Web Scraping

Steve V. lannaccone Taryn Heilman

(modified by Jon Courtney)



- Motivation
- The Internet and the WWW
- HTML and CSS
- Beautiful Soup
- Web APIs

Learning Objectives



- Differentiate Internet vs. World Wide Web (www)
- Understand the basics of the client-server relationship in a web framework
- Explain the basic concepts of HTML and CSS
- Know how to use Python to pull content from a web page
- Know how to use Python to pull content using an API



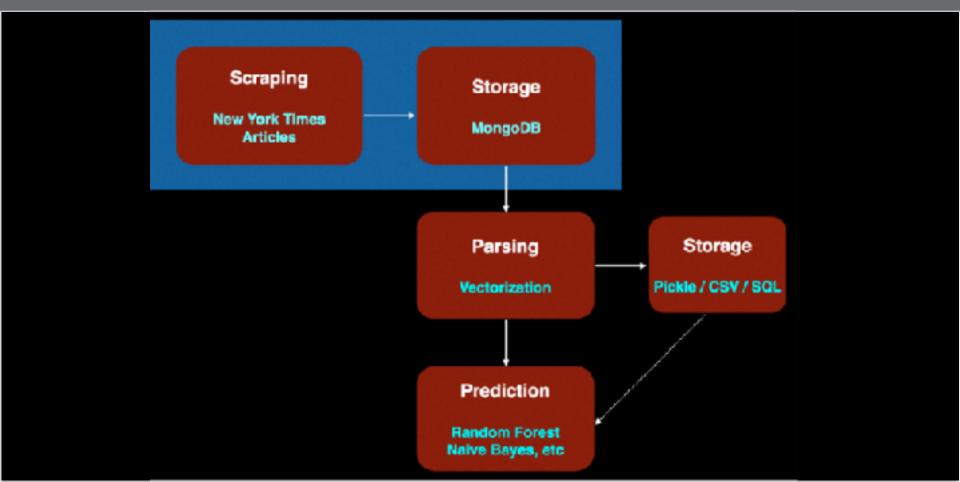
Motivation: Why do we want to do web scraping?

- Any time that you want data from the web and it doesn't have a clickable link, you will have to pull down that data via the command line (e.g., using curl) or via a program (e.g, using Python)
- The web is an enormous database of text/image/video training data*

^{*} Be aware of any Terms of Service when using a website.

Some websites really don't like scrapers... https://techcrunch.com/2016/08/15/linkedin-sues-scrapers/

Typical Web Scraping + Machine Learning Pipeline



WWW vs Internet



- The world wide web is a global collection of interconnected hypermedia documents hosted on web servers
- The *Internet* is the global network that connects them (using TCP/IP)
- Think of web servers as islands existing all over the globe, while the Internet provides bridges connecting those islands

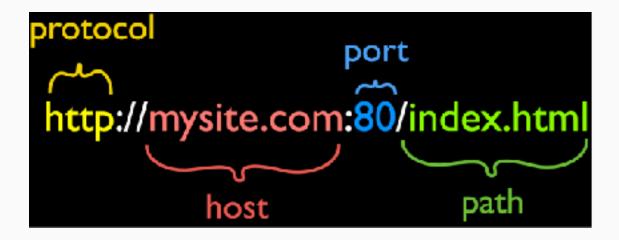


Less helpful:

- "...the Internet is not something you just dump something on. It's not a big truck. It's a series of tubes."
- Senator Ted Stevens, opposing net neutrality
 June 28, 2006



- URL Uniform Resource Locator. Used to specify the location of a document within the world-wide web.
- Each url has a few different parts...





- Protocol specifies the means for communicating with the web server, typically http or https. Often gets automatically filled in in your web browser, but this is usually not the case when web scraping.
 See http://www.realifewebdesigns.com/web-resources/web-protocols.html for examples.
- Host points to the name of the web server you want to communicate with.
 A host name is associated with a specific IP address.
- Port holds additional information used to connect with the host. (Think: host is the city name, port is street address)
- Path Indicates where on the server the file you are requesting lives



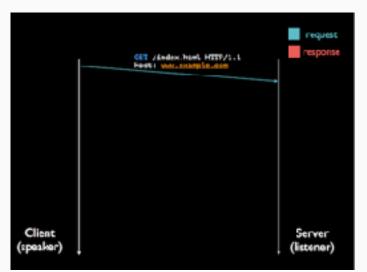
Client-Server Relationships

- At any point in time, any person or computer connected to the internet can be either a server or a client
- The client is the requesting party (requesting some info, like a webpage)
- The server is the party providing that information and responding to requests from a client

