combine.md

Digital Tools for Data-Driven Investigation

In this session we will explore the basics of web-architecture and see how websites are built. This will involve opening up chrome's developer tools in the browser to see how websites are structured. We will also discuss what makes for a well structured data driven web app, learn how to hit an API for data, and host a simple website of our own on github. We will also go through some examples of how to extract data from websites, how to find data behind a chart if it is on the page, and how to discover hidden or undocumented APIs.

link	description
setup.md	Setup Steps
lesson_1.md	Part 1: Anatomy of a Webpage
lesson_2.md	Part 2: Unpacking Websites & Hunting for Data
recap.md	Recap lessons

Registration Link: https://www.eventbrite.com/e/digital-tools-for-data-driven-investigation-tickets-32194610948

Part 1: Anatomy of a Web Page

How websites work

https://projects.propublica.org/graphics/images/data-institute/presentations/how-websites-work.pdf

Git and GitHub

A quick aside: github.md

HTML, CSS, & JavaScript

- HTML
- CSS
- JavaScript
  - Quick aside: this is a great JavaScript tutorial for newcomers, it will only take 2 hours or so. http://jsforcats.com/

Simple website example: https://dmil.github.io/simple-website/

Simple website code: https://github.com/dmil/simple-website

A Simple Website

Fork my simple website

1. Fork the simple website github repo https://github.com/dmil/simple-website

0:11 hrs

2. Turn on github pages in your repository so that it can serve up the webpage. In your fork of this repository on github, there is a "settings" tab. Click that tab and under "github pages" set the "source" to the master branch.
3. Make sure its being served in github pages by navigating to the URL of the new site.
4. Clone your fork onto your desktop with the github desktop app.

Inspect Element

1. Lets navigate to your website and explore the chrome developer tools for this simple website. How is it being put together?

Manipulating the Site

Lets explore together what happens when we comment out different parts of the code (HTML, CSS, and JavaScript)

1. Open the folder containing this code in sublime text.
2. Find the folder where the code is and double click on `index.html` to open.
3. Watch on-screen, don't worry about making the changes yourself, we will have have some time shortly for you to try it yourself.

Try It

1. Change the text of the headline
2. Change the image on the page to an image of your choice
3. Make the headline green
4. Change the behavior on the page. Right now when you click the image it says "Ouch, that hurt! Don't Poke Me" in an alert box. Instead make it so that when you click the headline, it says "Hey! You clicked the headline"

Bonus

1. When you're done with these things - discuss conceptually what you think is going on with your partner.
2. Go bananas - have some fun modifying the website.

Push your changes back to github

1. Open the github desktop app and add the files you modified to the "staging area"
2. Give the files a descriptive commit message and commit them to your local git repository.
3. Push the new commit back up to the remote repository by using the github desktop app to sync your local repository with your remote repository.
4. View the changed website online. (changes will show up instantly on github, could take a minute or two to render on your website which is hosted by "github pages".

CSS Selectors

CSS Selectors can be used among other things to:

- style the webpage
- select an element using javascript assign behavior to it
- select an element using javascript to manipulate it
- select an element on a page to scrape its data

CSS Selector Reference [https://www.w3schools.com/cssref/css_selectors.asp]

Types of Selectors

0:11 hrs

- element selector- https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Simple_selectors#Type_selectors_aka_element_selectors
- id selector - https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Simple_selectors#ID_selectors
- class selector - https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Simple_selectors#Class_selectors
- There are others, like attribute selectors

Try It

1. Navigate to my simple website: https://dmil.github.io/simple-website/
2. Click on the the "SelectorGadget" chrome extension
3. See if you can figure out selectors for the following
   - select the image on the page
   - select all quotes
   - select only the mainquote
   - select all paragraph elements
   - select only paragraph elements next to the image
   - select only paragraph elements that are not quotes

Using selectors to manipulate the DOM (Document Object Model)

Open up the "console" window in chrome. This is a console where you can write javascript code. This snippet of javascript for example uses the `#profilepic` selector to grab the image on the page, and applies logic to make it disappear.

```
$$('#profilepic')[0].style.display = "none";
```

The following will loop through

```
for (var x of $$('p')) {
    x.textContent="potato potato"
}
```

