# *Going Pro with HTML/CSS*

The Coding Bootcamp

# It’s Okay!

# *Admin Items*

# Where to Get Help

- **Practice, Practice, Practice:** Work Individually or in Groups
- **Review In Class Material (Exercises and Slides):**
  <<<<PROVIDE LINK HERE>>>>>
- **Re-Watch Class Videos:**
  <<<<PROVIDE LINK HERE>>>>>
- **In Class Office Hours:** 45 minutes before class, 30 minutes after
- **One-on-One Sessions:** By Announcement through SSM
- **Contact Student Success:** Anytime!

# Homework #1 - Assignment

- **Also, at this point everyone should have access to the homework repository in GitHub.**

  <<<<< LINK HERE>>>>

- **Homework Assignment #1 this week**

  – **Due: (<<<<DATE HERE>>>>)**

# *Today's Class!*

# Today’s Objectives

- Students will make more HTML documents.
- Students will learn to properly use basic HTML tags.
- Students will implement basic CSS styling to HTML documents.

# Know Thyself

**If you are a *complete* beginner to HTML/CSS and Coding:**

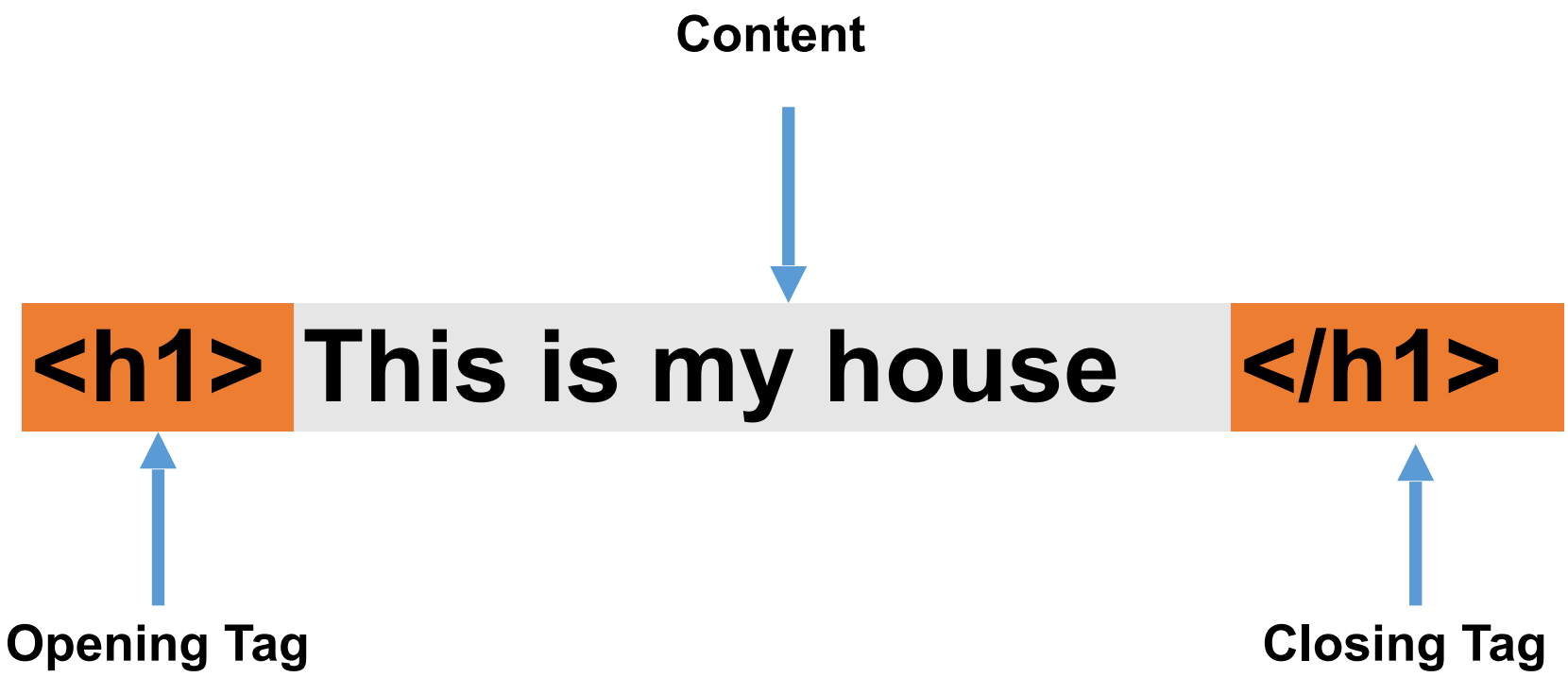
- Continue getting comfortable with HTML.
- Be able to completely write a basic HTML document (like in last class).
- Understand what CSS is, what it's for, and how it works with HTML.
- *Be able to use Git and GitHub to upload code.*

