++++  
+++  
+  
++++++  
++++++  
++
```

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/13>

Шаг 14

Корректный ip-адрес

На вход программе подаётся строка текста, содержащая 4 целых неотрицательных числа, разделённых точкой. Напишите программу, которая определяет, является ли введённая строка текста корректным ip-адресом.

Формат входных данных

На вход программе подаётся строка текста, содержащая 4 целых числа, разделённых точкой.

Формат выходных данных

Программа должна вывести «ДА» (без кавычек), если введённая строка является корректным ip-адресом, или «НЕТ» (без кавычек) в противном случае.

Примечание. ip-адрес является корректным, если все 4 числа находятся в диапазоне от 0 до 255 **включительно**.

► Тестовые данные

```
192.168.0.3
```

Sample Output 1:

```
ДА
```

Sample Input 2:

```
192.168.0.300
```

Sample Output 2:

```
НЕТ
```

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/14>

Шаг 15

Добавь разделитель

На вход программе подаётся строка текста и строка-разделитель. Напишите программу, которая вставляет указанный разделитель между каждым символом введённой строки текста.

Формат входных данных

На вход программе подаются строка текста и строка-разделитель, каждая на отдельной строке.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

▶ Тестовые данные

Sample Input 1:

```
1234567  
*
```

Sample Output 1:

```
1*2*3*4*5*6*7
```

Sample Input 2:

```
qwerty and password  
**
```

Sample Output 2:

```
q**w**e**r**t**y** **a**n**d** **p**a**s**w**o**r**d
```

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/15>

Шаг 16

Количество совпадающих пар

На вход программе подаётся строка текста, содержащая целые числа. Из данной строки формируется список чисел.

Напишите программу, которая подсчитывает, сколько в полученном списке пар элементов, равных друг другу. Считается, что любые два элемента, равные друг другу, образуют одну пару, которую необходимо посчитать.

Формат входных данных

На вход программе подаётся строка текста, содержащая целые числа, разделённые символом пробела.

Формат выходных данных

Программа должна вывести одно число – количество пар элементов, равных друг другу.

▶ Тестовые данные

Sample Input 1:

```
1 7 5 7 5
```

Sample Output 1:

```
2
```

Sample Input 2:

3 3 3 3 3

Sample Output 2:

10

Sample Input 3:

8 7 6

Sample Output 3:

0

Чтобы решить это задание откройте <https://stepik.org/lesson/324755/step/16>