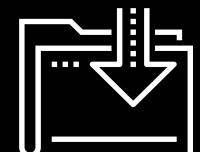




# Going Live

Web Development Boot Camp  
Lesson 1.4



# Objectives

---

In today's class, we will:

01

Recap the basic topics covered to date.

02

Offer a conceptual introduction to CSS reset and downloadable style sheets.

03

Work with Chrome DevTools for site inspection.

04

Utilize GitHub Pages for website deployment.

# Checking In

---

How's it going?



# Instructor Feedback

---

Seriously? Mind. Blown.



# Instructor Feedback

---

Things I've noticed people doing incredibly well:



All of you are handling an enormous volume of information.



All of you are asking the right questions.



You notice the right details.

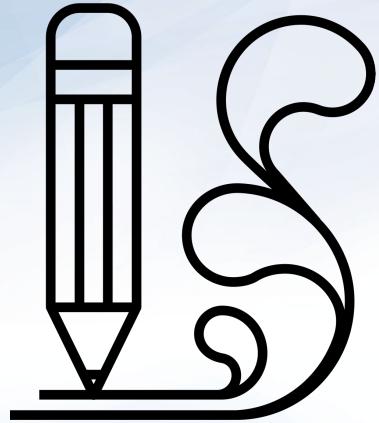


You all help each other out.



And, most importantly, you are figuring out things on your own.

# A Few Admin Things



## Homework is due!

Don't forget to submit both the GitHub  
and GitHub Pages links.

Due Date:



## **Office Hours and Additional Help:**

Office hours are held 45 minutes before class and 30 minutes after class.



Feedback #1: **The Pace Is Fast!**

# Feedback #1: Pace Is Fast

---

Here's the reality: the pace of this program is fast.



We could slow down and make it easier, but that wouldn't prepare you to be developers right out of the program.



Take up the challenge. Try to retain as much as you can during the course.



After the program is over, go back and review the concepts you need a refresher on.



Don't try to master every little detail.



Focus on the big picture and how to find what you need, when you need it.

# Feedback #1: Pace Is Fast

---

That said, as instructors and TAs, we are here to help!



Share your GitHub code as you run into issues. We can let you know if you're going in the right (or wrong) direction.



Come to office hours! We can review your code with you, offer tips on homework, or just give you a pep talk if that's what you need.



Reach out to Student Success if want to schedule a 1-on-1 meeting.

Feedback #2: **Books, Please!**

# Feedback #2: Books, Please!

---



During the program, concentrate all your efforts on what we're covering in class.



Books can be helpful when you're learning on your own, but in a boot camp like this, they can be distracting—especially when class is moving so quickly.



When you're at home, focus your efforts on completing your homework, re-completing in-class exercises, and going over slides.



Invest in books and outside tutorials after you finish the program.

That said, if you truly feel the need for outside resources...



[PluralSight.com](#) offers some of the best online courses for HTML, CSS, JavaScript, Node, Express, and Angular.

# Feedback #2: Books, Please!

---

Jon Duckett's books *HTML & CSS* and *JavaScript & jQuery* are two excellent primers. If you're a book person, these can be very helpful!



*HTML & CSS: Design and Build Websites*  
by Jon Duckett  
(Wiley, 2011)



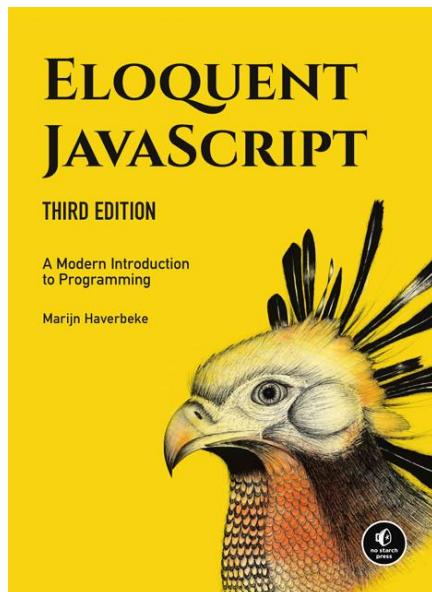
*JavaScript & JQuery: Interactive Front-End Web Development*  
by Jon Duckett  
(Wiley, 2014)

# Feedback #2: Books, Please!

---

*Eloquent JavaScript* is a terrific, free book that reviews the fundamentals of JavaScript.

You can find it here: [eloquentjavascript.net](http://eloquentjavascript.net)



*Eloquent JavaScript, 3rd Edition*  
by Marijn Haverbeke  
(No Starch Press, 2018)

Feedback #3: **The Homework Is Hard**

# Feedback #3: Homework Is Hard

---



Check your homework due dates in Boot Camp Spot.



Generally speaking, expect homework to be semi-challenging.



Make sure you're setting aside a solid block of time for each assignment.



If you find yourself working on an assignment for more than two hours without even a small amount of progress, let us know!



You don't win points in life for staring at a blank screen.

# Recap

# Recap

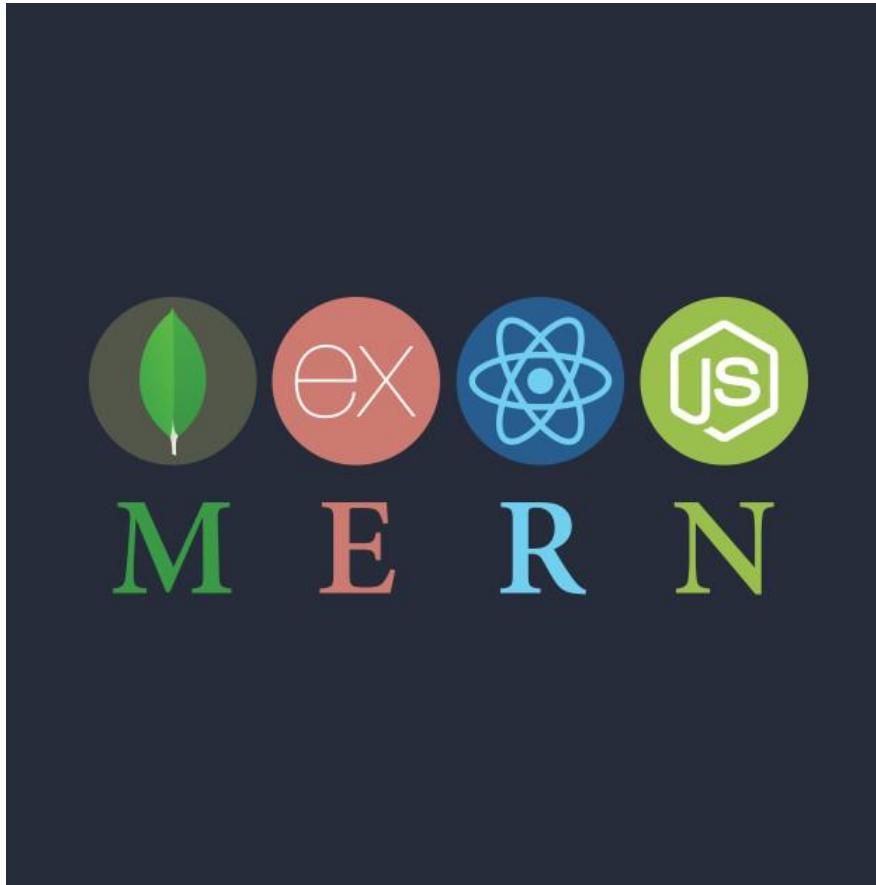
---

In just one whirlwind week we've covered:

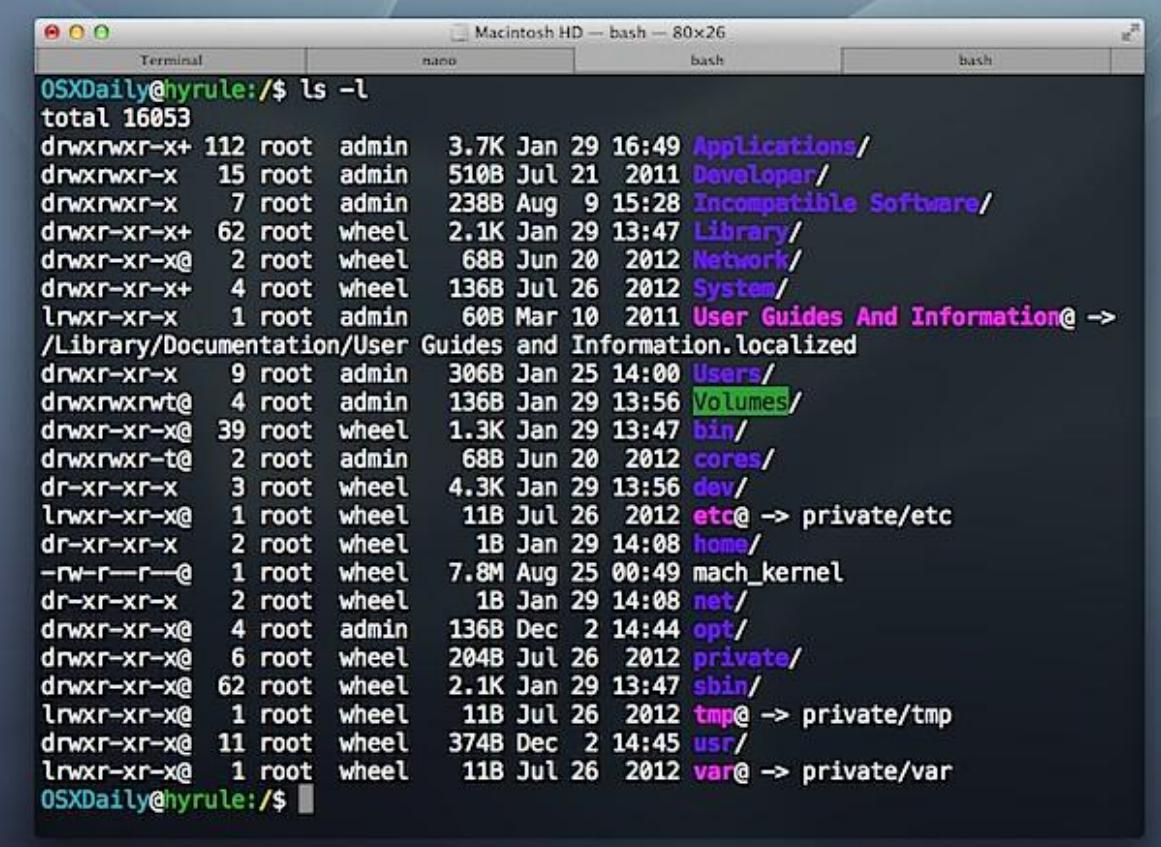
- Basics of full-stack development
- Terminal and Git Bash
- HTML syntax
- Git concepts and commands
- CSS purpose, syntax, and styles
- Floating
- Positioning
- Box model
- Chrome DevTools
- **How to learn on your own!**

# What Is Full-Stack Development?

---



# > Intro to Console



The screenshot shows a Mac OS X Terminal window titled "Macintosh HD — bash — 80x26". The window displays the output of the command "ls -l". The output lists numerous system directories and files, including Applications, Developer, Incompatible Software, Library, Network, System, User Guides And Information, Volumes, bin, cores, dev, etc, home, mach\_kernel, net, opt, private, sbin, tmp, usr, and var. The "Volumes/" directory is highlighted with a green rectangular selection.

```
OSXDaily@hyrule:/$ ls -l
total 16053
drwxrwxr-x+ 112 root admin 3.7K Jan 29 16:49 Applications/
drwxrwxr-x  15 root admin 510B Jul 21 2011 Developer/
drwxrwxr-x   7 root admin 238B Aug  9 15:28 Incompatible Software/
drwxr-xr-x+  62 root wheel 2.1K Jan 29 13:47 Library/
drwxr-xr-x@  2 root wheel 68B Jun 20 2012 Network/
drwxr-xr-x+  4 root wheel 136B Jul 26 2012 System/
lrwxr-xr-x   1 root admin 60B Mar 10 2011 User Guides And Information@ ->
/Library/Documentation/User Guides and Information.localized
drwxr-xr-x   9 root admin 306B Jan 25 14:00 Users/
drwxrwxrwt@  4 root admin 136B Jan 29 13:56 Volumes/
drwxr-xr-x@  39 root wheel 1.3K Jan 29 13:47 bin/
drwxrwxr-t@  2 root admin 68B Jun 20 2012 cores/
dr-xr-xr-x   3 root wheel 4.3K Jan 29 13:56 dev/
lrwxr-xr-x@  1 root wheel 11B Jul 26 2012 etc@ -> private/etc
dr-xr-xr-x   2 root wheel 1B Jan 29 14:08 home/
-rw-r--r--@  1 root wheel 7.8M Aug 25 00:49 mach_kernel
dr-xr-xr-x   2 root wheel 1B Jan 29 14:08 net/
drwxr-xr-x@  4 root admin 136B Dec  2 14:44 opt/
drwxr-xr-x@  6 root wheel 204B Jul 26 2012 private/
drwxr-xr-x@  62 root wheel 2.1K Jan 29 13:47 sbin/
lrwxr-xr-x@  1 root wheel 11B Jul 26 2012 tmp@ -> private/tmp
drwxr-xr-x@  11 root wheel 374B Dec  2 14:45 usr/
lrwxr-xr-x@  1 root wheel 11B Jul 26 2012 var@ -> private/var
OSXDaily@hyrule:/$
```

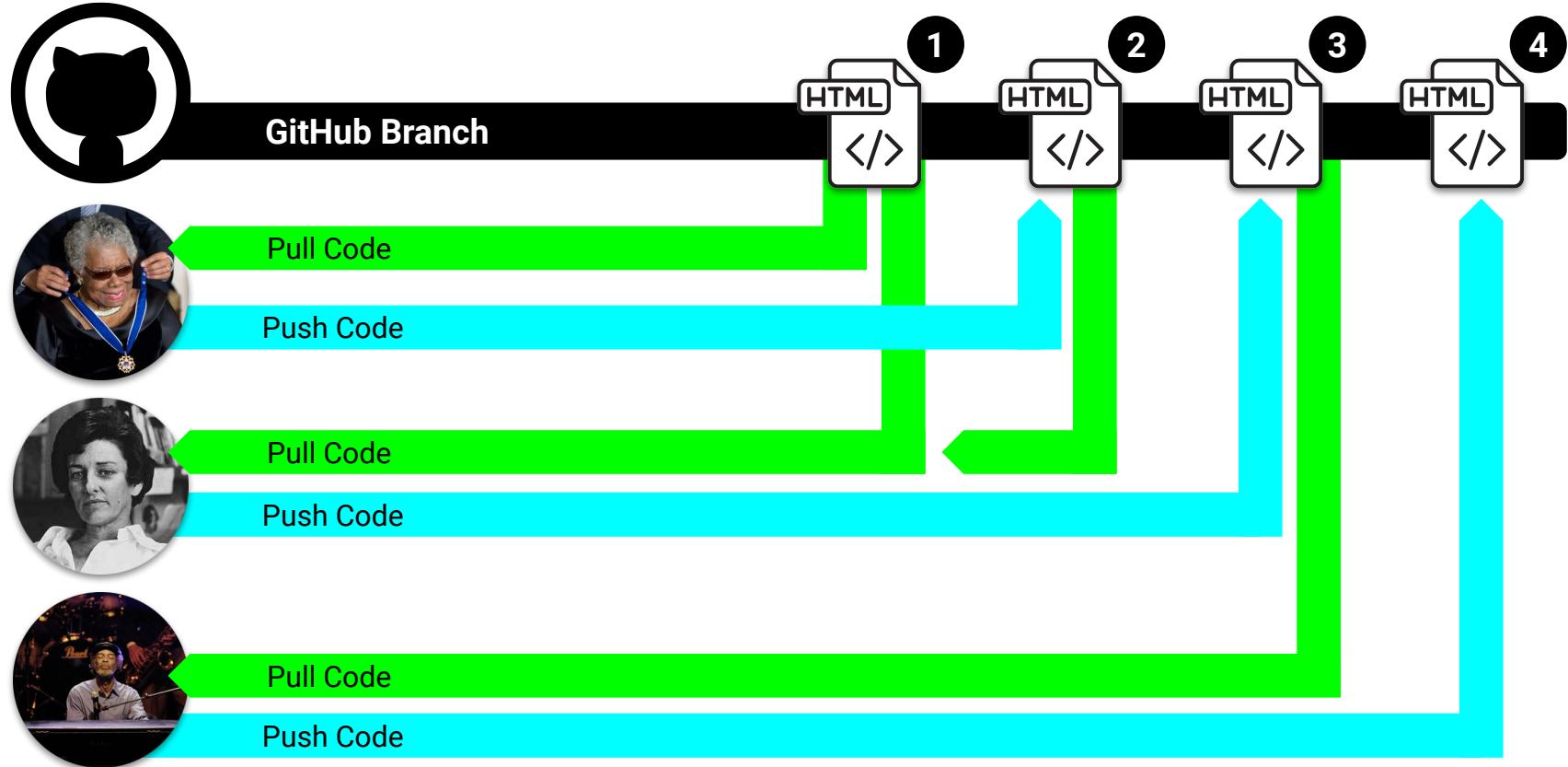
# <title> Intro to HTML </title>

---

HTML is one of the three base languages behind every website.  
It defines all of the basic content as well as a bit of formatting.