**If you've had past exposure and felt comfortable with the last lesson:**

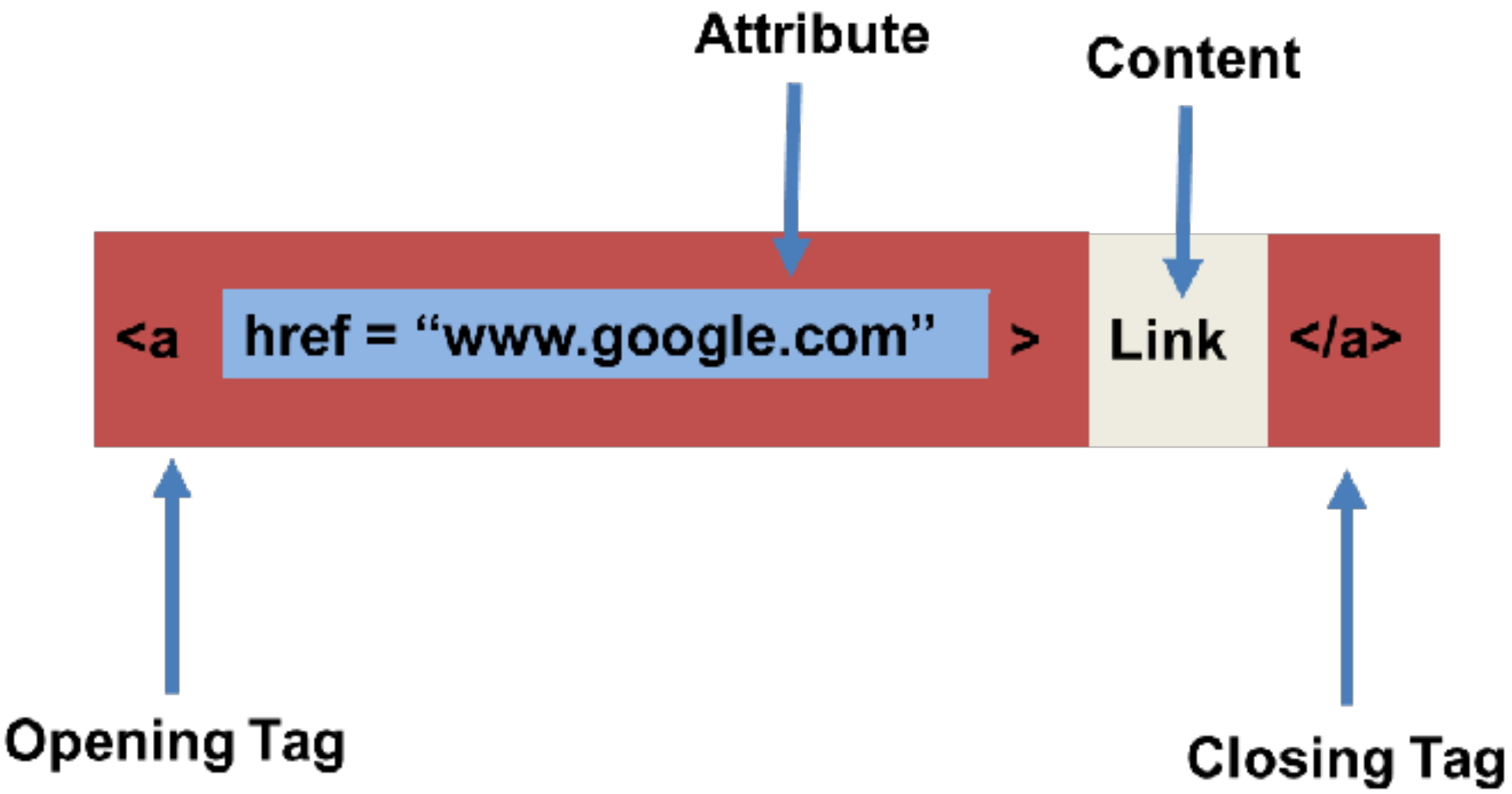
- Aim to build up your skills. Clear up any questions or confusions about HTML.
- Become knowledgeable about a wider range of HTML and CSS tags.
- Be able to selectively apply CSS to specific HTML elements.
- *Be able to use Git and GitHub to upload code.*

# *HTML Round 2*

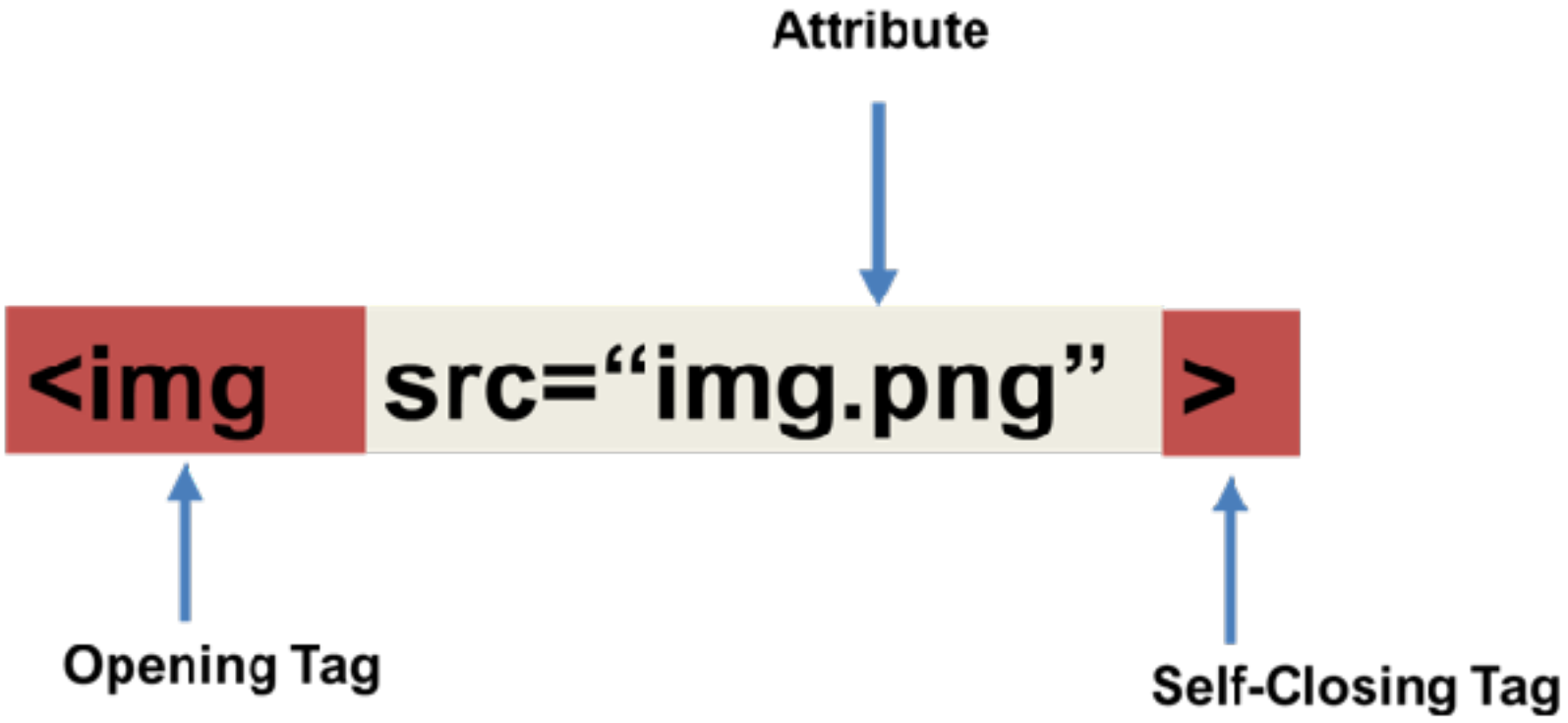
# HTML Syntax (Basic)



