Семинар 3. Основы парсинга

Парсинг

Парсинг - автоматическое извлечение информации из документов/файлов/сайтов/что-угодно в текстовом виде

RSS (Rich Site Summary), сводка сайта

RSS является своеобразной упрощенной версией сайта, которая более структурирована и допустима (с точки зрения политики новостных сайтов) для парсинга. Строгая структура и отсутствие препятствий со стороны новостного агрегатора делают парсинг проще

RSS для Яндекса

По миру - https://news.yandex.ru/world.rss

Доступные RSS (Яндекс решил скрыть свой RSS, поэтому пользуемся архивом)- http://web.archive.org/web/20190322194855if_/http://news.yandex.ru/export.html

После нажатия на интересный нам RSS, в адресной строке (где указан адрес сайта) необходимо убрать все, что идет до http://news.yandex...,

XML и HTML

HTML и XML имеют схожий формат, их различия для нас при парсинге не важны. XML мы получаем обычно как результат RSS и некоторых API (программный интерфейс - набор ссылок/кнопочек для получения данных). HTML представляет все существующие в интернете Web-страницы: научились парсить XML → научились парсить весь интернет

RSS выдает информацию в виде XML

Пример XML (RSS Яндекса по миру):

```
<?xml version="1.0" encoding="utf-8" ?>
<rss version="2.0">
  <channel>
    <title>Яндекс.Новости: В мире</title>
    <link>https://news.yandex.ru/world.html?from=rss</link>
    <description>Первая в России служба автоматической обработки и систематизации новостей. Сообщения ведущих российских
и мировых СМИ. Обновление в режиме реального времени 24 часа в сутки.</description>
    <image>
      <url>https://company.yandex.ru/i/50x23.gif</url>
      <link>https://news.yandex.ru/world.html?from=rss</link>
      <title>Яндекс.Новости: В мире</title>
    </image>
    <lastBuildDate>19 Sep 2021 12:12:42 +0000</lastBuildDate>

    <item>
      <title>Командование США поздравило свои ВВС с днем формирования логотипом с истребителями Су-27</title>
      <link>https://yandex.ru/news/story/Komandovanie_SSHA_pozdravilo_svoi_VVS_sdnem_formirovaniya_logotipom_sistrebitel
yami_Su-27--e91243a8488231c9957f566224861b17?lang=ru&amp;from=rss&amp;wan=1&amp;stid=JSE90L79gZhLSmbSlDQu</link>
      <guid>https://yandex.ru/news/story/Komandovanie_SSHA_pozdravilo_svoi_VVS_sdnem_formirovaniya_logotipom_sistrebitel
yami_Su-27--e91243a8488231c9957f566224861b17?lang=ru&amp;from=rss&amp;wan=1&amp;stid=JSE90L79gZhLSmbSlDQu</guid>
      <description>Южное командование Министерства обороны США поздравило американские Военно-воздушные силы с 73-летие
м, опубликовав изображение с очертаниями советско-российских истребителей Су-27. В Twitter командования 18 сентября появ
ился поздравительный логотип с надписью: &amp;quot;С днем рождения, ВВС США!&amp;quot;.</description>
      <pubDate>19 Sep 2021 10:42:34 +0000</pubDate>
    </item>

    <item>
      <title>В США выявили массовое заражение детей респираторным вирусом RSV на фоне пандемии COVID-19</title>
      <link>https://yandex.ru/news/story/VSSHA_vyyavili_massovoe_zarazhenie_detej_respiratornym_virusom_RSV_nafone_pande
mii_COVID-19--10b6e34ba2df6febaf065934fc1ce851?lang=ru&amp;from=rss&amp;wan=1&amp;stid=gMOYICvtPBexOzHx6imU</link>
      <guid>https://yandex.ru/news/story/VSSHA_vyyavili_massovoe_zarazhenie_detej_respiratornym_virusom_RSV_nafone_pande
mii_COVID-19--10b6e34ba2df6febaf065934fc1ce851?lang=ru&amp;from=rss&amp;wan=1&amp;stid=gMOYICvtPBexOzHx6imU</guid>
      <description>В двух детских больницах американской столицы (город Вашингтон) наблюдается рост числа пациентов с ре
```

```
спираторно-синцитиальным вирусом (RSV), сообщает РИА Новости.</description>
      <pubDate>19 Sep 2021 08:37:41 +0000</pubDate>
    </item>
  </channel>
</rss>
```

Визуально (немного упрощено, подсвечено основное) это выглядит примерно так:

Таким образом мы можем построить путь/маршрут до определенного нужного нам поля. Блоков item у нас два. Поэтому пронумеруем их (точно так же поступит и python) начиная с нуля. В нашем случае получится item №0 и item №1.

