Extracting Data From the Web Using Python

Web Data Extraction Workshop D-Lab UC Berkeley



Learning Objectives

- The purpose of web scraping and APIs when to use either
- HTML & CSS and their use in web scraping.
- Use the requests and BeautifulSoup libraries to acquire and parse data from websites.
- Use a 3rd-party Python library to make API calls.
- Working with data pulled from APIs.



Why Webscrape

- Tons of web data useful for social scientists and humanists
 - social media
 - news media
 - government publications
 - organizational records
- Two kinds of ways to get data off the web
 - Webscraping i.e. user-facing websites for humans
 - APIs i.e. application-facing, for computers



Webscraping v. APIs

- Webscraping Benefits
 - Any content that can be viewed on a webpage can be scraped. Period
 - No API needed
 - No rate-limiting or authentication (usually). Your IP can be blocked (403)
- Webscraping Challenges
 - Rarely tailored for researchers
 - Messy, unstructured, inconsistent
 - Entirely site-dependent
- Rule of thumb:
 - Check for API first. If not available, scrape.

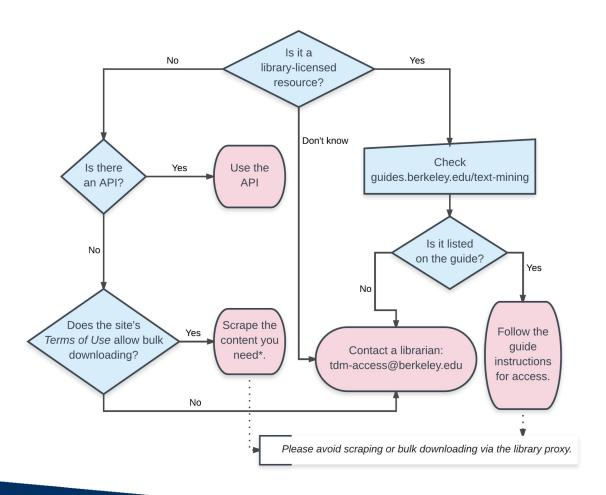


Some Disclaimers!!

- Check a site's terms and conditions before scraping.
- Be nice don't hammer the site's server.
 - Add a time delay in between batches of scrapping.
- Sites change their layout all the time. Your scraper will break.



Workflow

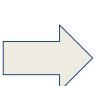




What's a website

- Some combination of codebase, database.
- The "front end" product is HTML + CSS stylesheets + javascript.
- Browser turns the left image into the right.