# HTML Syntax (with Attribute)



# Tricky Tags (Self-Closing)



# Important Common Tags

**Headings:**

- **<h1> </h1>** - Heading 1 (Largest Heading)
- **<h2> </h2>** - Heading 2 (Next Largest Heading)
- **<h3> </h3>** - Heading 3
- **…**

**Containers:**

- **<html> </html>** - Wraps the entire page
- **<head> </head>** - Wraps the header of the page
- **<body> </body>** - Wraps the main content
- ** ** - Logical Container ***
- ** ** - Wraps individual Paragraphs

**Others:**

- **** (bold), **** (emphasis)
- **<img>** (images)**, <a href>** (links)**, <li>** (list items) **, <title>** (title), **<br>** (line break), **<table>** (tables), **<!-- -->** (comments)

# Less Common Tags

- **All HTML Tags are listed here:** http://www.w3schools.com/tags/

- Don’t try to memorize them! Simply refer back to documentation as needed.

- Other tags:
  - <video> for Videos
  - <audio> for Audio files
  - <embed> for Embedded files
  - <code> for including computer code
  - <header> for headers
  - <nav> for navigation bars
  - <footer> for footers

# HTML for Forms

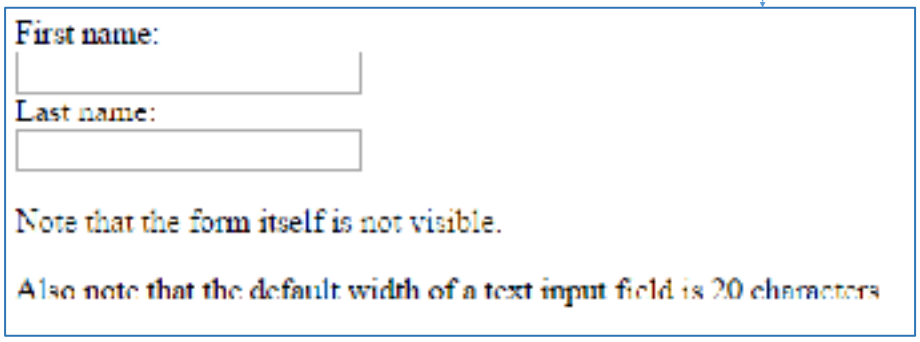
## Common UI (User Interface) Form Elements:

- **<form>** - Creates a form section in HTML
- **<input>** - Input boxes
- **<label>** - Labels for boxes
- **<button>** - Button
- **<textarea>** - Large textbox

# HTML for Forms

```
<!DOCTYPE html>
<html>
<body>

<form>
  First name:<br>
  <input type="text" name="firstname">
  <br>
  Last name:<br>
  <input type="text" name="lastname">
</form>

<p>Note that the form itself is not visible.</p>

<p>Also note that the default width of a text input field is 20 characters.</p>

</body>
</html>
```

Looks like this in the browser



# On Ugly HTML

firstfile.html | Quick.html

```
1  <!DOCTYPE html> <html> <head> <title>1.2.6 Exercise</title> </head>
   <body> <header> <img src="assets/RutgersBanner.png" alt="Rutgers
   Coding Bootcamp" width="797" height="107"> <h1>Student Bio</h1> </
   header> <div> <section> <h2>Your Name</h2> <img src="http://
   placehold.it/200x200" alt="Your Name"> <p>Write a short paragraph
   or two about yourself, or use placeholder text from <a href="http://
   www.lipsum.com/">www.lipsum.com</a></p> </section> <section> <h2>
   Contact Info</h2> <ul> <li><strong>Email:</strong> <a href="#">
   someplace@gmail.com</a></li> <li><strong>Github:</strong> <a href="#
   ">sampleName</a></li> <li><strong>Portfolio:</strong> <a href="#">
   coming soon</a></li> </ul> </section> </div> </body> </html>
```

- Don’t do this… Use proper indentation and sectioning.
- Readable code is easier to maintain.
- Invest time to get better about this now. It will pay dividends!

## Assignment

In this activity, you'll create a student bio using HTML. You will then add, commit, and push your completed HTML to GitHub for the world to see.

Additional instructions, sent via Slack.

