

# CODE IN ONE DAY

Fred Katona

FEWD Instructor, General Assembly

## INTRODUCTIONS

---

### INSTRUCTOR

- BS in Biology and Computer Science
- A lot of years developing web applications
- Works at DispatchHealth
- Develops healthcare related software as service products



**Let's connect!**

[linkedin.com/in/frederick-katona-9233b04](https://www.linkedin.com/in/frederick-katona-9233b04)

## INTRODUCTIONS

---

# WHO ARE YOU?

say hello!

---

## WELCOME TO YOUR CRASH COURSE

---

# What are we going to learn?!

### Talk the Talk:

- Basics of the internet & how web pages work
- HTML and CSS syntax
- Front-end code organization & project structure



### Walk the Walk:

- Modify the code of existing websites
- Building common types of websites like a landing page or marketing site
- Build a basic webpage from scratch
- How to get your website live on the internet
- Deploy a website to the internet



Talk the talk - about developing the same vocabulary by which we can explore and build

Walk the walk - apply that vocabulary and use tools to do so.

---

# CODE CRASH COURSE

---

LEARN THE BASICS

TEST THE BASICS

DO THE BASICS

---

## WELCOME TO YOUR CRASH COURSE

---

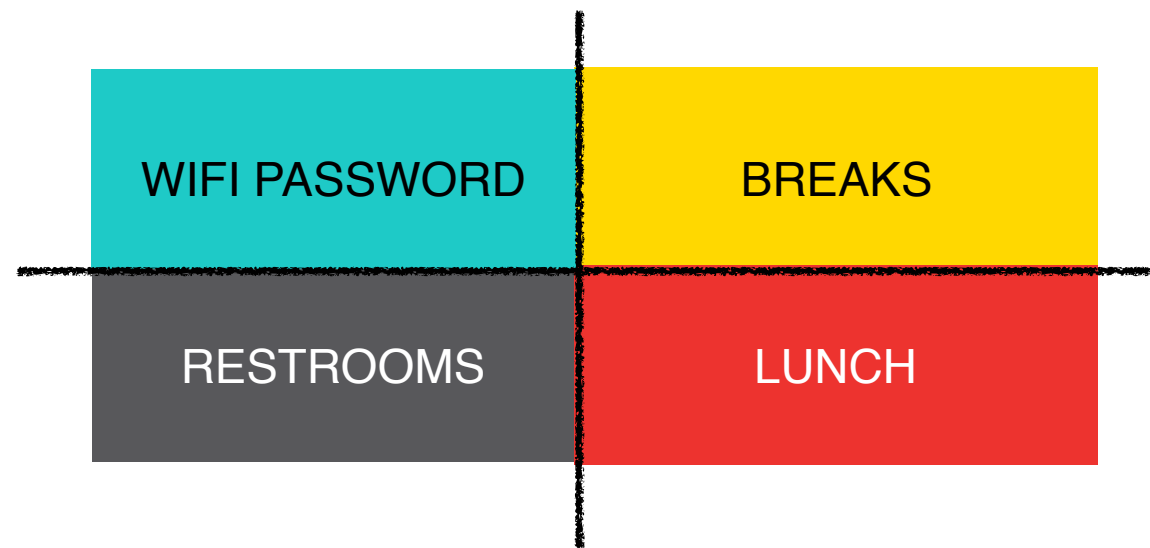
# CLASSROOM RULES

- Class participation is a must
- Keep a **positive** attitude
- Help your neighbor along
- High fives are great for celebrating
- Ask questions!

**Remember: Code is hard, that's why I'm here. Don't get frustrated, and if you need help ask a classmate or myself, I'm here to help you succeed.**

WELCOME TO YOUR CRASH COURSE

YOUR SURROUNDINGS



- Wifi password
- Restrooms
- Breaks
- Lunch

**Have you downloaded  
Sublime Text 3 & Google  
Chrome?**

**Have you made a GitHub  
account? Download github  
desktop?**

Let's take 10-20 minutes to make sure all our tools are set up.

Make sure that we have sublime text3 and github desktop installed



## DOWNLOADS

---

<https://www.sublimetext.com/3>

<https://desktop.github.com/>

make github account at  
<https://www.github.com>

---

**CODE CRASH COURSE**

---

# **INTERNET BASICS**

What is the internet?

---

## CODE CRASH COURSE

---

### Terms used

- Front-end vs. Back-end
  - Browser Basics(html, css, js)
  - Databases(backend)
- Servers
  - Requests
- Javascript
  - What makes the internet interactive

Draw front end and back end - computer and database

# An artistic picture of how the internet works

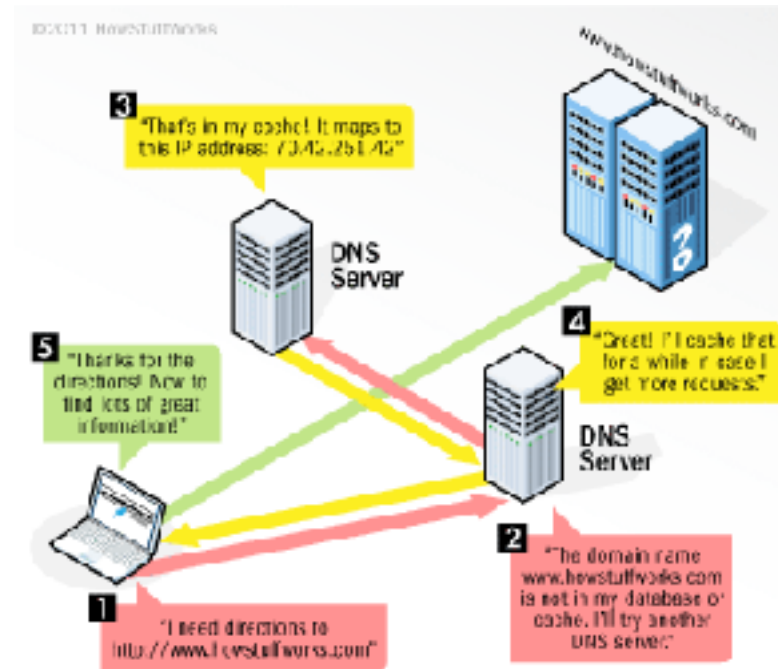
<https://github.com/den-materials/how-the-internet-works> (How does data travel through the internet? - Data Packet)

intertangling of wires that transmit data.

Seqway into data packets.