Найдем путь до заголовка самой первой новости (в случае с RSS Яндекса item представляет собой одну новость):

rss → channel → item**[0]** → title

Библиотека requests

Перед тем, как записать этот путь в терминах Python, нам необходимо получить саму XML

Для простого выполнения интернет-запросов для Python придумали библиотеку (набор готовых функций)

Установка библиотеки

В python встроена утилита под названием pip. Для установки библиотека присутствует команда `pip install <имя библиотеки/пакета>`

В нашем случае необходимо выполнить команду `pip install requests`

Эту команду возможно запустить в отдельной ячейки сразу из jupyter notebook используя восклицательный знак (!).

1. Создайте новую ячейку
2. Пропишите `!pip install requests`
3. Выполните и дождитесь окончания установки requests

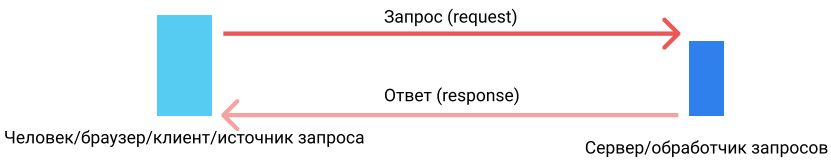
Подключение библиотеки

Подключение библиотек в Python осуществляется при помощи команды import <имя библиотеки/пакета>

Выполните `import requests`

Коротко об HTTP запросах

HTTP (протокол передачи гипертекста) запрос работает примерно так (упрощенная схема)



Существует 4 типа HTTP запросов: GET, POST, UPDATE, DELETE. На текущем этапе нам понадобится знать только GET.

GET запросы, как следует из названия, служат для простого запроса информации (аля "покажи главную страницу сайта по ссылке XXX", "покажи RSS по ссылке YYY").

Выполнение GET запроса при помощи requests

С помощью GET запроса мы можем получить RSS по ссылке (так же называется URL, универсальный указатель ресурса).

В примерах будем использовать https://news.yandex.ru/world.rss

Для того, чтобы выполнить HTTP запрос, необходимо вызвать одноименную (get, post, put, delete) функцию из requests. Для каждого из запросов параметры отличаются. Для функции get требуется передать только ссылку. В нашем случае это будет requests.get('https://news.yandex.ru/world.rss'). Результат запроса (ответ от сервера, response) сохраним в переменную r (так как response)

Выполните `r = requests.get('https://news.yandex.ru/world.rss')`

Теперь можем просмотреть ответ вызвав свойство text из r (буквально получить текст ответа) `r.text`. Сохраним этот xml в переменную xml `xml = r.text`

Увидим, что в переменной у нас хранится такой же RSS, какой и отображается в браузере

Теперь мы имеем готовый XML для парсинга, на этом работа с requests прекращается

Библиотека Beautifulsoup4

BeautifulSoup - библиотека, которая преобразовывает XML/HTML (в нашем случае XML) в объект Python. Объект Python на данном этапе можно воспринимать как словарь, где доступ к переменным (в терминах объекта Python они называются свойствами) с обращением по точке (dot notation, `object.property`).

Установка

Установка выполняется аналогично requests, имя пакета - `beautifulsoup4`

Подключение библиотеки

В питоне библиотека уже меняет свое название на bs4 (почему? потому что.), поэтому импортировать возможно только bs4. Кроме того, вся работа происходит с классом BeautifulSoup, который хранится в bs4. Конечно, мы могли бы всегда писать bs4.BeautifulSoup, но так как это единственная вещь которой мы будем пользоваться + она достаточно говорящая о библиотеке (в отличии от get в requests), то мы можем импортировать лишь BeautifulSoup при помощи команды `from bs4 import BeautifulSoup`

Типы парсеров

Перед тем, как натравить BeautifulSoup на наш XML мы должны определиться с типом парсера, который должен использовать BeautifulSoup. Типы парсеров указаны в документации, но там будет достаточно лишь одного - `html.parser`. Этот вид парсера хорошо парсит как XML, так и HTML.

