Структурирование данных
Сбор информации из интернета

Лекция 2

Электронная почта

From: Subhojit Banerjee <subhojit20@gmail.com>
To: peter@romov.ru
Date: Mon, 7 Aug 2017 10:15:16 +0200
Subject: ydf-recsys2015-challenge

Hello Peter,
A quick question about your recsys approach: if you didn't have access to matrix net (yandex package), could you suggest packages which can be used instead(to solve the large number of categorical features)?

Kind regards,
Subhojit

Электронная почта

Delivered-To: romovpa@gmail.com
Received: by 10.37.34.68 with HTTP; Mon, 7 Aug 2017 01:15:16 -0700 (PDT)
Message-ID:
<CAOi4bd0A8T25bT6SZZ0RUS2GHd6bfBXH=GLTb+Pf9vsoZbfLkg@mail.gmail.com>

From: Subhojit Banerjee <subhojit20@gmail.com>
To: peter@romov.ru
Date: Mon, 7 Aug 2017 10:15:16 +0200
Subject: ydf-recsys2015-challenge

Hello Peter,
A quick question about your recsys approach: if you didn't have access to matrix net (yandex package), could you suggest packages which can be used instead(to solve the large number of categorical features)?

Kind regards,
Subhojit

Multipurpose Internet Mail Extensions

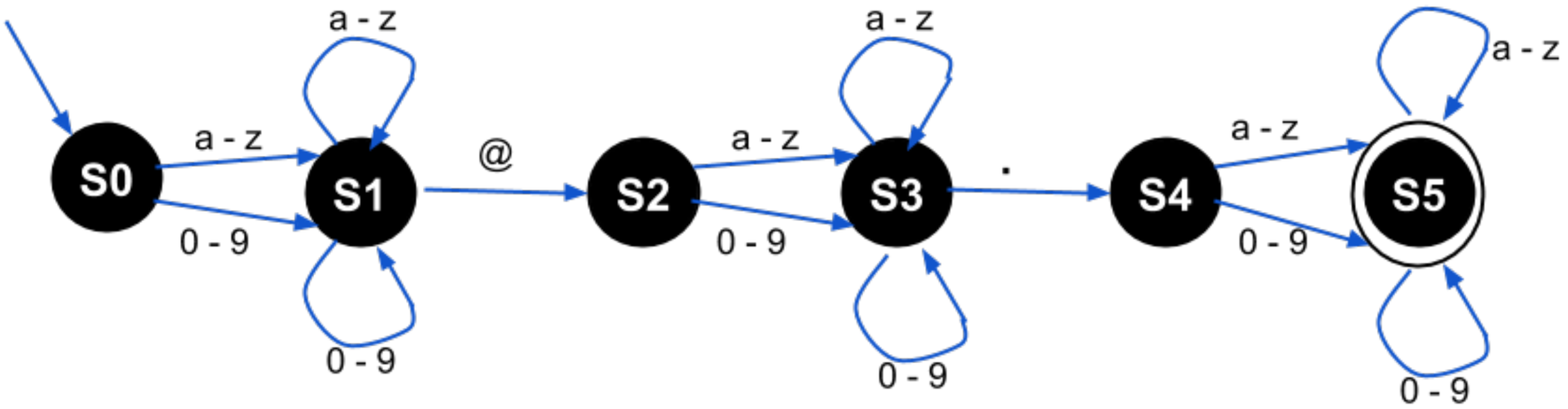
```
MIME-Version: 1.0

Delivered-To: romovpa@gmail.com
Received: by 10.37.34.68 with HTTP; Mon, 7 Aug 2017 01:15:16 -0700 (PDT)
Message-ID: <CAOi4bd0A8T25bT6SZZ0RUS2GHd6bfBXH=GLTb+Pf9vsoZbfLkg@mail.gmail.com>

From: Subhojit Banerjee <subhojit20@gmail.com>
To: peter@romov.ru
Date: Mon, 7 Aug 2017 10:15:16 +0200
Subject: ydf-recsys2015-challenge
Content-Type: multipart/alternative; boundary="94eb2c07dc38711836055625722f"

--94eb2c07dc38711836055625722f
Content-Type: text/html; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

<div dir=3D"ltr">Hello Peter,<div>A quick question about your recsys approa=
ch: if you didn&#39;t have access to matrix net (yandex package), could you=
 suggest packages which can be used instead(to solve the large number of ca=
tegorical features).</div><div><br></div><div><br></div><div>Kind regards,<=
/div><div>Subhojit</div></div>

--94eb2c07dc38711836055625722f
Content-Type: text/plain; charset="UTF-8"

Hello Peter,
A quick question about your recsys approach: if you didn't have access to
matrix net (yandex package), could you suggest packages which can be used
```

Логи веб-сервера

```
123.123.123.123 - - [26/Apr/2000:00:23:48 -0400] "GET /pics/wpaper.gif HTTP/1.0" 200 6248
"http://www.jafsoft.com/asctortf/" "Mozilla/4.05 (Macintosh; I; PPC)”

123.123.123.123 - - [26/Apr/2000:00:23:47 -0400] "GET /asctortf/ HTTP/1.0" 200 8130
"http://search.netscape.com/Computers/Data_Formats/Document/Text/RTF" "Mozilla/4.05
(Macintosh; I; PPC)”

123.123.123.123 - - [26/Apr/2000:00:23:48 -0400] "GET /pics/5star2000.gif HTTP/1.0" 200
4005 "http://www.jafsoft.com/asctortf/" "Mozilla/4.05 (Macintosh; I; PPC)”

123.123.123.123 - - [26/Apr/2000:00:23:50 -0400] "GET /pics/5star.gif HTTP/1.0" 200 1031
"http://www.jafsoft.com/asctortf/" "Mozilla/4.05 (Macintosh; I; PPC)”

123.123.123.123 - - [26/Apr/2000:00:23:51 -0400] "GET /pics/a2hlogo.jpg HTTP/1.0" 200
4282 "http://www.jafsoft.com/asctortf/" "Mozilla/4.05 (Macintosh; I; PPC)”

123.123.123.123 - - [26/Apr/2000:00:23:51 -0400] "GET /cgi-bin/newcount?
jafsof3&width=4&font=digital&noshow HTTP/1.0" 200 36 "http://www.jafsoft.com/asctortf/"
"Mozilla/4.05 (Macintosh; I; PPC)"
```

Регулярные выражения (regexp)

```
[a-z0-9]+@[a-z0-9]+\.[a-z0-9]+
```




```
(?P<username>[a-z0-9]+)@(?P<server>[a-z0-9]+\.[a-z0-9]+)
```

alice@acme.com -> {“username”: “alice”, “server”: “acme.com”}
http://google.com -> NOT MATCHED (недопустимый переход)
aliceacme.com -> NOT MATCHED (завершение не в терминальном состоянии)

Регулярные выражения (regexp)

Множество символов

```
[abc]
[^z]
[\d] = [0-9]
[\D] = [^0-9]
a = [a]
\s = [\s] = [ \t\n\r…]
\w = [a-zA-Z0-9_]
. = [^\n]
```

Повторы

```
[ab]?
[ab]{1,3} = [ab][ab]?[ab]?
[ab]+ = [ab]{1,}
[ab]* = [ab]{0,}
```

