WELCOME TO FRONT-END BOOTCAMP!

INTRODUCTION

JESEEKIA VAUGHN

jeseekia@grandcircus.co | @metadevgirl | jeseekia

INTRODUCTION

DAVID WOLVERTON

david@grandcircus.co | @dwolverton | dwolverton

INTRODUCTION

ADAM DAVID

adam@grandcircus.co | @adamAtGC

COURSE EXPECTATIONS

What we expect from you...

BE PRESENT

Listen, soak up the information (there's a lot of it), process it.

ASK QUESTIONS

I'm not big on formality. Feel free to ask any question that comes to mind. I'll also stop frequently for questions.

SUPPORT EACH OTHER

I'm a big proponent of pair programming. Any in-class exercise or project work may be paired and certain labs *will* be paired or in groups. Larger final projects will be group projects.

DO THE WORK

There's really only one way to learn to code: Write code! We expect you to turn in every assignment complete and on time.

HAVE FUN

See previous comment on my stance on formality and ceremony. Bored people learn nothing. I'll do what I can to make this fun as well as instructive.

OUR GOALS FOR YOU

OUR GOALS FOR YOU

Inculcate you with vital tools such as version control, specifically git and github.

OUR GOALS FOR YOU

Initiate you into the world of JavaScript.

OUR GOALS FOR YOU

Learn the basics of web and mobile architecture by seeing it in practice.

COURSE OUTLINE

(What are we doing here anyway?)

'THE PLAN'

- Week 1 - HTML / CSS / JavaScript
- Week 2 - JavaScript
- Week 3 - jQuery / AJAX / JSON
- Week 4 - Angular JS / TDD
- Week 5 - Angular JS / Node JS
- Week 6 - Express / SQL
- Week 7 - Project
- Week 8 - Project / Demo Day

THE PLAN

- Each week will be split among a number of lectures focused around a weekly topic.
- There will be practical exercises every day.

PRACTICAL WORK

The practical work is divided into two categories

- *Exercises* are short & focused to practice a specific topic or technique
- *Labs* are broader in scope and incorporate more of the material into a single problem.

YOUR NEW JOB

YOUR NEW JOB

As of right now, this class is your new job and I am your boss. The hours are 9am - 5pm Monday through Friday with some work in the evenings and weekends required, and as much optional work as you can stomach.

YOUR NEW JOB

Your primary job responsibility is to

FIGURE IT OUT

YOUR NEW JOB

Developers are essentially well-paid problem solvers. If you come across something you're not sure about, you need to make an effort to solve the issue yourself.

YOUR NEW JOB

A checklist for when you get stuck:

1. *First*, ask google.
2. *Second*, ask a fellow bootcamper.
3. *Third*, ask another fellow bootcamper.
4. If you're *still* stumped, ask Instructor or the TA.

YOUR NEW JOB

In the class GitHub repo, we have provided a debugging checklist that you can use to diagnose common problems you may run into with your code.

If you come to us with a problem, we'll ask you what you've tried on that list.

YOUR NEW JOB

The rule about the checklist:

Don't be a hero and spin your wheels for hours without any progress. If you're genuinely stuck, ask for help and keep asking until you get it.

YOUR NEW JOB

One final thing...

Because we want you to learn how to diagnose problems, we are going to eventually pull back from helping in certain ways as the course progresses.

YOUR NEW JOB

Starting at the end of week 2 if your console is displaying syntax errors or files not found (such as scripts, CSS sheets, images not loading), we will ask you to run through your code where errors are found and fix it.

YOUR NEW JOB

Most importantly

PRACTICE!!

We won't always have homework for this class, but any extra time you spend working on this stuff will only deepen your understanding of the material. If you finish an assignment early, look for ways to go a step further. We have *loads* of resources for extra work.

YOU WILL GET OUT OF THIS WHAT YOU PUT INTO IT

GETTING STARTED

Note that some of this should already be done. This is mostly a checklist of things we will need for smooth(ish) sailing in this course.