Парсинг

Запустим BeautifulSoup на XML, который мы ранее сохранили в переменной `xml` и укажем типа парсера `html.parser`.

```
parsed = BeautifulSoup(xml, 'html.parser')
```

Теперь наш xml превратился в объект Python и мы можем "гулять" по XML с помощью путей, таких как мы видели выше (rss → channel → item**[0]** → title)

Давайте теперь получим заголовок самой первой новости уже на реальном RSS

Для этого выполним `parser.rss.channel.findChildren('item')[0].title`.

Обратите внимание на функцию findChildren. Она нужна для того, чтобы вернулся список itemов, при чем только тех, которые содержатся непосредственно в parser.rss.channel.

После выполнения команды мы получим блок title `<title>США, Великобритания и Австралия создали новый оборонный альянс AUKUS</title>`. Финальное его значение можно получить вызвав свойство text, которое будет содержать "США, Великобритания и Австралия создали новый оборонный альянс AUKUS"

Итого финальная команда выглядит так: `parser.rss.channel.findChildren('item')[0].title.text`

На этом быстрое ознакомление с bs4 - фсё. На самом деле она содержит множество крутых функций для более сложного парсинга, но все их можно написать руками зная лишь вещи описанные здесь. Изучайте новые функции по мере того, как вам надоест писать множество циклов и длинные пути.

