Министерство образования и науки РФ

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра Информационные технологии и автоматизированные системы

Вычислительная лингвистика

Лабораторная работа № 1

Выполнил: студент группы РИС-16-1м

Леонов Евгений Русланович

Проверил: к.т.н., доцент кафедры ИТАС

Курушин Даниил Сергеевич

г. Пермь – 2016

Скакчиваем, устанавливаем и подключаем BeautifulSoup

Сначала так,

[http://wiki.python.su/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D0%B8/BeautifulSoup](http://wiki.python.su/Документации/BeautifulSoup)

from BeautifulSoup import BeautifulSoup

не получилось

потом так, уже получилось

http://python-3.ru/page/parser-html-python

from bs4 import BeautifulSoup

Находим инструкцию по открыванию сайта.

http://python.su/forum/topic/23576/

from urllib.request import urlopen

Причем, это для 3 версии питона. Сначала находил для второй (from urllib import urlopen

). Открываем нужный нам сайт (https://news.yandex.ru/science.html).

http://python-3.ru/page/parser-html-python

html\_doc = urlopen('https://news.yandex.ru/science.html').read()

Полученный html записываем в переменную, через BeautifulSoup.

soup = BeautifulSoup(html\_doc, ' lxml')

BeautifulSoup – библиотека, которая служит оберткой для парсера. Парсер в данном примере lxml. Сущесвуют другие парсеры.

[http://wmasteru.org/threads/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B9-%D0%BF%D0%B0%D1%80%D1%81%D0%B5%D1%80-%D1%82%D0%BE%D0%B2%D0%B0%D1%80%D0%BE%D0%B2-%D1%81-amazon-%D1%81-%D0%BF%D0%BE%D0%BC%D0%BE%D1%89%D1%8C%D1%8E-beautifulsoup.22880/](http://wmasteru.org/threads/Простой-парсер-товаров-с-amazon-с-помощью-beautifulsoup.22880/)

Используя примеры в интернете.

<http://python-3.ru/page/parser-html-python>

<https://toster.ru/q/260869>

<http://stackoverflow.com/questions/19983165/parse-href-attribute-value-from-element-with-beautifulsoup-and-mechanize>

Ищем новосной блок.

result\_dic={}

list\_temp = []

list\_story = soup.select("div.story")#ссылки на новости

Внутри блока находим ссылку, а внури нее кратинку и ссылку на нее.

for story in list\_story:

list\_temp.clear()

a = story.find('a', class\_='link')

img = a.find('img', class\_='image')

list\_temp.append(img.attrs["src"]) # Картинка

Так же внутри блока новости находим заголовок новости.

div = story.find('div', class\_='story\_\_content')

h2 = div.find('h2', class\_='story\_\_title')

#a\_title = h2.find('a', class\_='link')

#list\_temp.append(a\_title.get\_text())# можно так, но и без него работает

list\_temp.append(h2.get\_text()) # Заголовок

И ее описание краткое.

div\_text = div.find('div', class\_='story\_\_text')

list\_temp.append(div\_text.get\_text())

Полученные результаты сохраняем в словарь, ключом в котором является ссылка на новость, а значением массив. В массиве хранятся адрес на картинку, заголовок и описание, которые мы извлекали выше.

Адресс на ссылку, которая является кулючем словаря извлекли из тега a и его атрибута href сразу же после выделения новостного блока.

result\_dic["https://news.yandex.ru"+a.attrs["href"]] = list\_temp # записали ссылку на новость

Нашли урок по JSON.

<https://pythonworld.ru/moduli/modul-json.html>

Подключаем

import json

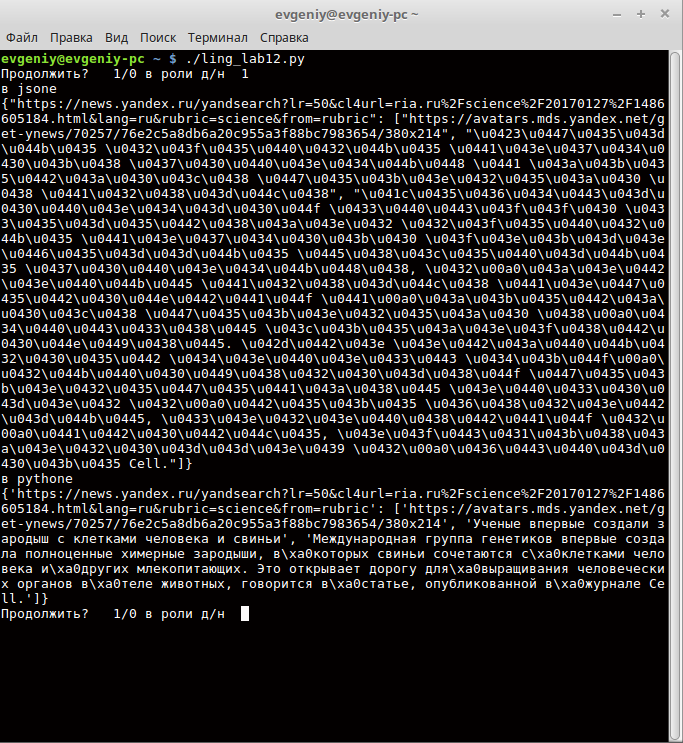
Пример записи в json.

temp=json.dumps(result\_dic)

Пример извлечения из json.

temp2=json.loads(temp)

Ниже на рисунке представлен фрагмент результата работы.

Рис. 1 — Результат работы при извлечении первой новости