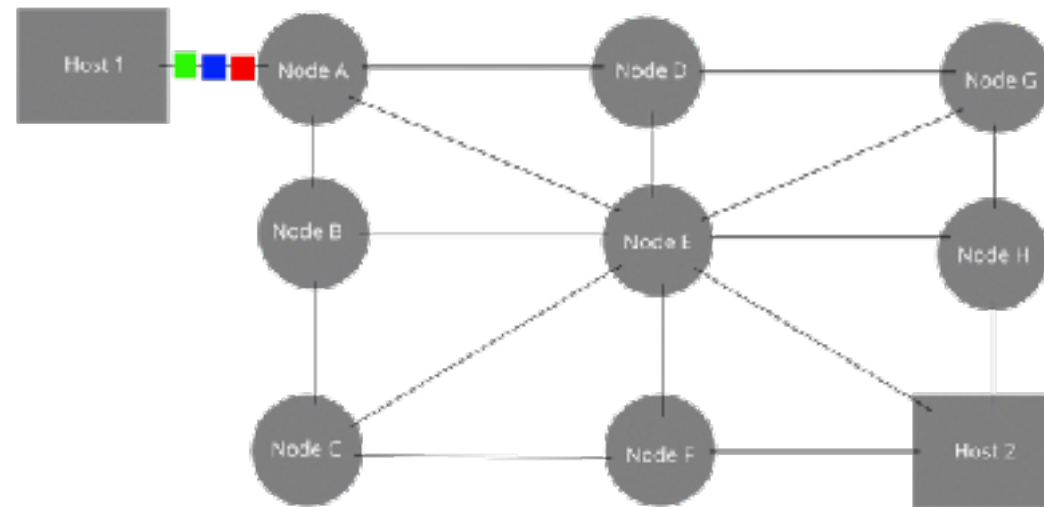
The data from servers is broken up into tiny pieces called packets. Each packet has the location where it needs to go. And as it goes along the internet towards its destination it can be broken up.

## CODE CRASH COURSE - BROWSER TO SERVER



## CODE CRASH COURSE

The original message is Green, Blue, Red.



Host 1 is our server sending pieces of data to Host 2(which is our client).

**What does your computer do with  
all these packets?**

Your browser is pretty amazing. It takes the re-ordered pieces and puts them in a file. And then it parses out each file. the html, css and javascript and renders the page viewable to you the end user.

## CODE CRASH COURSE

# Activity!

Demo how a page travels from a server to a browser

if time....create a message on piece of paper for rendering to someone else

What happens if one node on the internet is broken?

acknowledging packets

What happens if the dns server gets broken?

we can't get to twitter



---

**CODE CRASH COURSE**

---

# **HTML & CSS SYNTAX**

## **CODE CRASH COURSE**

---

▸ What is HTML?

▸ What is CSS?

## CODE CRASH COURSE

### ▸ What is HTML?

- The structure, or bones, of a webpage
- Not a programming language. Markup language.
- HTML tags, like `<div> </div>`, create a box on a page ...BUT the box needs to know how to look on a page

### ▸ What is CSS?

- What makes a webpage look pretty not like Craigslist
- Layout, colors, fonts, sizes
- Presentation

html is made of nouns(things). We call these things html elements. Our page is made up of many html elements.

CSS describes our html elements. It's like an adjective. If an element is a pizza element, then css might say "pepperoni" pizza element.

## CRASH COURSE

---

### Analogy to nouns, adjectives, and verbs

An Html page is made of things. We call these things html elements. We can say these are nouns.

CSS describes our html elements. It's like an adjective.

If an element is a pizza element, then css might say “pepperoni” pizza element.

Javascript is our verb. It makes the entire structure come together by performing actions. One action might be “validate” a form.

## CODE CRASH COURSE

---

### Most Basic Syntax

*<tagname>content</tagname>*

content is marked up or surrounded by tag names that give it meaning

Tags usually come in pairs

We said earlier that html was about markup. What does that mean?

We markup our content

---

## CODE CRASH COURSE

---

# HTML example

**every webpage is made up of HTML!**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

Where/What is the head?

Where/What is the body?

Can we guess what this will look like in the browser?

---

## CODE CRASH COURSE

---

# HTML

hyper text markup language

### Elements

- `<div></div>`
- `<p></p>`
- `<br>*`
- `<title></title>`
- `<body></body>`
- `<h1></h1>`
- `<img>*`
- `<a href="destination.com">LINK</a>`

NOTE: Every tag that is opened, needs to be closed with '/' in the second tag - unless it is self-closing(noted here with \*)

Some basic html elements. This is a non-exhaustive list

---

## CODE CRASH COURSE

---

# HTML files

- The stuff on the web is like the stuff on your computer, just a bunch of files
- We can make `.html` files and put them on any browser to view
- `index.html` is the root file of most pages

NOTE - In *general*, the main page of most html sites is in a file called `index.html`.



## CODE CRASH COURSE - HELLO WORLD

- Make a new folder **on your Desktop** called `CodeInOneDay`
- Open this folder in Sublime Text - drag fold into sublime text
- Create a new file, with command-n on mac
- Save the file as index.html
- Type in Hello World and save the file

PLEASE be careful where you make this folder.

Will we be able to see “Hello World”

-Goal, to tie editing file and viewing in browser

## CODE CRASH COURSE - HELLO WORLD

---

- Let's clear that out of the file and then...
- Type `<h` and then press `tab` key
- Write `<h1>Hello World</h1>` inside the `<body>` tag
- Let's add a title together too
- And let's take a look at the file in our browser, right-click on the page and choose "view in browser"

Sublime text has this great shortcut for filling out our page scaffold.

---

## CODE CRASH COURSE

---

# HTML ACTIVITY - using only html

▸ How would you make Hello World smaller than what we have it now?

(Right now it is an h1 tag)

▸ How would you italicize Hello World?

▸ How would you make “Hello” bold?

Google is your best friend.

let's try do all these operations in html only.

Hierarchy of tags. (H1 is going to be bigger than H2).

any problems with bold? use “strong” for bold. it add semantic meaning the page

---

## CODE CRASH COURSE

---

*Thought Experiment.*

**What would a hypothetical html element look like around song lyrics?**

This one is tough...think very abstractly

## CODE CRASH COURSE

---

*Thought Experiment.*

What would a hypothetical html element look like around song lyrics?

`<lyric>These are the lyrics to the best song ever</lyric>`

---

## CODE CRASH COURSE

---

Let's explore and add more html for reference:

Let's add more headings together - h1 through h6

Let's add a paragraph and add some text

Let's also put a break in here.

Let's also add an unordered list to our page, with items in it.

show shortcut for dummy text

---

## CODE CRASH COURSE

---

Why don't elements appear next to each other?

---

## CODE CRASH COURSE

---

Why don't elements appear next to each other?

Block vs Inline elements

### Block Elements

- start on a new line
- start on the left and take up the whole width
- include: <div>, <h1>-<h6>, <p>, <form>, <li>

### Inline Elements

- do not start on a new line
- expand out till they're done
- include: <span>, <img>, <a>



## CODE CRASH COURSE

### HTML Activity

Let's add inline elements to our page to see how they work.