# > YOUR TURN!

## Student Bio

### Your Name

Write a short paragraph or two about yourself, or use placeholder text from www.lipsum.com

### Contact Info

- **Email**: someplace@gmail.com
- **Github**: sampleName
- **Portfolio**: coming soon

# *CSS Stylin'*

# HTML / CSS Definitions (*yawn* unimportant)

- **HTML:** Hypertext Markup Language – (Content)
- **CSS:** Cascading Style Sheets – (Appearance)
- **HTML/CSS are the "languages of the web."** Together they define both the content and the aesthetics of a webpage – handling everything from the layouts, colors, fonts and content placement. (JavaScript is the third – handling logic, animation, etc.)

# HTML / CSS Analogy

## HTML Alone

- Like writing papers in “Notepad.”
- Can only write unformatted text.

## HTML / CSS

- Like writing papers in Microsoft Word.
- Can format text, page settings, alignment, etc. based on “highlighting” and menu options.

# Basic HTML Page

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>My First Website!</title>
</head>

<body>

    <h1>Awesome Header</h1>
    <h2>Smaller Awesome Header</h2>
    <h3>Even Smaller Header</h3>

    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
        incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
        exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
    <img src="https://vignette.wikia.nocookie.net/Bill_Murray.jpeg/" alt="bill murray" width="25%" >

    <h3>Menu Links</h3>
    <ul>
        <li><a href="http//www.google.com">Google</a></li>
        <li><a href="http//www.facebook.com">Facebook</a></li>
        <li><a href="http//www.twitter.com">Twitter</a></li>
    </ul>

</body>
</html>
```

# Basic HTML Page - Result

# Awesome Header

## Smaller Awesome Header

### Even Smaller Header

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quidem consequatur unde aut dolores odio hic, accusamus recusandae ipsam illum enim voluptatibus obcaecati totam tempora eum quod sapiente. Corporis, quidem, culpa?

### Menu Links

- Google
- Facebook
- Twitter

# Basic HTML Page - Result

## Awesome Header

### Smaller Awesome Header

#### Even Smaller Header

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quidem consequatur unde aut dolores odio hic, accusamus recusandae ipsam illum enim voluptatibus obcaecati totam tempora eum quod sapiente. Corporis, quidem, culpa?

**Very boring…**

#### Menu Links

- Google
- Facebook
- Twitter

# Enter CSS

```
<style>
    h1 {
        font-size: 60px;
        text-align: center;
        margin-bottom: 15px;
        text-decoration: underline;
        background color: black;
        color: white;
    }

    h2 {
        font-size: 40px;
        text align: center;
        margin-top: 15px;
        margin-bottom: 15px;
    }

    h3 {
        font-size: 20px;
        text-align: center;
        margin-top: 15px;
    }
    img {
        display: block;
        margin-left: auto;
        margin-right: auto;
    }

    p {
        text-align: center;
        font-size: 20px;
        font-weight: bold;
    }

    ul {
        text-align: center;
        font-size: 35px;
        list-style-position: inside;
        border-style: solid;
        border-width: 5px;
    }
</style>
```

# Enter CSS - Result

# Awesome Header

## Smaller Awesome Header

### Even Smaller Header

**Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quidem consequatur unde aut dolores odio hic, accusamus recusandae ipsam illum enim voluptatibus obcaecati totam tempora eum quod sapiente. Corporis, quidem, culpa?**

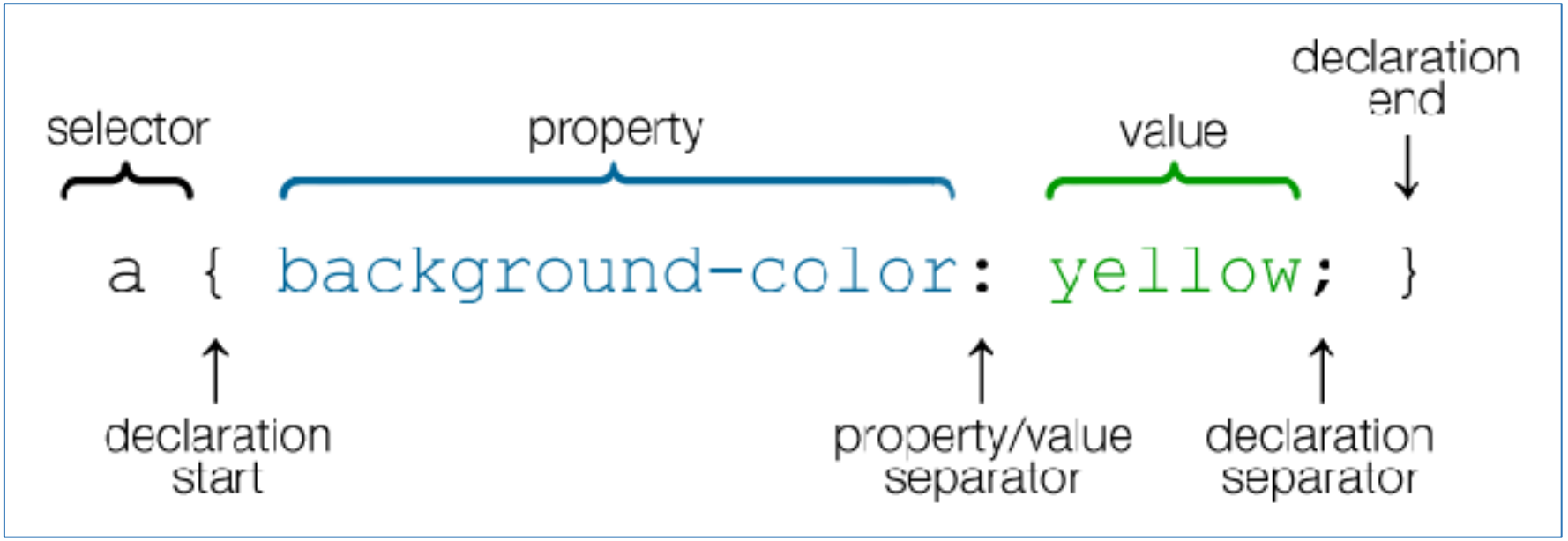
**Menu Links**

- Google
- Facebook
- Twitter

# 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.



# CSS Example

- In the below example the “Header” would be turned blue and MUCH larger because of the CSS.

- We can incorporate an element’s class or ID to apply a CSS style to a particular part of the document.
  - Just remember to include the necessary symbol before the CSS: “.” for class, “#” for ID.

**Example (HTML):**