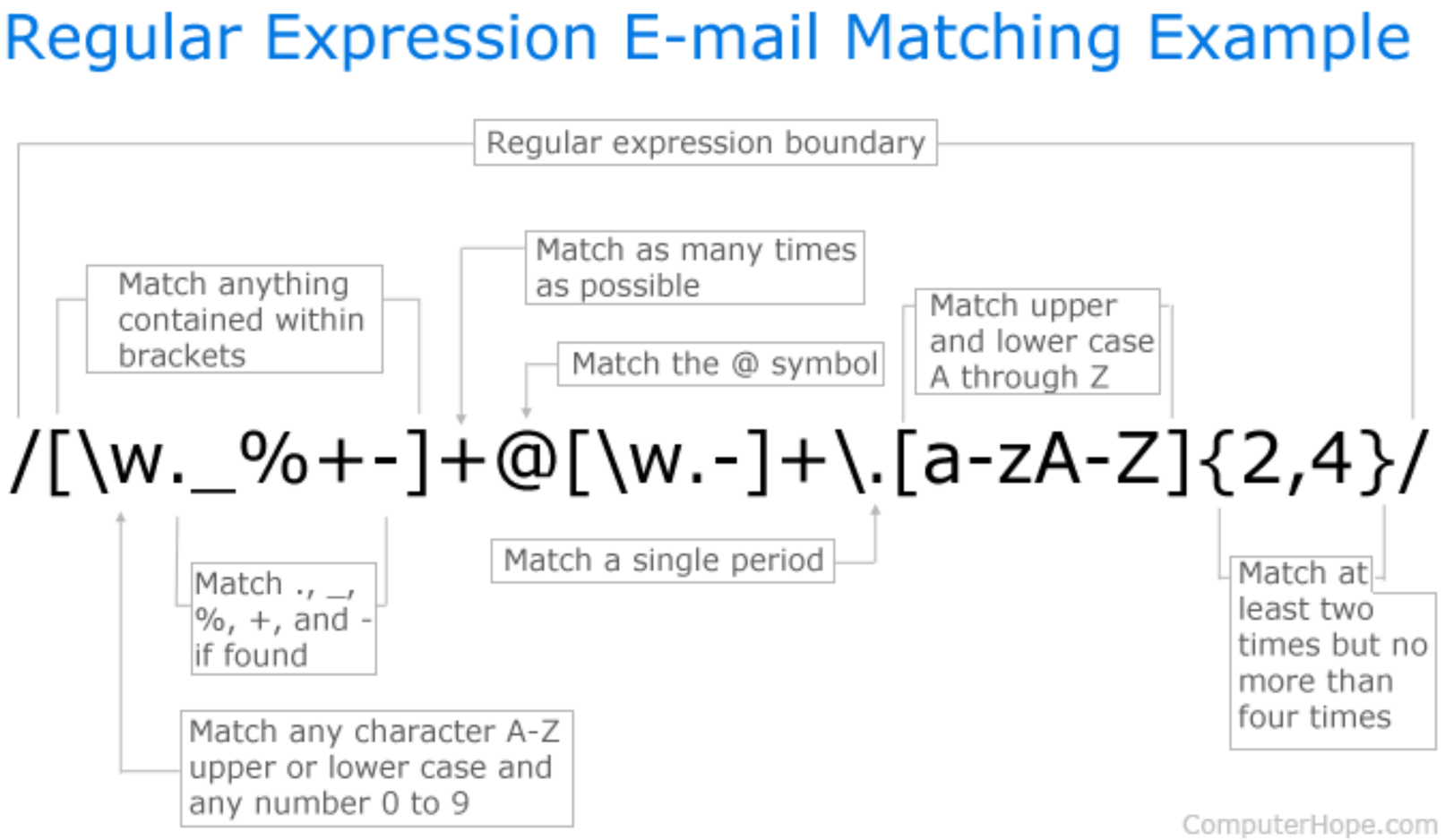
Условия

```
a|(b*a)
([ab]+c*)|([def]?g+)
```

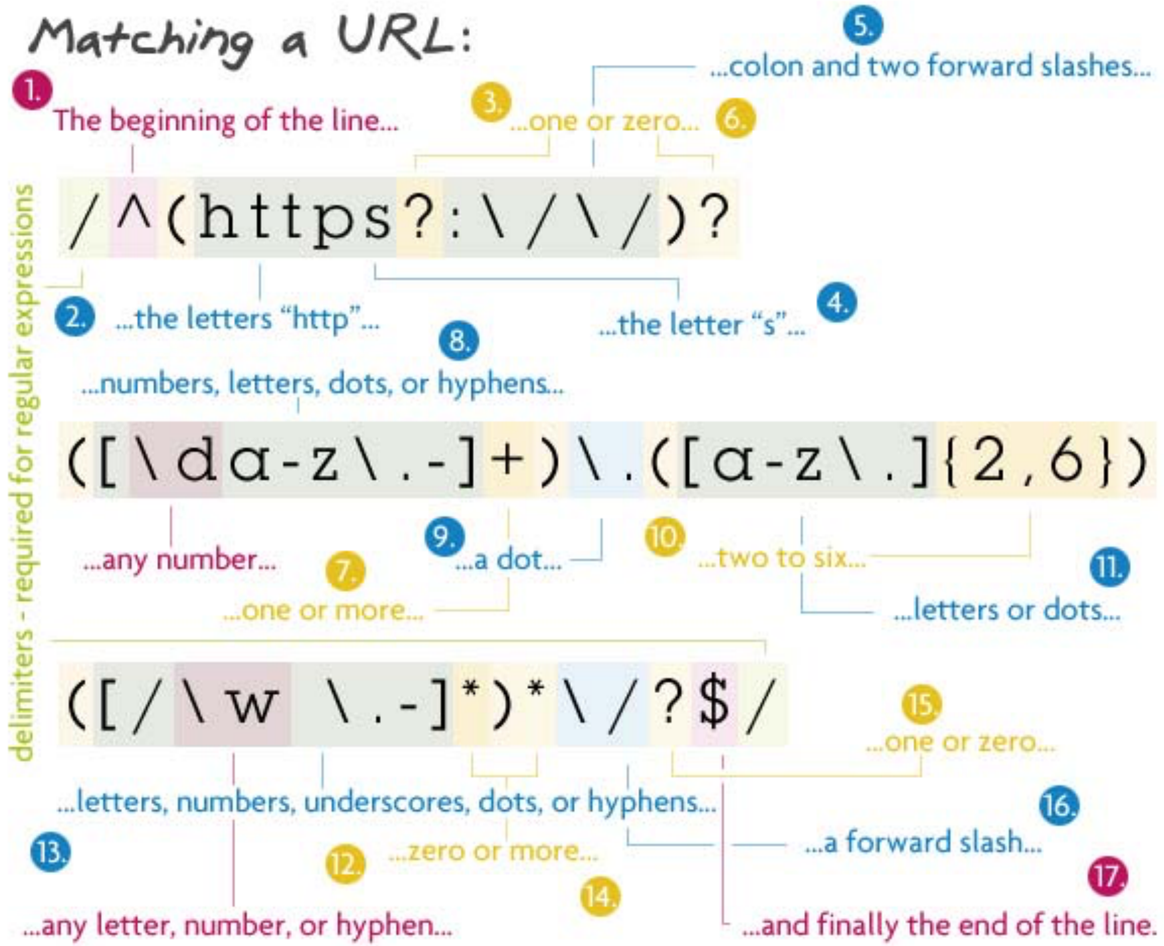
Начало и конец строки

```
^[ab]+$
^\d+.\d+.\d+.\d+
```