And finally we'll add a link that opens in a new tab.

What inline elements did you add in the earlier exercise?

I think we made text strong?

We can also add an image

lorempixel

target="\_blank"

## CODE CRASH COURSE

---

HTML Tag

`<tagname>content</tagname>`

HTML Tag with more information

`<tagname attributename="attributevalue">content</tagname>`

We can add other attributes to html tags for increased meaning and styling(preparing for css)

- All tags can have attributes
- Placed in the starting tag of element
- key/value pair (key="value")

**`<h1 class="title">Welcome</h1>`**

right now the attributes just seem like extra. It seems like only an "a" tag element has an attribute that merits any value, but when we dive into css you can see where they come in handy.

## **CODE CRASH COURSE - REVIEW/ADVANCED**

---

Let's apply what we learned and search for new tags to use:

navigate in your browser to:

<https://github.com/bigkat73/codeinaday>

hit the clone or download button. download the zip file.

Drag this folder into sublime text and open the starter\_code/  
press\_release/press\_release.html

Let's examine and add html to the content in this file.

html tags to this content

Did you discover any new elements?

What is interesting about these new elements.

---

## CODE CRASH COURSE - HTML5 DEEP TOPIC

---

HTML is great for adding structure

HTML5 introduced meaning for the structure

What exactly does this mean?

## HTML5 SEMANTIC TAGS

If we look at these tags, they describe the content they wrap(not exhaustive list)

<header>

<footer>

<section>

<article>

<nav>

header would wrap the header of a page or content

nav is for navigation

article might be used for online publishing or a blog

section is a little more generic version of this

footer would be a footer to a page or content

## HTML SEMANTIC TAGS

Let's go back to our first index.html page

Together let's add where we might add semantic html5 tags to our content.

header

articles around content

The big thing is to not overthink using html5 tags. They are great and give more meaning to the content. But, it's best to stick to the more used ones that make sense for your content.

There's nothing wrong with a generic old div.

---

## CODE CRASH COURSE - FOR SELF STUDY

---

Other elements for you to check out.

Button

Lists - ordered and unordered list

Forms and form entry elements(inputs, selects)

---

## CODE CRASH COURSE

---

Let's take a minute and review what we just did.

We took content. Added html structure around that content to add a meaningful structure around it.



---

## CODE CRASH COURSE

---

Let's take a minute and review what we just did.

We took content. Added html structure around that content to add a meaningful structure around it.

Our browsers gave it some basic styling given the meaning of the content. But we want to make it look a lot better.

## CODE CRASH COURSE

---

Let's take a minute and review what we just did.

We took content. Added html structure around that content to add a meaningful structure around it.

Our browsers gave it some basic styling given the meaning of the content. But we want to make it look a lot better.

Let's add some style. Specifically, some style sheets.

---

## CODE CRASH COURSE

---

# CSS

### cascading style sheets

```
body {  
    background-color: red;  
}  
  
h1 {  
    color: white;  
    text-align: center;  
}  
  
p {  
    font-family: verdana;  
    font-size: 20px;  
}
```

example css - can we predict what it's doing?

---

## CODE CRASH COURSE

---

# Why CSS?

History Lesson -



css used be less important. people used to embed style into the html itself.

So?

example css - can we predict what it's doing?

---

## CODE CRASH COURSE

---

# Why CSS?

History Lesson -



css used be less important. people used to embed style into the html itself.

So?

Html should only describe the content. Not include style.

CSS was created to take styling out of the html.

It also let's you define styles across multiple pages all at once.

There's also some cool new features in css like animations.

History lesson 2. Back when css first was created, the popular method was adding inline css. Which adds styling directly to each element. What everyone learned pretty quickly is that this was hard to maintain. So, please don't do it.

# CSS

---

## Cascading Style Sheets

- Also, not a programming language
- Used for layout and styling

Standard way to add styles is by linking external stylesheets to our html document.

## CODE CRASH COURSE

# CSS

- Your selector is the HTML tag you want to manipulate
- For each selector, there is a property
- Every property has a **value**

selector      declaration

p {font-size: 12px}

property      value

CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

---

## CODE CRASH COURSE

---

# CSS

```
selector {  
    property: value;  
    property: value;  
}
```

generic version. Styles multiple properties for a given selector.



## CODE CRASH COURSE



A word about CSS - I love it, but it is difficult. It's more like poetry than prose

## CODE CRASH COURSE

### html:

```
<h1>Hello World!</h1>
```

### CSS:

```
h1 {  
  color: red;  
}
```

What is the css selecting?

What property is it updating in the html.

Not only can a selector select by element name, it can select elements using classes and id's.

Id's should be unique for each html page. classes can be used multiple times.

---

## CODE CRASH COURSE

---

### 1). In Sublime Text:

create styles.css using command-n

### 2). In HTML:

```
<link href='styles.css' rel='stylesheet'>
```

### 3). In Sublime styles.css:

enter in a css selector, property, and value!

**Note: Don't forget to save every sublime file (Cmd + S)!**

Using index.html example that we used earlier.

---

## CODE CRASH COURSE

---

# CSS ACTIVITY

- How would you make Hello World purple?
- How would you make the background of Hello World yellow?
- How would you make Hello World appear on the right hand side of your browser window?

**Remember: Google is your BFF**

---

## CODE CRASH COURSE

---

# CSS ACTIVITY

- Let's work in groups of two to update our previous press release.
- Let's link a stylesheet to it and change 3 styles using css.
- Let's share what we've changed and any other discoveries

Go over how to select by element type  
class  
id  
every html element being a box

## CSS

properties we can change(non-exhaustive list)

color(font)

background-color

font-family

font-size

font-style

font-weight

line-height

text-decoration

text-transform

letter-spacing

word-spacing

Let's change properties in our hello world exercise.

width

height

Apply styles to different elements.

Show how we can reuse selectors.

let's find default web fonts - google

## CSS EXTRA TRICKS

---

Select elements by a class name

```
.topline{ font-size: 32px }
```

```
<header class="topline">
```

```
....
```

```
</header>
```

Select elements by id

```
#left-thing{ font-size: 20px }
```

```
<div id="left-thing">
```

```
....
```

```
</div>
```

example for reference

## CSS CRASH COURSE

---

So we added style to our html...but what about moving things around the page, you know, layout.

layout is where things get tricky. It helps to have a layout page without content to envision where all our boxes go.



## CSS EXTRA TRICKS

How do we center a div?

How can we add a border?