Try it

1. Make the first and second paragraphs disappear

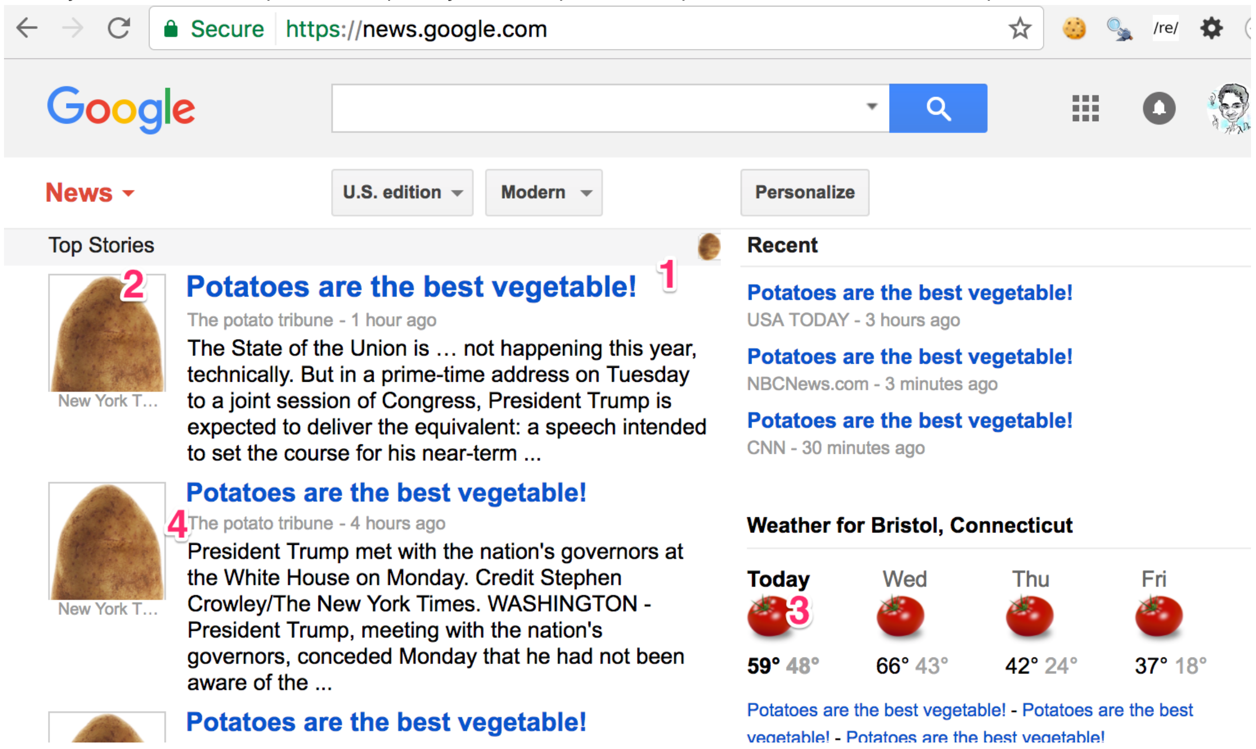
Try It

Lets see if you can make potato google news

1. modify all the headlines
2. modify all the images
3. modify only the weather images

0:11 hrs

4. Modify the name of the publication, in my case I replaced the publication name with "The potato tribune"



Potatoes are the best vegetable!
The potato tribune - 1 hour ago
The State of the Union is … not happening this year, technically. But in a prime-time address on Tuesday to a joint session of Congress, President Trump is expected to deliver the equivalent: a speech intended to set the course for his near-term ...
New York T...

Potatoes are the best vegetable!
The potato tribune - 4 hours ago
President Trump met with the nation's governors at the White House on Monday. Credit Stephen Crowley/The New York Times. WASHINGTON - President Trump, meeting with the nation's governors, conceded Monday that he had not been aware of the ...
New York T...

Potatoes are the best vegetable!

Recent

Potatoes are the best vegetable!
USA TODAY - 3 hours ago

Potatoes are the best vegetable!
NBCNews.com - 3 minutes ago

Potatoes are the best vegetable!
CNN - 30 minutes ago

Weather for Bristol, Connecticut

Today	Wed	Thu	Fri
59° 48°	66° 43°	42° 24°	37° 18°

Potatoes are the best vegetable! - Potatoes are the best vegetable! - Potatoes are the best vegetable!

