УДК 004

Разработка программы автоматического поиска информации в Google через командную строку на языке Python

Кизянов Антон Олегович
Приамурский государственный университет имени Шолом-Алейхема
студент

Аннотация
В данной статье продемонстрирован процесс создания программы нахождения информации в Google через командную строку.
Ключевые слова: Python, requests, sys, webbrowser, bs4.

Development of the program automatic information search on Google from the command line in Python language

Kizyanov Anton Olegovic
Sholom-Aleichem Priamursky State University
Student

Abstract
This article demonstrated the process of creating a program of finding information on Google from the command line.
Keywords: Python, requests, sys, webbrowser, bs4.

Выполнив поиск в Google, никогда не начинаю сразу же просматривать все полученные ссылки одна за другой. Вместо этого щелкаю средней кнопкой мыши на несколько первых ссылок для открытия каждой на новой вкладке, чтобы просмотреть их позже. Поскольку поисковиком Google пользуюсь довольно часто, описанный порядок поиска – открытие браузера, поиск по заданному поисковому термину и последующие щелчки средней кнопкой мыши на нескольких ссылках – оказываются достаточно трудоемким. Было бы неплохо,если бы мог просто ввести поисковый термин в командной строке, а компьютер автоматически открыл бы браузер с первыми несколькими результатами поиска, размещенными на отдельны вкладках.

Для ознакомления с языком программирования Python требуется познакомиться со следующими статьями. 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 чем приступить к написанию кода, необходимо определить URLадрес страницы с результатами поиска. Взглянув на адресную строку браузера после выполнения поиска в Google, вы увидите там URL адрес вида https://www.google.com/search?q=поисковый_термин. Модель Requests может загружать эту страницу, после чего сможете использовать BeatifulSoup для нахождения ссылок на результаты поиска в HTML. Наконец, используете модуль webbrowser для открытия этих ссылок в отдельных вкладках браузера.

```
import requests, sys, webbrowser, bs4

print('Гуглим...')
res = requests.get('http://google.com/search?q=' + ' '.join(sys.argv[1:]))
res.raise_for_status()
```

Рис. 1

Поисковые термины будут представляться пользователем с помощью аргументов командной строки при загрузке программы. Эти аргумент будут сохраняться в виде строк в спискеsys.argv.

А теперь настал черед использовать модель BeatifulSoupдля извлечения нескольких первых ссылок на результаты поиска из загруженного HTMLдокумента. Однако как определить, какой селектор следует использовать для выполнения этой работы? Например, вы не можете просто отобрать все теги <a>, поскольку в полученном HTMLдокументе будет присутствовать множество ссылок, не представляющих для нас интереса. Вместо этого вы должны инспектировать страницу с результатами поиска с помощью инструментов разработчика, предоставляемых браузером, чтобы определить селектор, который отберет лишь нужные вам ссылки.

Используя для поиска в Googlestроку BeatifulSoupв качестве поискового термина, вы можете открыть окно инструментов разработчика и исследовать некоторые из элементов, содержащих ссылки.

Вам достаточно определить лишь шаблон, который является общим для всех ссылок. Однако в этом элементе <a> нет ничего, что позволило бы легко отличить его от элементов<a>, не имеющих никакого отношения к поиску.

```
soup = bs4.BeautifulSoup(res.text)

linkElems = soup.select('.r a')
```

Рис. 2

Однако, оглядевшись в окрестностях элемента <a>, вы заметите элемент наподобие <h3 class=”r”>. Просмотр оставшихся частей HTMLкода позволяет предположить, что класс r используется исключительно для

ссылок на результаты поиска. Вам необязательно знать, что собой представляет CSSкласс ri что он делает. Вы просто будете использовать его в качестве маркера элемента <a>, который ищете. Вы можете создать объектBeautifulSoup из HTMLтекста загруженной страницы, а затем использовать селектор ‘.ra’ для нахождения всех элементов <a>, которые вложены в элемент, имеющий CSS класс r.

Наконец, нам необходимо, чтобы программа открыла в браузере отдельныевкладки для каждого из результатов поиска.

```
numOpen = min(5, len(linkElems))
for i in range(numOpen):
    webbrowser.open('http://google.com' + linkElems[i].get('href'))
```

Рис. 3

По умолчанию вы открываете с помощью модуля webbrowser новые вкладки для пять результатов поиска. Однако пользователь мог выполнить поиск, дающий менее пяти результатов. Вызов soup.select() возвращает список всех элементов, соответствующих селектору ‘.ra’, поэтому количество открываемых вкладок либо будет равно 5, либо будет определяться длиной указанного списка(в зависимости от того, что меньше).

Встроенная функция Pythonmin() возвращает наименьшее из переданных ей целых или вещественных чисел. Можно использовать функцию min() для того, чтобы выяснить, содержит ли список менее пяти ссылок, исохраняет количество ссылок, подлежащих открытию, в переменной numOpen. После этого можно выполнить цикл for, вызвав функцию range(numOpen).

На каждой итерации цикла вы открываете новую вкладку в браузере с помощью вызова weebbrowser.open(). Обратите внимание на то, что значение атрибута href в возвращенных элементах <a>не содержит начальную часть URLадреса http://google.com, поэтому вы должны конкатенировать ее со строкой значение атрибута href.

```
import requests, sys, webbrowser, bs4

print('Гуглим...')
res = requests.get('http://google.com/search?q=' + ' '.join(sys.argv[1:]))
res.raise_for_status()

soup = bs4.BeautifulSoup(res.text)

linkElems = soup.select('.r a')
numOpen = min(5, len(linkElems))
for i in range(numOpen):
    webbrowser.open('http://google.com' + linkElems[i].get('href'))
```

Рис. 4

На рисунке 4 можно увидеть полный код нашей программы.

Вывод: Написали программу автоматическогонахождения и открывания ссылок нужной нам информации по термину из командной строки.

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. Beautiful Soup. [Электронный ресурс]. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (дата обращения: 25.01.2017)