Let's make the corners rounded.

let's style our links

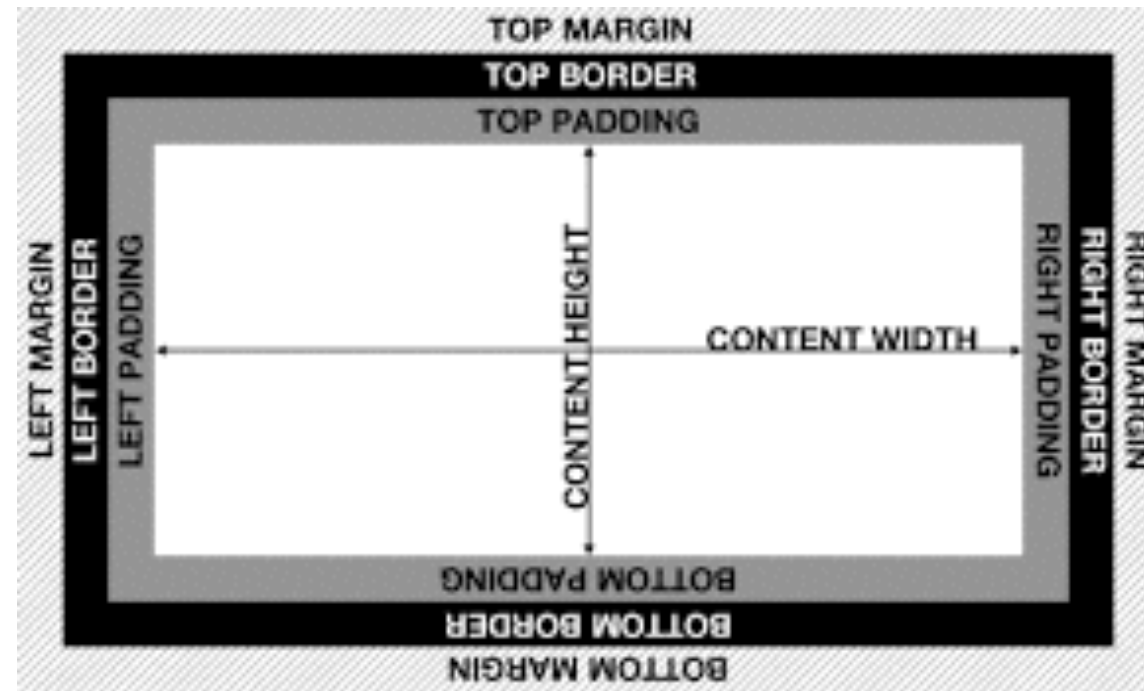
let's create a holder div for this in our reference

set a width for a block(div) element

add margin:auto

## CSS LAYOUT

### Box Model



## CSS BOX MODEL DEMO

Demo box model in hello world example.

We will use this idea of boxes to layout our page later today.

Show flattened boxes with no content.

Put two boxes side by side

Up next review and have fun with a tool that can help you learn more advanced selectors.

**CODE CRASH COURSE**

**GAME TIME!**

<http://flukeout.github.io/>

limit to 10 minutes - walk through the first few.

Flukeout is great place to practice selectors and learn new ones. Some people don't like that this is an abstraction of a page where animated items are what we are trying to select...so if you find it strange...disregard.

---

**CODE CRASH COURSE**

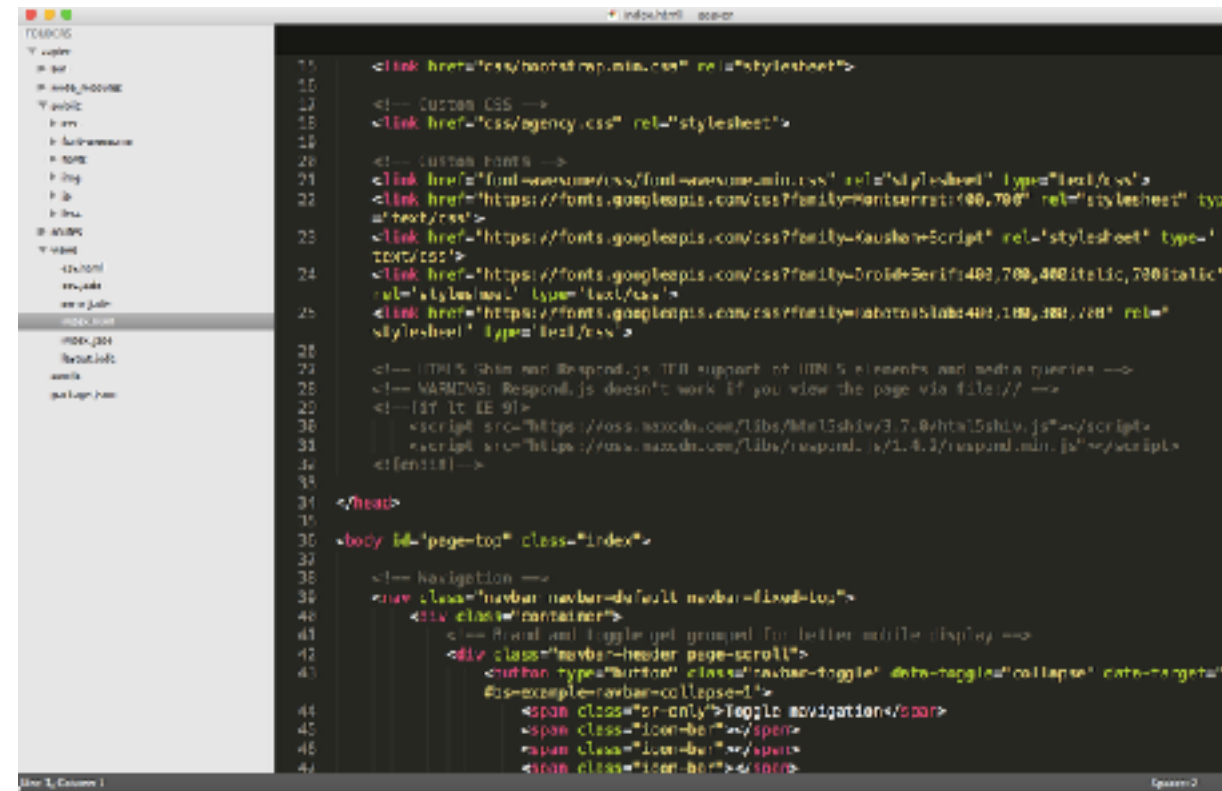
---

**LUNCH TIME!**

Be back in 60 minutes!