Additional Resources

- Mozilla tutorials - whatever you want to learn about basic web dev, first check if there is a mozilla tutorial
- https://jsfiddle.net/ - for playing around with HTML/CSS/Javascript snippets

Part 2: Unpacking Websites & Hunting for Data

Data on the Web

Many pages on the web that serve data will do so through APIs. These apis may be internal or external. Before we talk about scraping and data collection, we're going to breifly delve into the world of APIs to better understand how a data-driven website is constructed.

CRUD

Most data-driven web applications at their core will be what programmers call "CRUD" apps.

CRUD apps perform the following functions, they allow you to Create Read Update and Delete resources.

Someone once told me "The government is essentially just a newtork of crud apps"

- https://en.wikipedia.org/wiki/Create,_read,_update_and_delete

We won't linger on these terms, but I would recommend spending some time googling these, this should get you to some good literature / forums about the basics of website architecture.

APIs

APIs are what web apps use to communicate. Just as you read a web-page and synthesize data presented on the page via its visual interface, the API is a way for the website to serve the same information in a more structured interface. Usually this interface is consumed by another program and structured either as XML or JSON format, though the trend has been towards JSON and away from XML for most simple CRUD apps.

0:11 hrs

> ... what the world might be like when our software turns to the Web just as frequently and casually as we do. Today, of course, we can see the faint, future glimmers of such a world.There is software that phones home to find out if there's an update.There is software where part of its content—the help pages, perhaps, or some kind of catalog—is streamed over the Web. There is software that sends a copy of all your work to be stored on the Web. There is software specially designed to help you navigate a certain kind of web page. There is software that consists of nothing but a certain kind of web page. There is software—the so-called "mashups"—that consists of a web page combining information from two other web pages. And there is software that, using "APIs," treats other web sites as just another part of the software infrastructure, another function it can call to get things done.
>
> Our computers are so small and the Web so great and vast that this last scenario seems like part of an inescapable trend. Why wouldn't you depend on other web sites whenever you could, making their endless information and bountiful abilities a seamless part of yours? And so, I suspect, such uses will become increasingly common until, one day, your computer is as tethered to the Web as you yourself are now.
>
> - Swartz: Introduction

How Web Apps are Structured

> The other week I made one of my rare excursions from my plushly-appointed bed and attended a local party. There I met a man who made a website for entering and visualizing data. I asked him whether he had an API, since it seemed so useful for such a data-intensive site. He didn't, he said; it would be too much work to maintain both a normal application and an API.
>
> I tell you this story because the fellow at the party was wrong, but probably in the same way that you are wrong, and I don't want you to feel bad. If even welldressed young startup founders at exclusive Williamsburg salons make this mistake, it's no grave sin.
>
> See, the mistake is, that if you design your website following the principles in this book, the API isn't a separate thing from your normal website, but a natural extension of it. All the principles we've talked about—smart URLs, GET and POST, etc.—apply equally well to web sites or APIs. The only difference is that instead of returning HTML, you'll want to return JSON instead.
>
> - Swartz (Chapter 5)

Resource

Most APIs will follow the REST convention. Rest stands for "Representational state transfer" and was famously outlined in Roy Feilding's Dissertation.

Key takeaway from knowing about REST is the concept of a resource.

> "Web resources" were first defined on the World Wide Web as documents or files identified by their URLs, but today they have a much more generic and abstract definition encompassing every thing or entity that can be identified, named, addressed or handled, in any way whatsoever, on the Web
>
> - Wikipedia artickle on REST

Lets take an example:

- http://crunchbase.com/ - a website serving data about companies and investors

Explore another example with your neighbor. What are the "resources" in this web application?

- https://beta.fec.gov/data/

Further Reading:

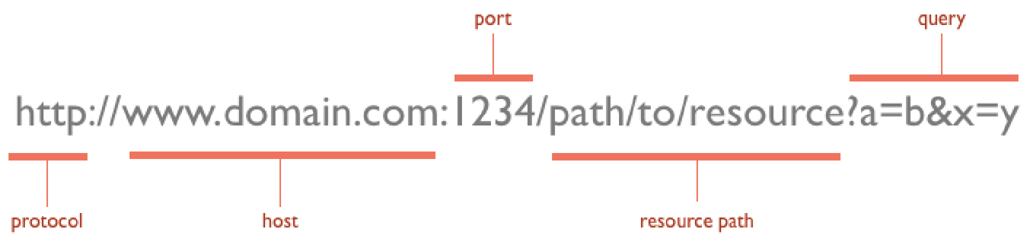
- https://www.programmableweb.com/api-university/what-are-apis-and-how-do-they-work
- https://www.thoughtworks.com/insights/blog/rest-api-design-resource-modeling

