

Автоматическое удаление заголовков из csv-файлов с помощью языка программирования Python

Кизянов Антон Олегович

*Приамурский государственный университет имени Шолом-Алейхема
студент*

Аннотация

В данной статье продемонстрирован процесс создания системы автоматического удаления заголовков из csv-файлов.

Ключевые слова: Python, csv, os.

Automatically remove headers from a csv file using the Python programming language

Kizyanov Anton Olegovic

*Sholom-Aleichem Priamursky State University
student*

Abstract

This article demonstrated the process of creating a system for automatic removal of headers from a csv file.

Keywords: Python, csv, os.

Предположим, предстоит выполнить рутинную работу по удалению первых строк из нескольких сотен CSV-файлов. Возможно, вы собираетесь передавать эти файлы какому-то автоматизированному процессу, которому требуются только данные без заголовков столбцов. Вы могли бы открывать каждый файл в приложении Excel, удалять первую строку таблицы и заново сохранять файл, но это отняло бы вас много часов времени. Гораздо лучше написать программу, которая выполнит эту работу вместо вас.

Программа должна будет открывать каждый из файлов с расширением .csv, находящийся в текущем рабочем каталоге, читать содержимое CSV-файла и перезаписывать содержимое без первой строки в файл с тем же именем. В результате старое содержимое CSV-файла будет заменено новым, в котором заголовки столбцов таблицы отсутствуют.

Для ознакомления с языком программирования Python прочтите следующие статьи. В.А. Машков, В.И. Литвиненко рассказали о применении языка программирования python для решения задач самодиагностики на системном уровне [1]. Г.Д. Бухарова, и П.С. Комельских рассказали о важности и необходимости внедрения языка программирования Python в процесс обучения студентов [2]. Г.С. Сейдаметов продемонстрировал особенности использования языка программирования python в подготовке

будущих инженеров-программистов [3]. Э.А.Усеинов продемонстрировал использование объектно-ориентированного программирования в рамках дисциплины «язык программирования python» [4].

Первое, что должна сделать ваша программа, - это организовать цикл по всем CSVфайлам, находящимся в текущем рабочем каталоге.

```
import csv, os

os.makedirs('headerRemoved', exist_ok=True)

for csvFilename in os.listdir('.'):
    if not csvFilename.endswith('.csv'):
        continue

    print('Removing header from ' + csvFilename + '...')
```

Рис. 1

Вызов `os.makedirs()` создает папку `headerRemoved`, в которую будут записаны все CSVфайлы, не содержащие заголовков. Цикл `for` по элементам списка `os.listdir('.')` частично решает эту задачу, однако он итерирует по всем файлам, находящимся в рабочем каталоге, поэтому в начале цикла необходимо добавить код, обеспечивающий пропуск файлов, имена которых не заканчиваются расширением `.csv`. Если в цикле встречается такой файл, то инструкция `continue` обеспечивает его пропуск и переход к следующему имени файла.

Далее следует вызов функции `print()`, которая выводит на экран имя текущего обрабатываемого CSVфайла, что позволяет контролировать ход выполнения программы.

В действительности программа не удаляет первую строку каждого CSVфайла. Вместо этого она создает новую копию, но уже без первой строки. Поскольку имя файла-копии совпадает с именем исходного файла, он перезаписывает оригинал.

В программе необходимо предусмотреть проверку, позволяющую определить, является ли текущая строка в цикле первой.

```
csvRows = []
csvFileObj = open(csvFilename)
readerObj = csv.reader(csvFileObj)
for row in readerObj:
    if readerObj.line_num == 1:
        continue
    csvRows.append(row)
csvFileObj.close()
```

Рис. 2

Для отслеживания номера строки CSVфайла, которая читается в данный момент, можно использовать атрибут `line_num` объекта `Reader`. Другой

цикл `for` итерирует по строкам, возвращенным объектам `Reader`, и все строки, кроме первой, присоединяются к списку `csvRows`.

