

# Lab #4: Web Scraping

---

Or, “How to get banned from Yelp”

Oct 17, 2019

# What is web scraping?



# What is web scraping?

- you need data (for research, personal interest, etc.)
- it's on the web
- there doesn't seem to be an API for it

# What is web scraping?

Get from here...

IMDb Charts

## Top Rated Movies

Top 250 as rated by IMDb Users

Showing 250 Titles

Sort by: Ranking

Rank & Title	IMDb Rating	Your Rating
1. <a href="#">The Shawshank Redemption</a> (1994)	★ 9.2	☆
2. <a href="#">The Godfather</a> (1972)	★ 9.1	☆
3. <a href="#">The Godfather: Part II</a> (1974)	★ 9.0	☆
4. <a href="#">The Dark Knight</a> (2008)	★ 9.0	☆
5. <a href="#">12 Angry Men</a> (1957)	★ 8.9	☆
6. <a href="#">Schindler's List</a> (1993)	★ 8.9	☆
7. <a href="#">The Lord of the Rings: The Return of the King</a> (2003)	★ 8.9	☆

...to here

```
0 title rating
1 The Shawshank Redemption 9.2
2 The Godfather 9.1
3 The Godfather: Part II 9.0
4 The Dark Knight 9.0
5 12 Angry Men 8.9
6 Schindler's List 8.9
7 The Lord of the Rings: The Return of the King 8.9
8 Pulp Fiction 8.9
9 The Good, the Bad and the Ugly 8.8
10 Fight Club 8.8
11 The Lord of the Rings: The Fellowship of the Ring 8.8
12 Joker 8.8
13 Forrest Gump 8.8
14 Inception 8.7
15 Star Wars: Episode V - The Empire Strikes Back 8.7
16 The Lord of the Rings: The Two Towers 8.7
17 The Matrix 8.6
18 One Flew Over the Cuckoo's Nest 8.6
19 Goodfellas 8.6
20 Seven Samurai 8.6
21 Se7en 8.6
22 City of God 8.6
23 Life Is Beautiful 8.6
24 The Silence of the Lambs 8.6
25 It's a Wonderful Life 8.6
26 Star Wars: Episode IV - A New Hope 8.6
```

# Before we begin

All materials available at:

<https://github.com/5harad/css/tree/master/web-scraping>

Credit to Jongbin Jung (2017 TA) and Joe Nudell (SCPL) for materials and inspiration.

# Outline

- Motivation
- Approaches
- Example 1: IMDB Top 250
  - HTML, live demo
  - Exercise: Wikipedia – Lakes of California
- Example 2: Public Notices
  - Chrome devtools, live demo
- Scraping ethics
- Postscript: Selenium WebDriver

# Some motivating examples

- Building a business directory for the United Kingdom
  - sources: tripadvisor, timeout, etc.
- Building a graph of actors (nodes) and movies (edges)
  - “six degrees of Kevin Bacon”
  - sources: IMDB, ...
- Writing a bot to check Craigslist for apartments
- Determining which subreddits tend to contain more hate speech, etc.

# Approaches

- **Standard**

- load page, parse HTML

- **Front door**

- public API

- **Back door**

- decipher internal structure (e.g. hidden API)

- **The imposter**

- control a browser automatically



# Approaches

- **Standard**

- load page, scrape HTML

● ~~Front door~~ ← We've seen this before (Twitter API)

○ ~~public API~~

- **Back door**

- decipher internal structure (e.g. hidden API)

- **The imposter**

- control a browser automatically

# Approaches

- **Standard**

- load page, scrape HTML

- ~~Front door~~ ← We've seen this before (Twitter API)

~~○ public API~~

- **Back door**

- decipher internal structure (e.g. hidden API)

- **The imposter**

- control a browser automatically

← This is really hard!

# Example 1: IMDB Top 250

---

# Example 1: IMDB Top 250

The “standard approach”

**Goal:** Collect the **cast overview** (actor + character played) for each of the **top 10 movies** on the **IMDB Top 250** list.

[https://www.imdb.com/chart/top?ref\\_  
\\_nv\\_ch\\_250\\_4](https://www.imdb.com/chart/top?ref_=nv_ch_250_4)

---

(live demo)

# Top Rated Movies

Top 250 as rated by IMDb Users

Showing 250 Titles

Sort by: Ranking

Rank & Title	IMDb Rating	Your Rating
1. <a href="#">The Shawshank Redemption</a>		
2. <a href="#">The Godfather</a> (1972)		
3. <a href="#">The Godfather: Part II</a> (1973)		
4. <a href="#">The Dark Knight</a> (2008)		
5. <a href="#">12 Angry Men</a> (1957)		
6. <a href="#">Schindler's List</a> (1993)	★ 8.9	☆

Open Link in New Tab

Open Link in New Window

Open Link in Incognito Window

Save Link As...