GET CONNECTED

wifi: Madison

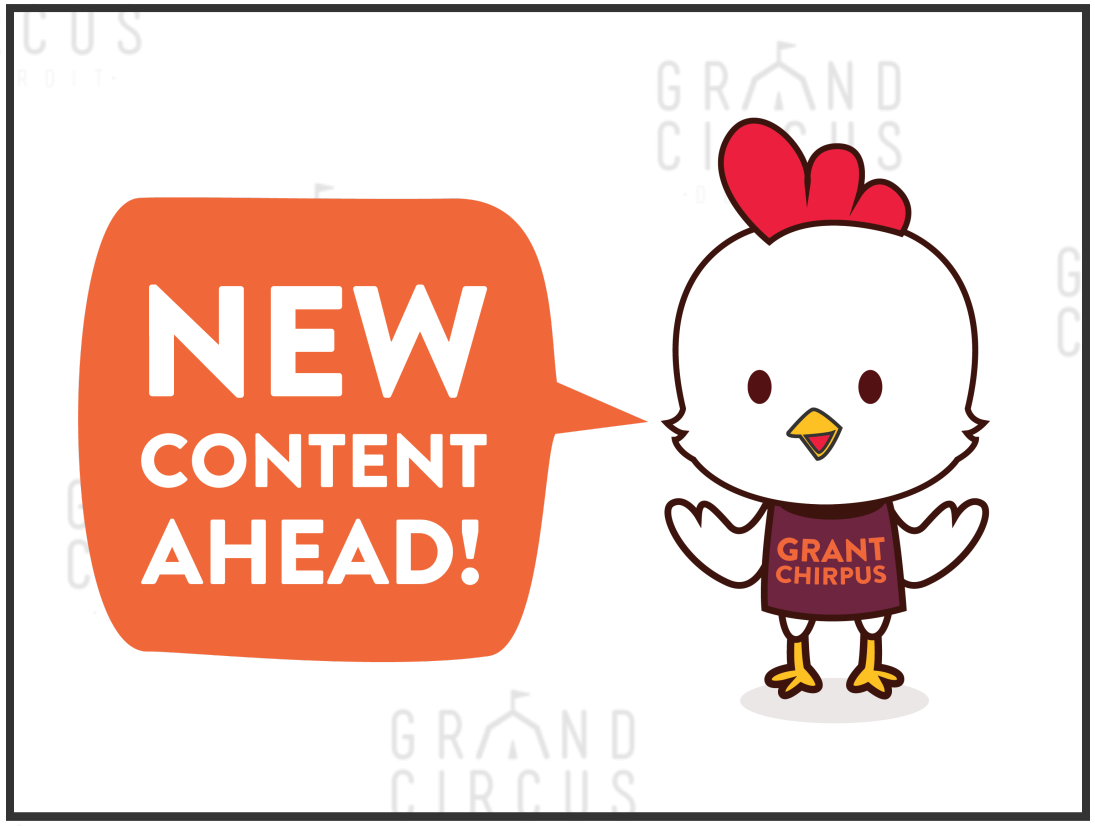
pw: Bro@dw@y#

CHECKLIST

- Google Chrome
- Sublime Text / Atom.io
- Slack
- Github / git
- NodeJS

QUESTIONS?

OUR CLASS AGILE BOARD


NEW
CONTENT
AHEAD!
GRANT
CHIRPUS

GOALS FOR THIS UNIT

1. The Website as Communication
2. Introduction to HTML & CSS
3. HTML 101
4. **Exercise:** Build a static HTML site from scratch

ASSOCIATED READING

From HTML & CSS:

- Introduction: 2-11
- Chapter 1: 12-39
- Chapter 2: 40-61
- Chapter 3: 62-73
- Chapter 4: 74-93
- Chapter 5: 94-119
- Chapter 6: 126-143
- Chapter 8: 176-199

Every website on the internet uses HTML & CSS.