Client-Server Relationships



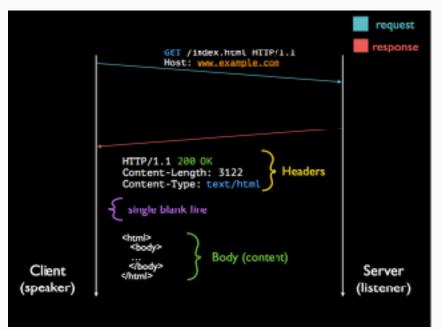
- If we visit www.example.com in our browser, then we are the client and www.example.com is the server.
- The interaction starts with the client issuing a GET request to the server, indicating that it would like some specific piece of information



Client-Server Relationships



 Once the server gets this request, it will send back a response with the information requested in the body (and a header with a status code in it)





Http Status Codes

- In general...
- 2xx successful (usually 200)
- 3xx redirect ultimately successful
- 4xx client side error aka the user's fault (common: 404 you are looking for a file that doesn't live where you think it does, or wrong permissions)
- 5xx server side error aka the service's fault

For more info, see http://www.restapitutorial.com/httpstatuscodes.html



HTML: Hypertext Markup Language

- The majority of web pages are formatted using Hypertext Markup Language (HTML)
 - Transmitted via Hypertext Transport Protocol (HTTP) over TCP/IP
- HTML uses tags, where each tag describes different document elements

Basic HTML Tags



Tag	Description
<h1> - <h6></h6></h1>	Heading
>	Paragraph
<i>></i>	Italic
	Bold
<a>	Anchor (links)
 & < <l> < < <l> < < <l> < < < <l> < < < <l> < < < <l> < < <l> < </l></l></l></l></l></l></l></l></l></l></l></l></l></l></l></l></l></l></l></l></l></l>	Unordered List & List Item
<blockquote></blockquote>	Blockquote
	Image
<div></div>	Division



HTML and CSS

- Permits clean separation of content from presentation style
- Radical site redesigns are possible by modifying just the style and not rewriting any of the content



http://www.csszengarden.com

```
51 <body id="css-zen-garden">
52 <div class='page-wrapper'>
                      <section class="intro" id="zen-intro">
                                   <header role="banner">
                                               <h1>CSS Zen Gardens/h1>
                                               <h2>The Beauty of <abbr title="Cascading Style Sheets">C3S</abbr> Design</h2>
                                   </header>
                                   <div class="summary" id="zen-summary" role="article">
                                               A demonstration of what can be accomplished through <abbr title="Cascading Style Sheets">CSS<//i>
51
                                               Oownload the example <a href="/examples/index" title="This page's source HTML code, not to be to be
                                   </div>
                                   <div class="preamble" id="zen-preamble" role='article">
                                               <h3>The Road to Enlightenment</h3>
                                               Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible <all
                                               Ye must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless.
                                               The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. 1
                                   \langle div \rangle
                      </gention>
```



The Beauty of CSS Design Download the evan sleinted file and cap file

The Road to Enlightenment

liftering in leit and thoughout legitle post efficient to wave specific Eags, in impositive IRANs, but widths input di, and also the lift investes.

We must describe mind after a sec. Web enlightenmost has been address distribute the trades efforts of fall the the $\frac{1}{2}$ $\frac{1}{2}$

The CSS Zer Gerden Invited you to reflect and modified a self-or in particular to see of the months. Region to permedically in the process to see the to well control the forequest to seen and one group the facilities. Secrement as well to the section.





The Beauty of CSS Design



A demonstration of enterior can be arranged than storing of \$100 beautiful ranges before any object and \$100 beautiful to had also that a gap. Done had the accomple block (Beautiful)

Days load the as ear file

The Road to Enlightenment

Utilizing a diskund dietaly roud lay the peak relies of browner open for tags, incompanion ususy, notions assumption, and accurate transfers.

We must clear the mind of the past Neb enlights mant has been addressed thanks to the theres will be of fulfille the pts. 强强。 and the major breast products.

The LM verticement invites you to relact and meditate on the important exposit of the masters, usign to see with disregulation or use the disself-dispredicting local the disself-dispredicting feetback Security translating local terms.

So What is This About?

This is a continuing meat to show the power of QSS. The Zer C Sandon after the continue to their recently provided particles on in grip, all we are well described by the gypothethe that Shalling in any sension contains only show one one sees soon. The Hilliam remains the extra the only thing that has around a life Parental containing search.