0:11 hrs

URLs (URIs)

Uniform Resource Locator (Uniform Resource Identifier)

What is a URL | Anatomy of a URL



source: https://code.tutsplus.com/tutorials/http-the-protocol-every-web-developer-must-know-part-1--net-31177

What a URL reveals about a web app

> Furthermore, URLs have to last. Those t-shirts and links and blogs will not disappear simply because you decided to reorganize your server, or move to a different operating system, or got promoted and replaced by a subordinate (or voted out of office). They will last for years and years to come, so your URLs must last with them.
>
> Moreover, URLs do not just exist as isolated entities (like “http:// example.org/lunch/bacon.html”). They combine to form patterns (“bacon.html”, “lettuce.html”, “tomato.html”). And each of these patterns finds its place in a larger path of interaction (“/”, “/lunch/”, “/lunch/bacon.html”).
>
> Because of all this, URLs cannot be some side-effect or afterthought, as many seem to wish. Designing URLs is the most important part of building a web application and has to be done first. Encoded in their design are a whole series of implicit assumptions about what your site is about, how it is structured, and how it should be used; all important and largely-unavoidable questions.
>
> - Swartz (Chapter 2:: Building for Users: Designing URLs)

More about URLs

URL Encodings (when you see strange characters in a URL, consult this table):

- https://www.w3schools.com/tags/ref_urlencode.asp

Think in terms of "Resources" if you are navigating a well built site, or especially if you're hitting a REST api (which most APIs are). For example, what are the resources on this site? What assumptions if any, can you make about how their data is structured?

Try It

1. Navigate to https://beta.fec.gov/data/advanced/
2. Observe how the URLs change across the site and discuss with the person sitting next to you. Try to map out the "resources" on their site.
3. Lets discuss as a group what this says about their site and how they store their data.
4. Together lets look at their API https://api.open.fec.gov/developers/ and how the structure of the FEC API reflects on their site, and again what this reveals about their data.

Bonus

1. Lets hit the FEC API, get some JSON, convert it to CSV using an online tool
2. Get an API key, submit the same request via postman

Why this is useful

0:11 hrs

You're steps away now from writing a program that can paginate through an API and collect all the data you need. The first step is exploring the URLs, manipulating them, and figuring out where the data you want is.

Even if you're not quite at the place where you can write the code, this will help you at least know what is possible, and a request like "paginate through these urls and give me these fields" is a really small ask. Also once you've done it once, you can do it again.

HTTP

HTTP Verbs: GET & POST

http://slides.com/dhrumilmehta/how-to-tell-a-story-with-data-tools-of-the-trade#/6/4

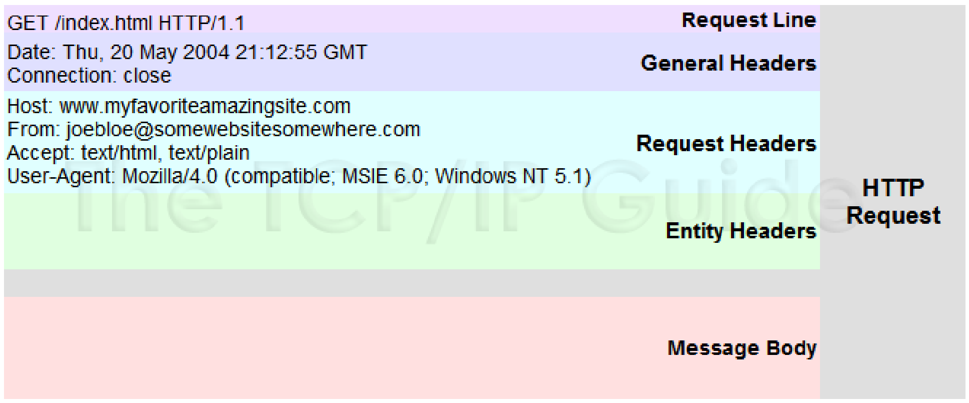
GET	POST
Requests data from a specific resource.	Submits data to be processed by a specific resource.
Data is submitted as part of the URL	Data is submitted in the request body
Less secure but faster	More secure but slower
Can be cached by browser	Not Cached by Browser
Length Limited by URL size	MaxLength determined by server

