

## **Разработка программы автоматического извлечения адресов электронной почты из блока текста на языке программирования Python**

*Кизянов Антон Олегович*

*Приамурский государственный университет имени Шолом-Алейхема  
студент*

### **Аннотация**

В данной статье продемонстрирован процесс создания программы нахождения адресов электронной почты из блока текста.

**Ключевые слова:** Python, re, pyperclip.

## **Development of the program automatic retrieval of email addresses from a block of text in the Python programming language**

*Kizyanov Anton Olegovic*

*Sholom-Aleichem Priamursky State University  
Student*

### **Abstract**

This article demonstrated the process of creating a program to find e-mail addresses from a block of text.

**Keywords:** Python, re, pyperclip.

Предположим, вам предстоит заняться рутинной работой – найти все адреса электронной почты, которые содержатся на длинной веб-странице или в документе большого размера. Если прокручивать страницу вручную, то на такой поиск у вас может уйти много времени. Но если бы у вас была программа, способная выполнять поиск электронных адресов в буфере обмена, то можно было бы просто нажать комбинацию клавиш <Ctrl+A> для выделения всего текста и комбинацию клавиши <Ctrl+C> для копирования выделенного текста в буфер, а затем выполнить программу. Такая программа смогла бы заменить находящийся в буфере текст найденными адресами электронной почты.

Для работы нам понадобятся следующие библиотеки re: стандартная библиотека регулярных выражений языка программирования Python, pyperclip [5]: сторонняя библиотека для получения из буфера обмена текста и вставки его.

Для ознакомления с языком программирования Python прочтите следующие статьи. В.А. Машков, В.И. Литвиненко рассказали о применение языка программирования python для решения задач самодиагностики на системном уровне [1]. Г.Д. Бухарова, и П.С. Комельских рассказали о важность 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].

Прежде всего, необходимо создать регулярное выражение для поиска адресов электронной почты.

```
emailRegex = re.compile(r'''(
    [a-zA-Z0-9._%+-]+      # имя
    @                      # символ @
    [a-zA-Z0-9.-]+        # доменное имя
    (\.[a-zA-Z]{2,4}){1,2} # точка и текст
)''', re.VERBOSE)
```

Рис. 1

На рисунке 1 изображен код регулярного выражения на языке программирования Python. Часть адреса, содержащая имя пользователя, включает один или более символов, которыми могут быть любые из следующих символов: буквы в верхнем регистре или нижнем регистре, цифры, точка, символ подчёркивания, знак процента, знак «плюс» и дефис. Все эти символы можно указать в виде символьного класса: [a-zA-Z0-9.\_%+-].

Имя домена отделяется от имени пользователя символом@. Доменное имя представляется с помощью более узкого класса, включающего только буквы, цифры, точку и дефис: [a-zA-Z0-9.-]. А последняя часть может содержать только точку и любой текст. Эта часть может включать от двух до четырех символов.

Теперь, когда у нас есть регулярное выражение для поиска адресов электронной почты, можно поручить модулю re выполнить утомительную работу по поиску в буфере обмена всех строк, соответствующих составленным регулярным выражениям. Метод ruperclip.paste() получит строковое значение текста, хранящегося в буфере обмена, а метод findall() вернет список кортежей.

Теперь, когда адреса электронной почты сохраняются в переменной matches, их необходимо скопировать в буфер обмена. Функция ruperclip.copy() принимает одиночное строковое значение, а не список строк, поэтому для переменных matches называются метод join().

Чтобы упростить проверку работы программы, выведем все найденные совпадения на экран. А если адресов электронной почты в тексте не найдено, будет выведено соответствующее сообщение.

Получилась следующая программа, изображенная на рисунке 2.

```
import pyperclip, re

emailRegex = re.compile(r'''(
    [a-zA-Z0-9._%+-]+      # имя
    @                      # символ @
    [a-zA-Z0-9.-]+        # доменное имя
    (\.[a-zA-Z]{2,4}){1,2} # точка и текст
)''', re.VERBOSE)

text = str(pyperclip.paste())

matches = []
for groups in emailRegex.findall(text):
    matches.append(groups[0])

if len(matches) > 0:
    pyperclip.copy('\n'.join(matches))
    print('Из буфера обмена:')
    print('\n'.join(matches))
else:
    print('Ничего не найдено.')
```

Рис. 2

Результат её работы можно наблюдать на рисунке 3. Для теста была скопирована страница <http://pgusa.ru/> .

```
Из буфера обмена:
rectorat@pgusa.ru
abiturient@pgusa.ru
```

Рис. 3

Эта программа очень сильно облегчит работу с поиском электронных адресов. Так же её можно переделать под себя изменив только регулярное выражение.

Вывод: Написали программу автоматического нахождения электронных адресов.

### 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)
  5. puperclip. [Электронный ресурс]. URL: <https://pypi.python.org/pypi/puperclip> (дата обращения: 25.01.2017)