Copy Link Address

Copy

Search DuckDuckGo for "The Shawshank Redemption"

Print...

Block element...

Inspect

Speech Services

Elements Console Sources Network Performance

```
top250movie">
  > <colgroup>_</colgroup>
  > <thead>_</thead>
  > <tbody class="listner-list">
    > <tr>
      > <td class="posterColumn">_</td>
      > <td class="titleColumn">
        "
        1.
        "
        <a href="/title/tt0111161/?
        pf_rd_m=A2FGELUUN00JNL&pf_rd_p=e31d89dd-322d-4646-8962_
        0PZKZXZJX9FFPS&pf_rd_s=center-
        1&pf_rd_t=15506&pf_rd_i=top&ref=chttp_tt_1" title="Frank
        Darabont (dir.), Tim Robbins, Morgan Freeman">The
        Shawshank Redemption</a> == $0
        <span class="secondaryInfo">(1994)</span>
        </td>
      > <td class="ratingColumn imdbRating">_</td>
      > <td class="ratingColumn">_</td>
      > <td class="watchlistColumn">_</td>
    </tr>
  </tbody>
</table>
... #pagecontent div #content-2-wide #main div span div div div table tbody tr td a
```

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter

element.style {

a:visited {

color: #70579D;

text-decoration: none;

a:link {

color: #136CB2;

text-decoration: none;

a:visited {

color: #70579D;

text-decoration: none;

a:link {

color: #136CB2;

text-decoration: none;

margin

border

padding

auto x auto

Filter

Show all

background-color

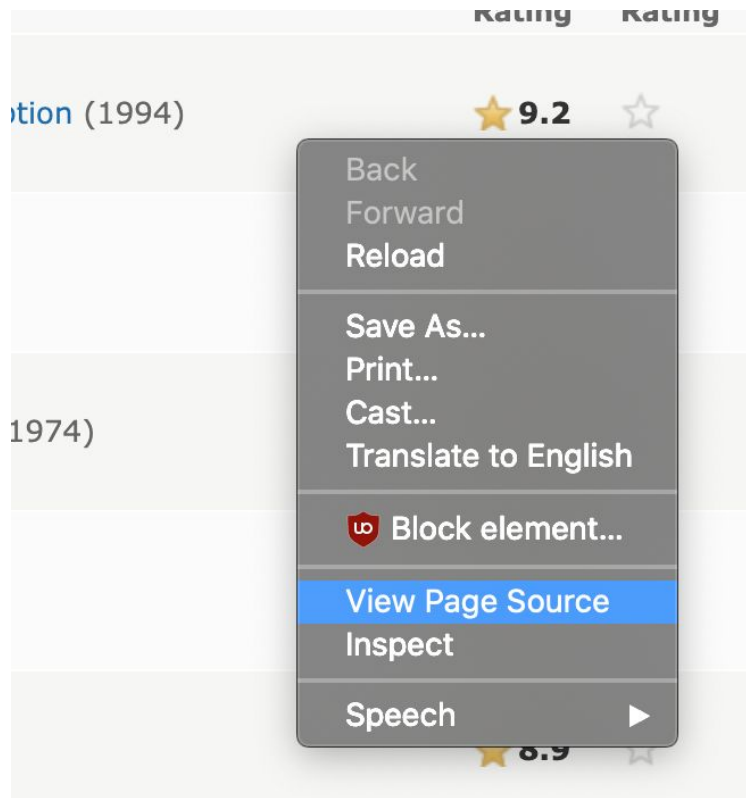
rgba(0, 0, 0, 0)