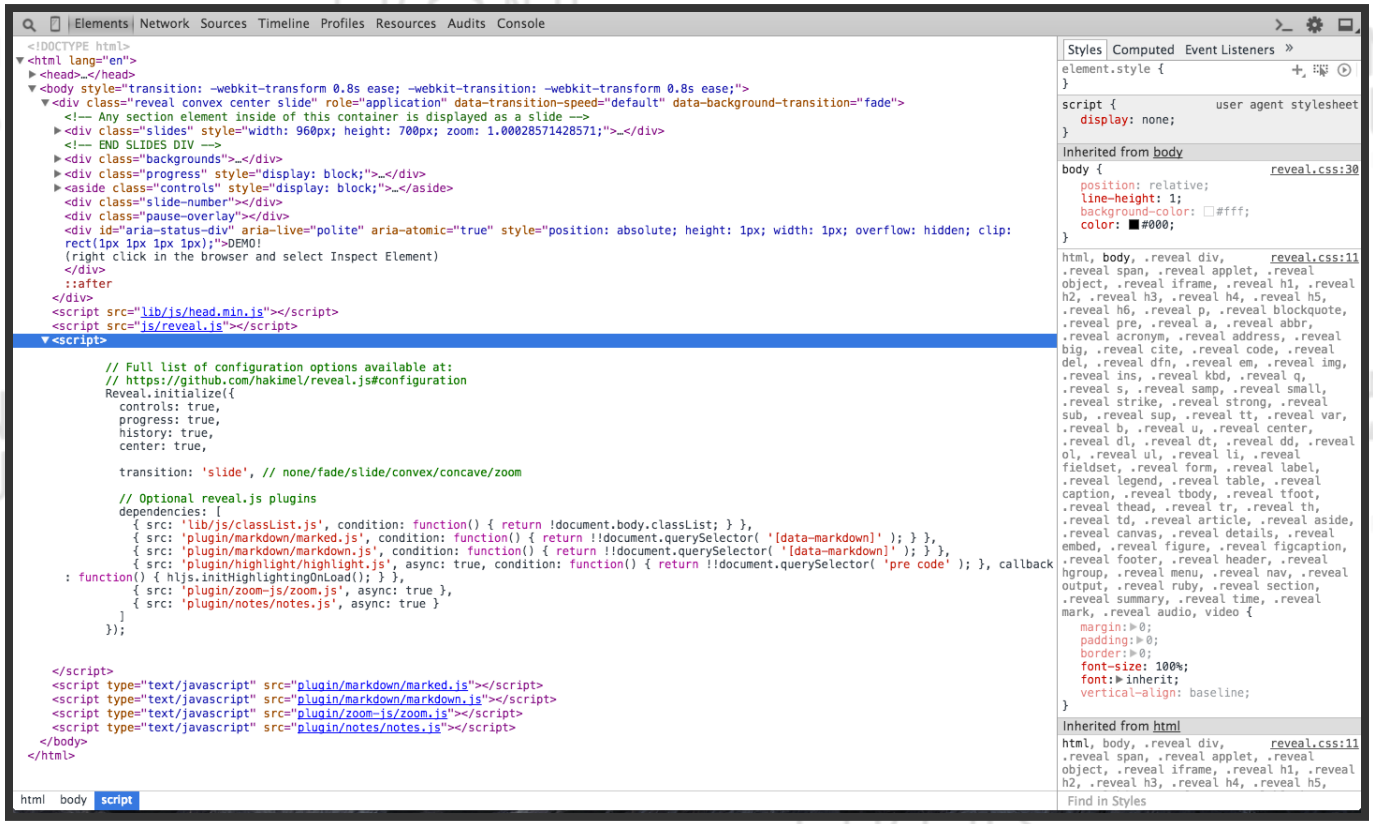
...most of them use JavaScript as well, in one form or another.

MEET YOUR NEW BEST FRIEND!

It's called the Dev Tools Inspector

(right click in the browser

and select `Inspect Element`)


Elements Network Sources Timeline Profiles Resources Audits Console
<!DOCTYPE html>
<html lang="en">
<head>…</head>
<body style="transition: -webkit-transform 0.8s ease; -webkit-transition: -webkit-transform 0.8s ease;">

<!-- Any section element inside of this container is displayed as a slide -->
…
<!-- END SLIDES DIV -->
…
…
<aside class="controls" style="display: block;">…</aside>

DEMO!
(right click in the browser and select Inspect Element)

