

Intro to Web Scrapping

November 1, 2018 GW Libraries

Slides: go.gwu.edu/webscraping

Dan Kerchner kerchner@gwu.edu

Laura Wrubel lwrubel@gwu.edu

Install the Scraper Chrome extension: bit.ly/chrome-scraper

Objectives

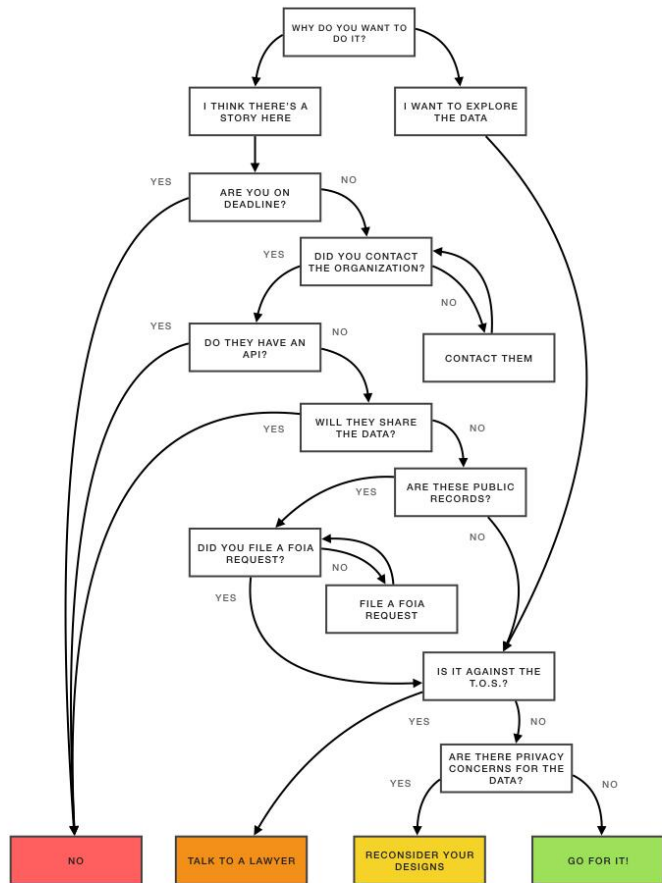
- What is web scraping and what is it good for?
- Technical and ethical considerations
- Using Scraper on a web page (hands-on)
- Demo of Tabula on PDF pages
- Discussion of using Python for web scraping

Install the Scraper Chrome extension: bit.ly/chrome-scraper

What is web scraping?

Extracting data from a web page, using cut-and-paste, code, or another tool that parses the HTML.

Should You Build a Scraper?



Considerations

- Is this data available some other way? (bulk download, API)
- Is the data well-structured on the website? Is it all on one page? Is the page dynamic/interactive?
- Are there terms of service concerning the website?
- Are there copyright concerns?
- What am I planning to do with this data? Be cautious about sharing data.

Structure of a web page: **HTML + CSS + JavaScript**

HTML provides the basic structure of a page. The HTML is enhanced and modified by CSS and JavaScript.

CSS is used to control styling: presentation, formatting, and layout.

JavaScript is used to control the behavior of different elements.

Structure of a web page

```
.bold-paragraph {  
  font-weight: bold;  
  color: red  
}
```

```
<html>  
  <head>  
    <link href="css_file.css" rel="stylesheet" type="text/css" media="all">  
  </head>  
  <body>  
    <div id="text-section1" class="box-around">  
      <p class="bold-paragraph" style="font-size:16">  
        Here's some bold text and href here is an attribute of the tag  
        <a href="https://library.gwu.edu" id="library-link">a link to GW Libraries</a>  
      </p>  
      <p class="bold-paragraph extra-large">  
        Here's another paragraph, even bigger  
      </p>  
    </div>  
    <table id="table1">  
      <tr>  
        <td>Stuff inside a table cell</td>  
      </tr>  
    </table>  
  </body>  
</html>
```