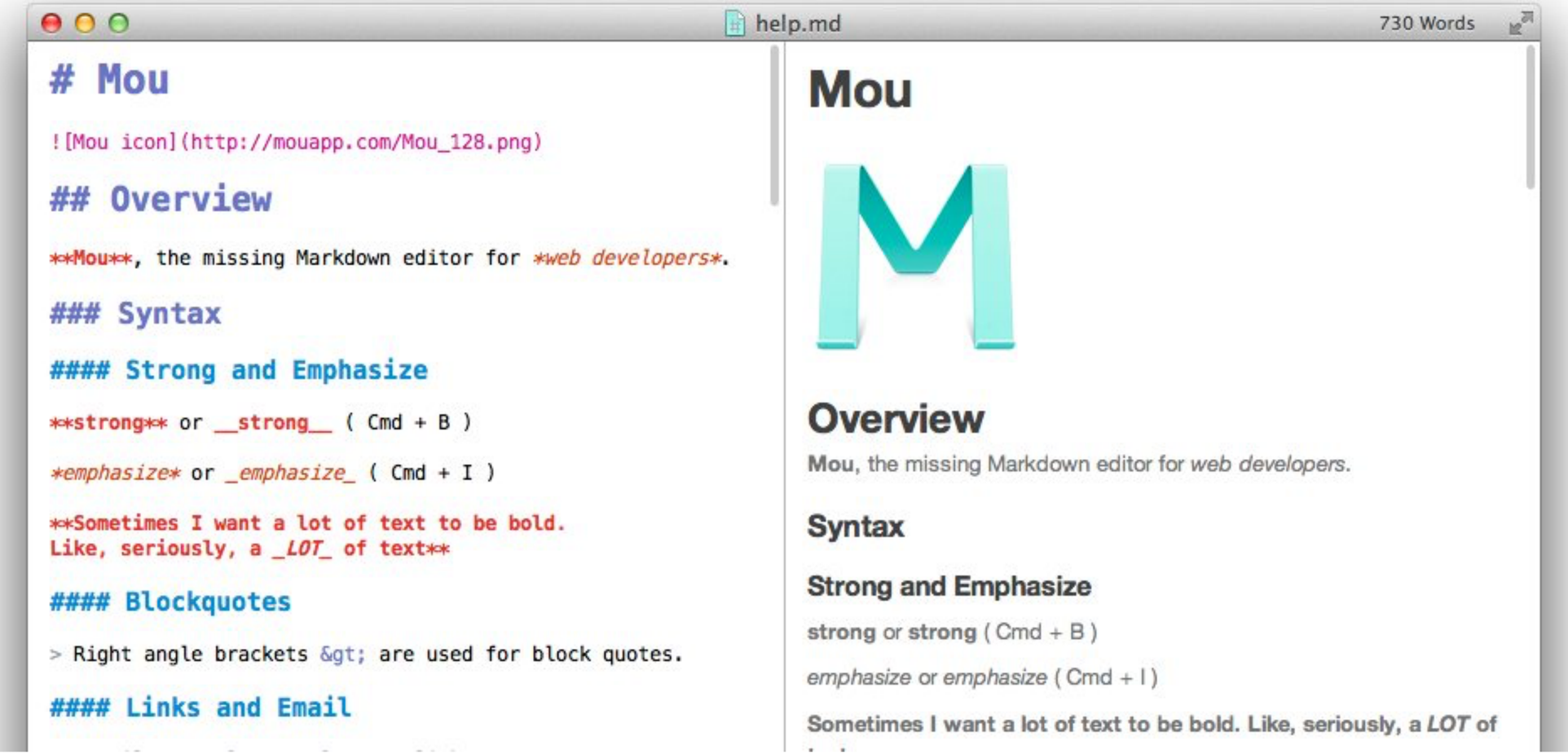
Регулярные выражения (regexp)



Регулярные выражения (regexp)

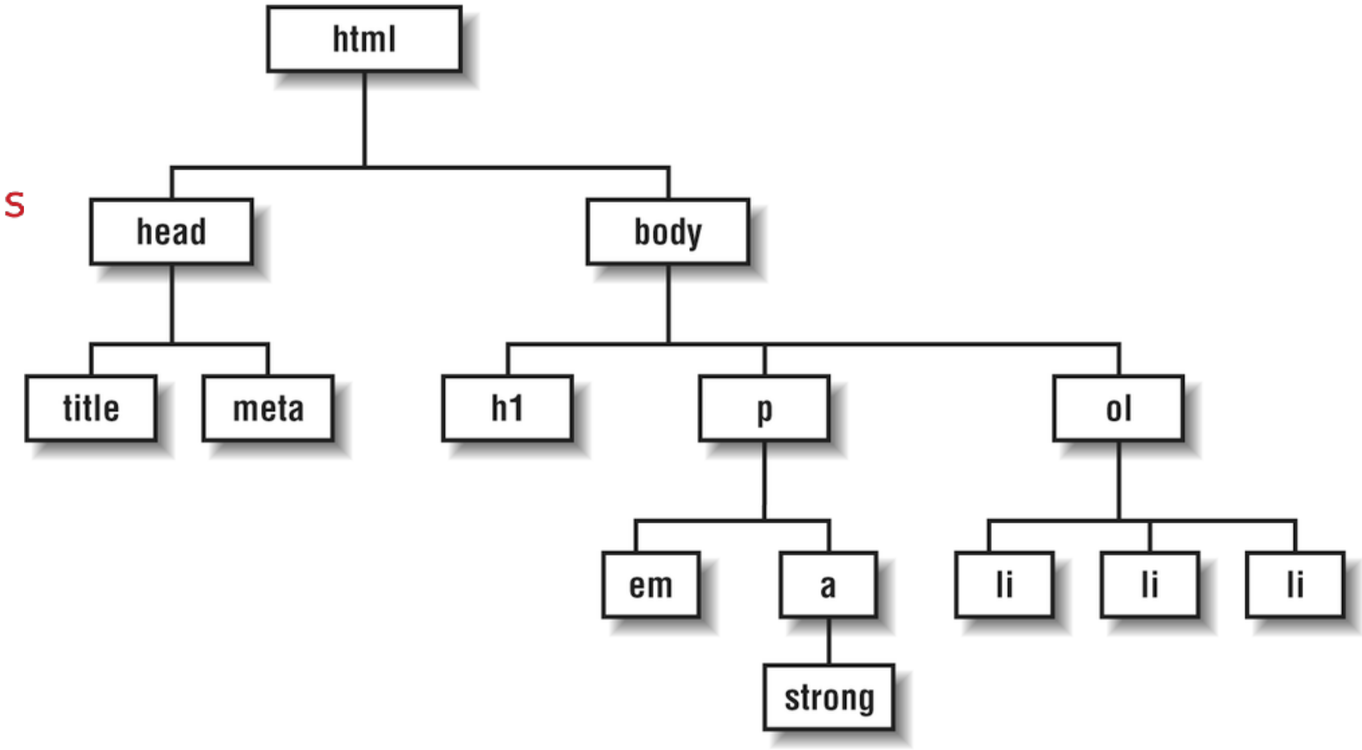


Markdown


help.md
730 Words
Mou
![Mou icon](http://mouapp.com/Mou_128.png)
Overview
Mou, the missing Markdown editor for *web developers*.
Syntax
Strong and Emphasize
strong or __strong__ (Cmd + B)
emphasize or _emphasize_ (Cmd + I)
**Sometimes I want a lot of text to be bold.
Like, seriously, a _LOT_ of text**
Blockquotes
> Right angle brackets > are used for block quotes.
Links and Email
Mou
Overview
Mou, the missing Markdown editor for web developers.
Syntax
Strong and Emphasize
strong or strong (Cmd + B)
emphasize or emphasize (Cmd + I)
Sometimes I want a lot of text to be bold. Like, seriously, a LOT of

HyperText Markup Language (HTML)