::after

<script src="lib/js/head.min.js"></script>
<script src="js/reveal.js"></script>
<script>
// Full list of configuration options available at:
// https://github.com/hakimel/reveal.js#configuration
Reveal.initialize({
controls: true,
progress: true,
history: true,
center: true,
transition: 'slide', // none/fade/slide/convex/concave/zoom
// Optional reveal.js plugins
dependencies: [
{ src: 'lib/js/classList.js', condition: function() { return !document.body.classList; } },
{ src: 'plugin/markdown/marked.js', condition: function() { return !!document.querySelector('[data-markdown]'); } },
{ src: 'plugin/markdown/markdown.js', condition: function() { return !!document.querySelector('[data-markdown]'); } },
{ src: 'plugin/highlight/highlight.js', async: true, condition: function() { return !!document.querySelector('pre code'); }, callback
: function() { hljs.initHighlightingOnLoad(); } },
{ src: 'plugin/zoom-js/zoom.js', async: true },
{ src: 'plugin/notes/notes.js', async: true }
]
});
</script>
<script type="text/javascript" src="plugin/markdown/marked.js"></script>
<script type="text/javascript" src="plugin/markdown/markdown.js"></script>
<script type="text/javascript" src="plugin/zoom-js/zoom.js"></script>
<script type="text/javascript" src="plugin/notes/notes.js"></script>
</body>
</html>
html body script
Styles Computed Event Listeners
element.style {
}
script { user agent stylesheet
display: none;
}
Inherited from body
body { reveal.css:30
position: relative;
line-height: 1;
background-color: #fff;
color: #000;
}
html, body, .reveal div, reveal.css:11
.reveal span, .reveal applet, .reveal
object, .reveal iframe, .reveal h1, .reveal
h2, .reveal h3, .reveal h4, .reveal h5,
.reveal h6, .reveal p, .reveal blockquote,
.reveal pre, .reveal a, .reveal abbr,
.reveal acronym, .reveal address, .reveal
big, .reveal cite, .reveal code, .reveal
del, .reveal dfn, .reveal em, .reveal img,
.reveal ins, .reveal kbd, .reveal q,
.reveal s, .reveal samp, .reveal small,
.reveal strike, .reveal strong, .reveal
sub, .reveal sup, .reveal tt, .reveal var,
.reveal b, .reveal u, .reveal center,
.reveal dl, .reveal dt, .reveal dd, .reveal
ol, .reveal ul, .reveal li, .reveal
fieldset, .reveal form, .reveal label,
.reveal legend, .reveal table, .reveal
caption, .reveal tbody, .reveal tfoot,
.reveal thead, .reveal tr, .reveal th,
.reveal td, .reveal article, .reveal aside,
.reveal canvas, .reveal details, .reveal
embed, .reveal figure, .reveal figcaption,
.reveal footer, .reveal header, .reveal
hgroup, .reveal menu, .reveal nav, .reveal
output, .reveal ruby, .reveal section,
.reveal summary, .reveal time, .reveal
mark, .reveal audio, video {
margin: 0;
padding: 0;
border: 0;
font-size: 100%;
font: inherit;
vertical-align: baseline;
}
Inherited from html
html, body, .reveal div, reveal.css:11
.reveal span, .reveal applet, .reveal
object, .reveal iframe, .reveal h1, .reveal
h2, .reveal h3, .reveal h4, .reveal h5,
Find in Styles

TERMINOLOGY

TERMINOLOGY

Some key terms or phrases that are used as a matter of course in the software industry. You may know some or all of these, or you may have heard the terms but be unclear about their actual meaning. They are common jargon among developers

WEB DEVELOPMENT

Web development is a broad term for the work involved in developing a web site for the Internet. In industry parlance, 'web development' usually refers to the more code-related tasks such as programming JavaScript, coding HTML and CSS. It can even extend to tasks related to the back end server infrastructure such as creating web services and handling business logic for a company or product.

WEB DESIGN

The process of planning & structuring a website; specifically, the visual aspects and assets for the site. Recently, this job description has also begun to include interaction design. That is, designing the user experience (UX), information architecture, and the flow of the application or site.

WEB SITE

A largely informational web page. While they may include dynamic elements and react to user inputs. The general purpose of a web site is to provide information about a person, business, product, or service.

WEB APPLICATION

A more recent term to indicate a web site whose sole purpose is not just informational, but rather functional. Web applications have become robust enough to do everything from our taxes, manage our personal calendars, or even do standard desktop publishing tasks.

FRONT END

The visible and interactive parts of a website or application.

BACK END

The 'invisible' or inner functionality of a website or application. Examples include costly calculations, interacting with a database or making use of web service end points. While we're at it...

WEB SERVICE

A software function provided at a network address over the Web. The W3C defines a Web service generally as:

"A software system designed to support interoperable machine-to-machine interaction over a network."

DATABASE

A software system for storing data long term. This is also sometimes referred to as a 'persistance layer'.

VERSION CONTROL

A software tool for managing changes to a set of files, website, application or any collection of files and for reconciling the differences between those files when conflicts emerge.

APPLICATION PROGRAMMING INTERFACE (API)

An API is a set of routines, protocols, and tools for building software applications. An API expresses a software component in terms of its operations, inputs, outputs, and underlying types. An API defines functionalities that are independent of their respective implementations, *which allows definitions and implementations to vary without compromising each other.*

QUESTIONS?

NESTING

HTML elements 'nest' inside of one another. The element that opens first closes last.