a tag, or node

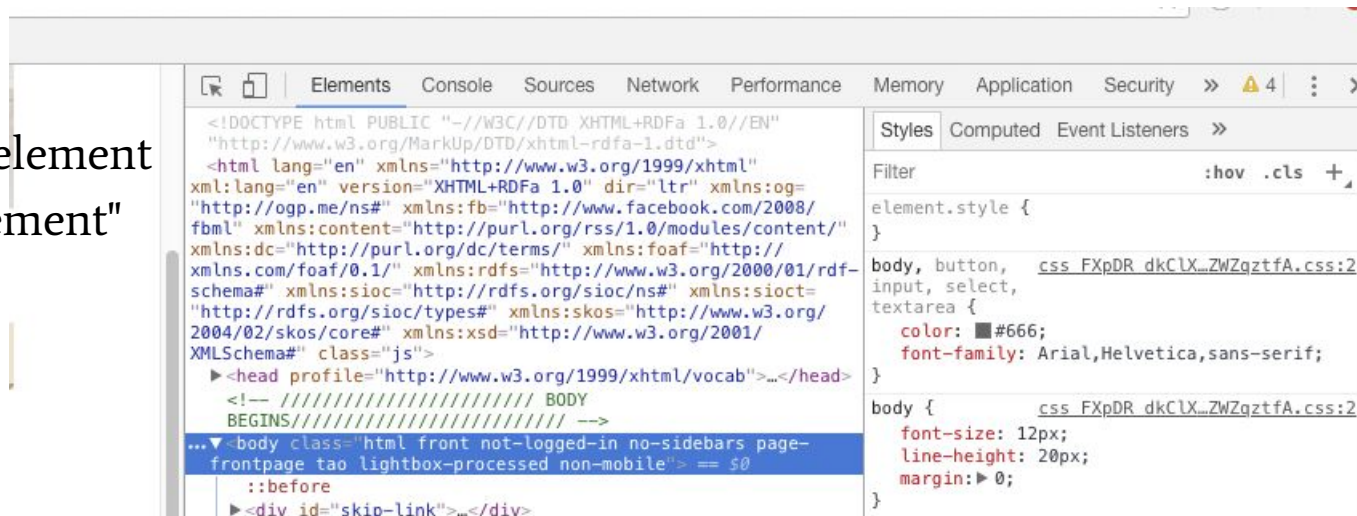
Working with a web page

In Chrome (other browsers have a similar tool):

View → Developer → Developer Tools

OR

Right-click on a page element
and select "Inspect Element"



XPath

DOM = Document Object Model

"A language for addressing parts of an XML document"
(a web page is an HTML document is an XML document)

Example:

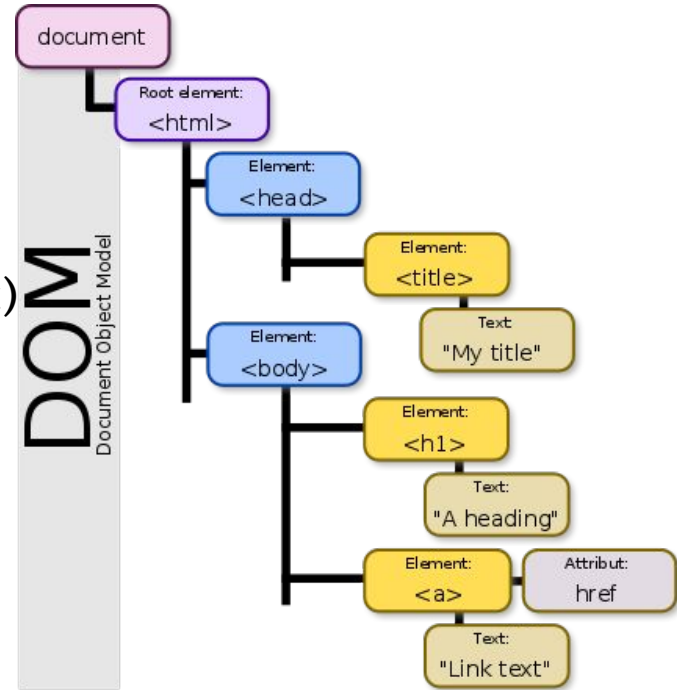
```
//div[@id='text-section1']/p/text()
```

