

Алгоритмы и примеры парсинга сайтов для извлечения данных

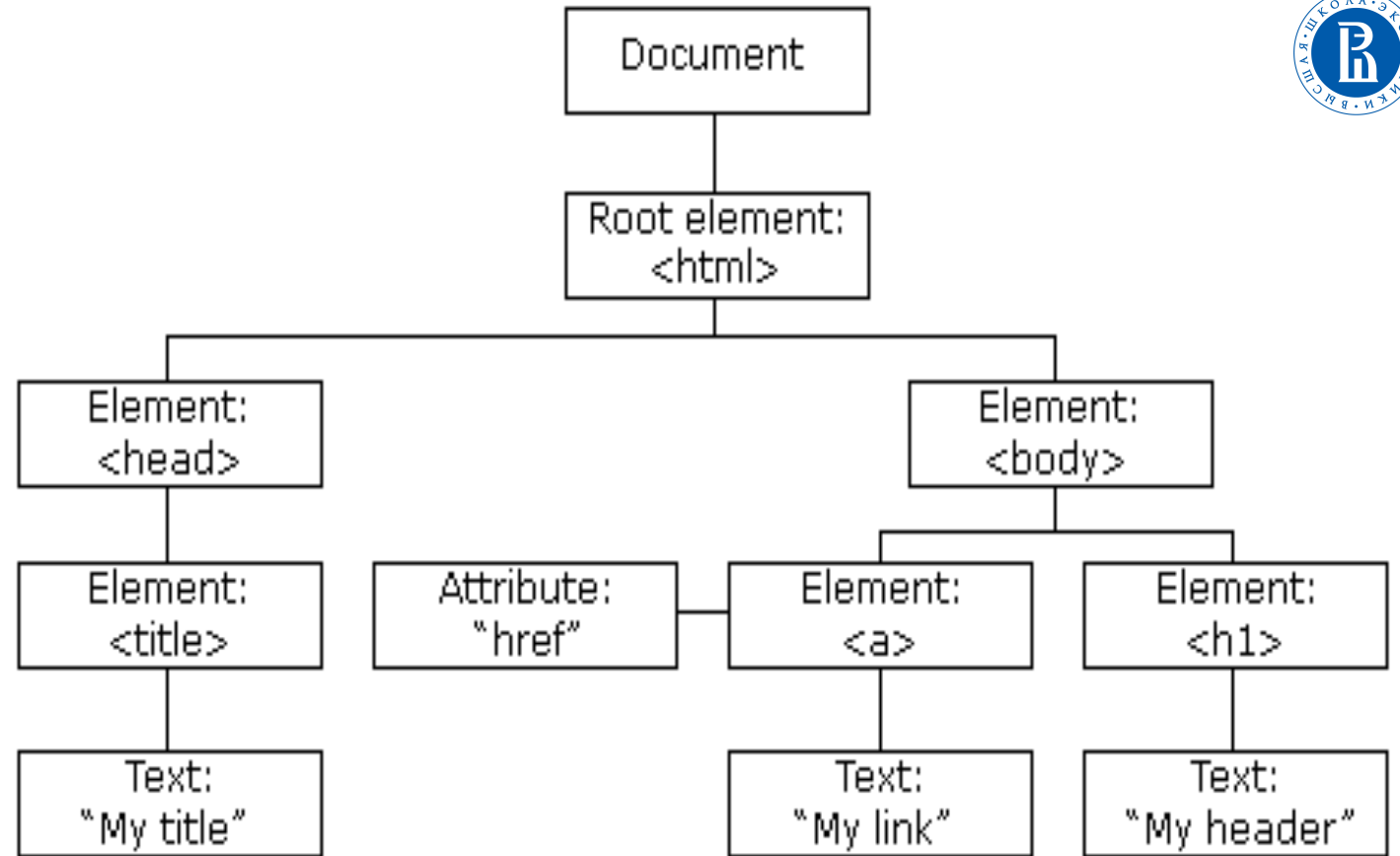


Докладчик:
Анастасия Родыгина

С чем имеем дело?



```
<!DOCTYPE html ...>
<html xmlns="...">
  <head>
    <title>My title</title>
  </head>
  <body>
    <a href="...">
      My link
    </a>
    <h1>My header</h1>
  </body>
</html>
```



DOM (Document Object Model) – объектная модель, используемая для XML/HTML-документов; представление документа в виде дерева тегов.



Задача?

- Быстро достать большое количество нужных данных

Как ее можно решать на питоне?

- Scrapy – создает «веб-паука»
- BeautifulSoup – может преобразовать даже неправильную разметку в дерево синтаксического разбора
- API (application programming interface) – программный интерфейс приложения. Упрощает написание кода, т.к. предлагает набор уже готовых классов, функций или структур для работы с имеющимися данными.



Задача?

- Быстро достать большое количество нужных данных

Каков алгоритм действий?

- Понять, где именно в дереве лежат данные (веб-инспектор)
- Проверить, как осуществляется переход на следующую страницу (page=1)
- Пройтись по нужным страницам, выделить нужные элементы (например, заголовки, значения в ячейках таблиц)
- Записать полученные данные в файл (форматы JSON, CSV, XML)

Применимость в социологических исследованиях



- Анализ данных социальных сетей
 - Данные о пользователях (соцдем)
 - Предпочтения пользователей (например, отношение к курению)
 - Данные о группах (количество подписчиков,
- Контент-анализ
- Создание базы данных для формирования выборки
- ... Ваши идеи?



Ограничения

- Нужно освоить подходящий язык программирования
- Нужно освоить не только язык, но и нужные библиотеки
- Страницы могут увидеть в вас бота (вы и есть бот) и заблокировать
- Сайты могут быть не только статичными (серверный рендеринг), но и динамическими (клиентский рендеринг)
- Нужно быть аккуратным с юридической точки зрения (например, не нарушать закон об авторском праве)
- Пользователи могут предоставлять ложную информацию
- Аккаунты-боты и контент, генерируемый ими

BeautifulSoup & Демоскоп Weekly



http://www.demoscope.ru/weekly/ssp/rus_gub_97.php

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

Задача:

Собрать в одну таблицу данные по всем губерниям (+ уезды).

Включать только те строки, где указана площадь.

```
import requests  
import bs4  
import csv
```

Scrapy & knife.media



<https://knife.media/category/how-to>
<https://docs.scrapy.org/en/latest/>

Задача:

Собрать ссылки на посты с одной страницы и их заголовки

- Открываем терминал
- Создаем новую директорию (папку) `mkdir *dirname*`
- `cd *dirname*`
- `scrapy startproject *projectname*`
- `cd *projectname*`
- `scrapy genspider *spidername* *link*`
- Переходим в папку `*dirname*`, ищем там папку `spiders`, в ней находим файл `*spidername*`
- Идем на сам сайт, в веб-инспекторе пытаемся написать селектор в формате `$$(*selector*)`
- Пишем в `*spidername*` код, который будет делать то, что хотим (или не совсем то)
- `scrapy crawl *spidername* -o *filename*.json`



Пример 3. API VK

<https://vk.com/dev>

Мои приложения => Создать приложение => Standalone-приложение => Подключить приложение => подтвердить по телефону => Сервисный ключ доступа: => CTRL+C => CTRL+V

https://vk.com/dev/first_guide

<https://vk.com/dev/methods>

Играемся в браузере

Пробуем писать код в джупитере



```
In [1]: print('Есть ли у вас вопросы?')  
Out [1]: Есть ли у вас вопросы?
```

```
In [2]: print('Спасибо за внимание!')  
Out [2]: Спасибо за внимание!
```