CONTENT TAGS

COMMON CONTENT TAGS

Tag	Description
`div`	defacto container element
`p`	used for body copy
`h1 thru h6`	designating titles/subtitles
`ol`	create a numbered list
`ul`	create an unordered list
`li`	list elements
`a`	link to other pages or sites
`link`	import external documents

SEMANTIC HTML5 TAGS

class	Description
`section`	Container tag used for page organization
`header`	Container for introductory and navigational stuff
`footer`	Container for footer content (site map, internal links, etc.)
`nav`	Container for a major block of navigation links
`audio`	Multimedia tag for playing audio files
`video`	Multimedia tag for playing video files
`canvas`	element can be used to draw graphics via JavaScript

HTML COMMENTS

Like any other good coding language, HTML offers comments. They operate like comments in any other language. They are ignored by the browser engine.

```
<!-- Hello, I am a comment. -->
```

TABLES

A NOTE ON TABLES

You may be thinking it now or you may think later that tables would be a great way to position content for site...

Don't do that. Just... don't. Trust me on this. It's a bad idea. It will give us an ulcer.

(srsly, don't)

WHAT'S WRONG WITH THIS CODE?

Look at the following examples and tell me what is wrong with the code.

```
<html>
  <head>
  <body>
  </head>

  </body>
</html>
```

```
<html>
  <head>
    <title>The Best Site Evar!!
  </head>
  <body>

    <p>Check out this riveting content!</p>

  </body>
</html>
```

```
<p style=hotStuff>Check out this riveting content!</p>
```

FOLDER STRUCTURE

This stuff is not exciting but it's *important*.

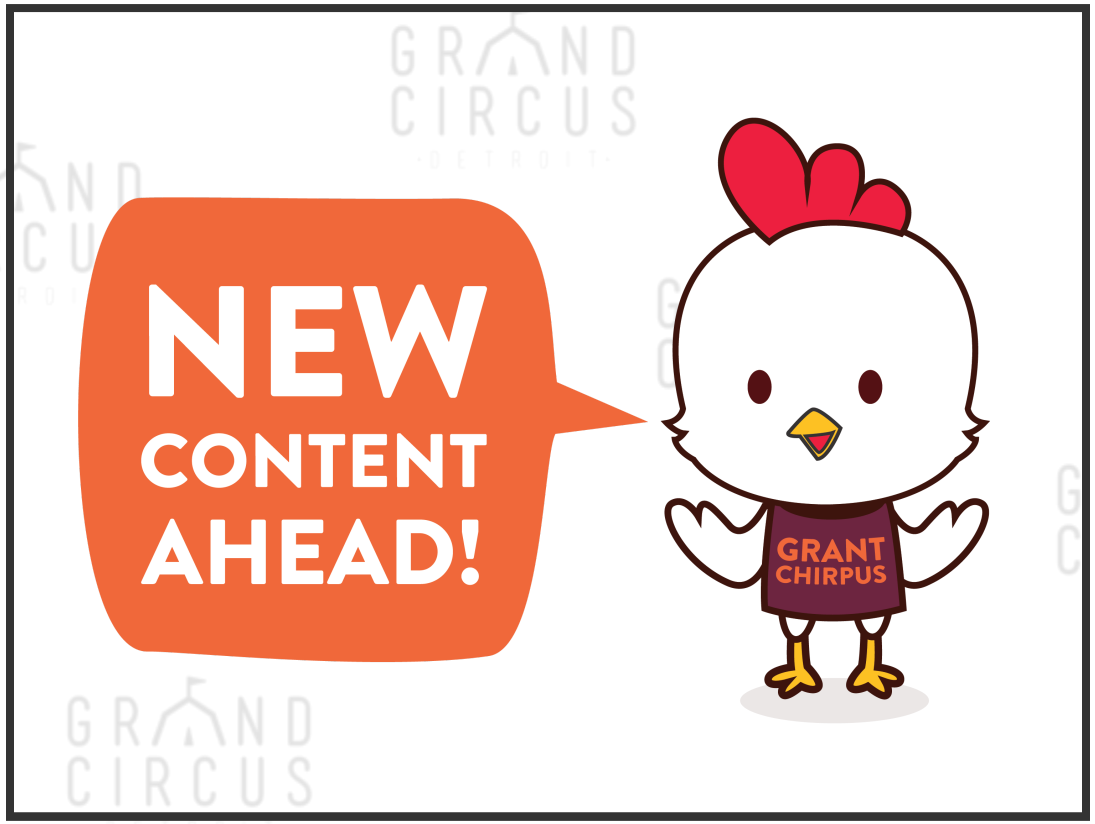
FOLDER STRUCTURE

The Rules of *Threes*

RECAP

You should understand and be able to use:

- HTML elements
- Proper nesting
- HTML Comments
- Correct folder structure


NEW
CONTENT
AHEAD!
GRANT
CHIRPUS

CSS 101

CSS COMMENTS

Just like HTML, CSS offers comments:

```
/* This is a CSS comment,
      it can be multi-line */
```

CSS DECLARATIONS

Declarations are made up of the property and value of the style you want to apply.