// - At any level down in the "tree"

div - any div tag, with an id attribute of 'text-section1'

/p - get back all child <p> tags

/text() - and give me the text of each tag



en.wikipedia.org/wiki/Document_Object_Model#/media/File:DOM-model.svg

Web scraping with Scraper

- Chrome extension for scraping web pages.
- Uses XPath to identify elements in HTML.
- Works best if data is on a single page and HTML is well-structured.
- Doesn't work with PDFs.

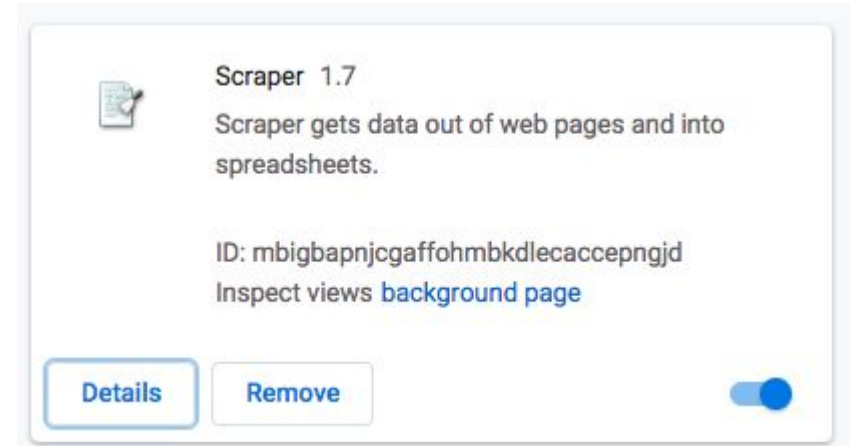
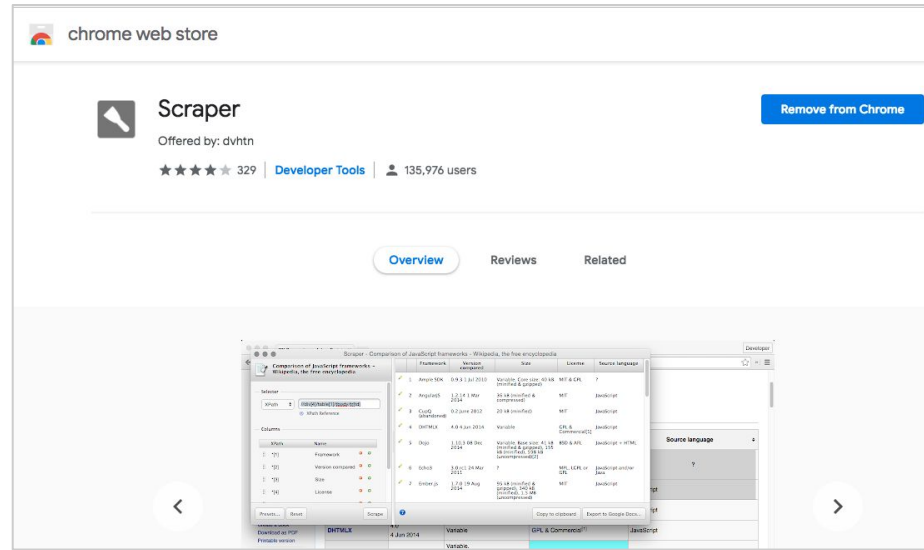
Install the Scraper Chrome extension:

bit.ly/chrome-scraper

Installing Scraper

In the Chrome Webstore:
bit.ly/chrome-scraper

Installation directions and tips:
go.gwu.edu/gmuscraper



Scraper steps

1. Load your target web page.
2. Highlight and right-click on a part of the web page and click “Scrape similar...”
3. Tweak the XPath to get the elements you need.
4. Export to Google Docs or copy into Excel.