```
<p class=“bigBlue”>Header</p>
```

**Example (CSS):**

```
.bigBlue
{
   font-size: 100px;
   color: blue;
}
```

# Key CSS Attributes

**Font / Color:**

- **color**: Sets color of text.
- **font-size**: Sets size of the font.
- **font-style**: Sets italics.
- **font-weight**: Sets bold.

**Alignment / Spacing:**

- **padding (top/right/bottom/left):** Adds space between element and its own border.
- **margin (top/right/bottom/left):** Adds space between element and surrounding elements.
- **float:** Forces elements to the sides, centers, or tops.

**Background:**

- **background-color:** sets background color.
- **background-image:** sets background image.

# Powerful Duo

Believe it or not, HTML / CSS is all you need to develop a vivid, full-blown website.

# INSTRUCTOR DEMO

***Instructor: Demo***

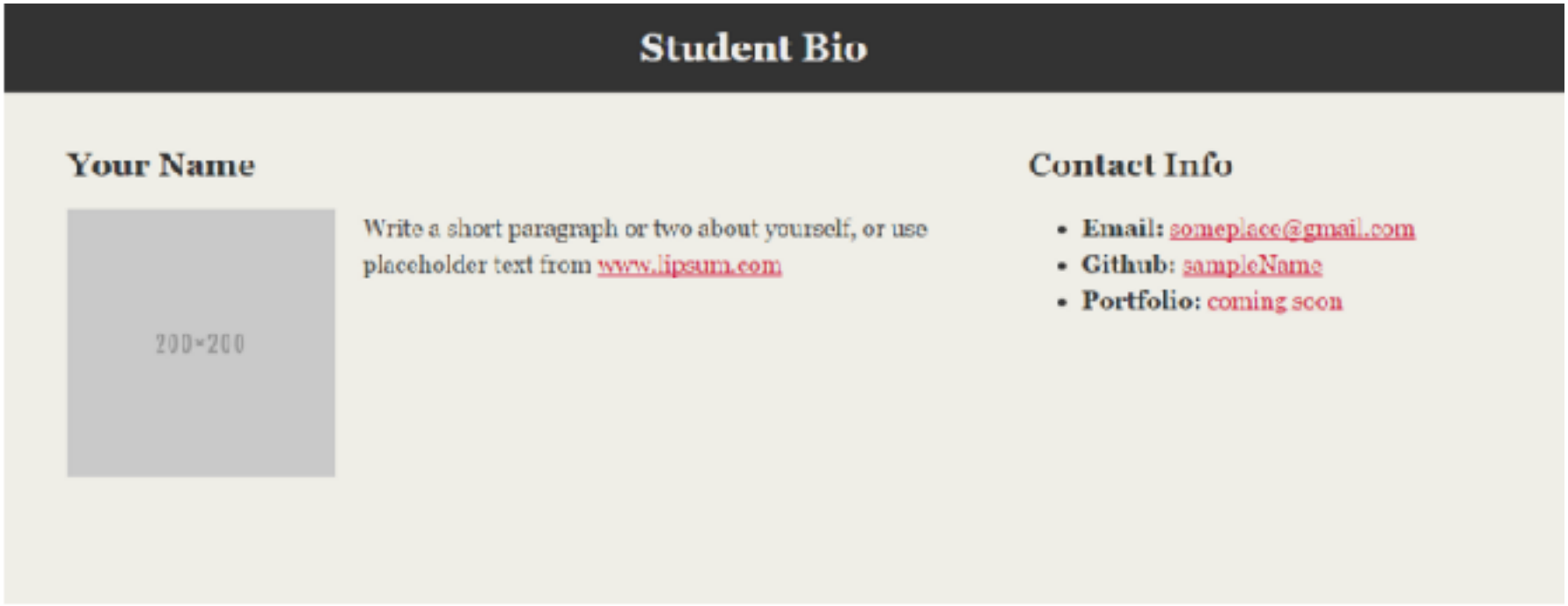
*(quick-example-internal-css.html | 05 -BasicCSS)*