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Example<title>
5          <link rel="stylesheet" href="s
6      </head>
7      <body>
8          <h1>
9              <a href="/">Header</a>
10         </h1>
11         <nav>
12             <a href="one/">One</a>
13             <a href="two/">Two</a>
14             <a href="three/">Three</a>
15         </nav>
```

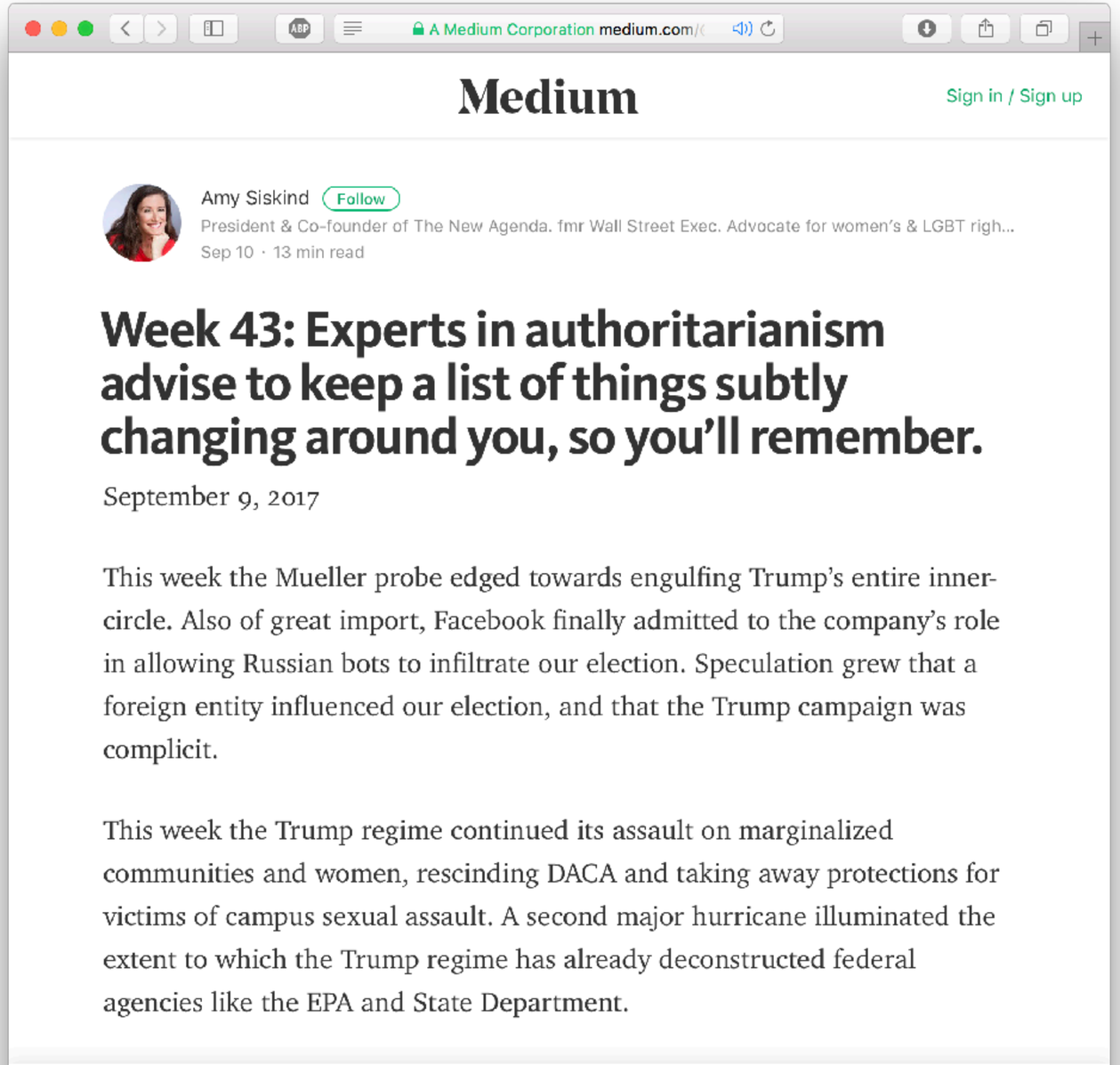


XML, XPath

```xml
<?xml version="1.0" encoding="UTF-8"?>

<bookstore>

<book category="cooking">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>50.00</price>
</book>

<book category="children">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>

</bookstore>
```

```
/bookstore/book[1]

/bookstore/book[last()]

/bookstore/book[last()-1]

/bookstore/book[position()<3]

//title[@lang]

//title[@lang='en']

/bookstore/book[price>35.00]

/bookstore/book[price>35.00]/title
```

Веб-страницы



Веб-страницы

```
▼ <div class="section-inner sectionLayout--insetColumn">
  ▶ <h1 name="9a35" id="9a35" class="graf graf--h3 graf--leading graf--title">…</h1>
    <p name="70e0" id="70e0" class="graf graf--p graf-after--h3">September 9, 2017</p>
  ▼ <p name="a0ea" id="a0ea" class="graf graf--p graf-after--p">
      "This week the Mueller probe edged towards engulfing Trump's entire inner-circle. Also of great import, Facebook finall
      company's role in allowing Russian bots to infiltrate our election. Speculation grew that a foreign entity influenced o
      the Trump campaign was complicit."
    </p>
  ▼ <p name="af08" id="af08" class="graf graf--p graf-after--p">
      "This week the Trump regime continued its assault on marginalized communities and women, rescinding DACA and taking awa
      victims of campus sexual assault. A second major hurricane illuminated the extent to which the Trump regime has already
      agencies like the EPA and State Department."
    </p>
  ▼ <ol class="postList">
    ▼ <li name="c9d6" id="c9d6" class="graf graf--li graf-after--p">
        ::before
        "Late Saturday over Labor Day Weekend, the "
        <a href="http://www.politico.com/story/2017/09/02/obama-trump-tower-wiretap-no-evidence-242284" data-href="http://www
        2017/09/02/obama-trump-tower-wiretap-no-evidence-242284" class="markup--anchor markup--li-anchor" rel="noopener nofol
        "_blank">DOJ unceremoniously announced there is no evidence Obama wiretapped Trump Tower</a>
        ". Trump did not apologize to Obama for this frequently repeated, false claim."
      </li>
    ▶ <li name="d253" id="d253" class="graf graf--li graf-after--li">…</li>
    ▶ <li name="9194" id="9194" class="graf graf--li graf-after--li">…</li>
    ▶ <li name="f113" id="f113" class="graf graf--li graf-after--li">…</li>
    ▶ <li name="ec3b" id="ec3b" class="graf graf--li graf-after--li">…</li>
```

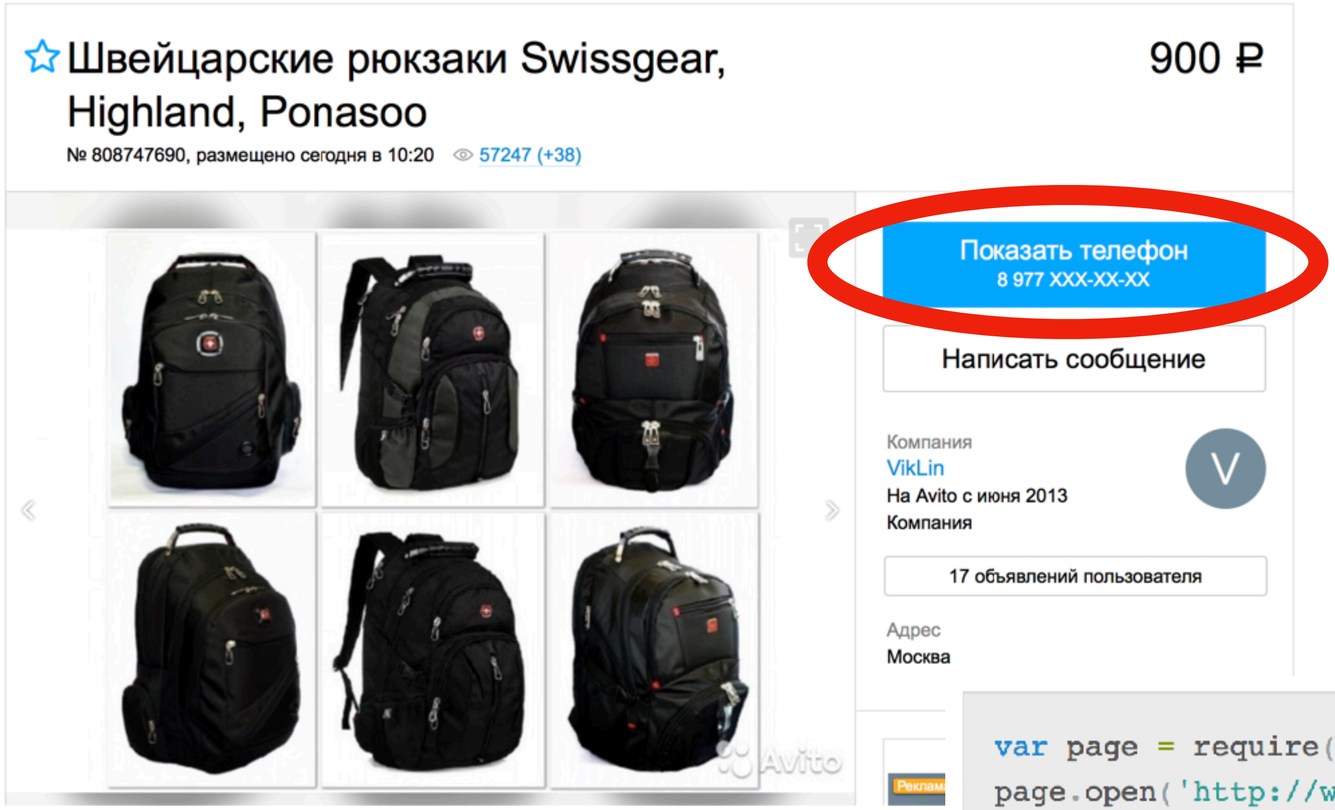
Язык CSS-селекторов

