

Ashoka Summer School

Web Scraping

Outline

Web scraping

Web page

BeautifulSoup Library

Practical Works

Web Scraping - What is it ?

Data Scraping ?

- Automated process
- Explore and download raw data
- Grab content
- Convert data in usable format for analysis
- Store data in database or text file

Web Scraping = Data Scraping of web pages

Web Scraping - What is a web page ?

Components of a web page

- HTML - Organize and contain the main content of a web page
- CSS - Add styling to make the page looks nicer
- JS - Javascript files add interactivity to web pages
- Media files - Images, Sounds, Videos, etc.

Interesting content for web scraping = **HTML**

Web Scraping - HTML

HTML is used to create documents on the Web

Very simple and logical

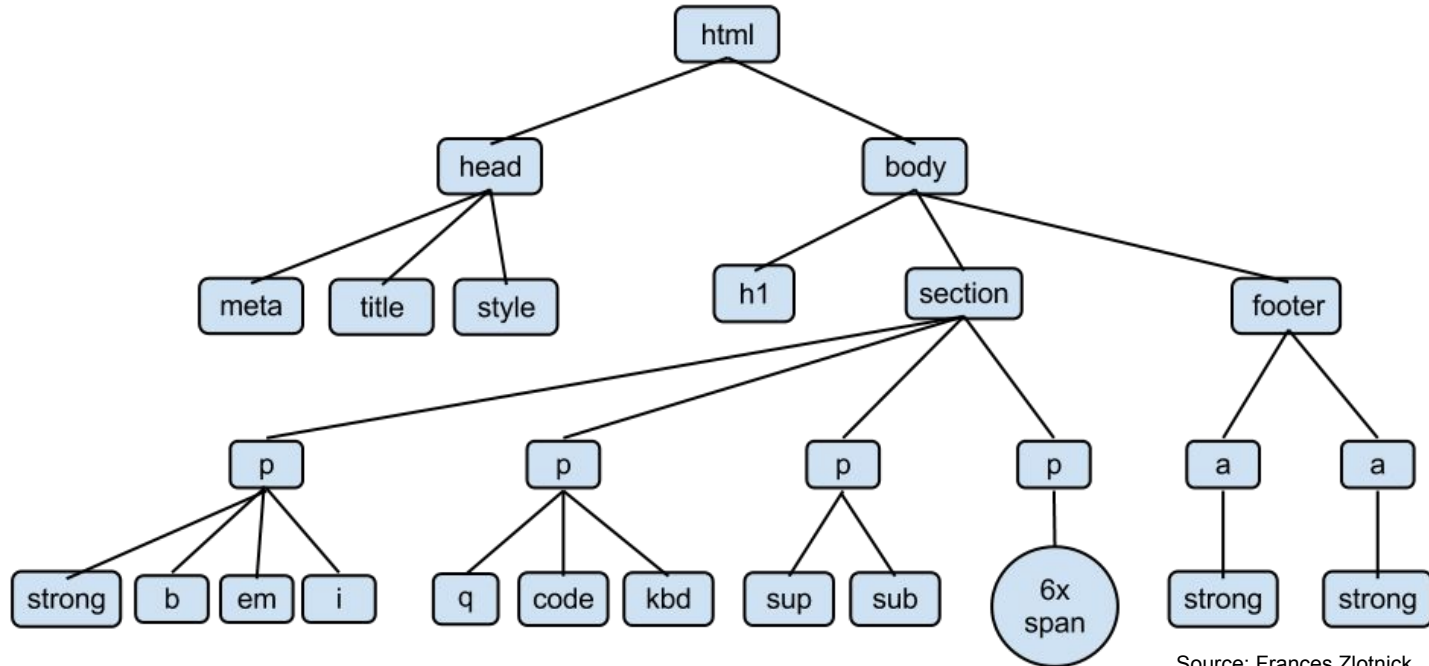
NOT a programming language but a **markups** language which use <tags> like this

The websites you view are basically HTML files rendered by web browsers

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Example</title>
5          <link rel="stylesheet" href="style.css">
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>
```

Web Scraping - HTML

HTML is organized like a hierarchical tree



Source: Frances Zlotnick

Web Scrapping - Inspect the source

Inspect the element

Find de HTML node

<table> defines a table

<tr> defines a row in a table

<th> defines a table header cell

<td> defines a cell in table

Use BeautifulSoup to grab it

Delhi MCD Election Results 2017

MCD Election Results 2017

MCD Election Results – Party Wise

Party Name	Leading/Won (2017)	2012 Results
AAP	47	AAP did not contest in 2012.
BJP	184	138
Congress	30	77
Others	10	57

Ward Wise MCD Election 2017 Results