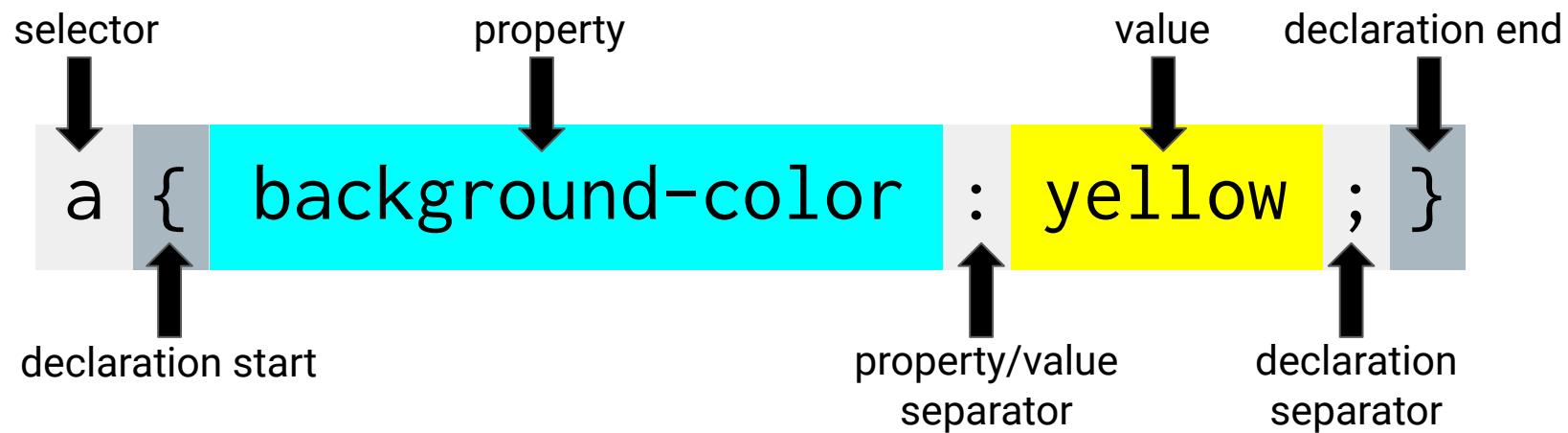
# Pushing and Pulling to GitHub



# CSS Syntax

CSS works by hooking onto **selectors** added into HTML using **classes** and **identifiers**.

Once hooked, we apply **styles** to those HTML elements using CSS.

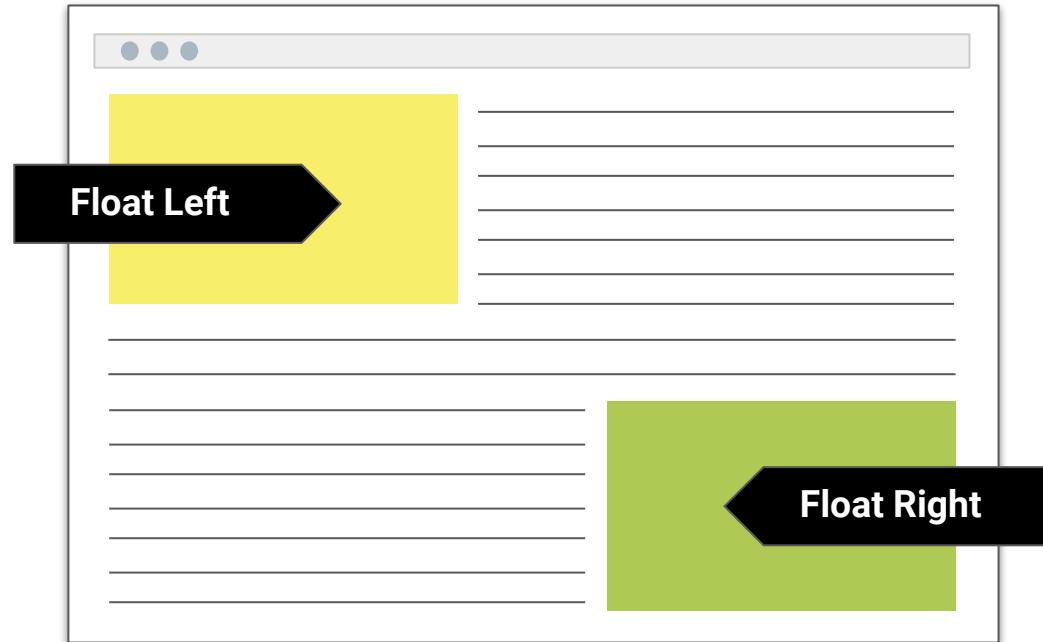


# The Concept of Flow

---

By default, every HTML element displayed in the browser is governed by a concept called **flow**.

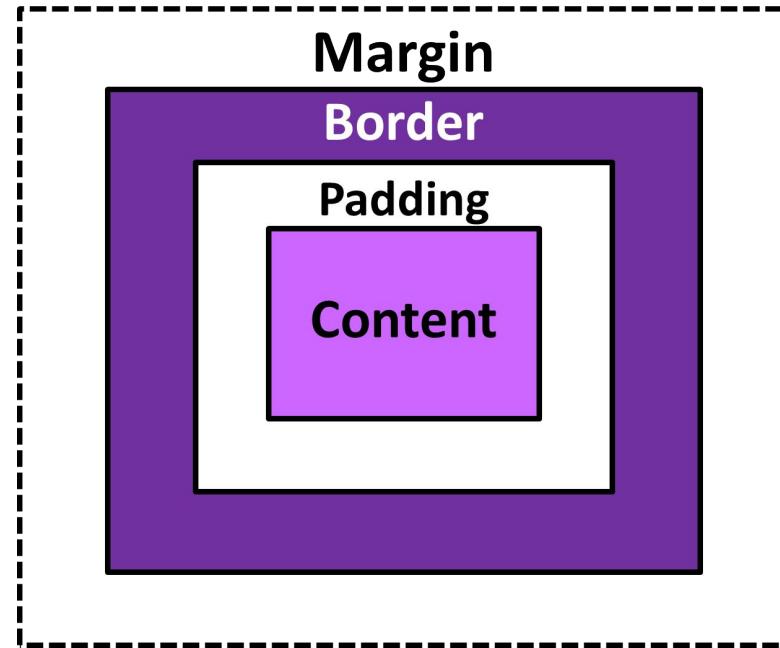
This means that HTML elements force their adjacent elements to flow around them.



# The Box Model

---

The CSS **box model** wraps every HTML element in a box. This box consists of padding, border, margin, and content, and allows developers to modify spacing styles.



# CSS Positioning

---

We can orient our HTML elements in relation to space with CSS positioning (static, relative, fixed, absolute).

<div>  
position: fixed

ading

ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse a  
erit ex, at blandit sapien. Lorem ipsum dolor sit amet, consectetur  
adipiscing elit. Quisque finibus felis sem, non pulvinar odio fermentum vel.  
Nunc varius tempus scelerisque. Curabitur congue magna vitae velit dictum,  
eu finibus neque bibendum. In hac habitasse platea dictumst. Aliquam  
fermentum lobortis felis, in feugiat diam congue ac.

<div>  
position: relative

<div>  
position: absolute

Nulla tempor ornare diam, vitae volutpat erat bibendum eget. Nunc sagittis  
placerat velit sit amet interdum. Nam in iaculis purus, quis tristique velit.  
Cras ut nisl vitae orci malesuada placerat non sed magna. Nulla ultrices,  
dolor at aliquam volutpat, lorem magna pharetra arcu, eget feugiat nisi libero  
at nunc. Phasellus finibus elit at sapien vehicula varius. Maecenas in dapibus  
leo. Aliquam molestie vulputate metus. Morbi sed posuere quam, et sodales  
felis. Proin augue nulla, pellentesque at venenatis vel, sagittis eget nibh.  
Maecenas libero velit, luctus eu velit vitae, eleifend convallis felis.

# Resources

---



[stackoverflow.com](https://stackoverflow.com)



[w3schools.com](https://www.w3schools.com)



[designshack.net](https://designshack.net)



[css-tricks.com](https://css-tricks.com)



[smashingmagazine.com](https://smashingmagazine.com)



[sitepoint.com/community](https://sitepoint.com/community)



[developer.mozilla.org/en-US](https://developer.mozilla.org/en-US)



[cssnewbie.com](https://cssnewbie.com)

# **General Questions/Issues?**

# Double-Take

# Divs and Sections

---

[stack\*\*overflow\*\*.com](http://stackoverflow.com)



What is the difference between `<section>` and `<div>`?