Source:

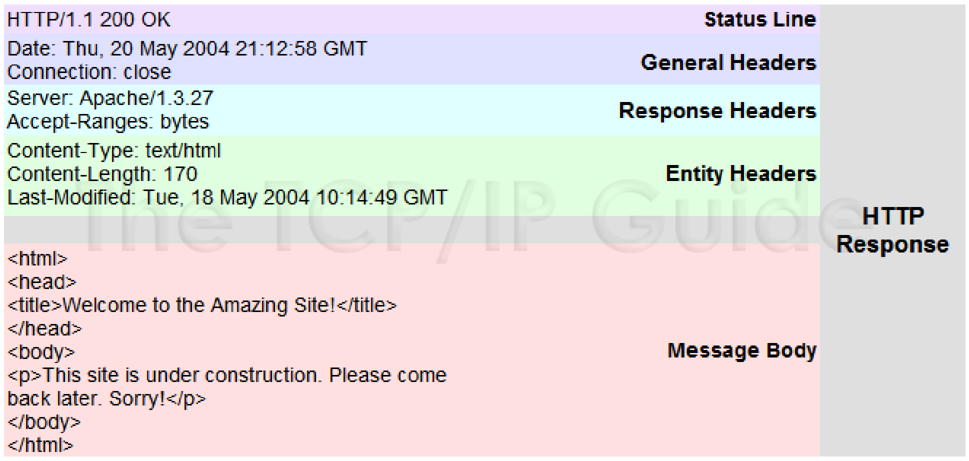
http://slides.com/dhrumilmehta/how-to-tell-a-story-with-data-tools-of-the-trade#/6/5

http://www.w3schools.com/tags/ref_httpmethods.asp

HTTP: Headers



```
GET /index.html HTTP/1.1
Date: Thu, 20 May 2004 21:12:55 GMT
Connection: close
Host: www.myfavoriteamazingsite.com
From: joebloe@somewebsitesomewhere.com
Accept: text/html, text/plain
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
```



```
HTTP/1.1 200 OK
Date: Thu, 20 May 2004 21:12:58 GMT
Connection: close
Server: Apache/1.3.27
Accept-Ranges: bytes
Content-Type: text/html
Content-Length: 170
Last-Modified: Tue, 18 May 2004 10:14:49 GMT

<html>
<head>
<title>Welcome to the Amazing Site!</title>
</head>
<body>
<p>This site is under construction. Please come
back later. Sorry!</p>
</body>
</html>
```

0:11 hrs

http://www.tcpipguide.com/free/t_HTTPResponseMessageFormat-3.htm

HTTP: Status Codes

https://www.w3schools.com/tags/ref_httpmessages.asp

https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

HTTP: Additional References

https://code.tutsplus.com/tutorials/http-the-protocol-every-web-developer-must-know-part-1--net-31177

http://httpbin.org https://requestb.in/

Try It

1. Open the postman app
2. Open http://httpbin.org in chrome
3. Lets try the following examples of GET requests together. First in chrome, then in postman
    - `/` This page.
    - `/ip` Returns Origin IP.
    - `/user-agent` Returns user-agent.
    - `/get` Returns user-agent.
    - `/response-headers?key=value` Returns given response headers.
4. Lets try the following POST requests, first on the site, then via postman
    - `/post` Returns POST data
    - `/forms/post` HTML form that submits to /post
5. Make a GET request to the following URL in postman

    https://mdn.github.io/learning-area/javascript/oojs/json/superheroes.json

    Drop the JSON into a JSON to CSV converter and download the CSV file that results. Open that file in excel

HTTP requests from the terminal with `curl` & `wget`

Print results of HTTP request body to the terminal

```
curl https://pp-projects-static.s3.amazonaws.com/congress/staffers/2016Q3-house-disburse-detail.csv
```

Save results of HTTP request body to file

```
wget https://pp-projects-static.s3.amazonaws.com/congress/staffers/2016Q3-house-disburse-detail.csv
```

Save results of several HTTP requests to several files

```
wget https://pp-projects-static.s3.amazonaws.com/congress/staffers/20{09,10,11,12,13,14,15,16}Q{1,2,3,4}-ho
```

Why this is Useful

- You can get data with curl
- If you know how to manipulate URLS, you can get data with postman

0:11 hrs

- Writing scrapers is essentially just the art of figuring out which GET and POST requests to make in which order and selecting data with CSS selectors. Figuring out those things means you've already done most of the legwork leading up to writing a scraper.