# **CODE ORGANIZATION & PROJECT STRUCTURE**

# CODE CRASH COURSE



```
15 <link href="css/hotstart.min.css" rel="stylesheet">
16
17 <!-- Custom CSS -->
18 <link href="css/agency.css" rel="stylesheet">
19
20 <!-- Custom Fonts -->
21 <link href="font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
22 <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet" type="text/css">
23 <link href="https://fonts.googleapis.com/css?family=Kausan+Script" rel="stylesheet" type="text/css">
24 <link href="https://fonts.googleapis.com/css?family=Roboto+Slab:400,700,400italic,700italic" rel="stylesheet" type="text/css">
25 <link href="https://fonts.googleapis.com/css?family=Roboto+Slab:400,700,400italic,700italic" rel="stylesheet" type="text/css">
26
27 <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
28 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
29 <!--[if lt IE 9]>
30 <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
31 <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
32 <![endif]>
33
34 </head>
35
36 <body id="page-top" class="index">
37
38 <!-- Navigation -->
39 <nav class="navbar navbar-default navbar-fixed-top">
40 <div class="container">
41 <!-- Brand and toggle get grouped for better mobile display -->
42 <div class="navbar-header page-scroll">
43 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">
44 <span class="sr-only">Toggle navigation</span>
45 <span class="icon-bar"></span>
46 <span class="icon-bar"></span>
47 <span class="icon-bar"></span>
```

example project - it's pretty complicated

## CODE CRASH COURSE

- Every project is a series of folders
- Inside those folders are files
- Each file contains code that gets executed
- Every web app has different project structure
- Folders for CSS and HTML and Javascript
- You want organization so other developers can find things easily

refer to picture of internet again.

Diagram that requests get made to server and are directed to web server that serves up these files.

You also want to organize the files because web servers also expect files in certain places.



---

## CODE CRASH COURSE

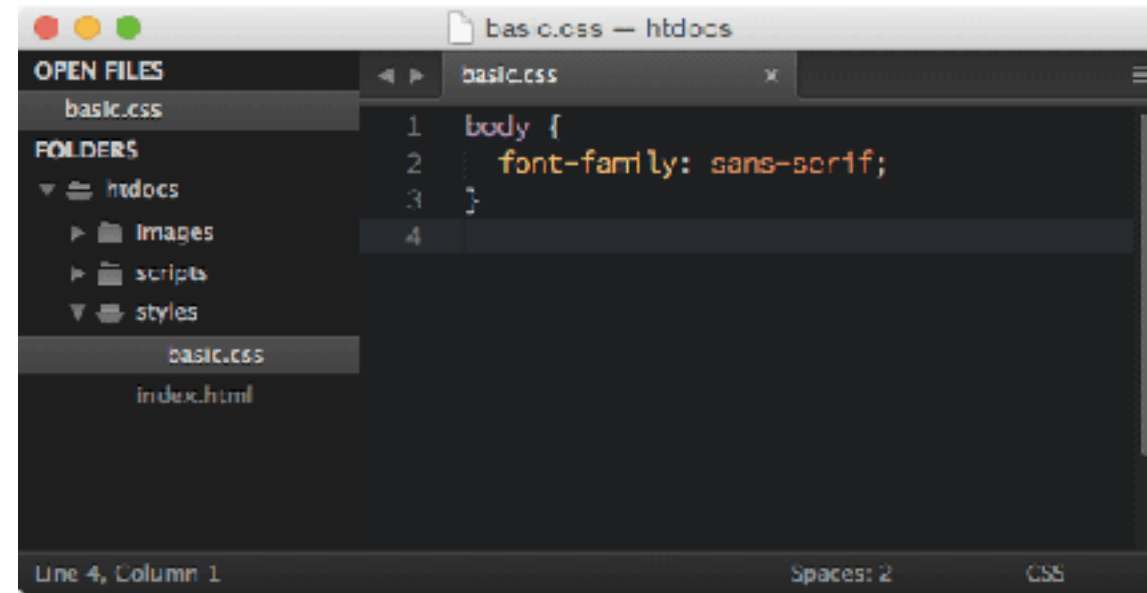
---

# FRONT END ORGANIZATION

```
|-- app(root)
|  |-- assets
|  |  |-- img
|  |  |-- js
|  |  |-- `-- app.js
|  |  |-- `-- css
|  |      |-- `-- style.css
|  index.html
```

a javascript application will have a structure that looks a lot like this one.

## CODE CRASH COURSE



an html, css, javascript site will look a lot like this. We want to use this structure for our websites

When we talked about servers just serving up files. Which file is sent if the main url is hit here.

---

## CODE CRASH COURSE

---

### ACTIVITY TIME

Review the file structure in `/extras/example_file_structure`

And then Let's organize our existing files into this new structure

walk through

why do we need the structure to look like this?

what happens if I rename the `index.html` to `fred.html`?

**CODE CRASH COURSE**

---

# **MODIFY CODE OF EXISTING WEBSITES**

We want to explore modifying existing website code to demonstrate tools we can use to understand how our own code is running.