```
h1
p
div.section-inner
li.graf.highlighted

#title
.graf

#element ::text()
a ::attr(href)

div.section-inner > h1.graf ::text()

div.section-inner li

li ::attr(data-price)
li ::attr(data-title)
```

У каждого узла есть тег (h1, p, …)

Узлы могут иметь поля
`class` (один или несколько)
`name` (уникальное имя узла)

JavaScript, Headless-браузеры




```
var page = require('webpage').create();
page.open('http://www.sample.com', function() {
  page.includeJs("http://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jqu
    page.evaluate(function() {
      $("button").click();
    });
    phantom.exit()
  });
});
```

Микроразметка

```
<li
    class="product__item product__item--1358482 product__item--catalog "
    data-category-id="mediamarktcataloginternet_863"
    data-category-title="Планшеты"
    data-article="1358482"
    data-ga-article="1358482"
    data-dl-params="Phab 2 Pro 4G 64Gb"
    data-price="31499"
>
```

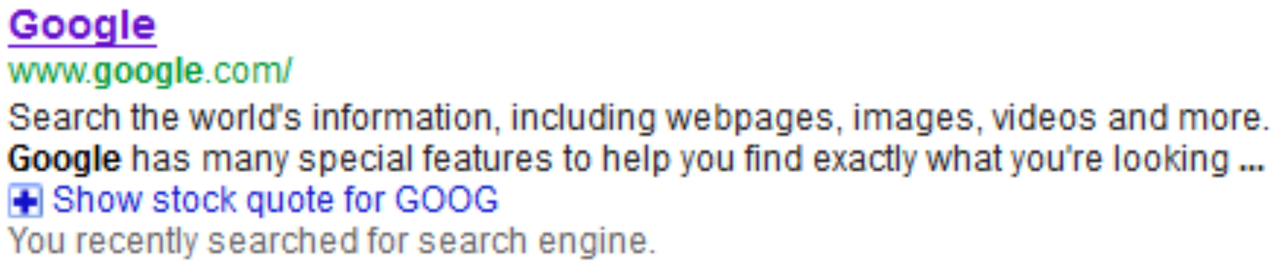
Микроразметка

Документ с разметкой Schema.org

```
<div itemscope itemtype="http://schema.org/Organization">
  <span itemprop="name">Яндекс</span>
  Контакты:
  <div itemprop="address" itemscope itemtype="http://schema.org/PostalAddress">
    Адрес:
    <span itemprop="streetAddress">Льва Толстого, 16</span>
    <span itemprop="postalCode"> 119021</span>
    <span itemprop="addressLocality">Москва</span>,
  </div>
  Телефон:<span itemprop="telephone">+7 495 739–70–00</span>,
  Факс:<span itemprop="faxNumber">+7 495 739–70–70</span>,
  Электронная почта: <span itemprop="email">pr@yandex-team.ru</span>
</div>
```

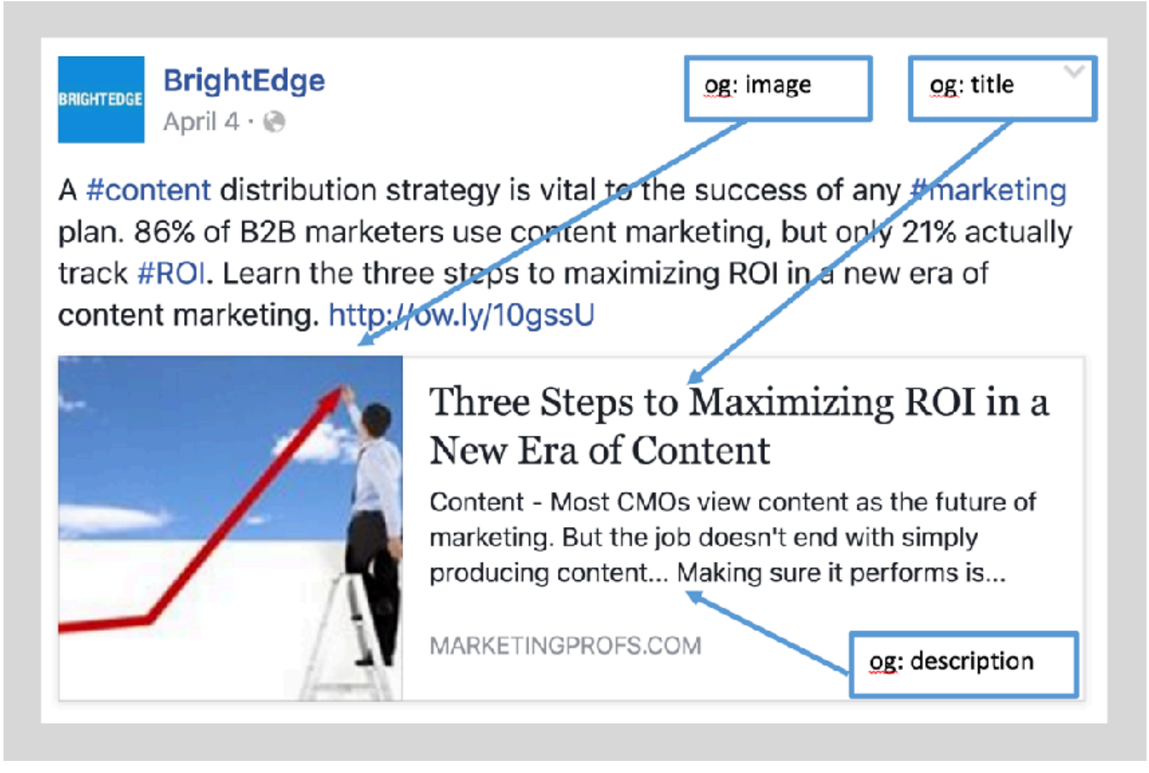
Meta-теги страницы