They can be grouped together so that more than one declaration may be applied to a selected element.

Declaration groups must be surrounded by curly brackets.

Declarations must end in a semicolon.

```
.selector {
  background-color: red;
  color: white;
  border: 1px solid black;
  border-radius: 5px;
}
```

CSS SELECTOR

The selector instructs the browser to search the page for any HTML element that matches the given criteria. It applies any applicable declarations to that element.

CSS SELECTOR - ELEMENT

Elements can be selected by their element name. In this case, all elements of that element type will be selected and have the styles applied.

CSS SELECTOR - ELEMENT

```css
p {
  position: absolute;
  top: 0px;
  left: -100px;
}
```

CSS SELECTOR - CLASS

Elements can be selected based on HTML attributes such as class. In this case all elements that have a matching class attribute will be selected.

CSS SELECTOR - CLASS

```css
.timer {
  position: absolute;
  top: 0px;
  left: -100px;
}
```

CSS SELECTOR - ID

Elements can also be selected based on HTML attribute ID. In this case only one element would be selected, as HTML IDs are intended to be unique.

CSS SELECTOR - ID

```css
#fluffy {
  position: absolute;
  top: 0px;
  left: -100px;
}
```

CSS SELECTOR - CHILD SELECTORS

Selectors can be combined to become more specific. This example selects searches for any paragraph tag that is nested inside a div tag.

```
div p {
  position: absolute;
  top: 0px;
  left: -100px;
}
```

CSS SELECTOR - MULTIPLE

In addition a set of declarations can be applied to more than one selector by listing a number of comma-separated selectors.

```
.timer, img, div p, #kitty {
  position: absolute;
  top: 0px;
  left: -100px;
}
```

POP QUIZ HOT SHOT

Look at the following examples and tell me which Elements (if any) would be returned by the following selectors.

POP QUIZ

```
p {
  /* blah blah */
}
```

POP QUIZ

```
#fuzzy {
  /* blah blah */
}
```

POP QUIZ

```
.blue {
  /* blah blah */
}
```

POP QUIZ

```
.blue, #fuzzy {
  /* blah blah */
}
```

POP QUIZ

```
div p {
  /* blah blah */
}
```

POP QUIZ

```
body div {
  /* blah blah */
}
```

QUESTIONS?

CSS PROPERTIES

CSS PROPERTIES

There are literally hundreds of css properties that are available for use. We don't have time to go over more than just a few. We will go over a few of the most common. However, the best strategy is to google for styling options as you're working.

COMMON CSS PROPERTIES

Property	Description
`background-color`	background color for an element
`color`	color of the *text* in an element
`font-family`	typeface for text
`font-size`	size for text (px, %, em, pt)
`font-weight`	used to bold text (if possible)
`text-decoration`	used for underline (mostly)
`height`	specifies the height of an element
`width`	specifies the width of an element

COLOR IN CSS

Method	Syntax	Description
color name	`white`	a list of 140 predefined colors
hexidecimal	`#FF0000`	RGB values in hex 00 - FF (0 - 255)
RGBA	`rgba(255, 0, 187, 0.5)`	RGB values with an added alpha (opacity) value

CSS UNITS

Method	Syntax	Description
em	1em	Scalable unit based on font size
pixels	16px	Fixed number of pixels
percent	120%	Percent value based on font size

HOMEWORK

From HTML & CSS:

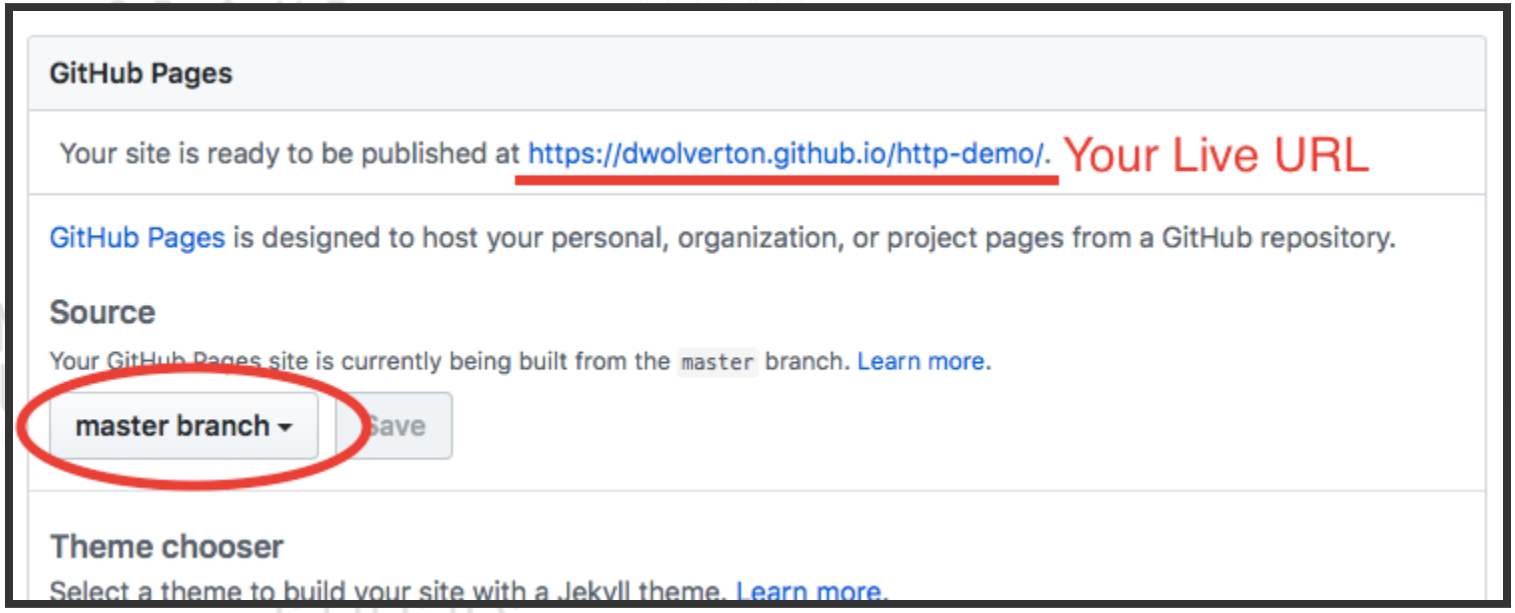
- Chapter 10: 226-245
- Chapter 11: 246-253
- Chapter 12: 264-287
- Chapter 13: 300-329
- Chapter 14: 330-340
- Chapter 15: 358-376
- Chapter 16: 406-418

GITHUB PAGES

MAKE YOUR PREWORK DELIVERABLE LIVE

GITHUB PAGES

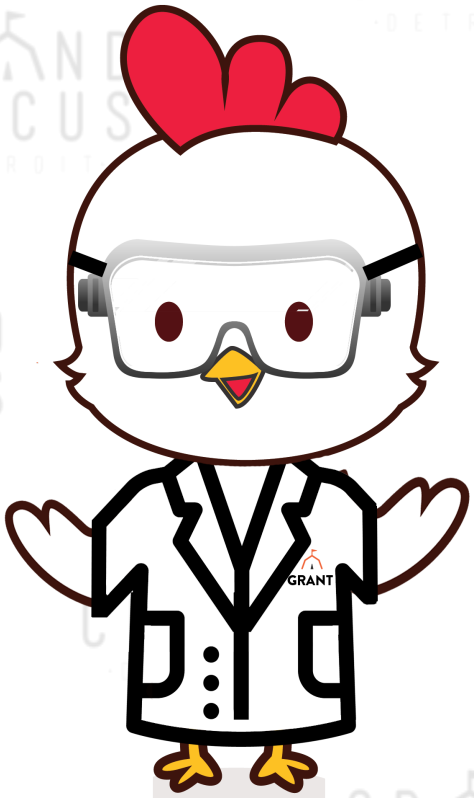
Settings tab at the top of your repo screen on GitHub.com



Note: This will only work well if you have an HTML file named exactly *index.html*. You should always use this name for your main file.