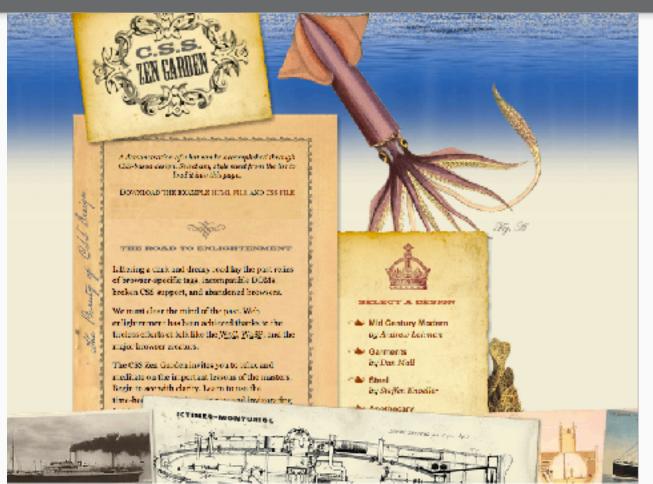
25 allows complete and seaf control over the system of systems, down risk. The only may the control flavored the may that gots are a neutral risk greatment transplant and to the personnel rest seaf placet in the residuant attracts either over construction shows the supplementation of the seaf control residuant of the seaf control resi

Participation

Drong view data in her silveys earn our focus frou are modifields. This page spectrum (SIS with arts reviewed yout but the own tow. The are commented with enough the west (SIS inches can use them are conting out to Macro are the (SIS findered duals for accessored a tree opening as a wholey (Mine to the productions are whose) (Mine to the silveys of the silv









CSS Classes, IDs and Selectors

- Web developers may define CSS classes (starting with .) to identify HTML elements for styling
 - The same CSS class can be assigned to multiple page elements
 - Applying a new style to that class in the CSS changes the appearance of all elements using that class
- CSS IDs (starting with #) identify a single, unique element within a page.
- CSS *selectors* are patterns used to select HTML element(s) (including those matching CSS classes and IDs) for formatting.



HTML & CSS Breakout

Go to:

https://www.w3schools.com/cssref/trysel.asp

Explore the various types of CSS selectors.



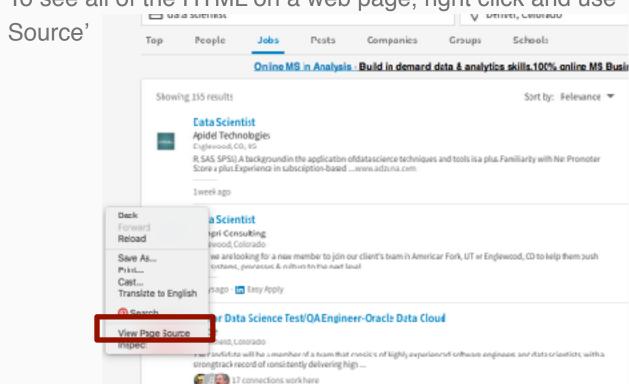
HTML and Web Scraping

- All those HTML tags, style classes and ids make great hooks for finding exactly the content you want in a web page.
- In practice, when web scraping, we will issue a get request for an entire page's html, and then pull specific elements either by their HTML tags or CSS selectors.

Viewing Page Source



• To see all of the HTML on a web page, right click and use 'View Page



Idayeago - 🔚 Issy Apply

Which yields this mess...