```
<head>
    ...
    <meta name="title" content="Google.com">
    <meta name=”description” content=”Search the world’s information, including web pages, images, videos and more. Google has many special features to help you find exactly what you’re looking for.“>
    ...
</head>
```

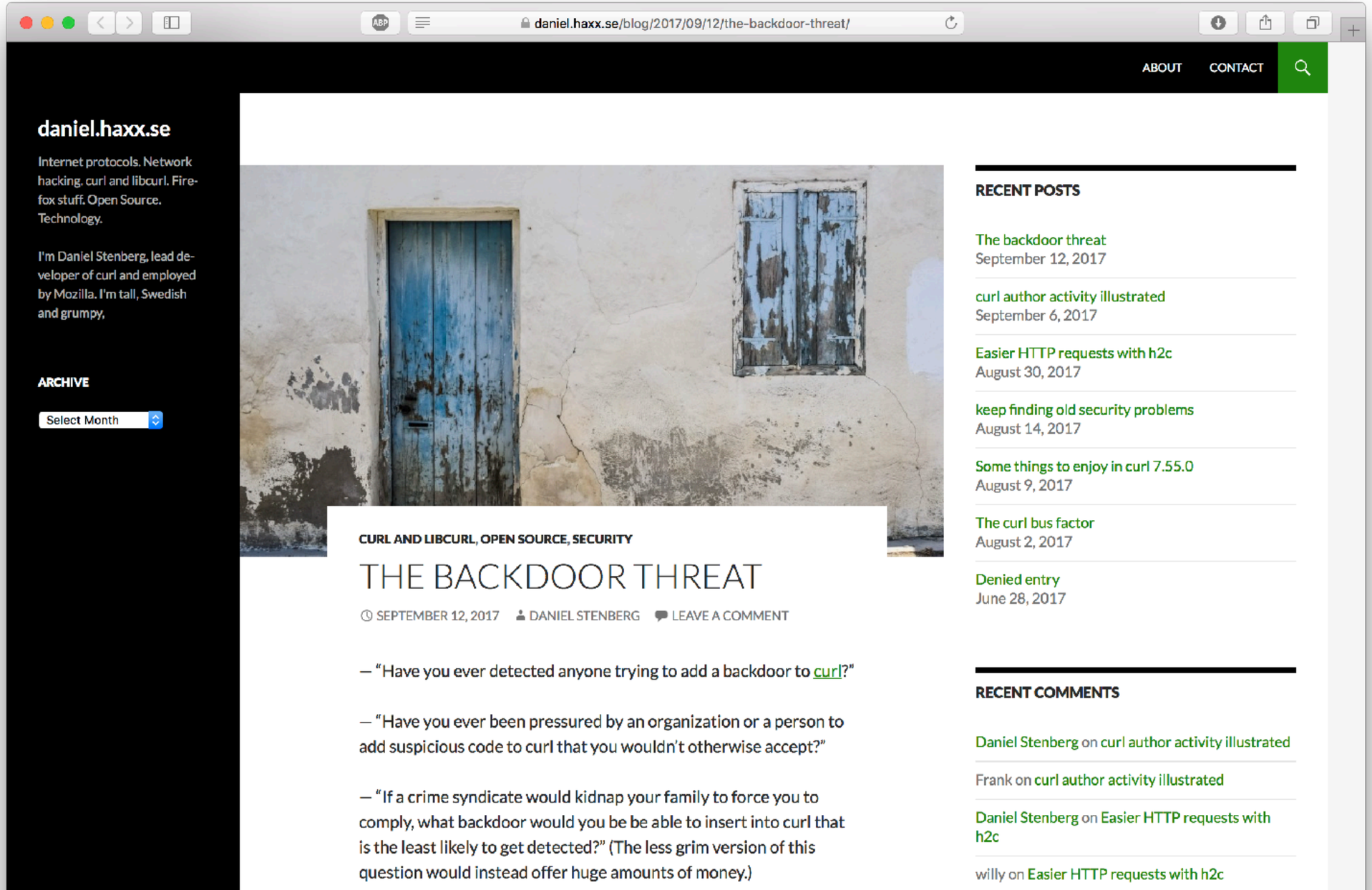


Разметка OpenGraph

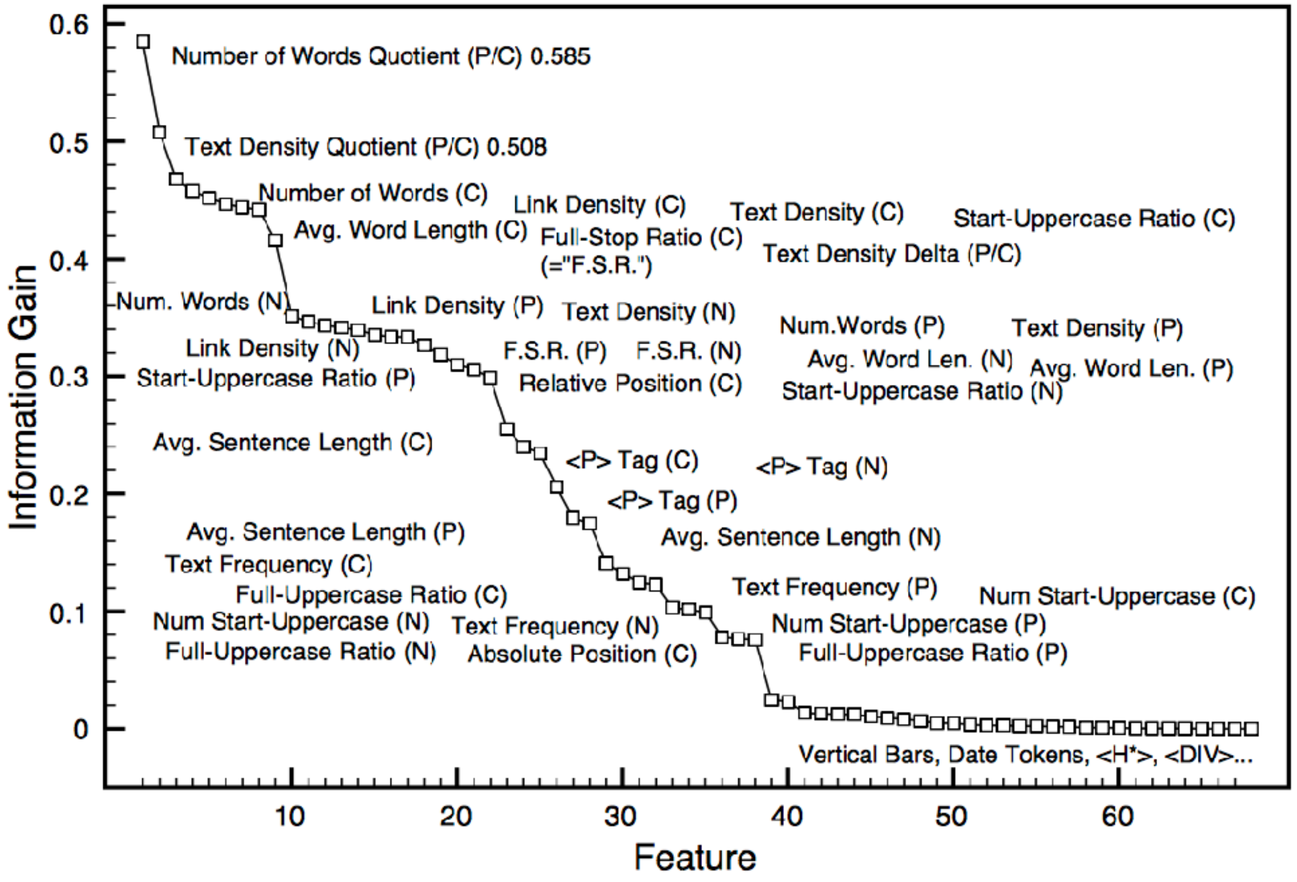
```
<meta property="og:title" content="Example title of article">
<meta property="og:site_name" content="example.com website">
<meta property="og:type" content="article">
<meta property="og:url" content="http://example.com/example-title-of-article">
<meta property="og:image" content="http://example.com/article_thumbnail.jpg">
<meta property="og:image" content="http://example.com/website_logo.png">
<meta property="og:description" content="This example article is an example of OpenGraph protocol.">
```



Извлечение текста из произвольной страницы



Извлечение текста из произвольной страницы



Algorithm 1 Densitometric Classifier

```
curr_linkDensity <= 0.333333
| prev_linkDensity <= 0.555556
| | curr_textDensity <= 9
| | | next_textDensity <= 10
| | | | prev_textDensity <= 4:  BOILERPLATE
| | | | prev_textDensity > 4:  CONTENT
| | | next_textDensity > 10:  CONTENT
| | curr_textDensity > 9
| | | next_textDensity = 0:  BOILERPLATE
| | | next_textDensity > 0:  CONTENT
| prev_linkDensity > 0.555556
| | next_textDensity <= 11:  BOILERPLATE
| | next_textDensity > 11:  CONTENT
curr_linkDensity > 0.333333:  BOILERPLATE
```

Christian Kohlschütter, Peter Fankhauser, Wolfgang Nejdl **“Boilerplate Detection using Shallow Text Features”**

https://github.com/kohlschutter/boilerpipe

Структурированные выгрузки веб-сайта