LAB 1

PERSONAL SITE



PERSONAL SITE

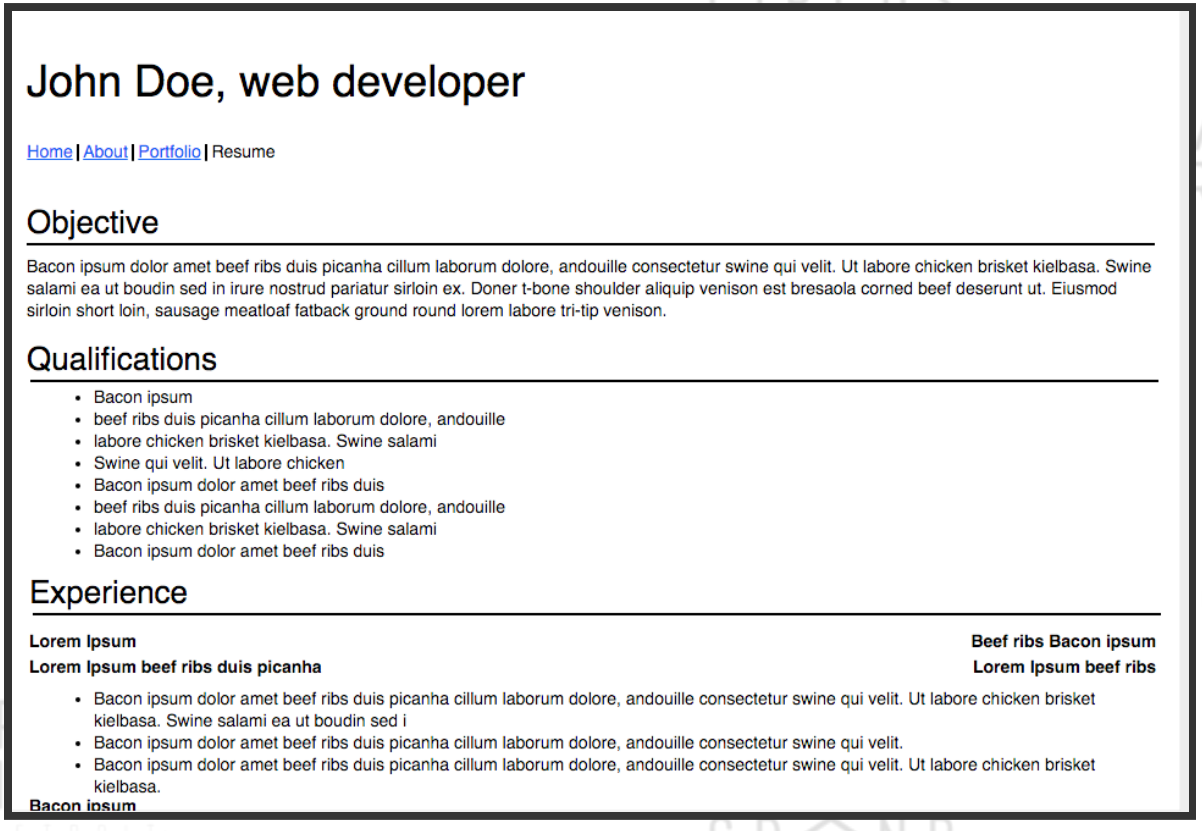
By the end of this course, you will have a personal website with which you can showcase all the hard work you've done. In this lab, you'll take the first steps toward creating your site by outlining the HTML.

Your site should contain the following elements:

- main page: an introduction to the site
- about page: a short bio and other relevant information about you
- portfolio page: this page will hold examples of projects you've done
- resume page: incorporate any feedback you've received on your portfolio site from the pre-work
- navigation: a list of links between pages

Bonus: Include elements we have not specifically outlined (e.g. a photo of you, tabular data).

PRE-GAME



John Doe, web developer

Home | About | Portfolio | Resume

Objective

Bacon ipsum dolor amet beef ribs duis picanha cillum laborum dolore, andouille consectetur swine qui velit. Ut labore chicken brisket kielbasa. Swine salami ea ut boudin sed in irure nostrud pariatur sirloin ex. Doner t-bone shoulder aliquip venison est bresaola corned beef deserunt ut. Eiusmod sirloin short loin, sausage meatloaf fatback ground round lorem labore tri-tip venison.

Qualifications

- Bacon ipsum
- beef ribs duis picanha cillum laborum dolore, andouille
- labore chicken brisket kielbasa. Swine salami
- Swine qui velit. Ut labore chicken
- Bacon ipsum dolor amet beef ribs duis
- beef ribs duis picanha cillum laborum dolore, andouille
- labore chicken brisket kielbasa. Swine salami
- Bacon ipsum dolor amet beef ribs duis

Experience

Lorem Ipsum **Beef ribs Bacon ipsum**

Lorem Ipsum beef ribs duis picanha **Lorem Ipsum beef ribs**

- Bacon ipsum dolor amet beef ribs duis picanha cillum laborum dolore, andouille consectetur swine qui velit. Ut labore chicken brisket kielbasa. Swine salami ea ut boudin sed i
- Bacon ipsum dolor amet beef ribs duis picanha cillum laborum dolore, andouille consectetur swine qui velit.
- Bacon ipsum dolor amet beef ribs duis picanha cillum laborum dolore, andouille consectetur swine qui velit. Ut labore chicken brisket kielbasa.

Bacon ipsum