What is the purpose of new HTML5 elements like `<section>` and `<article>`?



Why would you use an HTML5 semantic tag instead of `<div>`?



When do you use a `<div>` element, and when do you use a `<section>` element?

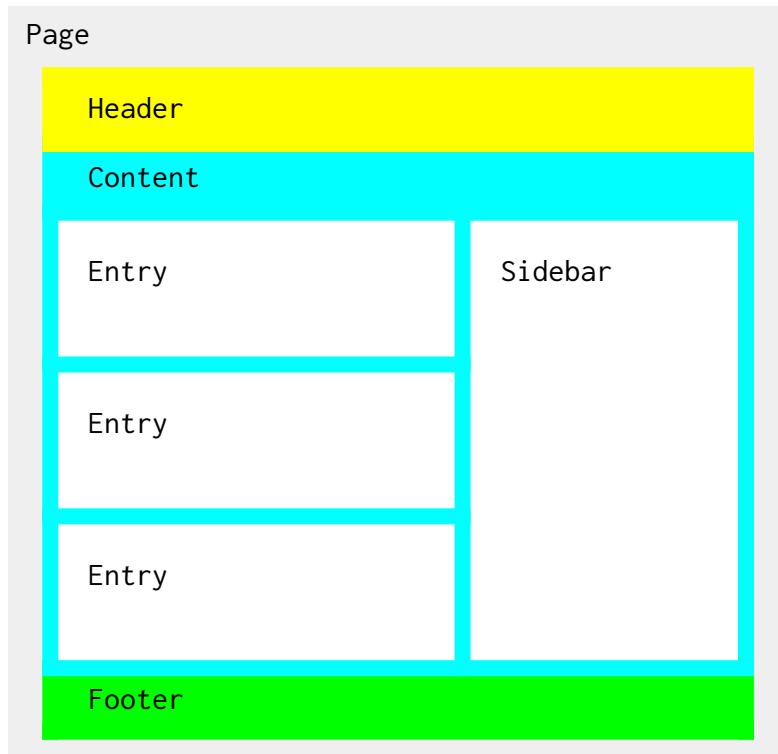


Which should you use for semantic markup and page layout: `<section>` or `<div>`?

# Divs and Sections

---

All web layouts are made up of containers, traditionally called **divs**.



# Divs and Sections

---

HTML5 introduced the concept of **semantic layouts**, meaning divs could be given more meaningful names. In theory, this helps with organization and search engine optimization.

## Website Layout Using HTML5:

HTML5 offers new semantic elements that define different parts of a webpage:

<header>	Defines a header for a document or a section
<nav>	Defines a container for navigation links
<section>	Defines a section in a document
<article>	Defines an independent self-contained article
<aside>	
<footer>	Defines a footer for a document or a section
	Defines additional details
	Defines a heading for the <details> element

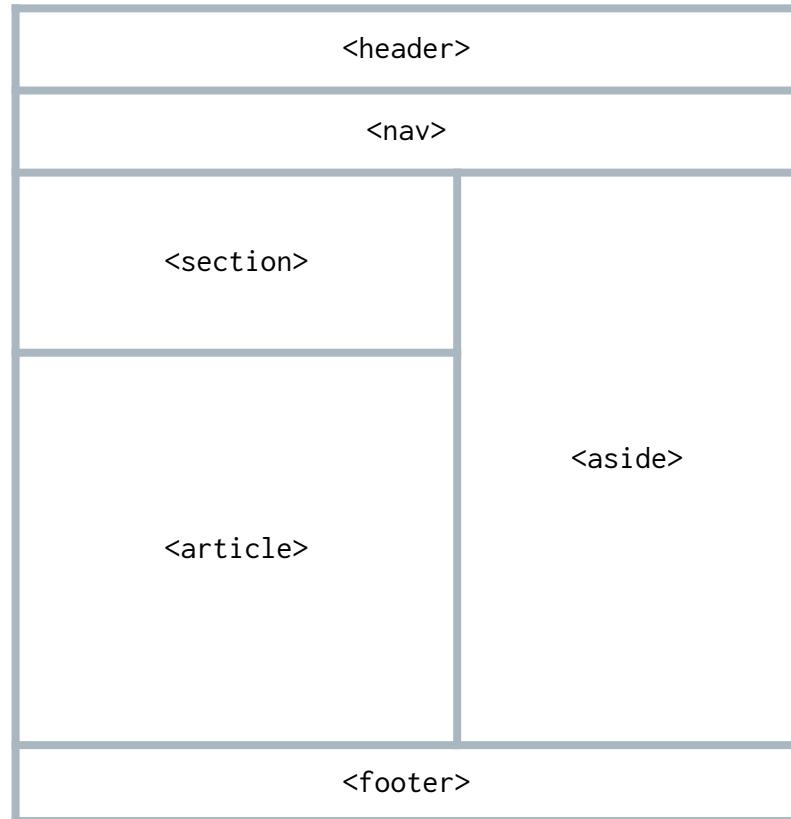
# Divs and Sections

---

That said, it seems many, if not most, websites still use basic divs.

There are reasons for this that we'll showcase in later lessons.

Additionally, it's possible to include semantics by using ID names and classes.



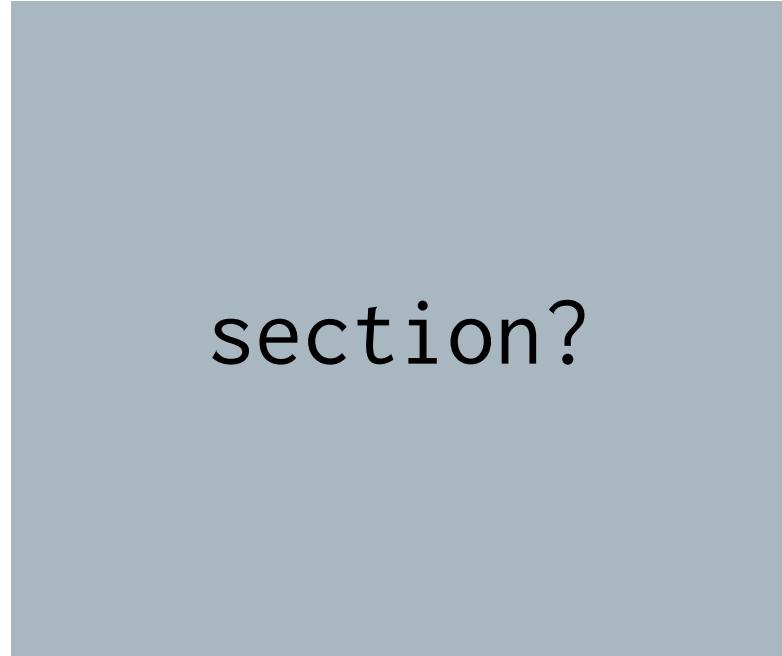
# Divs and Sections

---

**Bottom line:** Follow the homework instructions. And when you're out in the real world, follow the conventions of where you work!



div?



section?

# Classes vs. IDs

**When choosing between a CSS ID and a CSS class, follow the convention:**

**Classes (.classname)** are used if the same style will be used on multiple HTML elements.

---

**IDs (#idname)** are used if a style is unique to a certain HTML element.



# Modifying Sites

---

You can edit any page's HTML and CSS with Chrome DevTools.  
And you'll see your results instantly.



# Practice Through Frustration

---

Keep practicing. It gets better!



# CSS Typography

# Typography Is Huge

---

Typography can play a huge role in the visual aesthetic and emotional identity of a webpage.



# Typographic Properties

---

Typographic properties describe attributes of the type on the page.