GWU Schedule of Courses: <https://my.gwu.edu/mod/pws/>

Spring 2019: Main Campus > pick a department



OFFICE OF THE REGISTRAR

[SCHEDULE OF CLASSES HOME](#)[COURSE SEARCH](#)[RENUMBERED COURSE KEY](#)[COURSE SYLLABUS](#)[OFFICE OF THE REGISTRAR](#)

Schedule of Classes

[HOME](#) » [MAIN CAMPUS](#) - [SPRING 2019](#) » [AMERICAN STUDIES](#)

Result Page: 1 - 2

[Next Page >>](#)

Helpful Hints:

Subject: Click on the course number to view the Bulletin description

Bldg/Rm: Click on the building to view the street address

XLlist: Click to view the same course offered by another department

Linked: Click to view associated discussions, labs, etc.

[PRINT ALL](#) | [PRINT THIS PAGE](#)

STATUS	CRN	SUBJECT	SECT	COURSE	CREDIT	INSTR.	BLDG/RM	DAY/TIME	FROM / TO	
OPEN	47078	AMST 1000	10	Zombie Capitalism	3.00	Orenstein, D	PHIL 108	M 12:45PM - 03:15PM	01/14/19 - 04/29/19	
Comments: Registration restricted to CCAS freshmen only. For more information about Dean's Seminars click on http://go.gwu.edu/ccasdeansseminars Course Attributes										Find Books

American Studies Courses Spring 2019 (Main Campus)
| The George Washington University

Selector

XPath [? XPath Reference](#)

Columns

XPath	Name	
*[2]	CRN	
*[3]	Course Code	
*[4]	Section	
*[5]	Course	
*[6]	Credit	
*[7]	Instructor	
*[8]	Room	

Filters

☒ Exclude empty results

Presets...

Reset

Scrape

	CRN	Course Code	Section	Course	Credit	Instructor	Room
1	Result Page: 1 - 2		Next Page >>				
2	47078	AMST 1000	10	Zombie Capitalism	3.00	Orenstein, D	PHIL 108
3	47271	AMST 1000	11	Bodies of Work	3.00	Ivy, N	GELM 402
4	48074	AMST 1000	12	Washington Sex Scandals	3.00	Heap, C	MON 250
5	47821	AMST 1160	10	Race, Gender, and Law	3.00	Rule, E	1957 E B12
6	47276	AMST 2000	10	Politics of "Saving Africa"	0.00 OR 3.00	McAlister, M	1957 E 311
7	46568	AMST 2011	80	Modern American Cultural History	3.00	Orenstein, D	FNGR 108
8	43262	AMST 2020W	80	Washington, DC: History, Culture, and Politics	3.00	Klemek, C	1957 E 112
9	46040	AMST 2071	80	Introduction to the Arts in America	3.00	Bjelajac, D	SMTH 114
10	44220	AMST 2120W	80	Freedom in American Thought and Popular Culture	3.00	Anker, E	PHIL B156
11	47284	AMST 2210	10	The African American Experience	3.00	Musser, A	FNGR 222
12	44145	AMST 2380	80	Sexuality in U.S. History	3.00	Heap, C	PHIL B156
13	47288	AMST 2490	10	American Contagions	3.00	Ivy, N	ROME 350
14	42532	AMST 2521	80	American Architecture II	3.00	Jacks, P	SMTH 114
15	47289	AMST 2710	80	The United States in Global Context, 1898-Present	3.00	McAlister, M	1957 E 213
16	44820	AMST 3361	80	African American History Since 1865	3.00	Clement, B	MON 115
17	48373	AMST 3811	80	Historical Archaeology	3.00	Cressey, P	
18	46049	AMST 3901	10	Examining America	3.00	Anker, E	P 201
19	44822	AMST 3950	10	USConstructionsofMidEast	3.00		DUQUES 250
20	42501	AMST 4400	10	Independent Study	3.00	Wald, G	
21	46278	AMST 4701W	80	Epidemics in American History	3.00	Gamble, V	COR 204
22	\$10.00						
23	\$10.00						



Copy to clipboard

Export to Google Docs...