## Assignment

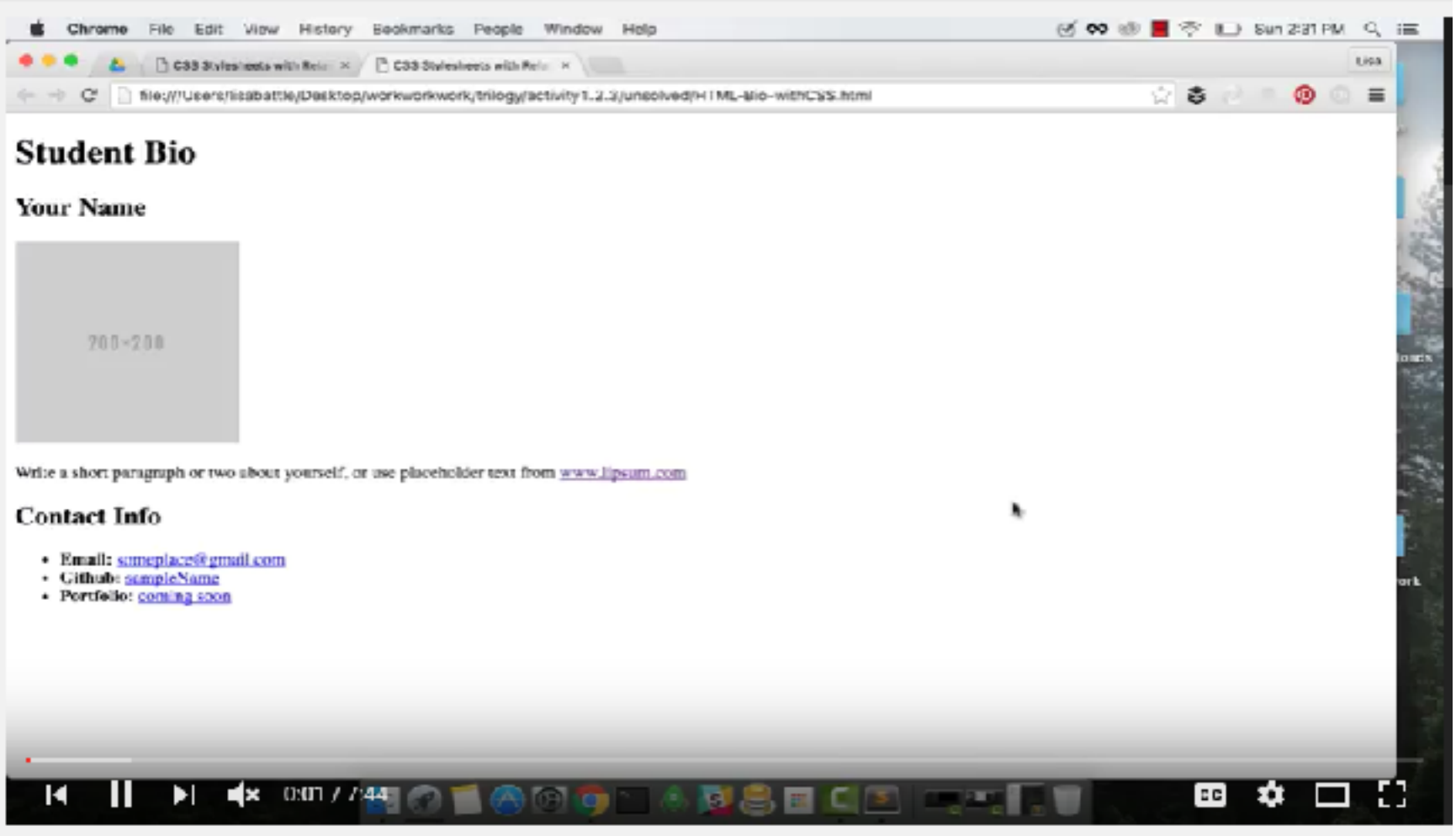
In this activity, you'll upgrade your previous HTML bio-page using CSS style rules. Once you're done, commit and push up your changes to GitHub.

We'll send you additional instructions via Slack.

# > YOUR TURN!


Student Bio
Your Name
200×200
Write a short paragraph or two about yourself, or use placeholder text from www.lipsum.com
Contact Info
Email: someplace@gmail.com
Github: sampleName
Portfolio: coming soon

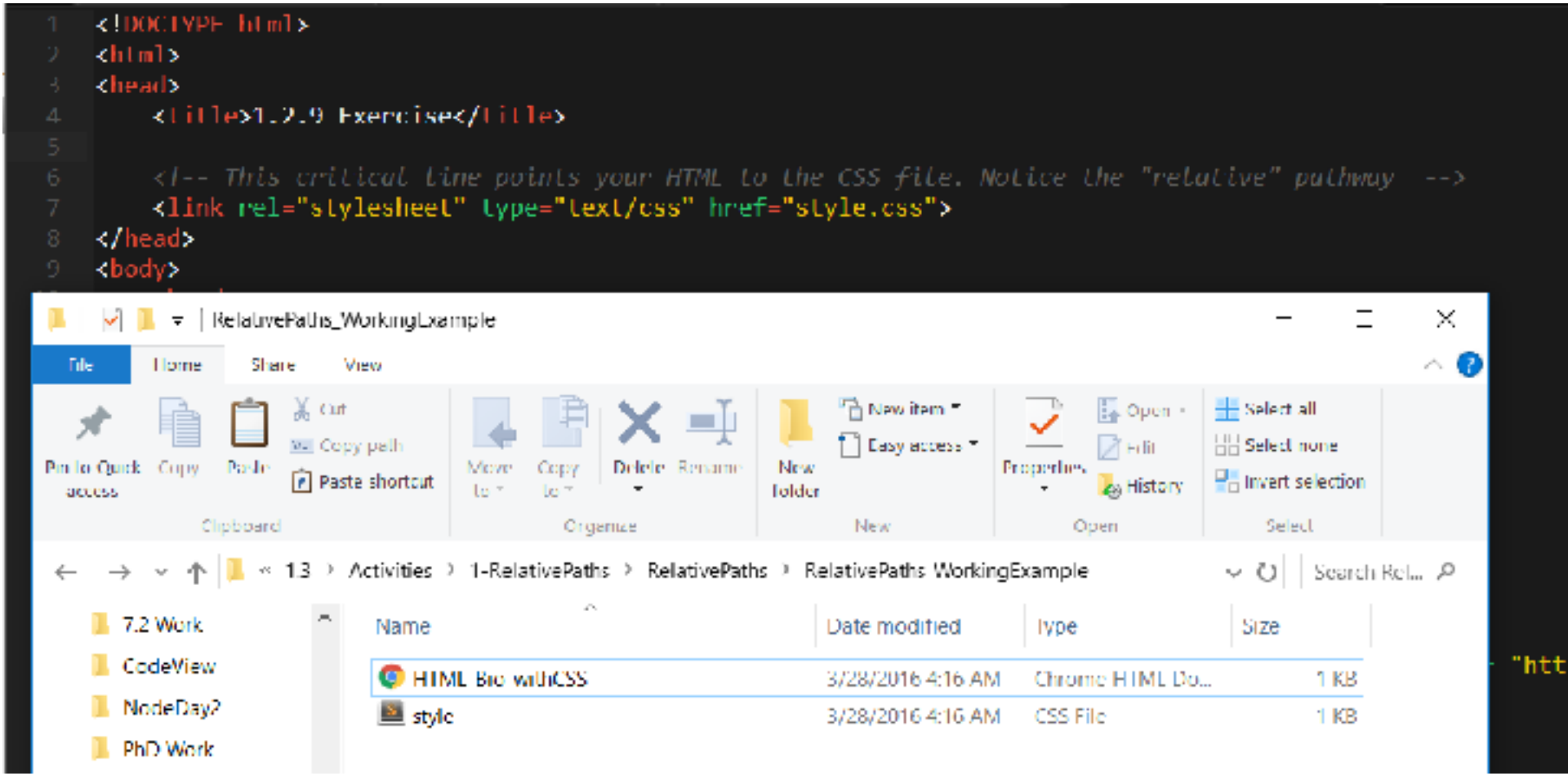
# Video Walkthrough!!




https://www.youtube.com/watch?v=kMBinXTCrXI&list=PLgJ8UgkiorCnMLsUevoQRxH8t9bt7ne14&index=2

# *Relative File Paths*

# Relative File Paths




- **Relative file paths** connect us with other files in our working directory. In this case, style.css is in the same folder as our html document.

# Absolutely No Absolute Paths

## VERY VERY BAD