"elementary/force/force/advise author/154" burdlers white " aligne conter" classe detail "650/force/advise author/154" "Berffungerigherkemmigenti widthe" ble" byedner"BBBBF "lanar"detail "en konfe"/amana/tanatus aug/Sachtanabar(be) [1976] " byedner"BBBBF "diger"netter" alamar"detail "eta konfe"betaburbilla aug/Bankar(be)[7776][1877847] bederkt widthe" ass-"detail" ha hast-"bediessibless asplicated D-2174" Committant/at-The D-704-03 size-"118" legislat-"20000 Film righter /ndring white "174" hooker "88888" alique "contes" classe "detail "right/better/adverture transfer at the "414" hoo utorkilla aspreminellellilriekillatys-trinot-tradevak sidkorfikki handiseriskisi alignetannot alampetantalliva konteina 1215-konmuntamanturetrantetak middetilde jaki handiseriskinet alignetsmannet elampetannot elampetak midderik midderik hotel "do/facto/sportprote width "404" sporter "2000" slamar data; "na brah"/amara/mator magida/fabbasa/Dajid Na" Sporter "20000" align="amar" data; "na brah" beatrahila ag/Mandar[20100" blian/ar/Mator/ubrah align=" Cases Serial Little Areas Serial Serial State and Resident De 2004 "Attended American Serial oraclinde/fontes/totalest algebrille "gentine" (1900) aligne" content classe "detail "Selfborger/total/agentine (1904) base engine applications and interior and an arms and the property of the second sector of the second sector and a sect Plane Shini Print Brade Sentential La and Sentential State Shill be April Sent Sentential State Sentential Sen "ENTER" slight outset class "dear" dearl' be / house/labe/labe/labeld middle "FD" baselor "stile" class "dearl' bas brack / house / ho width "IN" bycologi"stits" align" paints" class" data! """s trats" biologictum, asymmetric 21: "" align" back of data! """ align" back of data! "" align" back of data! """ align """ align """ align "" align """ align Classe Scinit Ten Analy Senatural Lie, and Charges 25-1887 181 (art/art/funt) Library and the 1987 agon to "Stratte" D'allips'instar' class'delail"99/fest-Chirchiretrend midde"354, application'slave delail"em hydr'/assata/6 G127-despublica E. Collinaria-Chirchiretd widde"154' applica-"abita" align-'corter' class-'detail"em traf-'benato -153°-953]etteriformyldend widde's N° Spoologe'wite' allgor'ester' class'detel'ete bod'e Modospittee, septemb N° (N° Ngoslor-Wile' digor'ester' sjass'denel'tjerifontrifond sidno'154° bpolog-white' allgor'ester' sjass state "densitive head-"tensorabile anglements-1212" willergate/basheratora eight-"14" housing-market aliesmillione and Monter (1910) ("riconsistement and forces (millioned widolo" 154" bycolose "689909" alligan testees" "WHERE" aligne "dentar" aligne "detail "ele/factive/age/furvers/end widths "life" inpution "white" aligne "detail "ele/factive" /accete/fact HE "John J. Callathar-Variational State-Valle-Valle and the "IM" Species " white " aligne content cleaner default " a hear- decade ("Advid width"))) bysiser"stife" elige "ester" sisee "detsiller best "betownitee eegiteeber15-150 vomentees/ar/finst "adite" elige "ester" elese "detsiller/detsi width"))) bysiser"stife" elige "ester" sisee "detsiller/detsinon"



Current Senate Members 99th General Assembly Leadership Officers Senate Seating Chart Democrats: 39 Republicans: 20

Senator	Bills	Committees	District	Party
Pamela J. Althoff	Bills	Committees	32	R
Neil Anderson	Bills	Committees	36	R
Jason A. Barickman	Bills	Committees	53	R
Scott M. Bennett	Bills	Committees	52	D
Jennifer Bertino-Tarrant	Bills	Committees	49	D
Daniel Biss	Bills	Committees	9	D
Tim Bivins	Bills	Committees	45	R
William E. Brady	Bills	Committees	44	R
Melinda Bush	Bills	Committees	31	D
James F. Clayborne, Jr.	Bills	Committees	57	D
Jacqueline Y. Collins	Bills	Committees	16	D
Michael Connelly	Bills	Committees	21	R
John J. Cullerton	Bills	Committees	6	D