American Studies Courses Spring 2019 (Main Campus) | The George Washington University

Selector

XPath

//table[@class="courseListing"]/tbody/tr[1]

[XPath Reference](#)

Columns

XPath	Name
*[1]	Column 1
*[2]	Column 2
*[3]	Column 3
*[4]	Column 4
*[5]	Column 5
*[6]	Column 6
*[7]	Column 7
*[8]	Column 8
*[9]	Column 9
*[10]	Column 10
*[11]	Column 11

Filters

☒ Exclude empty results

Presets...

Reset

Scrape

		Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8
	1	OPEN	47078	AMST 1000	10	Zombie Capitalism	3.00	Orenstein, D	PHIL 108
	2	OPEN	47271	AMST 1000	11	Bodies of Work	3.00	Ivy, N	CELM 402
	3	OPEN	48074	AMST 1000	12	Washington Sex Scandals	3.00	Heap, C	MON 250
	4	OPEN	47821	AMST 1160	10	Race, Gender, and Law	3.00	Rule, E	1957 E 812
	5	OPEN	47276	AMST 2000	10	Politics of "Saving Africa"	0.00 OR 3.00	McAlister, M	1957 E 311
	6	OPEN	46568	AMST 2011	80	Modern American Cultural History	3.00	Orenstein, D	FNGR 108
	7	OPEN	43262	AMST 2020W	80	Washington, DC: History, Culture, and Politics	3.00	Klemek, C	1957 E 112
	8	OPEN	46040	AMST 2071	80	Introduction to the Arts in America	3.00	Bjelajac, D	SMTH 114
	9	OPEN	44220	AMST 2120W	80	Freedom in American Thought and Popular Culture	3.00	Anker, E	PHIL 8156
	10	OPEN	47284	AMST 2210	10	The African American Experience	3.00	Musser, A	FNGR 222
	11	OPEN	44145	AMST 2380	80	Sexuality in U.S. History	3.00	Heap, C	PHIL 8156
	12	OPEN	47288	AMST 2490	10	American Contagions	3.00	Ivy, N	ROME 350



Copy to clipboard

Export to Google Docs...

Tabula for PDFs

<https://tabula.technology>

- Free, open-source application for identifying and extracting data tables from PDFs.
- Useful when tables don't cleanly cut-and-paste into a spreadsheet, or are on many pages.
- Exports data as CSV, TSV, or JSON.
- Does not work with "image" PDFs. Must be OCR.
- Data may require further clean-up.

Tabula



Tabula is a tool for liberating data tables locked inside PDF files.

[View the Project on GitHub](#)
tabulapdf/tabula

Download for
Windows

Download for
Mac

View source on
GitHub

Current Version: 1.2.1

Other Versions: [pre-releases & archives](#)

Need help? Open an [issue on Github](#).

Donate: Help support this project by [backing us on OpenCollective](#).