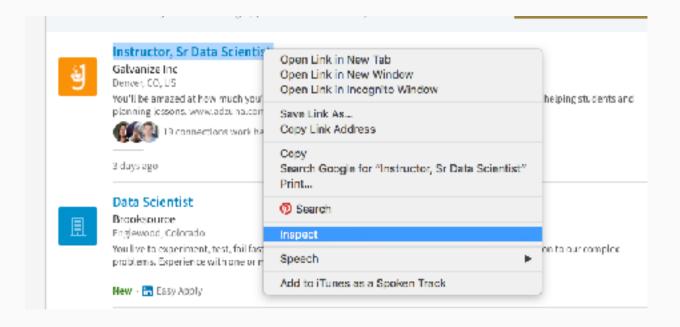


```
1 <1000TYPE heal>
a shtml lang="en">
   <heed>
      Secript type='application/israsoript'>ifunction(i.m){void 0!==1.addEventListener#Evoid 0!==1.hidden&&(n.liVisibilityChangeListener=function()(1.hidden&&
  (n.liMasWindowHidden=:0)},i.addEventListener("visibilitychange",n.liVisibilityChangeListener))}(document,vindow);/script>
      <meta charact="atf-8">
      cmeta http-equiv="X-UA-Compatible" content="IE-edge">
      <title>LinkedIn</title>
      <meta name="description" content="">
      Speta name="viewport" content="width-device-width, initial-scale=1.0, minimum-scale=1.0">
      %meta name="theme-color" content="#007785">
 strets name="extended/config/environment"
  content="978922moduleProfix92293A922wxtended92292C922enviroment92293A922production92292C922lix92293A975922test992293A955922cts.rump.article-
  reader$22$22$22cublishin$22$2C$22vovacer.feed.web.mative-video-ble$22$2C$22vovacer.feed.web.enable-so-sequence-number$22$2C$22vovacer.feed.web.article-mini-
  tags%22%2C%22voyager.feed.web.likes-comments-fetch-override-to-rerc%22%2C%22voyager.feed.web.enable-edited-comment-
  flag82292C$22pnblishin822$22voyager.feed.web.sharing.show-original-author-context$2282C$22voyager.feed.web.sharing.drag-and-
  drop$22920$22voyager.feed.olient.bashtags$22$20$22voyager.feed.olient.flex-beight-irage$22$20$22voyager.feed.olient.vnfollox-bub-
  filters%22%20%2voyager,search,web.gostsVertical%22%20voyager,search,web.richt-rail-ness-module%22%200voyager,web.feed.richt-rail.fellos-
  recommendations%22%2C%22voyager.feed.web.right-rail.follow-recommendations-recent-shares%22%2C%22voyager.feed.web.incorrectly-rentioned-in-news-undate-
  action%22%20%22veyager.feed.web.follow-fpr-page%22%20%22veyager.feed.web.unfollow-channel-action%22%20%22veyager.feed.web.remove-su-
  topbar#2292C$22voyager.feed.web.auto-
  play.v2%22%2C%22voyager.feed.video.autoplay.support%22%2C%22voyager.feed.video.autoplay.ugc%22%2C%22voyager.feed.video.istwitterDisabled%22%2C%22voyager.gublishi
  video.player-beacon$22$26522voyager.feed.web.video-upload$22$26522voyager.feed.web.video-init$22$2622voyager.feed.web.video-frosted-
  background%22%2C%22vovacer.sharing.web.allew-disable-
  comments922522522voyager.sharing.web.mentions52992922voyager.sharing.web.mentions.api522522voyager.sharing.web.moltiphoto-
  oploads#93%20%22voyager.veb.feed.fie.visibleHeight#95%20%22voyager.feed.seb.cached-feed-dats#93%20%22voyager.veb.feed.gerf.layered-rendering%22%20%32learning.rec
  hites-campaign, flagship%22%2C%27weyeper.web.feed.right-rail.learning-bites%22%2C%22weyeper.feed.web.right-rail-learning-bites-above-
  ad$22$20$22voyager.web.feed.enable@aPixelCall$22$20$22voyager.feed.web.su-osn$22$20$22voyager.feed.web.lynda-update-social-actions$22$20$22voyager.feed.web.lynda
  apdate-widec-player%22%2C%22voyaper.premius.api.my-premius-access%22%2C%22voyaper.feed.web.identity-module-premius-branding-entitlement-
  change82292C$22voyager.common.web.entity-
```

Inspecting a single element



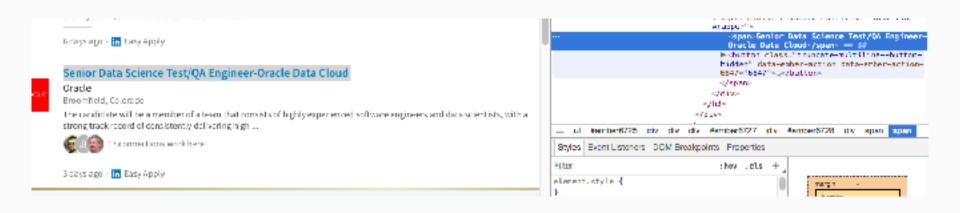
Better: right click and choose 'Inspect' (or 'Inspect Element')
 (Chrome, Firefox and Safari)



Resulting HTML from Inspect or Inspect Element



 Shows you the html that corresponds to the individual element you chose to look at