В процессе выполнения итераций цикла `for` по строкам код проверяет, является ли значение атрибута `readerObj.line_num` равным 1. Если это так, то инструкция `continue` осуществляет переход к следующей строке, не присоединяя текущую строку к списку `csvRows`. Для всех последующих строк условие будет иметь ложное значение, и они будут присоединяться к указанному списку.

Теперь, когда списке `csvRows` содержатся все строки, кроме первой, его нужно записать в CSV-файл, который будет находиться в папке `headerRemoved`.

```
csvFileObj = open(os.path.join('headerRemoved', csvFilename), 'w', newline='')
csvWriter = csv.writer(csvFileObj)
for row in csvRows:
    csvWriter.writerow(row)
csvFileObj.close()
```

Рис. 3

Объект `Writer` записывает список `csvRows` в CSV-файл в папке `headerRemoved`, используя переменную `csvFilename`. В результате этого исходный файл будет перезаписан.

Состав объект `Writer`, мы организуем цикл по всем подчиненным спискам, хранящимся в `csvRows`, и записываем каждый из них в файл.

После выполнения блока кода внешний цикл `for` перейдет к следующему имени файла из списка `os.listdir('.')`. По выполнении этого цикла программа завершается.

Данная программа должна выводить имя файла всякий раз, когда из CSV-файла исключается первая строка.

```
import csv, os

os.makedirs('headerRemoved', exist_ok=True)

for csvFilename in os.listdir('.'):
    if not csvFilename.endswith('.csv'):
        continue

    print('Removing header from ' + csvFilename + '...')

    csvRows = []
    csvFileObj = open(csvFilename)
    readerObj = csv.reader(csvFileObj)
    for row in readerObj:
        if readerObj.line_num == 1:
            continue
        csvRows.append(row)
    csvFileObj.close()

    csvFileObj = open(os.path.join('headerRemoved', csvFilename), 'w', newline='')
    csvWriter = csv.writer(csvFileObj)
    for row in csvRows:
        csvWriter.writerow(row)
    csvFileObj.close()
```

Рис. 4

На рисунке 4 показан весь код программы.

Вывод: Разработана программа удаления заголовков из csv-файлов.

Библиографический список

1. Машков В.А., Литвиненко В.И. Использование языка программирования python 3 и системы компьютерной алгебры sympy на факультативных занятиях по теории чисел // В сборнике: Электротехнические и компьютерные системы Издательство: Одесский национальный политехнический университет (Одесса) С. 48-54 [Электронный ресурс]. URL: <http://elibrary.ru/item.asp?id=23422667> (дата обращения: 25.01.2017)
2. Бухарова Г.Д., Комельских П.С. важность и необходимость внедрения языка программирования python в процесс обучения студентов // В сборнике: новые информационные технологии в образовании Материалы VII международной научно-практической конференции. Российский государственный профессионально-педагогический университет. 2014 Издательство: Российский государственный профессионально-педагогический университет (Екатеринбург) С. 40-42. [Электронный ресурс]. URL: <http://elibrary.ru/item.asp?id=22278620> (дата обращения: 25.01.2017)
3. Сейдаметов Г. С. Особенности использования языка программирования python в подготовке будущих инженеров-программистов // В сборнике: INTERNATIONAL SCIENTIFIC REVIEW Издательство: Олимп (Иваново) С. 50-51 [Электронный ресурс]. URL: <http://elibrary.ru/item.asp?id=24983350> (дата обращения: 25.01.2017)
4. Усеинов Э.А. Объектно-ориентированное программирование в рамках

дисциплины «язык программирования python» // В сборнике: ученые записки крымского инженерно-педагогического университета
Издательство: Государственное бюджетное образовательное учреждение высшего образования Республики Крым "Крымский инженерно-педагогический университет" (Симферополь) С. 157-160. [Электронный ресурс]. URL: <http://elibrary.ru/item.asp?id=24836776> (дата обращения: 25.01.2017)