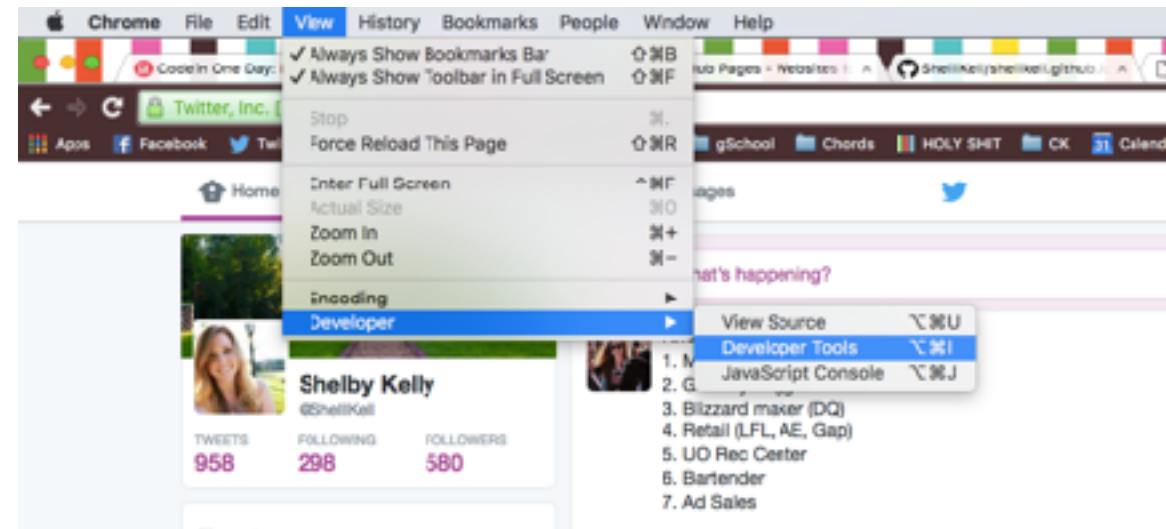
---

**CODE CRASH COURSE**

---

**Google Developer Tools!**

# CODE CRASH COURSE

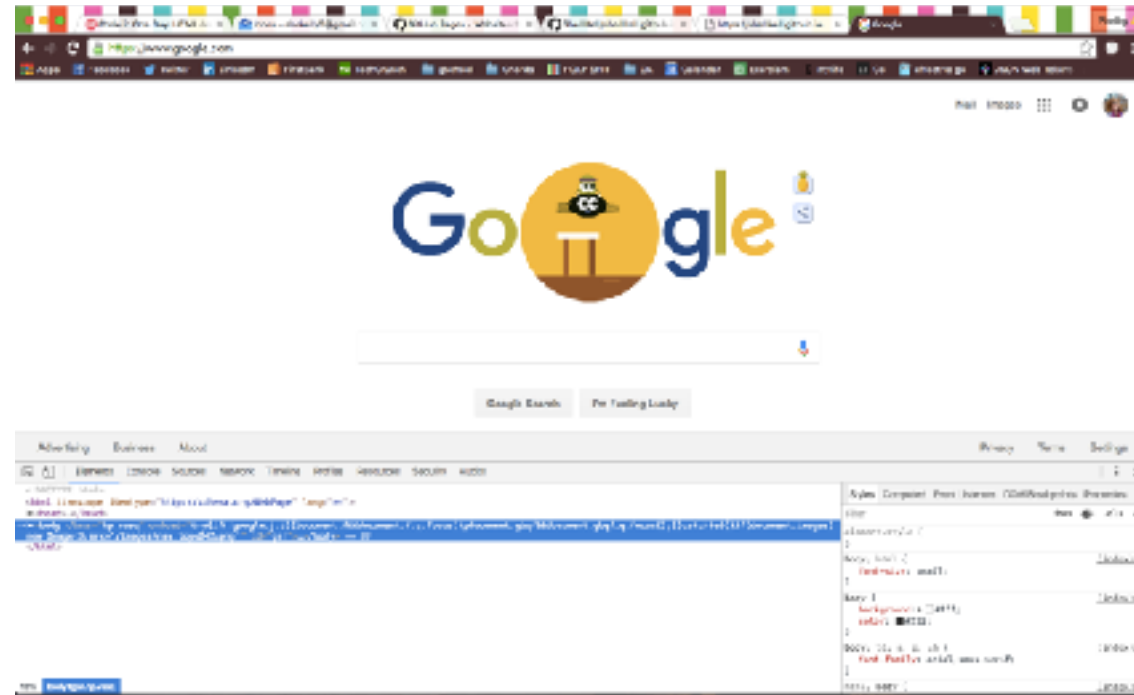


most browsers have built in dev tools

show firefox and safari too

## CODE CRASH COURSE

# Shortcut: command + option + j



## CODE CRASH COURSE

- Developer tools are great for inspecting a webpage
- Right click any element to inspect it
- You can change HTML and CSS on any live website to see what the changes would look like
- There are other tabs that are helpful for web development - but today we are going to focus on the ELEMENTS tab

Developer tools is a great way to debug not only our html, but our css as well.



---

## CODE CRASH COURSE

---

**Someone's favorite website?**

Demonstrate

## **Modify your favorite website**

- At least 3 changes
- Raise your hand when you're finished

I was able to see the css and why it was displayed from in my favorite website under the styles tab.

---

**CODE CRASH COURSE**

---

# **GETTING A SITE LIVE ON THE INTERNET**

we will be using our current codeinaday folder with our references

## CODE CRASH COURSE

# Downloading GIT:

- <https://desktop.github.com/>
- open github desktop, under menu-preferences-install command line tool



github desktop is a nice gui tool

---

## CODE CRASH COURSE

---

# Git

- Git is a source control tool that allows multiple developers to work on the same project at the same time, while also saving backups of previous versions of your site.
- Git is used in the command line, and is a tool that interacts with GitHub.com (they are not the same thing)
- Git is an industry standard for most web agencies

Git is basically google docs for writing software

---

## CODE CRASH COURSE

---

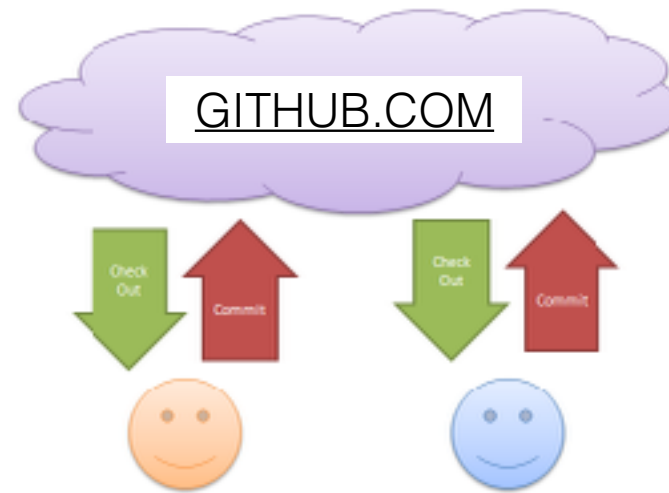
# GitHub

- When you write code locally on your machine, you can save backups using just git
- You can then 'push' that code to [GitHub.com](https://github.com), where it can be viewed and manipulated by other users
- Your code will live on GitHub, where it will be saved and stored
- When people say 'Open source', they are talking about things like GitHub

---

## CODE CRASH COURSE

---



---

## CODE CRASH COURSE

---

# GitHub Pages

- Allows you to turn your github.com site into a live, outward-facing website
- Instructions can be found here: <https://pages.github.com/>

NOTE: There are MANY options to deploy websites to the internet. This is one of the simpler ways for our needs today.

Walk through making this site GH-pages - it's easier than the link provided



---

## CODE CRASH COURSE

---

# Overview

git is hard \\_(ツ)\_/

- Git is hard, and will take some practice to become comfortable
- Today is more about using it to get to our end goal, not to master it
- Github WILL make working in groups on software much easier, but also more complex

**DEPLOY!**

---

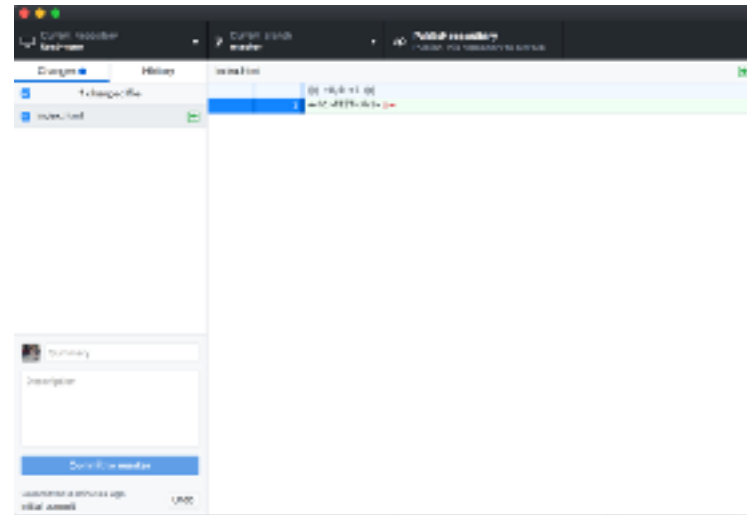
## CODE CRASH COURSE

---

1. open github desktop
2. Choose Add Local Repository
3. When it doesn't find the repo...click create repo here
4. And then publish