border-collapse

collapse

box-sizing

## Inspecting HTML



```

<body class="listner-list">
<tr>
<td class="posterColumn">
<span name="rk" data-value="1"></span>
<span name="ir" data-value="9.222197604788654"></span>
<span name="us" data-value="7.791552E11"></span>
<span name="nv" data-value="2147579"></span>
<span name="ur" data-value="-1.7778023952113458"></span>
<a href="/title/tt0111161/?pf_rd_m=A2FGELUUNOQJNL&pf_rd_p=e31d89dd-322d-4646-8962-327b42fe94b1&pf_rd_r=1&pf_rd_t=15506&pf_rd_i=top&ref=chttp_tt_1"
> 
</a>
<td class="titleColumn">
1.
<a href="/title/tt0111161/?pf_rd_m=A2FGELUUNOQJNL&pf_rd_p=e31d89dd-322d-4646-8962-327b42fe94b1&pf
l&pf_rd_t=15506&pf_rd_i=top&ref=chttp_tt_1"
title="Frank Darabont (dir.), Tim Robbins, Morgan Freeman" >The Shawshank Redemption</a>
<span class="secondaryInfo">(1994)</span>
</td>
<td class="ratingColumn imdbRating">
<strong title="9.2 based on 2,147,579 user ratings">9.2</strong>
</td>
<td class="ratingColumn">
<div class="seen-widget seen-widget-tt0111161 pending" data-titleid="tt0111161">
<div class="boundary">
<div class="popover">
<span class="delete">&nbsp;</span></div>
</div>
<div class="inline">
<div class="pending"></div>
<div class="unseeable">NOT YET RELEASED</div>
<div class="unseen"></div>
<div class="rating"></div>
<div class="seen">Seen</div>
</div>
</div>
</td>
<td class="watchlistColumn">
<div class="wlb_ribbon" data-tconst="tt0111161" data-recordmetrics="true"></div>
</td>
</tr>
</tbody>

```

Viewing HTML source



Learning how HTML works...



# Parsing HTML

```
import requests  
import bs4
```

```
IMDB_PAGE = "https://www.imdb.com/chart/top?ref\_=nv\_ch\_250\_4"
```

```
response = requests.get(IMDB_PAGE)  
soup = bs4.BeautifulSoup(response.content)
```

```
# for example  
soup.find_all('td', attrs={'class': 'titleColumn'})
```

# Exercise: Wikipedia

Using the “standard approach”

**Goal:** Collect the **name**, **county**, **type**, and **surface area** for each of the lakes in California (from Wikipedia).

[https://en.wikipedia.org/wiki/List\\_of\\_lakes\\_in\\_California](https://en.wikipedia.org/wiki/List_of_lakes_in_California)

---

## Example 2: Public Notices

---

# Public Notices: Background

- City of Mountain View is required by law to post public notices of certain city proceedings in a newspaper
- It does so in the *San Jose Post Record*
- The *San Jose Post Record* has a total print distribution of...

**49** copies

- Public notices also available on legaladstore.com (not a spam site, i swear!)

Ref: <https://www.mv-voice.com/news/2019/10/04/for-public-notices-mountain-view-turns-to-obscure-newspaper>

# Example 2: Public Notices

The “back door” approach

**Goal:** Collect **all government** notices for **Santa Clara County** posted since **2019-01-01**.

<https://www.legaladstore.com/>

---

(live demo)

View Ad

Ad

Close

**NOTICE TO CONTRACTORS**

Sealed written proposals are invited by the CITY OF SAN JOSE for:

9003 - San Jose Parking Management Command Center

In accordance with and as described and provided in the Plans and Specifications thereof and the proposed form of contract thereof, all of which are on file in the office of the Director of Public Works of the City, and which are made a part hereof.

**PLANS AND SPECIFICATIONS**

The City is using Biddingo, an online bid solicitation website, to facilitate this procurement. You must register with Biddingo to participate in this procurement. There is no cost associated with registering.

To register, bidders must go online to <https://www.biddingo.com/sanjose>. This procurement is registered under the bid number and bid name above and has the following commodity code classifications(s):

001000 - HVAC,  
017500 - Concrete  
100701 - Cons. Doors and Windows,  
100317 - Cons. Building,  
100313 - Utilities,  
031500 - Flooring,  
051000 - Miscellaneous

Search Legal Ads

Price Quote | Place an Ad

- Search our database of published legal ads by location, newspaper, notice type or order number.
- Ads are updated daily.

Click on following tabs to Search Ads by:

State/County/Type of Notice

Select State

CA

Select County

SANTA CLARA

Select Category

GOVERNMENT

Select Notice Type

ALL

Date Range From

10/09/2019

To

10/16/2019

(eg 02/10/2012)

Display

☒ First Published
 ☐ ALL Published

Search

Order No	Ad Description	First Published	Sale/Hearing/Bid Date	View Ad
3302957	9265 - Sanitary Sewer Condition Assessmnet 201	019	10/31/2019	View ...
3303221	172MCMURTRIE#58819	019		View ...
3304428	8732- Bailey Ave Storm Drain Inlet Repair Project	019	11/07/2019	View ...
3304442	9003- San Jose Parking Management Command Center	019	11/07/2019	View ...
3304435	9283 - 2019 Bridge Deck Treatment	019	11/07/2019	View ...
3304654	AF-1904-22008	019		View ...
3304089	AF-1909-06075	019		View ...
3302157	AF-1909-27078	019		View ...
3302543	AF-1910-05079	019		View ...
3303737	BUILDING AND FIRE CODES	019		View ...