```
<!-- BAD!!!! -->
<link rel="stylesheet" href="D:/trilogy/FullStack-Lesson-Plans/02-lesson-plans/01-
html css three days/1 Class Content/1.3/Activities/1 RelativePaths/RelativePaths/
RelativePaths WorkingExample/style.css">
```

**ALWAYS USE RELATIVE FILE PATHS.**

- If you deploy your sites without them, **all of your links will fail**.
  - The same will happen if you move your project from one folder to another.

- Remember, there is no such thing as a “C:” drive on the internet.

# Quick Demo

***Instructor: Demo***

*(RelativePaths_DEMO | 1-RelativePaths)*

## Assignment

1. Unzip the folder sent to you via Slack.

2. Edit the HTML files in all of the “RelativePaths” folders. You need to write relative paths that link the HTML documents with CSS stylesheets.

Tip: Check out the “RelativePaths_WorkingExample” folder.

# *LUNCH (30 mins)*

# *Box Model*

# Boxes Upon Boxes

**In CSS, every element rests within a series of boxes.**
Each box has customizable space properties:
margin, border, and padding.

Typical spacing value: 20px 10px 10px 20px (top, right, bottom, left)

# > YOUR TURN!!

**Suggested Time:** 10 min

```
#box {

    background-color: #1B5792;
    width: 400px;
    height: 440px;
    margin: 10px 30px 20px 50px;
    color: #fff;
    padding: 25px 10px 30px 20px;
    border-style: solid;
    border-width: 22px;
    border-color: #113152;

}
```

**How wide is the blue #box?**

**How tall is the blue #box?**

**Total element width** = content width + left padding + right padding + left border + right border + left margin + right margin

**Total element height** = content height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

# > YOUR TURN!!

```
#box {

    background-color: #1B5792;
    width: 400px;
    height: 440px;
    margin: 10px 30px 20px 50px;
    color: #fff;
    padding: 25px 10px 30px 20px;
    border-style: solid;
    border-width: 22px;
    border-color: #113152;