```
<link
    rel=“alternate"
    type=“application/rss+xml"
    title="daniel.haxx.se | Feed”
    href="https://daniel.haxx.se/blog/feed/" />
<link
    rel=“alternate"
    type=“application/rss+xml"
    title="daniel.haxx.se | Comments Feed”
    href="https://daniel.haxx.se/blog/comments/feed/" />
<link
    rel=“alternate"
    type=“application/rss+xml”
    title="daniel.haxx.se | The backdoor threat Comments Feed”
    href="https://daniel.haxx.se/blog/2017/09/12/the-backdoor-threat/feed/" />
```

RSS: Rich Site Summary

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
 <title>RSS Title</title>
 <description>This is an example of an RSS feed</description>
 <link>http://www.example.com/main.html</link>
 <lastBuildDate>Mon, 06 Sep 2010 00:01:00 +0000 </lastBuildDate>
 <pubDate>Sun, 06 Sep 2009 16:20:00 +0000</pubDate>
 <ttl>1800</ttl>

 <item>
  <title>Example entry</title>
  <description>Here is some text containing an interesting description.</description>
  <link>http://www.example.com/blog/post/1</link>
  <guid isPermaLink="false">7bd204c6-1655-4c27-aeee-53f933c5395f</guid>
  <pubDate>Sun, 06 Sep 2009 16:20:00 +0000</pubDate>
 </item>

</channel>
</rss>
```

Если нет RSS, то может быть **ATOM** — аналогичный популярный формат

Парсинг больших XML

1. **DOM** (Document Object Model)
- представление в виде дерева
- возможность делать запросы

2. **SAX** (Simple API for XML)
- вызов обработчиков событий “начало тега”, “конец тега”, “текст”

В случае, если документ большой:
- поиск нужных блоков SAX
- обработка содержимого блоков DOM

```
<?xml version="1.0" encoding="UTF-8"?>
<yml_catalog date="2017-02-05 17:22">
    <shop>
    <name>BestSeller</name>
    <company>Tne Best inc.</company>
    <url>http://best.seller.ru</url>
    <offers>

      <offer id="12346" available="true" bid="80" cbid="90" fee="325">
        <url>http://best.seller.ru/product_page.asp?pid=12348</url>
        <price>1490</price>
        <name>Вафельница First FA-5300</name>
      </offer>

      <offer id="9012" type="vendor.model" available="true" bid="80" cbid="90" fe
        <url>http://best.seller.ru/product_page.asp?pid=12345</url>
        <price>8990</price>
        <typePrefix>Мороженица</typePrefix>
        <vendor>Brand</vendor>
        <model>3811</model>
      </offer>

    </offers>
  </shop>
</yml_catalog>
```

Протокол HTTP

Request

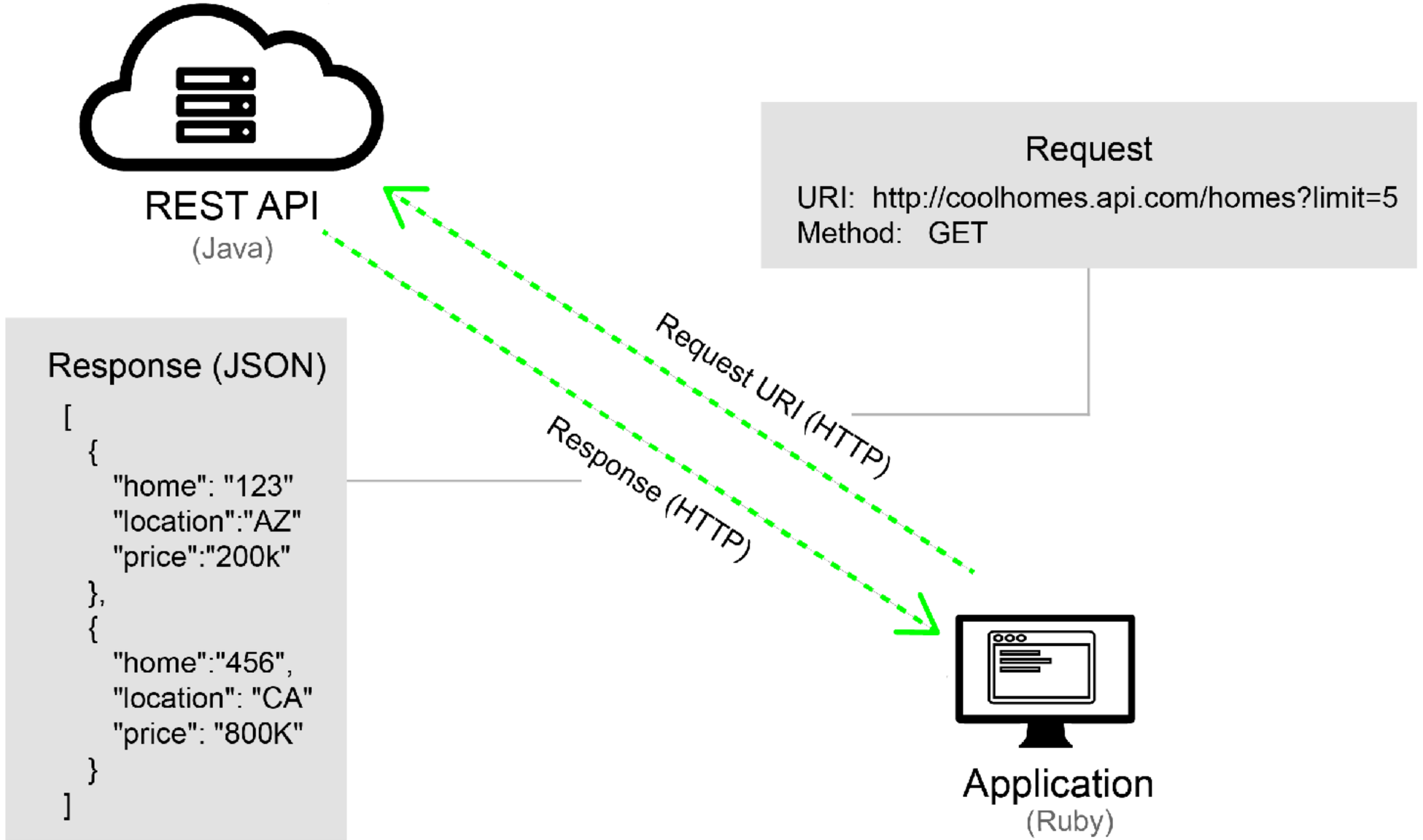
```
GET /index.html HTTP/1.1

Host: www.example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, …
Referer: http://google.com/q=example
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: fr-CH, fr;q=0.9, en;q=0.8, de;q=0.7, *;q=0.5
Cookie: name=value; name2=value2; name3=value3
```