RSS для Google Новостей

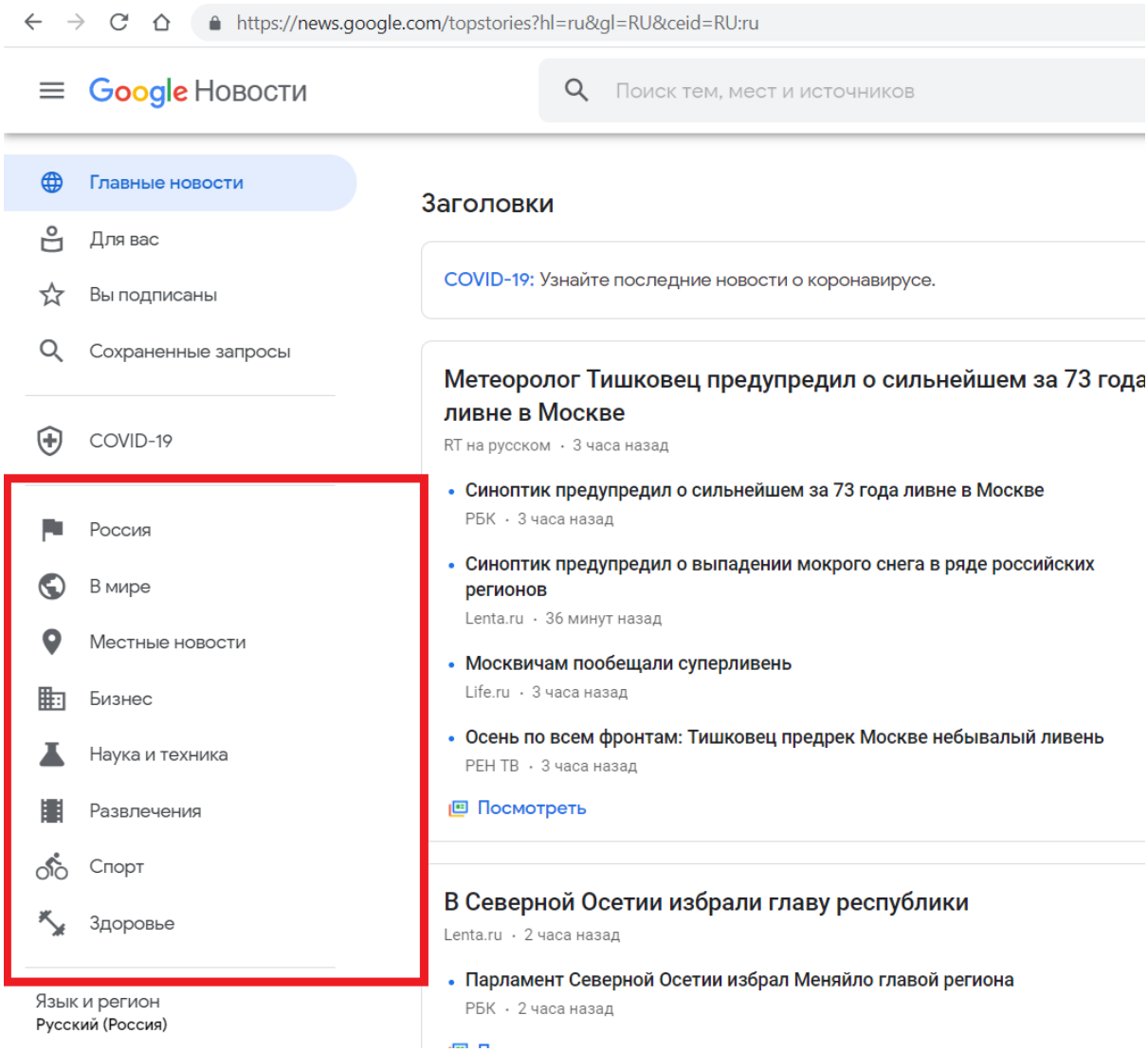
Топ новостей

Топ новостей в Google Новостях можно получить по ссылке https://news.google.com/rss

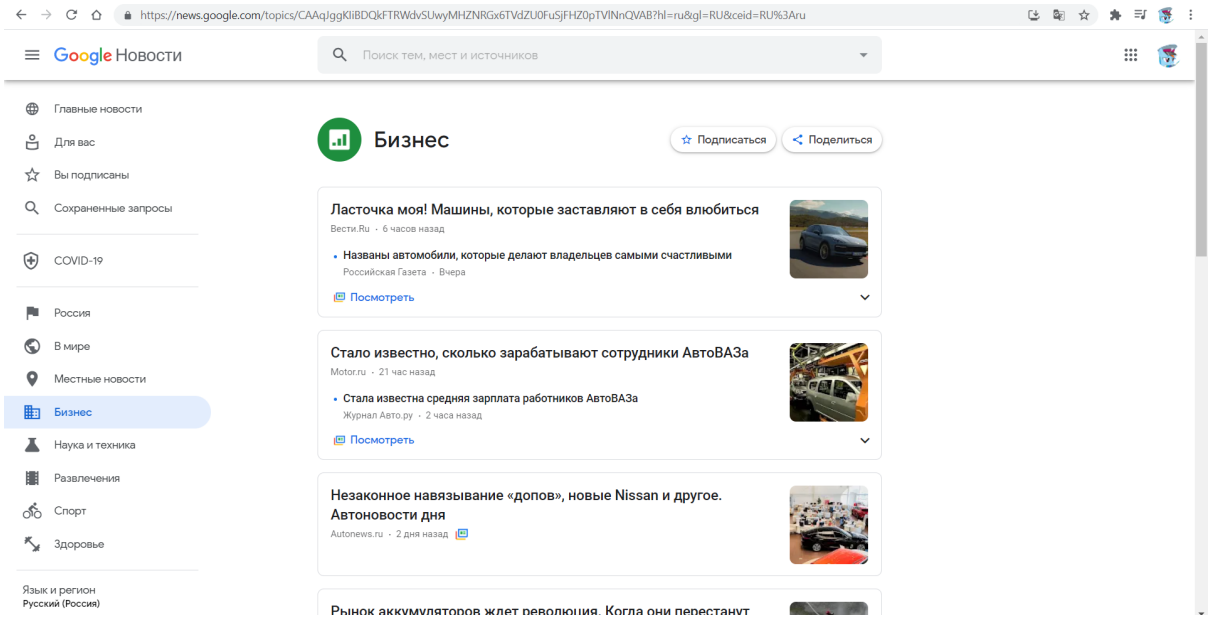
Новости по теме

Для получения новостей в Google Новостях необходимо сначала сформировать нужный URL (нужную ссылку)

1. Откройте https://news.google.com/
2. Нажмите на нужную тему



3. Посмотрим внимательно на страницу (я выбрал бизнес)



4. Здесь нам потребуется идентификатор в виде множества страшных букв в URL (слэш "/" и знак вопроса "?" в идентификатор не включаются)

В моем случае это будет CAAqJggKIiBDQkFTRWdvSUwyMHZNRGx6TVdZU0FuSjFHZ0pTVlNnQVAB

5. Ссылка на RSS Google Новостей имеет вид
   `https://news.google.com/rss/topics/<TOPIC_ID>`
6. Подставляем идентификатор в шаблон ссылки, у меня получится так
   https://news.google.com/rss/topics/CAAqJggKIiBDQkFTRWdvSUwyMHZNRGx6TVdZU0FuSjFHZ0pTVlNnQVAB

Теперь мы можем использовать полученную ссылку для парсинга!

Задания

- Получите заголовок второй новости из Google Новостей по теме Наука и Техника
- Получите все заголовки новостей со страницы Вышки Для Своих https://www.hse.ru/our/