**Line Height:** Distance between lines of text on a page (Golden Ratio of 1.5x font).



**Font Size:** Actual size of lettering. Should be at least 16 px on modern pages.

[smashingmagazine.com](http://smashingmagazine.com)



**Line Length:** Not a CSS property but rather a standard. Should be about 50–75 characters per line on a desktop. [baymard.com](http://baymard.com)



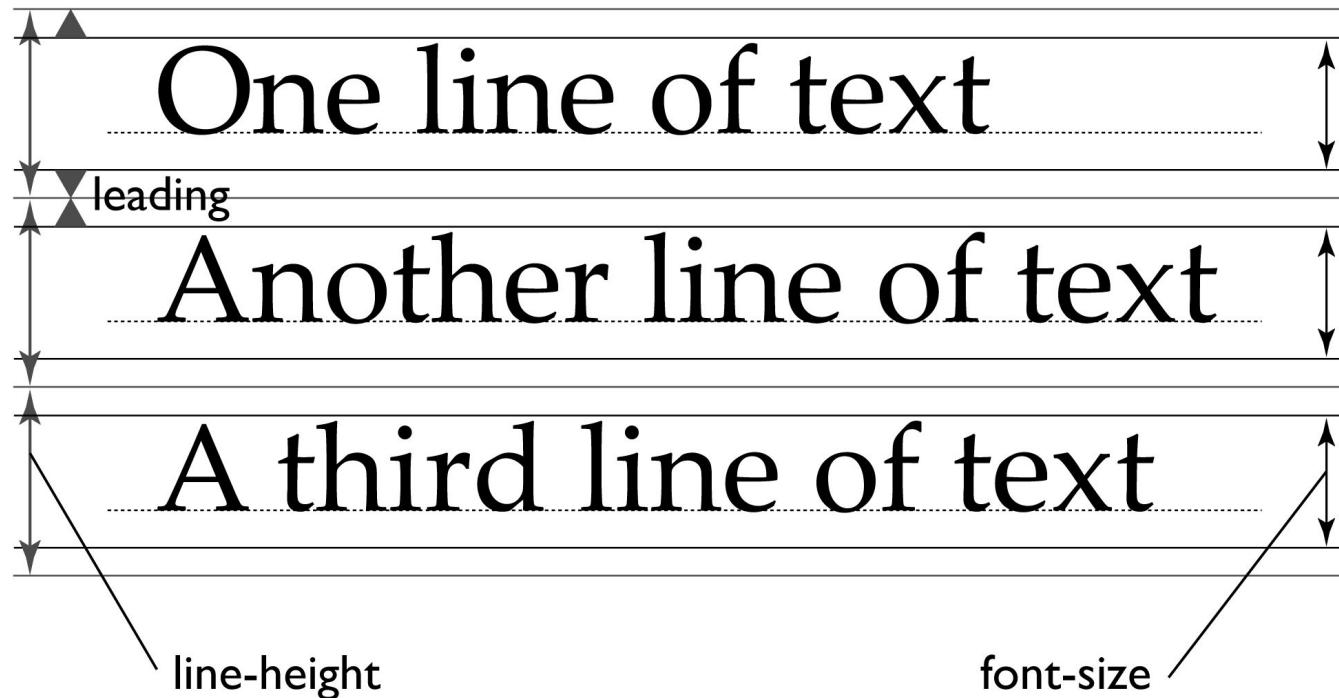
**Letter Spacing:** Spacing between individual letters. Try to avoid cramming letters together too closely.



**Sans-Serif vs. Serif:** Sans-serif are fonts *sans their serifs*—no tails. They're easier to read online.

# Line Height vs. Fonts

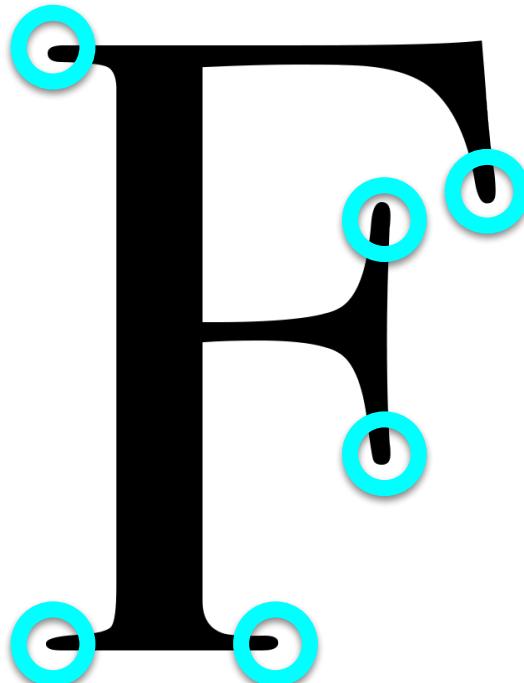
---



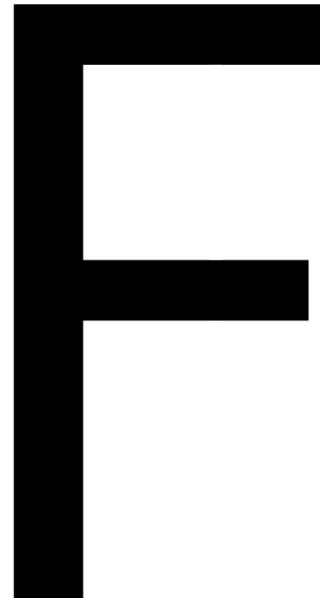
# Serif vs. Sans-Serif

---

Serif fonts include small lines attached to the end strokes of letters.



Sans-Serif ("without serif") do not include such strokes.



# Activity: CSS Typography

---

Using the `html-template` file provided to you, create an external CSS stylesheet to modify the typography of each of the body paragraphs.

**Use CSS styling to modify the following properties:**

- Font-family
- Font-size
- Font-weight
- Line-height
- Letter-spacing

**Bonus:**

If you finish early, read about TypePlate and see if you can figure out how to incorporate it into your site ([typeplate.com](http://typeplate.com)).

Suggested Time: 10 minutes



# Google Fonts

---

It's easy to incorporate [Google's custom fonts](#) into your site!

Roboto

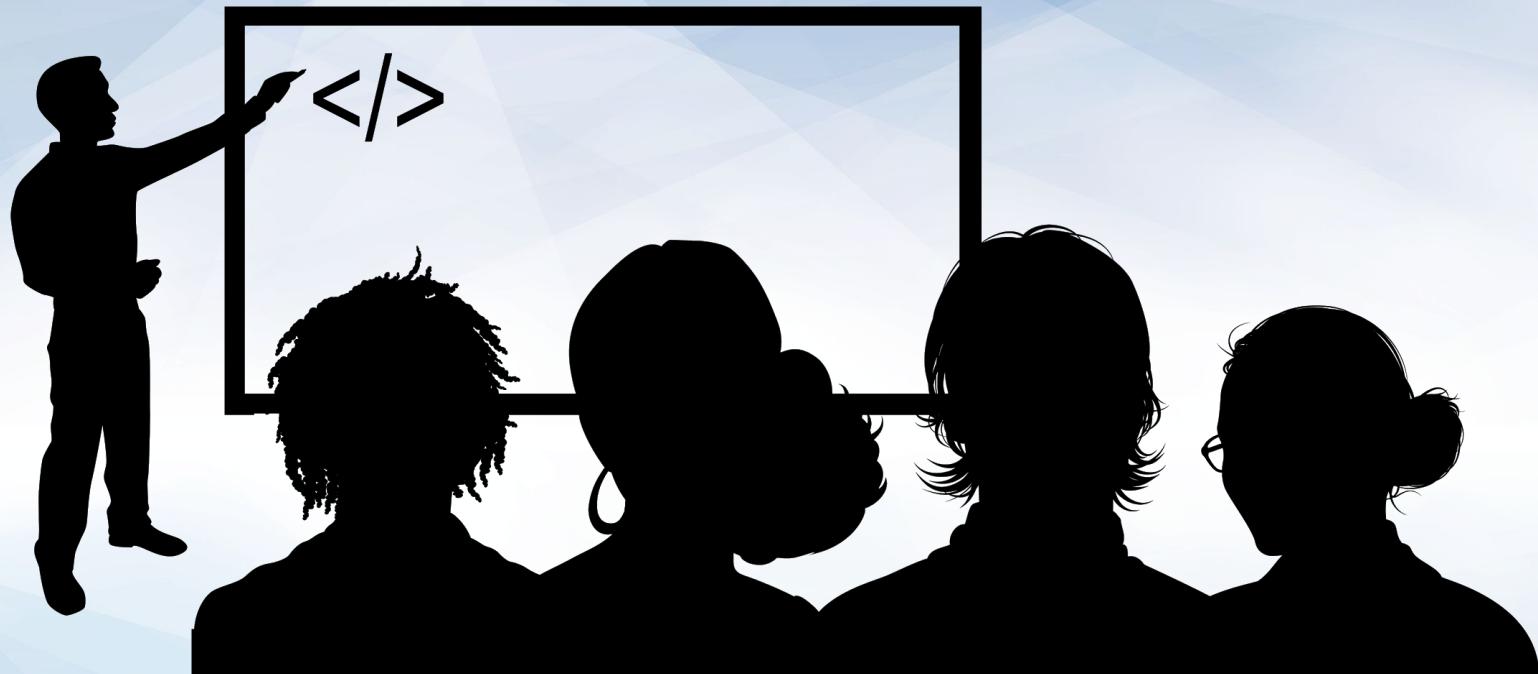
Christian Robertson

All their equipment  
and instruments  
are alive.

**coiny**

Marcelo Magalhães

**A red flare silhouetted  
the jagged edge of a  
wing.**



Instructor Demonstration  
Custom Fonts

# Activity: Custom Google Fonts

---

## Assignment:

Create (or reuse) an HTML document and then incorporate 2–3 custom Google fonts ([fonts.google.com](https://fonts.google.com)) into the page.