Протокол HTTP

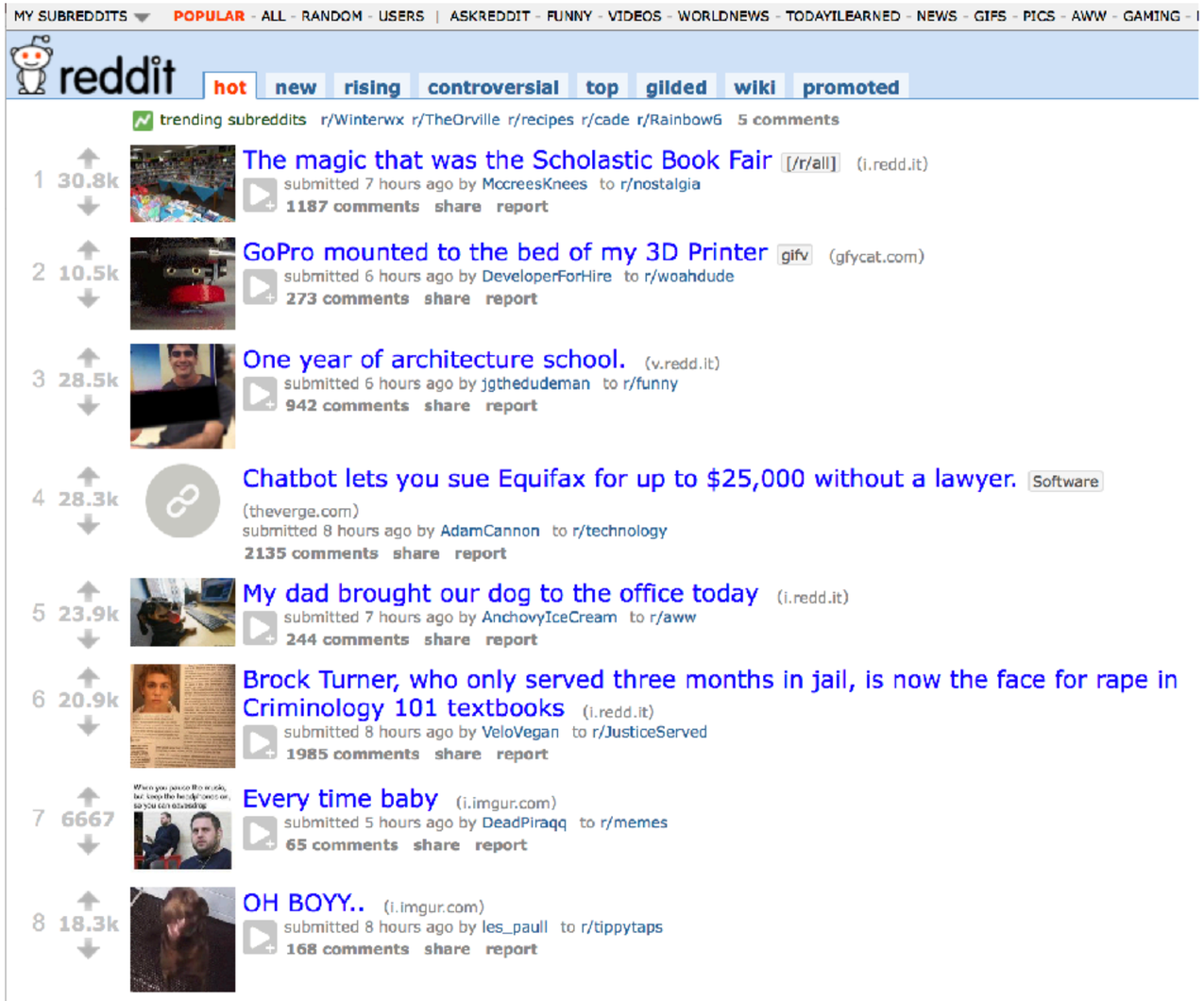
Response

```
HTTP/1.1 200 OK

Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Set-Cookie: tasty_cookie=strawberry

<html>
<head>
  <title>An Example Page</title>
</head>
<body>
  Hello World, this is a very simple HTML document.
</body>
</html>
```

HTTP API


REST API
(Java)
Request
URI: http://coolhomes.api.com/homes?limit=5
Method: GET
Request URI (HTTP)
Response (HTTP)
Response (JSON)
[
{
"home": "123"
"location":"AZ"
"price":"200k"
},
{
"home":"456",
"location": "CA"
"price": "800K"
}
]
Application
(Ruby)

Reddit API



API methods

by section | by oauth scope

account

/api/v1/me	oauth
/api/v1/me/blocked	oauth
/api/v1/me/friends	oauth
/api/v1/me/karma	oauth
/api/v1/me/prefs	oauth
/api/v1/me/trophies	oauth
/prefs/blocked	oauth
/prefs/friends	oauth
/prefs/messaging	oauth
/prefs/trusted	oauth
/prefs/*where*	oauth

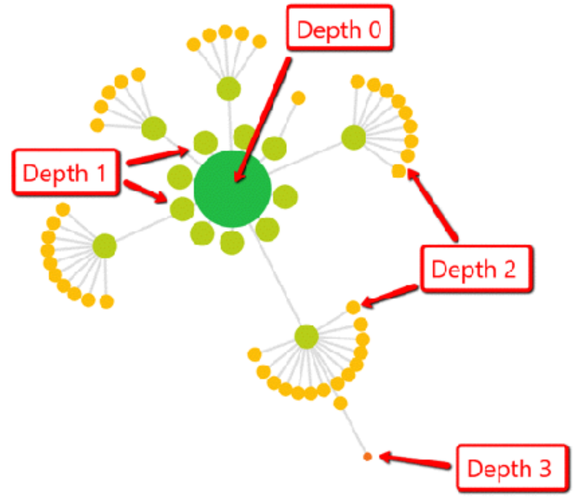
captcha

/api/needs_captcha	oauth

flair

/api/clearflairtemplates	oauth
/api/deleteflair	oauth
/api/deleteflairtemplate	oauth
/api/flair	oauth
/api/flairconfig	oauth
/api/flaircsv	oauth
/api/flairlist	oauth
/api/flairselector	oauth
/api/flairtemplate	oauth
/api/link_flair	oauth
/api/selectflair	oauth
/api/setflairenabled	oauth
/api/user_flair	oauth

Краулинг веб-сайтов



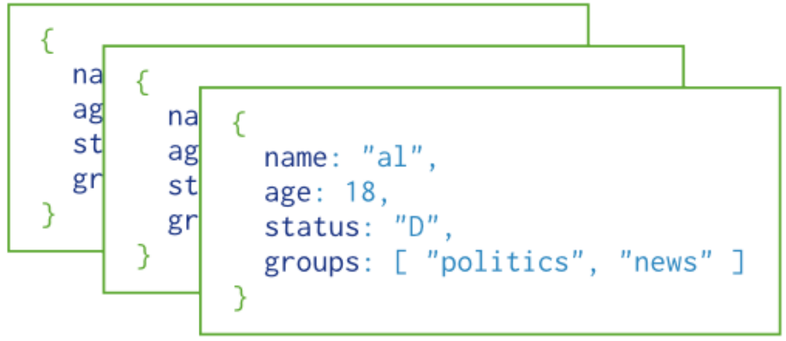
- Задается seed (стартовый набор адресов)
  например: список блогов, Alexa Top 1M
- Осуществляется DFS или BFS с ограниченной глубиной
  - возможно на разных машинах
- Частые проблемы, веб-сайт может:
  1. **не показывать содержимое неизвестным браузерам**
     - подделывать HTTP заголовки: User-Agent, Cookies
  2. **не выдерживать нагрузку**
     - ограничивать число запросов, или использовать режим Auto-Throttle
  3. **банить ip-адреса, показывать капчу**
     - использование списков прокси-серверов, TOR
  4. **использовать механизмы защиты от DDoS**
     - например, хитрый JavaScript код — его придется симулировать
  5. **требовать авторизацию и лимитировать число действий** (напр. Facebook)
     - создавать сеть фейковых аккаунтов, которые ведут себя как обычные люди

Рекомендуемые библиотеки / инструменты

- **Scrapy** (Python)
  - позволяет быстро писать и запускать пауков
  - удобный интерфейс для XPath/CSS-селекторов scrapy.Selector
- **Requests** (Python)
  - для HTTP запросов
- **PhantomJS**
  - headless-браузер, позволяет симулировать веб-страницу
- **LXML** (Python)
  - библиотека для парсинга HTML, XML; в том числе позволяет парсить большие XML файлы, HTML с ошибками
- **BeautifulSoup** (Python)
  - упрощенный интерфейс к содержимому HTML
- **Postman** — приложение для тестирования HTTP API
- **Chrome Developer Tools** — встроенный инструмент для ручного анализа дерева HTML, просмотра HTTP-запросов

Хранение частично-структурированных данных

- **В файлах / памяти**
- **MongoDB**
  - позволяет хранить JSON-объекты
  - максимально быстрое начало работы
  - очень быстрые выборки по запросам (нужно только построить индекс по нужным полям)
  - ограничение на размер документа 16Мб
- **PostgreSQL**
  - полноценный SQL, возможность делать агрегации
  - можно работать с JSON-объектами при помощи колонок специального типа
- **Neo4j**
  - графовая база данных, позволяет хранить объекты и связи различного типа (например: страницы, ссылки, именованные сущности)
  - специальный язык Cypher для запросов к графу



Collection

ID	Name	...	PhoneNumbers
1	John		[{"Number":"06472643","Type":"Work"}, {"Number":"01164322","Type":"Home"}]
2	Jane		[{"Number":"01726443","Type":"Work"}, {"Number":"06243344","Type":Mobile"}]
3	Jack		[{"Number":"01167343","Type":"Home"}]

Задание

Сделать себе корпус новостных статей для последующих экспериментов

1. Взять содержимое последних месяцев на HackerNews используя API **http://news.ycombinator.com**
2. Выгрузить текст и метаинформацию страниц по ссылкам
3. Сделать обзор получившегося корпуса