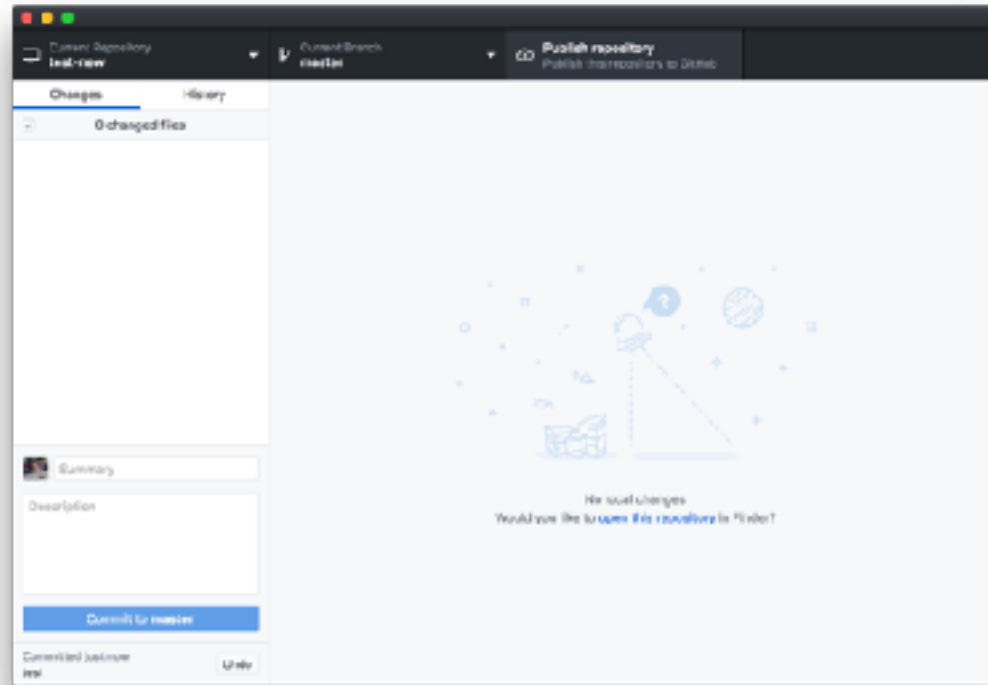
## CODE CRASH COURSE

6. commit to master with a message



## CODE CRASH COURSE

7. publish repository -  
check github to see repo

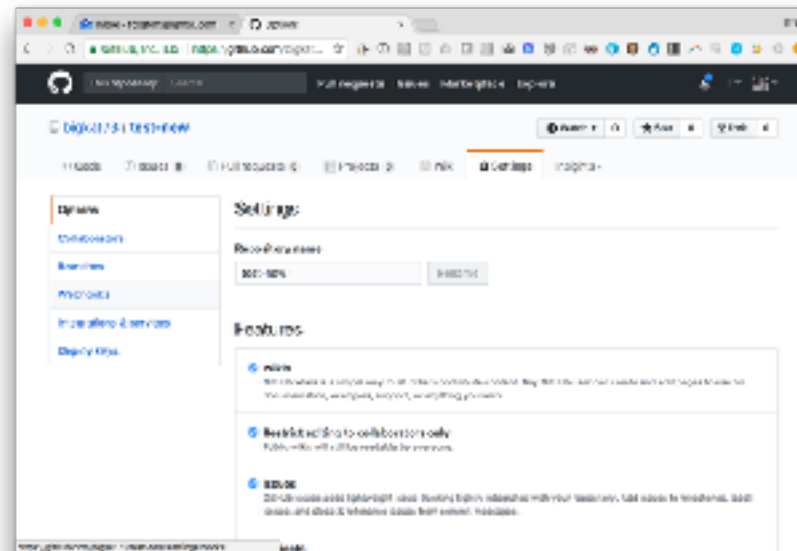
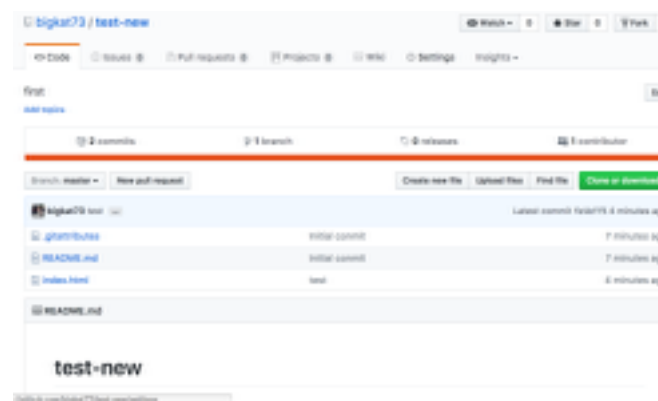