## Bonus:

If you finish early, read about Font Awesome icons ([fontawesome.com](https://fontawesome.com)) and see if you can figure out how to incorporate one onto your page.

Suggested Time: 10 minutes



# Take a Break!

---



# Pseudo Styles

# Pseudo-Classes

---

CSS provides keywords that can be added to selectors. These keywords can highlight the special states of the selected element.

**Great example:** CSS can hook onto the specific moment when a link is hovered over using the `:hover` pseudo-class.

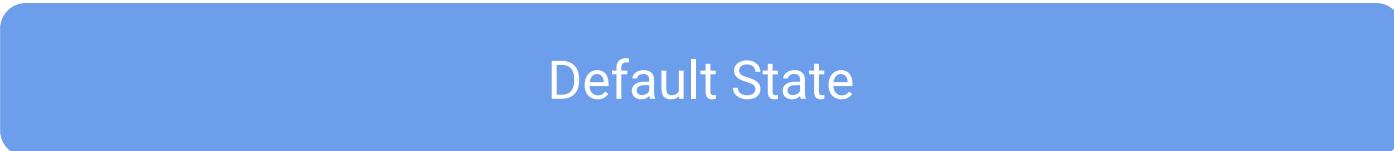
**Complete list of pseudo-classes:** [developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes)

Index of standard pseudo-classes				
<code>:active</code>	<code>:first-child</code>	<code>:last-child</code>	<code>:only-child</code>	<code>:root</code>
<code>:any</code>	<code>:first-of-type</code>	<code>:last-of-type</code>	<code>:only-of-type</code>	<code>:scope</code>
<code>:checked</code>	<code>:fullscreen</code>	<code>:left</code>	<code>:optional</code>	<code>:target</code>
<code>:default</code>	<code>:focus</code>	<code>:link</code>	<code>:out-of-range</code>	<code>:valid</code>
<code>:dir()</code>	<code>:hover</code>	<code>:not()</code>	<code>:placeholder-shown</code>	<code>:visited</code>
<code>:disabled</code>	<code>:indeterminate</code>	<code>:nth-child()</code>	<code>:read-only</code>	
<code>:empty</code>	<code>:in-range</code>	<code>:nth-last-child()</code>	<code>:read-write</code>	
<code>:enabled</code>	<code>:invalid</code>	<code>:nth-last-of-type()</code>	<code>:required</code>	
<code>:first</code>	<code>:lang()</code>	<code>:nth-of-type()</code>	<code>:right</code>	

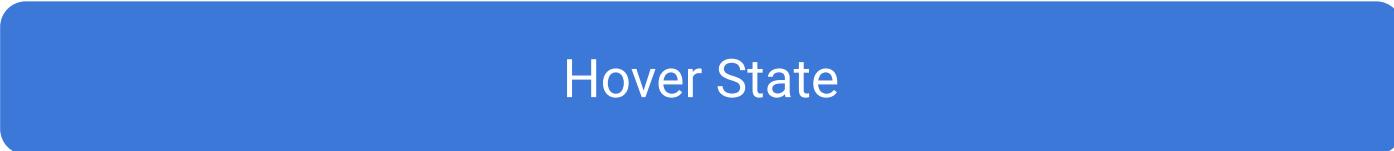
# Button with Various States

---

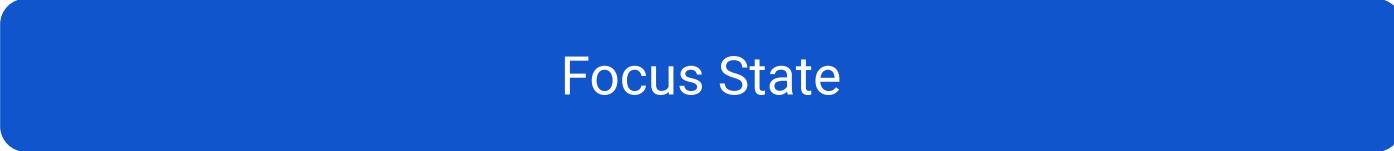
You can use pseudo-classes to define how a button appears in various states.



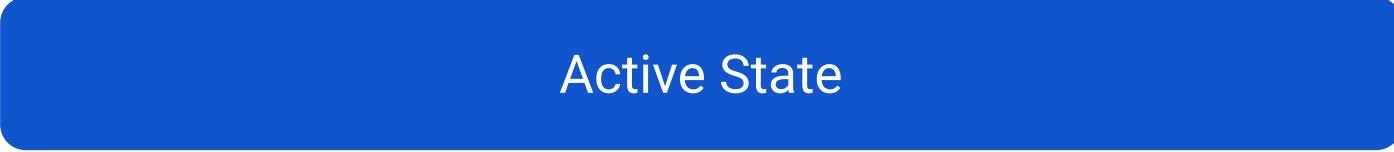
Default State



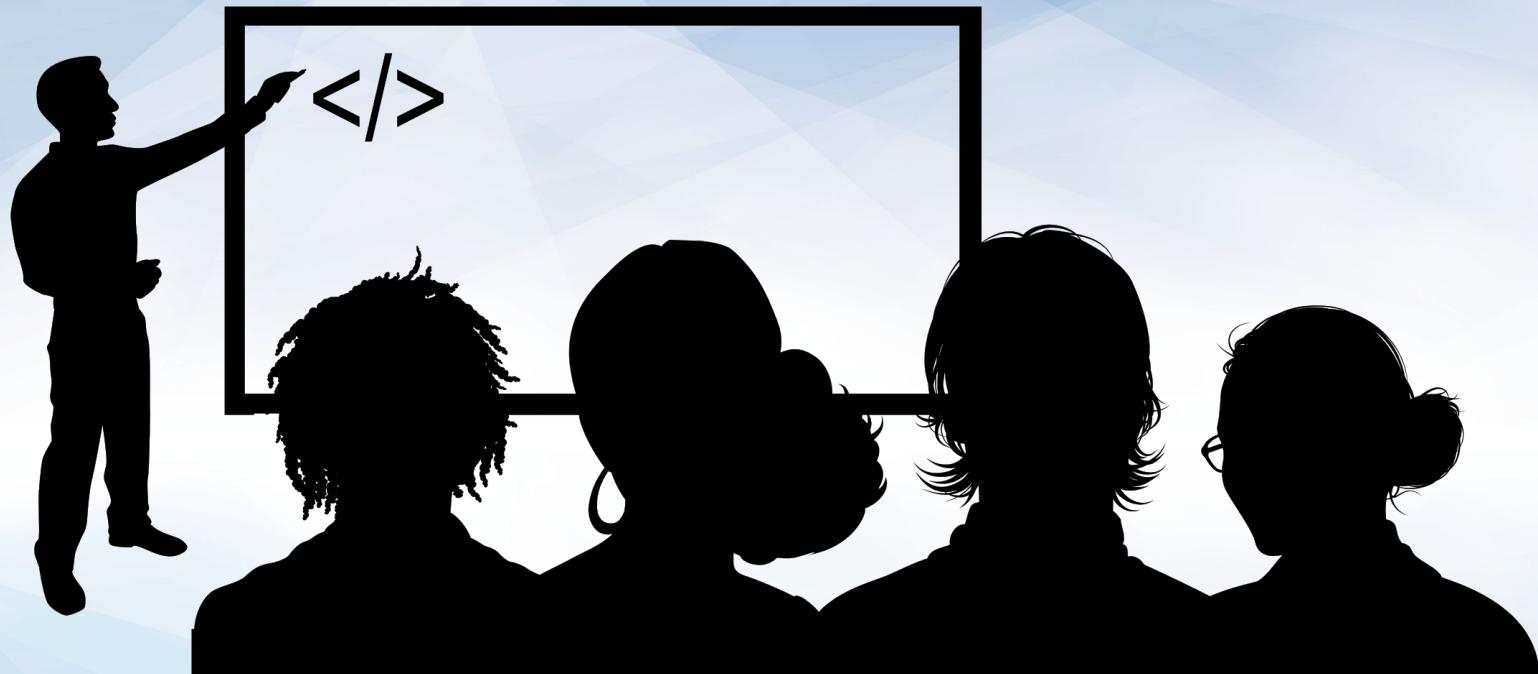
Hover State



Focus State



Active State



## Instructor Demonstration Pseudo Styles

# Activity: Pseudo Styles

---

## Assignment:

Create a series of links and modify the pseudo-classes associated with their default, active, hover, and focus states.

## Bonus:

If you finish early, incorporate a pseudo-class we haven't yet covered.  
Then Google pseudo-elements and try to incorporate one as well.