Webscraping returns HTML

- It's easy to pull HTML from a website
- It's much more difficult to find the information you want from that HTML
- So we have to learn how to parse HTML to find the data we want

```
<html lang="en-us" class="a-js a-audio a-video a-canvas a-svg a-drag-drop a-geolocation a-history a-webworker a-autofocus a-input-placeh</pre>
older a-textarea-placeholder a-local-storage a-gradients a-hires a-transform3d -scrolling a-text-shadow a-text-stroke a-box-shadow a-borde
r-radius a-border-image a-opacity a-transform a-transition a-ember" data-19ax5a9if="dingo" data-aui-build-date="3.21,9-2022-01-05">
   <!-- sp.feature.head-start -->
  ▶ <head>...</head>
   <!-- sp:end-feature:head-close -->
   <!-- sp:feature:start-body -->
  ▼<body class="a-m-us a-aui_72554-c a-aui_accordion_a11y_role_354025-c a-aui_killswitch_csa_logger_372963-c a-aui_launch_2021_ally_fixes
  _392482-c a-aui_pci_risk_banner_210084-c a-aui_preload_261698-c a-aui_rel_noreferrer_noopener_309527-c a-aui_template_weblab_cache_33340
  6-c a-aui_tnr_v2_180836-c a-meter-animate" data-new-gr-c-s-check-loaded="14.1043.0" data-gr-ext-installed>
   ▼ <div id="a-page">
     ▶ <script type="a-state" data-a-state="{"key":"a-wlab-states"}">...</script>
       <script>typeof uex === 'function' && uex('ld', 'portal-bb', {wb: 1})</script>
       <!-- sp:end-feature:start-body -->
     ▶ <script>...</script>
       <script>window.ue && ue.count && ue.count('CSMLibrarySize', 13275)</script>
       <!-- sp:feature:nav-inline-is -->
       <!-- NAVYAAN JS -->
     ▶ <script type="text/javascript">...</script>
     ▶ <script type="text/javascript">...</script>
       <img src="https://images-na.ssl-images-amazon.com/images/G/01/gno/sprites/nav-sprite-global-1x-hm-dsk-reorg. CB405937547 .png"</pre>
       style="display:none" alt>
       <script type="text/javascript">var nav_t_after_preload_sprite = + new Date();</script>
     ▶ <script>...</script>
       <!-- sp:end-feature:nav-inline-js -->
       <!-- sp:feature:nav-skeleton -->
       <!-- sp:end-feature:nav-skeleton -->
       <!-- sp:feature:navbar -->
       <!--Pilu -->
       <!-- NAVYAAN -->
       <!-- navmet initial definition -->
     ▶ <script type="text/javascript">...</script>
       <script type="text/javascript">window.navmet.tmp=+new Date();</script>
     ><script type="text/javascript">...</script>
       <style mark="aboveNavInjectionCSS" type="text/css"> div#navSwmHoliday.nav-focus {border: none;margin: 0;} </style>
     ▶ <script mark="aboveNavInjectionJS" type="text/javascript">...</script>
     ▶ <noscript>...</noscript>
       <script type="text/javascript">window.navmet.push({key:'PreNav',end:+new Date(),begin:window.navmet.tmp});</script>
       <a id="nav-top"></a>
       <a id="skiplink" tabindex="0" class="skip-link">Skip to main content</a>
       <script type="text/javascript">window.navmet.tmp=+new Date();</script>
       <!-- Navyaan Upnav -->
     ▶ <div id="nav-upnav" aria-hidden="true">...</div>
       <script type="text/javascript">window.navmet.push({key:'UpNav',end:+new Date(),begin:window.navmet.tmp});</script>
       <script type="text/javascript">window.navmet.main=+new Date();</script>
```



Basic strategy of webscraping:

- Find out what kind of HTML element your data is in. (Use your browser's "inspector")
- 2. Think about how you can differentiate those elements from other, similar elements in the webpage using HTML/CSS anatomy.
- 3. Use Python and add-on modules like BeautifulSoup to extract just that data.

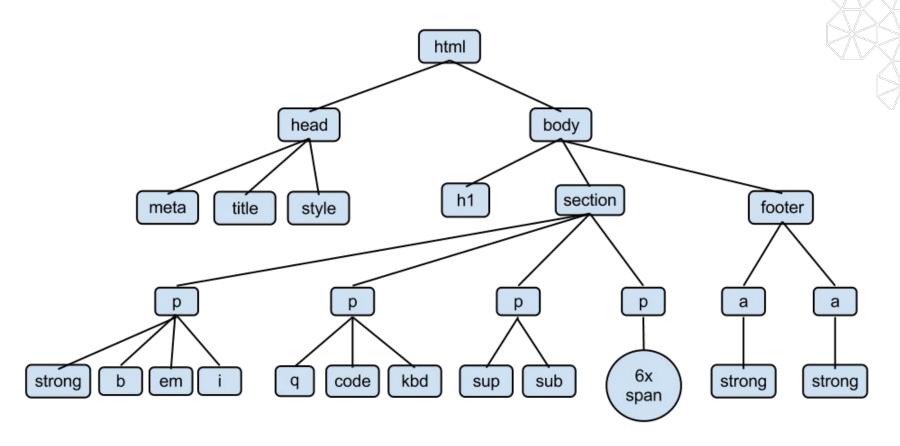


HTML: Basic structure

```
<!DOCTYPE html>
<html>
    <head>
       <title>Page title</title>
   </head>
    <body>
        Hello world!
    </body>
</html>
```