---

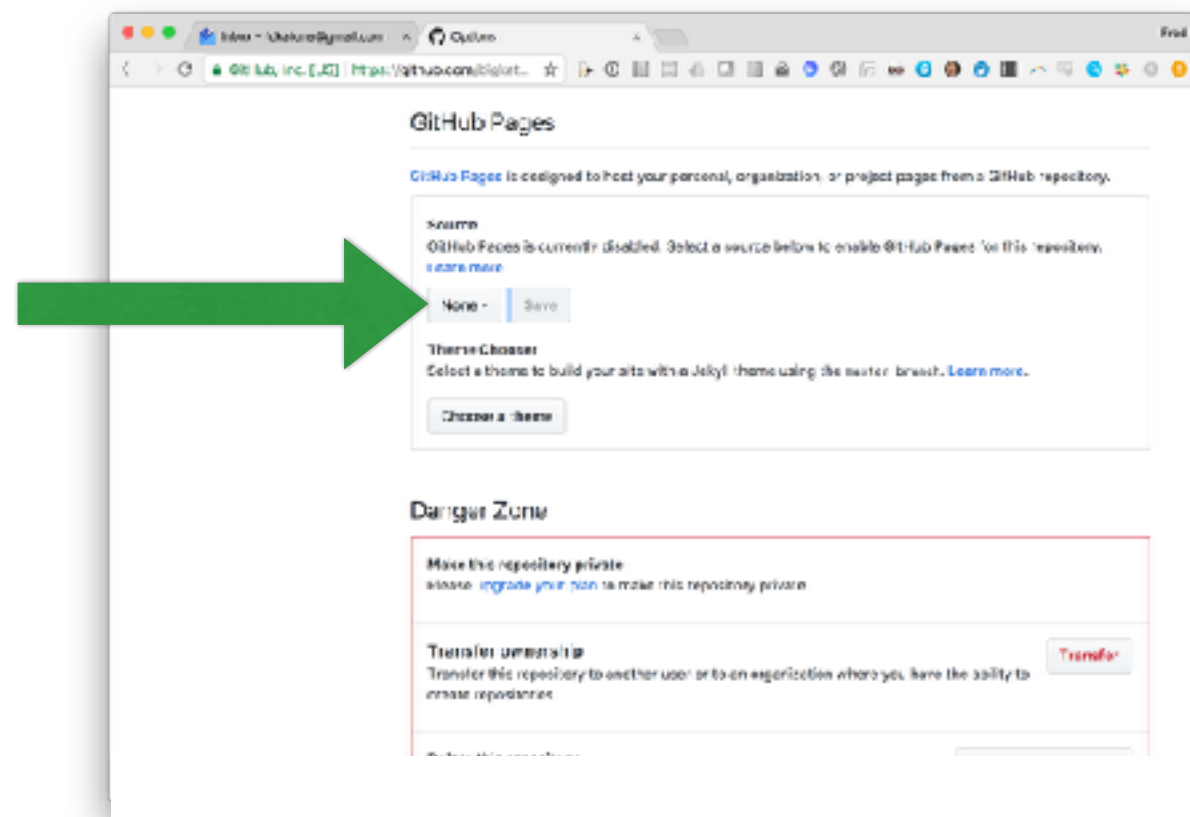
## CODE CRASH COURSE

---

8. In github online, check your repo.
9. Go into settings tab
10. Drill down to github pages
11. Set repo to master



very important to have index.html in the root folder here.





---

## CODE CRASH COURSE

---

Voila! Should be able to see your code hosted on the internets.

---

**CODE CRASH COURSE**

---

**BREAK TIME!**

Be back in 10 minutes!

---

**CODE CRASH COURSE**

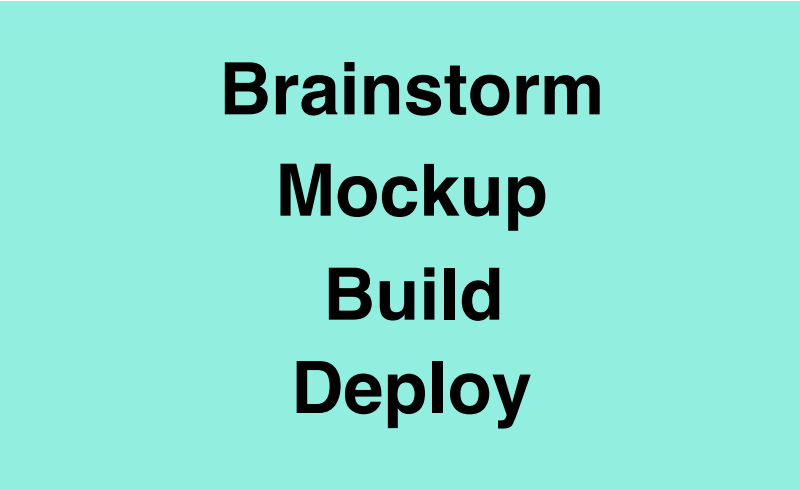
---

# **BUILD A WEBPAGE**

---

## CODE CRASH COURSE

---



**Brainstorm**  
**Mockup**  
**Build**  
**Deploy**

# Static Sites vs. Web Apps

**Static sites** are splash pages, landing pages

- They don't have forms to fill out, logins, etc.

**Web apps** have places for users to interact with the software

- Fill out forms, sign up, login, etc.

**We are focusing on building static sites today - not web apps**

# **Class exercise:**

## **Building a website together**

- What should our site be for?
- Colors? Pictures? Headers? Paragraphs?
- How do we see our site in my browser?

review the folder structure

Demo where the files will be found and what the first file is that will load.

Demo how to see in our browser

## WHAT WE WANT TO BUILD



this is responsive too.

## CODE CRASH COURSE

---

- Compare starter coder with mockup in the images directory
- What changes do we need to make
- Let's walk through all the changes that we need to make
- Starter\_code/landing\_page

let's copy this code to the desktop in it's own folder



---

**CODE CRASH COURSE**

---

**BREAK TIME!**

Be back in 10 minutes!

**CODE CRASH COURSE**

---

# **BUILD A WEBPAGE FROM SCRATCH**

## 5 Minute Brainstorming Session

- What kind of site do you want to build?
- What colors do you want to utilize?

*(psst . . . try [color.adobe.com](http://color.adobe.com))*

- Do you want to use pictures? etc.

*(psst . . . right click any image and choose 'Copy image address' to get it's path, then paste it into an img tag src!)*

# **10 Minute Mockup Session**

- Using paper or the whiteboards, design mockups
- Make a few different options

---

## CODE CRASH COURSE

---

**start with your project folder and  
github repository**

---

## CODE CRASH COURSE

---

**add file structure layout to the  
empty folder**

---

## CODE CRASH COURSE

---

# index.html

- Add `<!DOCTYPE>` on the very first line of the page
- Type `html` into a blank page, then hit tab
- This will give you the skeleton of your HTML
- We will need to include the CSS file in our HTML file:  
`<link href='styles.css' rel='stylesheet'>`

---

## CODE CRASH COURSE

---

# HTML

Remember that every webpage is made up of HTML!

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```



---

## CODE CRASH COURSE

---

# styles.css

- Add CSS to a file and manipulate the HTML you have
- Try googling different styles to apply to your HTML

---

## CODE CRASH COURSE

---

# CSS

### cascading style sheets

```
body {  
    background-color: red;  
}  
  
h1 {  
    color: white;  
    text-align: center;  
}  
  
p {  
    font-family: verdana;  
    font-size: 20px;  
}
```

# **30 Minute Build Session**

- Using Sublime Text & Terminal
- Refresh the page to see your changes  
(shortcut: `command + r`)

**Ask questions of your neighbors and me!**

---

**CODE CRASH COURSE**

---

**BREAK TIME!**

Be back in 10 minutes!

**DEPLOY!**

# SHOW & TELL

(if there's time)



---

**CODE CRASH COURSE**

---

# **WRAP UP - Q/A**

---

**THANKS!**

---

[https://www.linkedin.com/in/frederick-  
katona-9233b04/](https://www.linkedin.com/in/frederick-katona-9233b04/)



## **FURTHER RESOURCES**

---

- <https://dash.generalassemb.ly/>
- <https://www.codecademy.com/>
- <https://teamtreehouse.com>
- <http://codepen.io/>
- <https://generalassemb.ly/locations/denver>