```
<div class="left-links-bx"></div>
<!--end left link box section-->
<div class="ad160" style="display: none !important;"></div>
<div class="ad160" style="display: none !important;"></div>
<div class="ad160" style="display: none !important;"></div>
</div>
<!--Side panel starts-->
<div id="content-main">
  <div class="content-panel">
    <div class="header-base"></div>
    <div id="dump"></div>
    <div class="cl"></div>
    <div class="text">
      <!--/5535731/Elections-below-h1-new-->
      <div id="div-gpt-ad-1458386879591-0" style="height: 90px; width: 728px; display: none !important;"></div>
      <h2>MCD Election Results – Party Wise</h2>
      <table class="tableizer-table" style="background:none!important" cellspacing="0">
        <tbody>
          <tr>
            <th="">Party Name</th>
            <th=""></th>
            <th>2012 Results</th>
          </tr>
          <tr class="L_w">
            <td class="atable">AAP</td>
            <td class="atable">47</td>
            <td class="atable">AAP did not contest in 2012.</td>
          </tr>
          <tr class="L_w">
            <td class="atable">BJP</td>
            <td class="atable">184</td>
            <td class="atable">138</td>
          </tr>
          <tr class="L_w">
            <td class="atable">Congress</td>
            <td class="atable">30</td>
            <td class="atable">77</td>
          </tr>
          <tr class="L_w">
            <td class="atable">Others</td>
            <td class="atable">10</td>
            <td class="atable">57</td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>
```

Web Scraping - BeautifulSoup

Python library

Pull out data out of HTML/XML files

Designed for quick turnaround projects

Charged with some superb methods

Open-source, free & well documented

Web Scraping - Jump into the code

Grab the node with BeautifulSoup

```
from BeautifulSoup import BeautifulSoup
import urllib

raw_html =
urllib.urlopen('http://www.elections.in/delhi/mcd-elections/').read()

soup = BeautifulSoup(raw_html)

attrs = { 'class':'tableizer-table' }
tables = soup.findAll(attrs=attrs)
table = tables[0]
rows = table.findAll('tr')
```

} Import librairies

} Download data

} Instantiate
BeautifulSoup object

} Access the data

Web Scraping - Jump into the code

Use grabbed data to write a CSV file

```
import csv

with open('export.csv', 'wb') as f:
    writer = csv.writer(f, delimiter=';')
    for row in rows:
        csv_row = []
        headers = row.findAll('th')
        for header in headers:
            csv_row.append(header.text)
        cells = row.findAll('td')
        for cell in cells:
            csv_row.append(cell.text)
        writer.writerow(csv_row)
```

} Import the CSV library

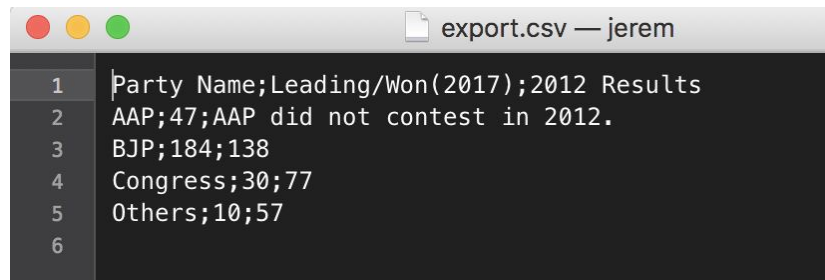
} Open a file with write permissions
} Handle it with CSV lib's methods

} Make loops for selecting data
inside table cells.
} Write them in a python list

} Write the list in the CSV handle file

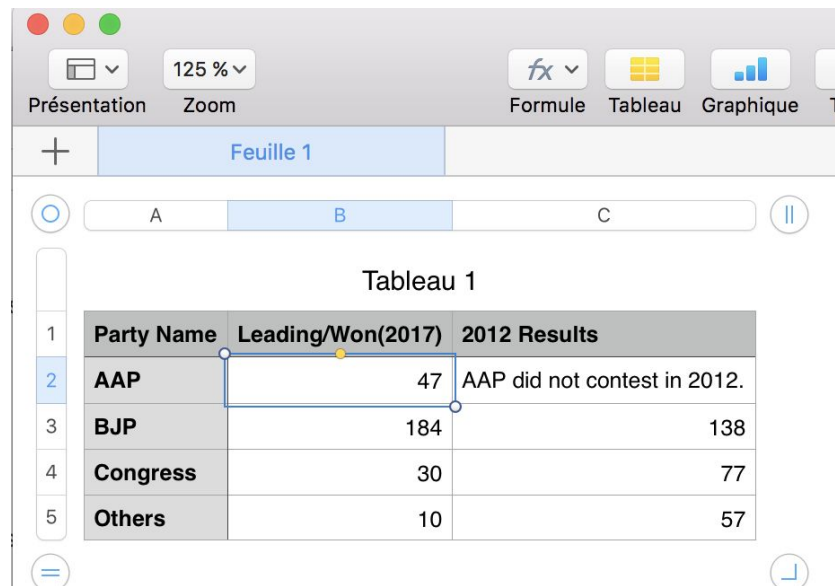
Web Scrapping - Jump into the code

Extraction Result



A screenshot of a text editor window titled "export.csv — jerem". The window contains a CSV file with 6 lines of data. The first line is a header, and the following five lines contain data for different political parties.

	Party Name;Leading/Won(2017);2012 Results
1	AAP;47;AAP did not contest in 2012.
2	BJP;184;138
3	Congress;30;77
4	Others;10;57
5	
6	



A screenshot of a Tableau interface showing a table view of the CSV data. The interface includes a toolbar with options for "Présentation", "Zoom", "Formule", "Tableau", and "Graphique". The table is titled "Tableau 1" and has three columns: "Party Name", "Leading/Won(2017)", and "2012 Results". The data is displayed in a table with 5 rows, corresponding to the data in the CSV file.

	Party Name	Leading/Won(2017)	2012 Results
1	AAP	47	AAP did not contest in 2012.
2	BJP	184	138
3	Congress	30	77
4	Others	10	57
5			

Let's play !

<https://ashoka.cdsp.sciences-po.fr>