Scraping Individual Elements



 Examine HTML corresponding to individual element we want to scrape, and then use either its HTML tag or CSS selectors to parse it out in Python

Workflow:

- 1) Find the page you want to scrape information from
- 2) Find the element(s) that you want to grab
- 3) Use inspect element to figure out what HTML tag or CSS selectors to use to grab the elements
- 4) Use Python to fetch those elements

galvanize

Beautiful Soup

"You didn't write that awful page. You're just trying to get some data out of it. Beautiful Soup is here to help. Since 2004, it's been saving programmers hours or days of work on quick-turnaround screen scraping projects."

https://www.crummy.com/software/BeautifulSoup/





Using Python and BeautifulSoup

• Use *requests* library to issue a 'GET' request on the url, then use *BeautifulSoup* to parse the html

```
import requests
from bs4 import BeautifulSoup

req = requests.get('www.example.com')
html = BeautifulSoup(req.content, 'html.parser')
```



Using Python and BeautifulSoup

 Next, use methods on the html object returned from BeautifulSoup to get content from the page

```
html.find('a') # Returns the **first** 'a' tag.
html.find_all('a') # Returns **all** 'a' tags.
html.find('a').text # Returns the text of the
# **first** 'a' tag.
```

Beautiful Soup: Selecting Tags



- .select() This method allows us to select tags using CSS selectors
- .find_all() This method allows us to select all tags matching certain parameters and returns them as a list. For example:
 - soup.find_all('div'): returns a list of all div tags in the soup object
 - soup.find_all('div', attrs={'class': 'content'}):
 returns only the div tags that also have class = content.
- .find() This method is the exact same as calling soup.find_all(limit=1). Rather than returning a list, it only returns the first match that it finds.

Beautiful Soup: Navigating the HTML Tree



Tags may contain strings and other tags. These elements are the tag's children. Beautiful Soup provides a lot of different attributes for navigating and iterating over a tag's children. For example:

```
head_tag = soup.head
title_tag = head_tag.contents[0]
for child in title_tag.children:
    print(child)
```

You can do .find() and .select() on a tag's contents too.

See https://www.crummy.com/software/BeautifulSoup/bs4/doc/#navigating-the-tree



Pandas and Web Scraping

Yes, pandas can read and parse HTML. So, if you only need data from tables from a single page this is probably the easiest approach.

```
tables = pd.read_html("http://www.website.com/page.html")
```

This returns a list of DataFrames where each DF is made from a table in the source page. But, depending on formatting quirks, this might be significantly worse than using *Beautiful Soup*.

Using Web APIs



Application Programming Interfaces (APIs)

Webmasters don't want you pinging their websites every 10ms to scrape some information.

Often times they build APIs to help you collect just the info you need (and so that you don't crush their servers).

It's win-win because you use less bandwidth on their server, and you get the data you need in a consistently formatted fashion.



API Keys

You will often need to register to gain access to a site's API.

Sometimes to to track usage, often to limit the number of calls, and occasionally as a business model where you pay per call or by amount of transferred data.

These are essentially your login credentials, so **never** publish your keys! Don't push up any script to GitHub that has your keys in plain text.



API Key Security

One option: Store your key as an environment variable on your local machine.

In your bash profile (~/.bash-profile, ~/.bashrc, or ~/.profile) include the following:

```
export API KEY="my api key"
```



Using API Keys

Now we can use the variable API_KEY in our Python scripts using the os package without fear of publishing our private key when we push our repos.

```
import os
my_key = os.environ['API_KEY']
```



APIs: They're All Different

You'll have to dig into the docs to figure out how to use the API you need. But, if you want an intro on the broad strokes of how to interact with a web API in Python, this blog does a good job:

https://www.dataquest.io/blog/python-api-tutorial/



Beautiful Soup, so rich and green,
Waiting in a hot tureen!
Who for such dainties would not stoop?
Soup of the evening, beautiful Soup!
Soup of the evening, beautiful Soup!
Beau—ootiful Soo—oop!
Beau—ootiful Soo—oop!
Soo—oop of the e—e—evening,
Beautiful, beautiful Soup!

'Beautiful Soup! Who cares for fish,
Game, or any other dish?
Who would not give all else for two
Pennyworth only of beautiful Soup?
Pennyworth only of beautiful Soup?
Beau—ootiful Soo—oop!
Beau—ootiful Soo—oop!
Soo—oop of the e—e—evening,
Beautiful, beauti—FUL SOUP!

galvanize