HTML as a Tree



Each branch of the tree is called an element



HTML Elements

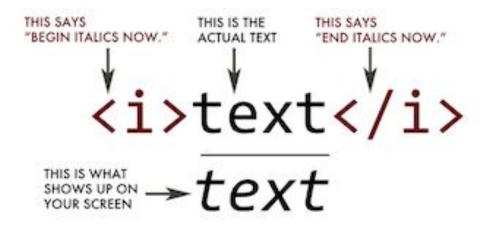
Generally speaking, an HTML element has three components:

- 1. Tags (starting and ending the element)
- 2. Attributes (giving information about the element)
- 3. Text, or Content (the text inside the element)





HTML: Tags

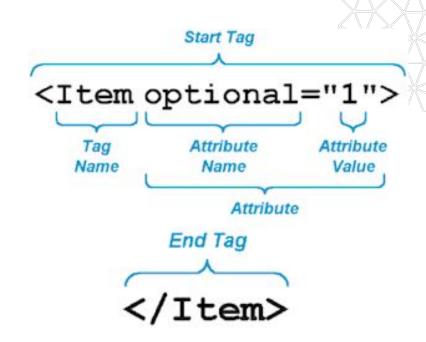


Tag	Meaning
<head></head>	page header (metadata, etc
<body></body>	holds all of the content
	regular text (paragraph)
<h1>,<h2>,<h3></h3></h2></h1>	header text, levels 1, 2, 3
ol,, ,	ordered list, unordered list, list item
</a 	link to "page.html"
,,	table, table row, table item
<div>,</div>	general containers



HTML Attributes

- HTML elements can have attributes.
- Attributes provide additional information about an element.
- Attributes are always specified in the start tag.
- Attributes come in name/value pairs like: name="value"





Finding HTML

- Sometimes we can find the data we want just by using HTML tags or attributes (e.g, all the <a> tags)
- More often, this isn't enough: There might be 1000
 <a> tags on a page. But maybe we want only the <a> tags inside of a tags.
- This is where CSS comes in.



CSS (Cascading Style Sheet)

- CSS defines how HTML elements are to be displayed
- HTML came first. But it was only meant to define content, not format it. While HTML contains tags like and <color>, this is a very inefficient way to develop a website.
- To solve this problem, CSS was created specifically to display content on a webpage. Now, one can change the look of an entire website just by changing one file.
- Most web designers litter the HTML markup with tons of classes and ids to provide "hooks" for their CSS.
- You can piggyback on these "hooks" to jump to the parts of the markup that contain the data you need.



CSS Selectors & Declarations

Туре	HTML	CSS Selector
Element	<a>,	a pa
Class	</a 	.blue a.blue
ID	</a 	#blue a#blue

Selector: a

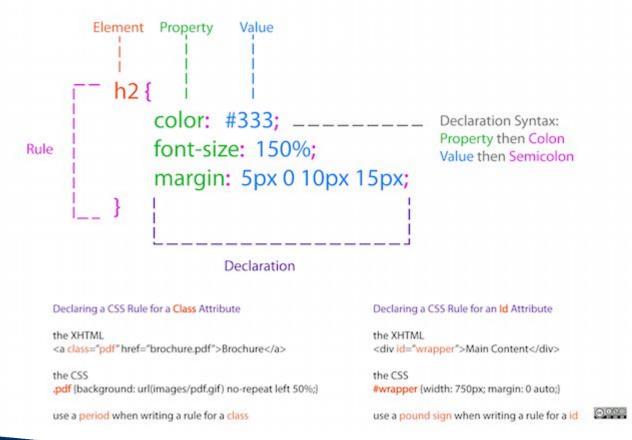
• Property: background-color

Value: yellow



CSS Hooks

Basic Anatomy of a CSS Rule





CSS meets HTML

What does the following HTML render to?

```
<body>
Kurtis
 McCoy
 Leah
 Guerrero
```



Relevance to Webscrapping

- It's necessary to cover HTML & CSS because we'll be using their tags to parse scrapped content.
- In the next section we'll be using the BeautifulSoup library to scrape web data.
- As previously mentioned, that's the easy part.
- Now that we know how tags work and what they're used for, we'll have an easier time sifting through the pile of HTML data