Page 1 of 4

Head to the console again (⇧-⌘-C, ⇧-⌘-C)

Elements Console Network >> 1 2

Filter ☐ Hide data URLs

All XHR JS CSS Img Media Font Doc WS Manifest Other

50 ms 100 ms 150 ms 200 ms

Name	St...	Ty...	Initiator	Size	1	Waterfall
cfcGetAdSearchDetails.cfc?	200	xhr	external	2	0	

https://www.legaladstore.com/CFC/cfcGetAdSearchDetails.cfc?method=getSearchData&&&&StateId=1&&&&CategoryId=2&&&&NoticeId=0&&&&County=SANTA%20CLARA&&&&dateFrom=10/09/2019&&&&dateTo=10/16/2019&&&&btnSearch=Search,Search&&&&dateType=1&pageSize=10&\_cf\_ajaxproxytoken=777C959853F79B89C31CC15A&\_cf\_clientid=435C81C8E6735E862CBAC805E1E4C40E&\_cf\_rc=18&\_cf\_nodebug=true&\_cf\_nocache=true&returnFormat=json&\_dc=1571289288525&start=0&limit=10&page=1&gridsortcolumn=&gridstartdirection=ASC

Elements Console Network >> 1 2

Filter ☐ Hide data URLs

All XHR JS CSS Img Media Font Doc WS Manifest Other

50 ms 100 ms 150 ms 200 ms

Name	St...	Ty...	Initiator	Size	1	Waterfall
cfcGetAdSearchDetails.cfc?	200	xhr	external	2	0	

- Look Up "cfcGetAdSearchDetails"
- Open in new tab
- Clear browser cache
- Clear browser cookies
- Copy

ViewAd.cfm?Pr...	200	xhr	cfajax...	1...	8	
------------------	-----	-----	-----------	------	---	--

https://www.legaladstore.com/ViewAd.cfm?Productid=4473992&Producer=1&\_cf\_containerId=WinVAD-body&\_cf\_nodebug=true&\_cf\_nocache=true&\_cf\_clientid=435C81C8E6735E862CBAC805E1E4C40E&\_cf\_rc=38

Watch the "Network" tab



# Through the back door we go!

```
import requests

search_results = requests.get(
    "https://www.legaladstore.com/CFC/cfcGetAdSearchDetails.cfc",
    params={
        "method": "getSearchData",
        "CategoryId": 2,
        "County": "SANTA CLARA",
        "dateFrom": "01/01/2019",
        "dateTo": "10/17/2019",
        "limit": 10000,
        # ... more parameters ...
    })
```

# Through the back door we go!

```
for record in search_results.json()["QUERY"]["DATA"]:
    ad_download = requests.get(
        "https://www.legaladstore.com/ViewAd.cfm",
        params = {
            "Productid": record[4],
            "Producer": 1,
            # ... more fields ...
        })
    # save the ad somewhere, e.g. write to a file or print
    print(ad_download.text)
```

# Scraping Ethics

---

# Scraping Ethics

- Throttling
  - Your script can visit web pages a lot faster than you can!
  - Compute resources are not unlimited
  - Fix: using `time.sleep` in a Python script, e.g.
  - If not careful, you can get a whole IP block banned from a site
- Robots
  - The `robots.txt` file exists as a guideline for web spiders, etc.; be a good internet citizen and abide by them!
  - Check out <https://en.wikipedia.org/robots.txt> for an extensive example
- Terms of service
  - Certain sites (Yelp, i'm looking at you!) explicitly prohibit scraping in their terms of service
  - However, this likely has no legal ramifications

# Postscript: Selenium WebDriver

---

# Selenium WebDriver

- Programmatically “drive” the browser
- Allows more flexibility (closer to human interaction)
- Possible applications:
  - Multi-step login
  - Filling out a CAPTCHA
- Difficult/error-prone. Prefer other methods when possible.

# Want more?

Materials for this section are at:

<https://github.com/5harad/css/tree/master/web-scraping>

A more in-depth tutorial on web scraping can be found here:

[https://github.com/stanford-policylab/iriss-workshop/blob/master/py/python\\_scraping-exercises.ipynb](https://github.com/stanford-policylab/iriss-workshop/blob/master/py/python_scraping-exercises.ipynb)

(rendered version can be viewed [here](#))