Hunting for Data

The Network Tab: Finding a Hidden API

- http://www.realclearpolitics.com/epolls/2016/president/mi/michigan_trump_vs_clinton_vs_johnson_vs_stein-6008.html
- http://www.gallup.com/poll/113980/Gallup-Daily-Obama-Job-Approval.aspx
- https://venmo.com/

Try It

Can you find the data?

- http://data.desmoinesregister.com/iowa-caucus/candidate-tracker/index.php
- http://polling.reuters.com/#poll/TR130
- https://obamawhitehouse.archives.gov/interactive-budget

Journalism from the Network Tab

http://money.cnn.com/2015/01/21/technology/security/obamacare-website-advertisers/

https://www.eff.org/deeplinks/2015/01/healthcare.gov-sends-personal-data

https://fivethirtyeight.com/features/fandango-movies-ratings/

Cleaning Data

Regular Expressions (find and replace)

- https://regexone.com/
- https://regex101.com/

Multi Line Cursor in Sublime Text

- Click & drag to select column(s): `Option⌥` + `LeftMouseBtn`
- Add other column(s) to selection by click & drag: `Option⌥` + `LeftMouseBtn`
- Add individual cursors: `Cmd ⌘` + `LeftMouseBtn`

alternatively

- select all with `Cmd` + `a`
- `Cmd ⌘` + `Shift` + `l` to get multiline cursor at the end of the line

moving the cursor

- one character a at a time: `←` and `→`
- one word at a time: `alt` + `←` and `alt` + `→`
- one line at a time: `Cmd ⌘` + `←` and `Cmd ⌘` + `→`

Try It (Together)

Clean the data using regular expressions and sublime text

0:11 hrs

Lets grab the data from Donald Trump's approval ratings on gallup:

link to website:

http://www.gallup.com/poll/201617/gallup-daily-trump-job-approval.aspx

link to data:

http://www.gallup.com/viz/v1/xml/6b9df319-97d6-4664-80e1-ea7fe4672aca/POLLFLEXCHARTVIZ/TRUMPJOBAPPR201617.aspx

and paste it into both sublime text

1. Remove all the parts of that document that do not contain the data you're looking for
2. write regular expressions to extract only the parts we want
3. use multi-line cursor or regular expressions to convert the document into a list of comma separated values (CSV) which can be opened in excel

Additional Resources

- Make a Simple Website
  - Best tutorial in my opinion is Mozilla:
    - Part 1: HTML
    - Part 2: CSS
    - Part 3: JavaScript
  - Code for my simple website to copy: github.com/dmil/simple-website
- JavaScript for Cats http://jsforcats.com/
  - Learn the basics of JavaScript using the console window in your browser. Takes only about an hour!
- Adding JS snippets to your console window.
  - `console.save` command: https://coderwall.com/p/prhwzg/add-console-save-to-chrome
  - More! http://bgrins.github.io/devtools-snippets/

Software to Know About

- Charles Proxy (Mac)
- Yesware
- WireShark
- Postman Interceptor

Recap

Part 1: Anatomy of a Webpage

- You can make simple webpages with HTML/CSS/and Javascript
  - HTML is the layout of the page
  - CSS can be applied to style the page using CSS selectors
  - JavaScript can use CSS selectors to modify the behaior of elements in the DOM (Document Object Model)
- You can store the code for those webpages and interact with other coders on github
- You can inspect websites using the chrome developer console

Part 2: Unpacking Websites & Hunting for Data

- Most apps that serve data on the web are CRUD apps
  - This involves creating, reading, updating, and deleting resources
- APIs are what programs use to communicate with each other. They frequently serve data in JSON format.

0:11 hrs

- You interact with websites using HTTP and and making GET and POST requests. APIs also communicate with one another using HTTP.
- You can hunt for data behind a website using the network tab.
- You can clean the data in a text editor like sublime text using regular expressions and the multi-line cursor

0:11 hrs