We'd love to hear from you! Say hi on Twitter at [@TabulaPDF](#)

Tabula for PDFs (demo)

[GW Daily Crime and Fire Log](https://safety.gwu.edu/daily-crime-and-fire-log) (<https://safety.gwu.edu/daily-crime-and-fire-log>)

 **DIVISION OF
SAFETY AND SECURITY**

APPLY TO BE A STUDENT ACCESS MONITOR

About the Division | Residence Hall Safety & Security | Health & Emergency Management | GW Police | Victim Services | Training

Home ▶ About the Division ▶ Reports and Records ▶ Daily Crime and Fire Log

Who We Are

Reports and Records

Daily Crime and Fire Log

Annual Security & Fire Safety Report

Clery Incident Reporting

GWPD Incident Report Request

Compliance

Emergency Communications

Complaints and Commendations

Daily Crime and Fire Log

The crime and fire logs are the daily records of all crimes and fires that have been reported to the GW Police Department. They are organized chronologically and are updated on a daily basis. Paper copies of both the crime and the fire logs are available upon request. Requests can be made in person in Rome Hall, Suite 101 during normal business hours.

Crime LogFire Log

- October 2018
- Septmenber 2018
- August 2018
- July 2018
- June 2018
- May 2018

More tools for web scraping

- Python libraries:
 - requests
 - bs4 (beautifulsoup)
 - scrapy
- R packages:
 - rvest
 - you'll also need the read_html() function (from xml2 package)
- Command-line tools (bash/Linux shell):
 - wget - gets a URL
- OpenRefine
 - Has some html parsing functions
- Other browser plug-ins

Web scraping with Python

Libraries you may need:

- `import requests` *# to retrieve the web page*
- `from bs4 import BeautifulSoup` *# to parse the HTML*
- `import scrapy` *# another web scraping library*

Do you need to interact with the page?

- No: Try web scraping with requests + beautifulsoup
- Yes: You may need something like Selenium WebDriver

Web scraping can be easy or hard

- Is the page static or dynamic? Do you have to interact with the page to get the content you want?
- A web page's structure can change without warning!
- Does the content you want require clicking through multiple pages?
- How well-written is the page's HTML? Do tags have 'id' attributes?

Web scraping code of conduct

Ask nicely. If your project requires data from a particular organisation, for example, you can try asking them directly if they could provide you what you are looking for. With some luck, they will have the primary data that they used on their website in a structured format, saving you the trouble.

Don't download copies of documents that are clearly not public. For example, academic journal publishers often have very strict rules about what you can and what you cannot do with their databases. Mass downloading article PDFs is probably prohibited and can put you (or at the very least your friendly university librarian) in trouble. If your project requires local copies of documents (e.g. for text mining projects), special agreements can be reached with the publisher. The library is a good place to start investigating something like that.

Check your local legislation. For example, certain countries have laws protecting personal information such as email addresses and phone numbers. Scraping such information, even from publicly available web sites, can be illegal (e.g. in Australia).

Don't share downloaded content illegally. Scraping for personal purposes is usually OK, even if it is copyrighted information, as it could fall under the fair use provision of the intellectual property legislation. However, sharing data for which you don't hold the right to share is illegal.

Share what you can. If the data you scraped is in the public domain or you got permission to share it, then put it out there for other people to reuse it (e.g. on datahub.io). If you wrote a web scraper to access it, share its code (e.g. on GitHub) so that others can benefit from it.

Don't break the Internet. Not all web sites are designed to withstand thousands of requests per second. If you are writing a recursive scraper (i.e. that follows hyperlinks), test it on a smaller dataset first to make sure it does what it is supposed to do. Adjust the settings of your scraper to allow for a delay between requests. By default, Scrapy uses conservative settings that should minimize this risk.

Publish your own data in a reusable way. Don't force others to write their own scrapers to get at your data. Use open and software-agnostic formats (e.g. JSON, XML), provide metadata (data about your data: where it came from, what it represents, how to use it, etc.) and make sure it can be indexed by search engines so that people can find it.

Resources

- [Programming Historian tutorials on web scraping](#)
- Lynda.com tutorials (lynda.it.gwu.edu) tutorials: search on web scraping
- Make an appointment for a coding consultation: calendly.com/gwul-coding
- Slides <https://go.gwu.edu/webscraping>

Dan Kerchner kerchner@gwu.edu

Laura Wrubel lwrubel@gwu.edu