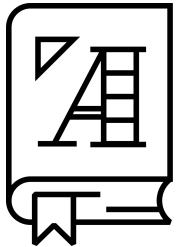
Suggested Time: 5 minutes



# Beyond CSS Hacking

# Quick Refresher on HTML/CSS

---



**HTML:** Hypertext Markup Language (Content)

**CSS:** Cascading Style Sheets (Appearance)

**HTML and CSS are the “languages of the web.”** Together, they define both the content and the aesthetics of a webpage, including colors, fonts, and content placement. (JavaScript is the third language of the web: it handles logic, animation, etc.)

# HTML vs. HTML/CSS

---

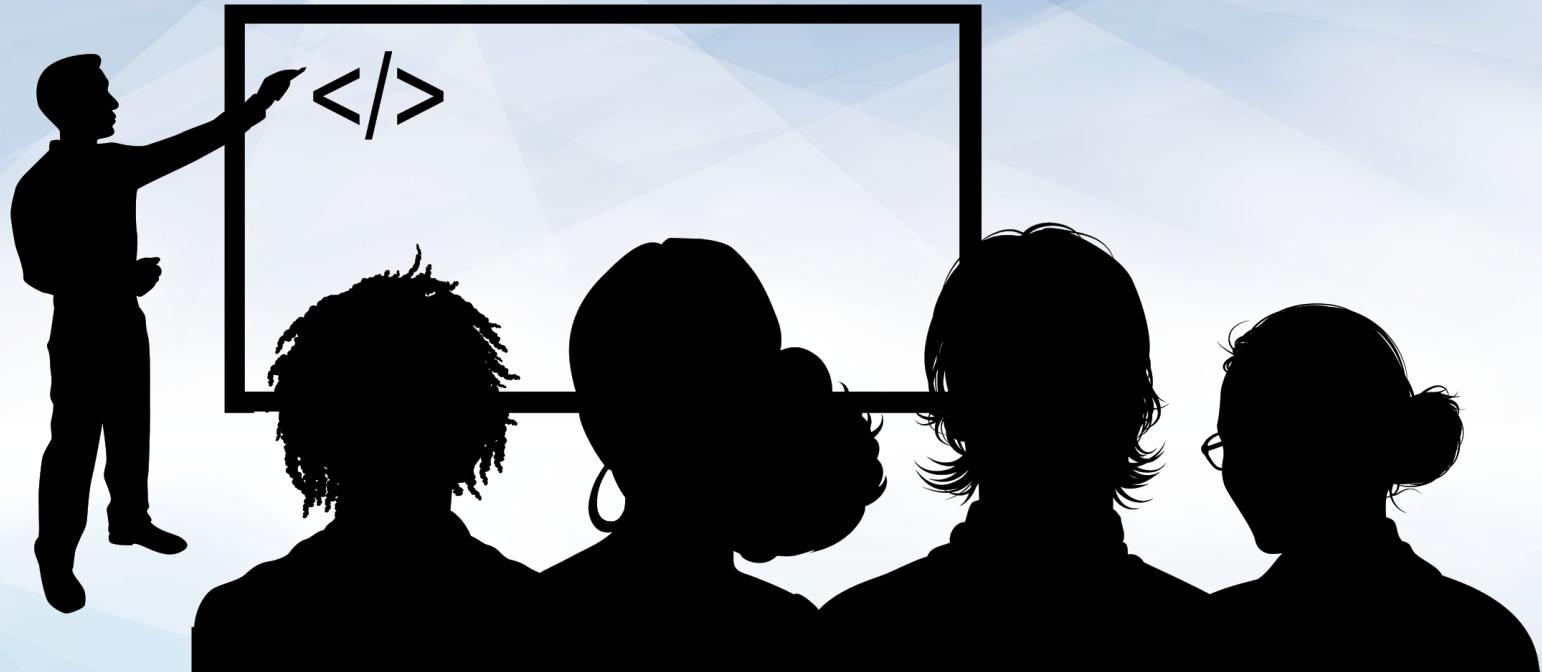
HTML Alone	HTML and CSS
Like writing papers in Notepad.	Like writing papers in Microsoft Word.
Used to write unformatted text (i.e, content only).	Used both to write the content <i>and</i> format it (color, font, alignment, layout, etc.).
	

# How You (Probably) Feel About CSS



A revelation. There is a better way!





Instructor Demonstration  
Where's the CSS?

# Partner Activity: Where's the CSS?

---

With those sitting nearest to you, take a close look at the Bootstrap-powered HTML document I just sent you.

Be prepared to answer the following two questions:

01

Where does this HTML document draw its styles from?

02

How does our browser know which HTML element to style, and in which way?  
(e.g., button face or background banner)

**HINT:** Think about how Google Fonts work.

Suggested Time: 5 minutes



# Powered by Bootstrap

---

External CSS Link:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>My First Website</title>
5      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
6          bootstrap/3.3.6/css/bootstrap.min.css">
7  </head>
7  <body>
```

Predefined CSS Class (e.g., jumbotron):

```
<div class="jumbotron">
    <h1 class="text-center"><strong>Awesome Header</strong></h1>
    <h2 class="text-center">Smaller Awesome Header</h2>
    <h3 class="text-center">Even Smaller Header</h3>
    <p class="text-center"><a class="btn btn-danger btn-lg" href="#" role="button">
        <span class="glyphicon glyphicon-envelope"></span> Learn more</a></p>
</div>
```

# Powered by Bootstrap

The Coding Bootcamp

Home

About

Contact

Dropdown ▾

Default

Static top

Fixed top

# Awesome Header

Smaller Awesome Header

Even Smaller Header

 Learn more

**class="jumbotron"**



*Placeholder text for the first column.*

Placeholder text for the first column.

Placeholder text for the first column.



*Placeholder text for the second column.*

Placeholder text for the second column.

Placeholder text for the second column.



Using Bootstrap gives us  
*predefined* CSS styles.

# Partner Activity: Predefined CSS

---



Turn to the person next to you and celebrate!



Then discuss the concept of Bootstrap to confirm that you both understand why you are celebrating.



**HINT:** You should be explaining why Bootstrap's predefined CSS a big deal.

**Suggested Time:** 5 minutes



# What Is Bootstrap?

# Bootstrap Definition

---



Bootstrap is a free collection of tools for creating websites and web applications.



It comes with a pre-built design template for typography, forms, buttons, navigation, UI elements, and JavaScript.



Bootstrap documentation: [getbootstrap.com](http://getbootstrap.com)

# Why Use Bootstrap?

---

## Reason #1: UI Kit



Familiarize yourself with the UI features Bootstrap offers by reading the documentation.



Once Bootstrap is active, you can simply copy snippets from the documentation to save yourself the time it takes to create elements from scratch!



# Why Use Bootstrap?

## Reason #2: Mobile Responsiveness



One of the most compelling reasons to use Bootstrap is that it is mobile-responsive by default.

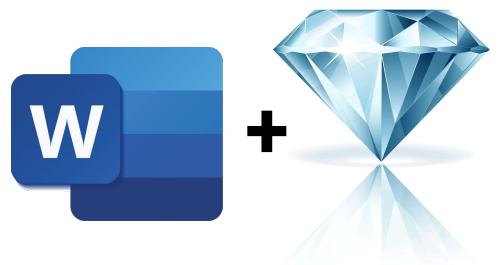


This means that your website will automatically look great when viewed on screens ranging from monitors to tablets to phones.



# HTML/CSS/Bootstrap Analogy

---

HTML Alone	HTML and CSS	HTML/CSS and Bootstrap
Like writing papers in Notepad.	Like writing papers in Microsoft Word.	Like writing papers in Microsoft Word <i>using a template</i> .
Used to write unformatted text (i.e., content only).	Used both to write the content <i>and</i> format it (color, font, alignment, layout, etc.).	You can still customize it, but you're starting with a pre-built style and aesthetic.
 A purple Notepad icon with a white 'N' on it.	 A blue Microsoft Word document icon with a white 'W' on it.	 A blue Microsoft Word document icon with a white 'W' on it, followed by a plus sign and a large blue diamond.

# Key Bootstrap Features

# Bootstrap Components

---

Bootstrap offers a wide range of components that you should consider including in your future web projects. Flip through the documentation and incorporate elements as you see fit.

Bootstrap Components				
Glyphicons	Navs	Badges	Progress bars	Wells
Dropdowns	Navbar	Jumbotron	Media object	
Button groups	Breadcrumbs	Page header	List group	
Button dropdown	Pagination	Thumbnails	Panels	
Input groups	Labels	Alerts	Responsive embed	

# Bootstrap Nav Bars

My First Website

Link 1

Link 2

Link 3

Example:

```
<nav class="navbar navbar-default navbar-fixed-top" role="navigation">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed"
              data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="page-scroll navbar-brand" href="#page-top">My First Website</a>
    </div> <!-- /.navbar-header -->
    <div id="navbar" class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li class="hidden">
          <a href="#page-top"></a>
        </li>
        <li><a href="http://www.google.com" class="nav-btn">Link 1</a></li>
        <li><a href="http://www.facebook.com" class="nav-btn">Link 2</a></li>
        <li><a href="http://www.twitter.com" class="nav-btn">Link 3</a></li>
      </ul>
    </div><!--/.navbar-collapse -->
  </div> <!-- /.container -->
</nav>
```

# Bootstrap Jumbotron

---

# Hello, World!

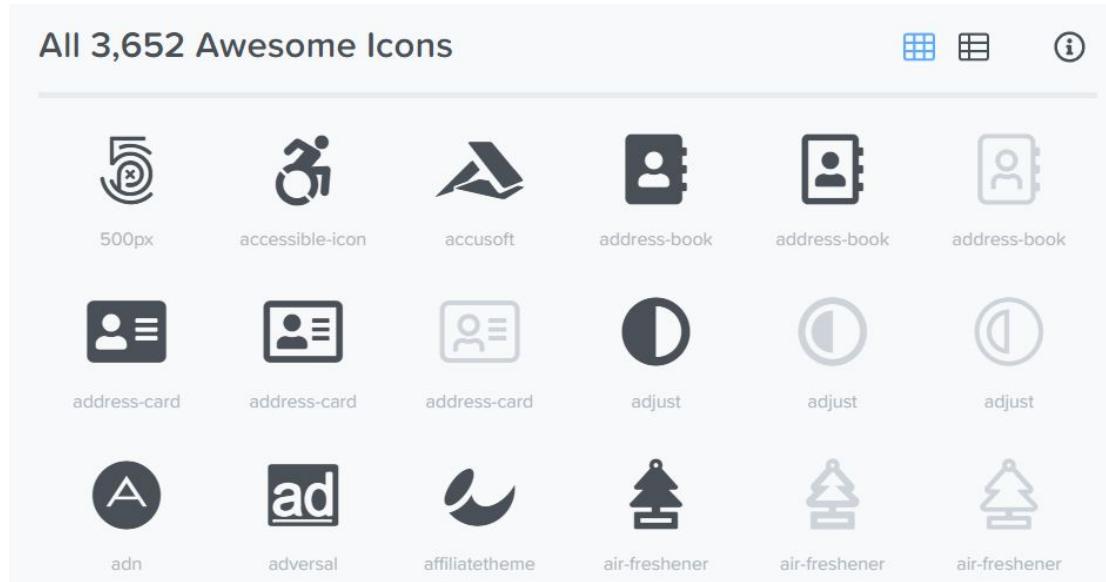
This is a simple hero unit, a simple jumbotron-style component for calling extra attention to featured content or information.

[Learn More](#)

```
<div class="jumbotron">
  <h1>Hello, world!</h1>
  <p>...</p>
  <p><a href="#" class="btn btn-primary btn-lg" role="button">Learn more</a></p>
</div>
```

# Bootstrap and Font Awesome

---



Example: `<span class="fa fa-envelope"></span> Learn more<a>`

# Bootstrap Table

---

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

Example:

```
<table class="table table-striped">
  <tr>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Username</th>
  </tr>
  <tr>
    <td>Mark</td>
    <td>Otto</td>
    </td>@mdo</td>
  </tr>
  ...
</table>
```

# Bootstrap Form

---

**Email address**

**Password**

**File input**

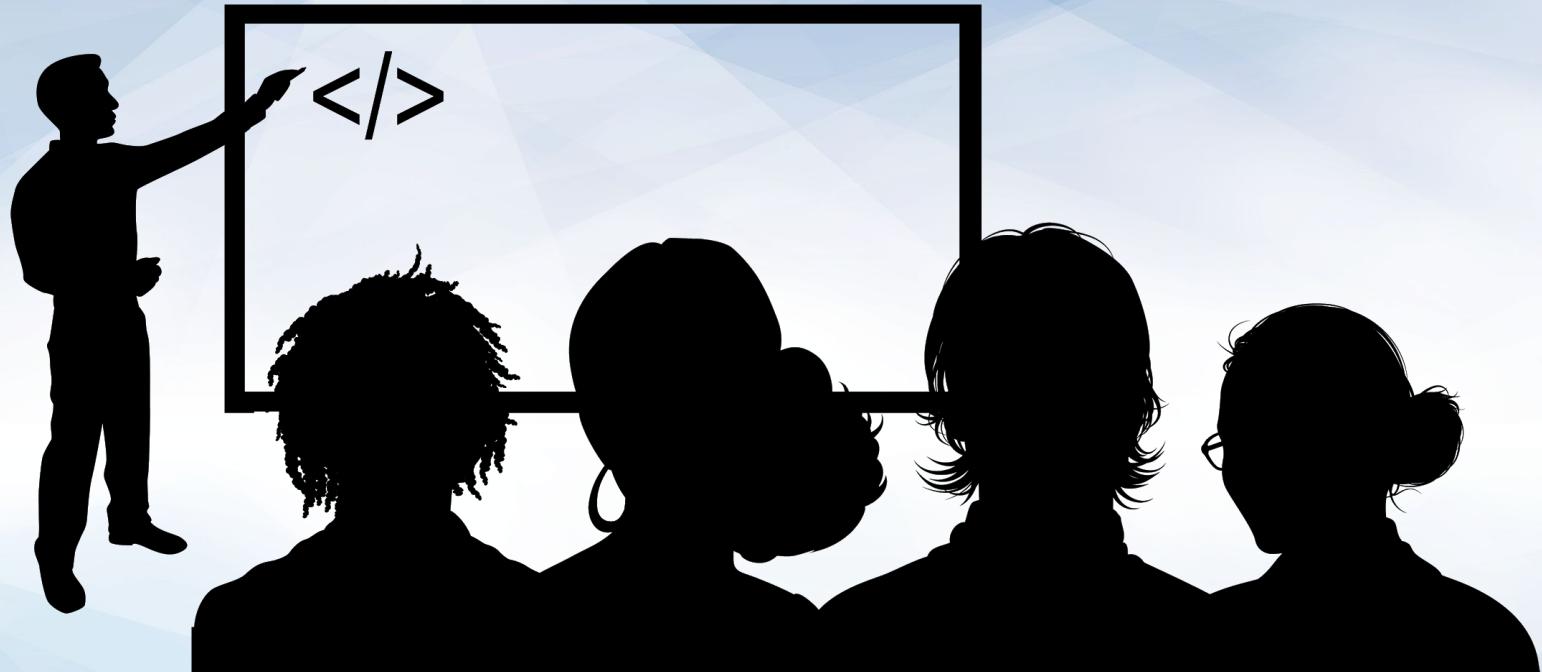
No file chosen

Example block-level help text here.

Check me out

## Example:

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" placeholder="Enter email">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">
  </div>
  <div class="form-group">
    <label for="exampleInputFile">File input</label>
    <input type="file" id="exampleInputFile">
    <p class="help-block">Example block-level help text here.</p>
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox"> Check me out
    </label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```



## Instructor Demonstration Bootstrap Components

# Partner Activity: Bootstrap Components

---

Working with someone sitting near you, do the following:

01

Navigate to the Bootstrap website.

02

Copy the link to the Bootstrap CSS file into one of your old HTML files.

03

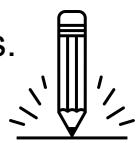
Look through the Bootstrap CSS or Components list and incorporate at least three Bootstrap elements into your page.

04

Slack a screenshot of your page to your section's channel when you finish.

**HINT:** First include a link to Bootstrap's CSS. Then visit the Bootstrap page on CSS or Components.

Suggested Time: 15 minutes



# Takeaways

# Takeaways

---

It's the end of class! By now, you should be able to:

01

Provide a loose definition of *typography* and list a few CSS properties that can be used to style it.

02

Offer a practical example of using pseudo-classes.

03

Understand generally what Twitter Bootstrap is and why predefined CSS frameworks are so powerful.

04

Know how to incorporate the Twitter Bootstrap style library into your website.

# Next Class: Layout Building!

---

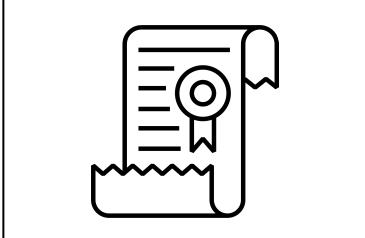
News      Home      About      Contact

## Awesome Stories



### The Greatest Website Ever Made

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla.



### The Greatest Website Ever Made

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla.



Questions?