# Преобразование списка в строку методом join

Метод `join` в Python отвечает за объединение списка строк с помощью определенного указателя. Часто это используется при конвертации списка в строку. Например, так можно конвертировать список букв алфавита в разделенную запятыми строку для сохранения.

Метод принимает итерируемый объект в качестве аргумента, а поскольку список отвечает этим условиям, то его вполне можно использовать. Также список должен состоять из строк. Если попробовать использовать функцию для списка с другим содержимым, то результатом будет такое сообщение: `TypeError: sequence item 0: expected str instance, int found` .

Посмотрим на короткий пример объединения списка для создания строки.

КОПИРОВАТЬ

```
vowels = ["a", "e", "i", "o", "u"]
vowels_str = ",".join(vowels)
print("Строка гласных:", vowels_str)
```

Этот скрипт выдаст такой результат:

```
Строка гласных: a,e,i,o,u
```

## Почему join() — метод строки, а не списка?

Многие часто спрашивают, почему функция `join()` относится к строке, а не к списку. Разве такой синтаксис не было бы проще запомнить?

```
vowels_str = vowels.join(",")
```

Это популярный вопрос на StackOverflow, и вот простой ответ на него:

> Функция `join()` может использоваться с любым итерируемым объектом, но результатом всегда будет строка, поэтому и есть смысл иметь ее в качестве API строки.

## Объединение списка с несколькими типами данных

Посмотрим на программу, где предпринимается попытка объединить элементы списка разных типов:

КОПИРОВАТЬ

```
names = ['Java', 'Python', 1]
delimiter = ','
single_str = delimiter.join(names)
```

Вывод:

```
Traceback (most recent call last):
  File "test.py", line 3, in <module>
    single_str = delimiter.join(names)
TypeError: sequence item 2: expected str instance, int found
```

Это лишь демонстрация того, что `join` нельзя использовать для объединения элементов разного типа. Только строковые значения.

Что бы избежать этой ошибки, превратите все элементы списка в строки:

КОПИРОВАТЬ

```
names = ['Java', 'Python', 1]
names = [str(i) for i in names]
```

## Разбитие строки с помощью join

Также функцию `join()` можно использовать, чтобы разбить строку по определенному разделителю.

КОПИРОВАТЬ

```
>>> print(",".join("Python"))
P,y,t,h,o,n
```

Если передать в качестве аргумента функции строку, то она будет разбита по символам с определенным разделителем.

## Обратное преобразование строки в список

Помимо `join()` есть и функция `split()`, которая используется для разбития строки. Она работает похожим образом. Посмотрим на код:

КОПИРОВАТЬ

```
names = ['Java', 'Python', 'Go']
delimiter = ','
single_str = delimiter.join(names)
print('Строка: {0}'.format(single_str))

split = single_str.split(delimiter)
print('Список: {0}'.format(split))
```

Вывод:

```
Строка: Java,Python,Go
Список: ['Java', 'Python', 'Go']
```

Вот и все что нужно знать об объединении и разбитии строк.



**Максим**

Я создал этот блог в 2018 году, чтобы распространять полезные учебные материалы, документации и уроки на русском. На сайте опубликовано множество статей по основам python и библиотекам, уроков для начинающих и примеров написания программ.

Мои контакты: Почта

Статьи по теме

Лямбда-функции и анонимные функции в Python

Когда стоит использовать yield вместо return в Python

Как извлечь кубический корень в Python

Python цикл for — for i in range

Полное руководство по замене элементов списка на Python

Функции в Python

СОДЕРЖАНИЕ

НОВОЕ В БЛОГЕ