}
```

**How wide is the blue #box?**

**How tall is the blue #box?**

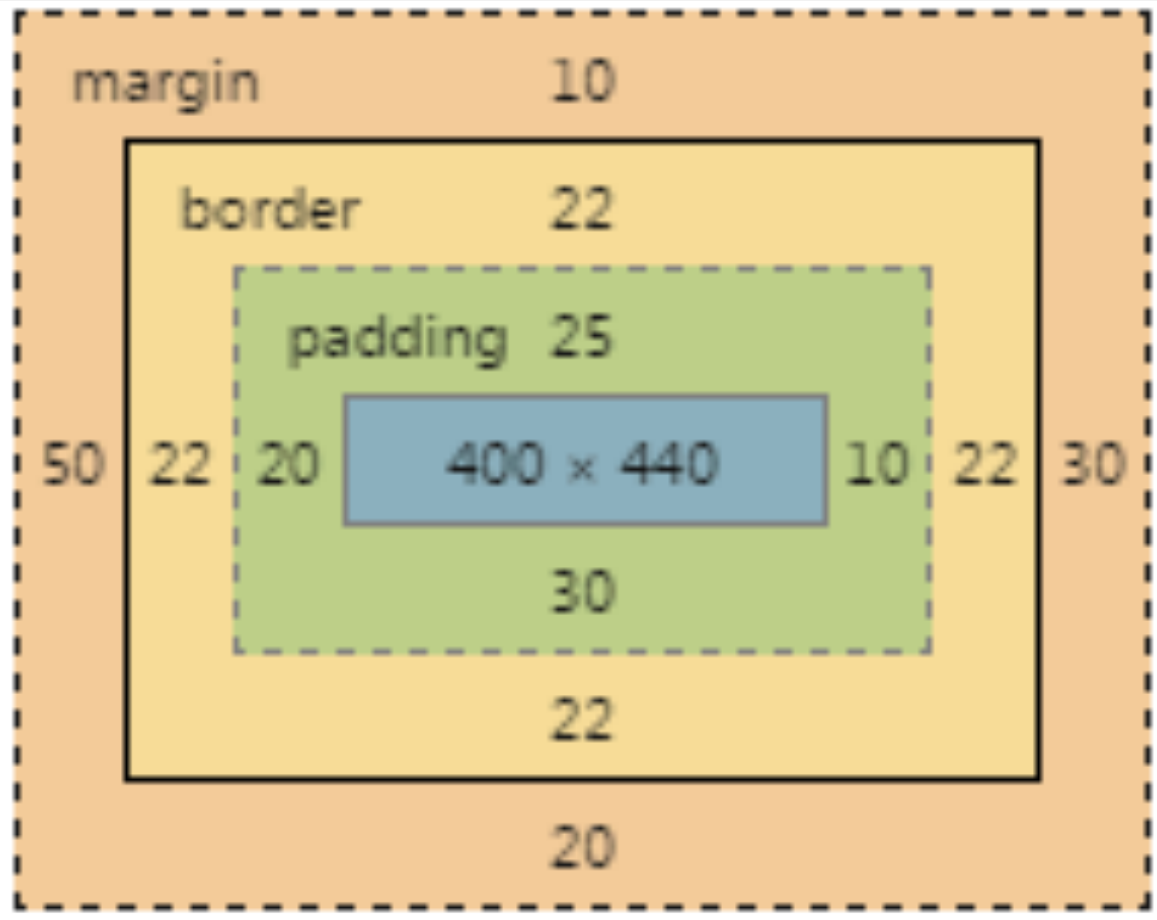
**Total element width** = content width + left padding + right padding + left border + right border + left margin + right margin

**Total element height** = content height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

**Answer**

**Width:** 474 px (no margin), 554 px (with margin)
**Height:** 539 px (no margin), 569 px (with margin)

# > YOUR TURN!!




**Answer**

**Width:** 474 px (no margin), 554 px (with margin)
**Height:** 539 px (no margin), 569 px (with margin)

# *We Be Floatin’*

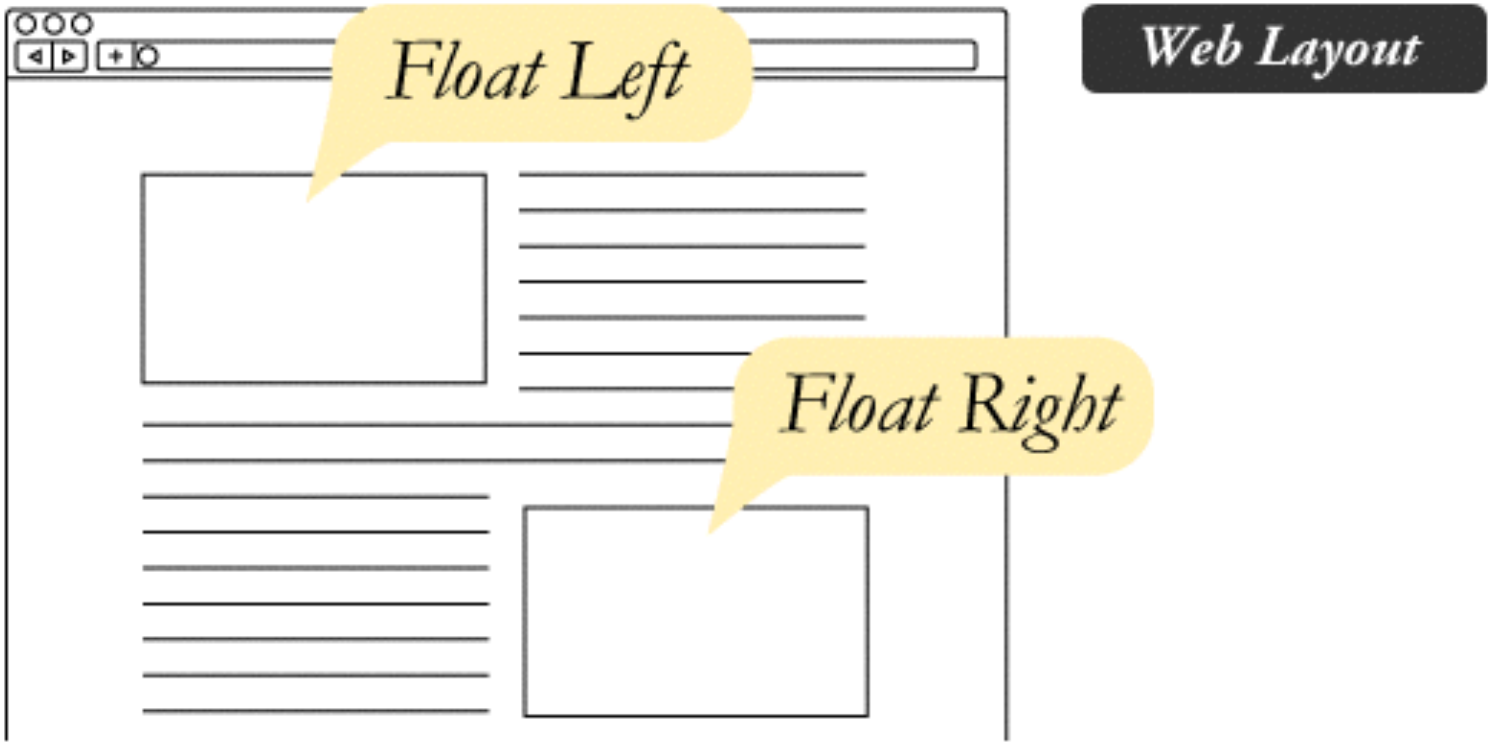
# Take a Facebook Break…

## Warning!

These next topics are fairly “tricky”… **but VERY IMPORTANT.**

Time to channel that inner genius.

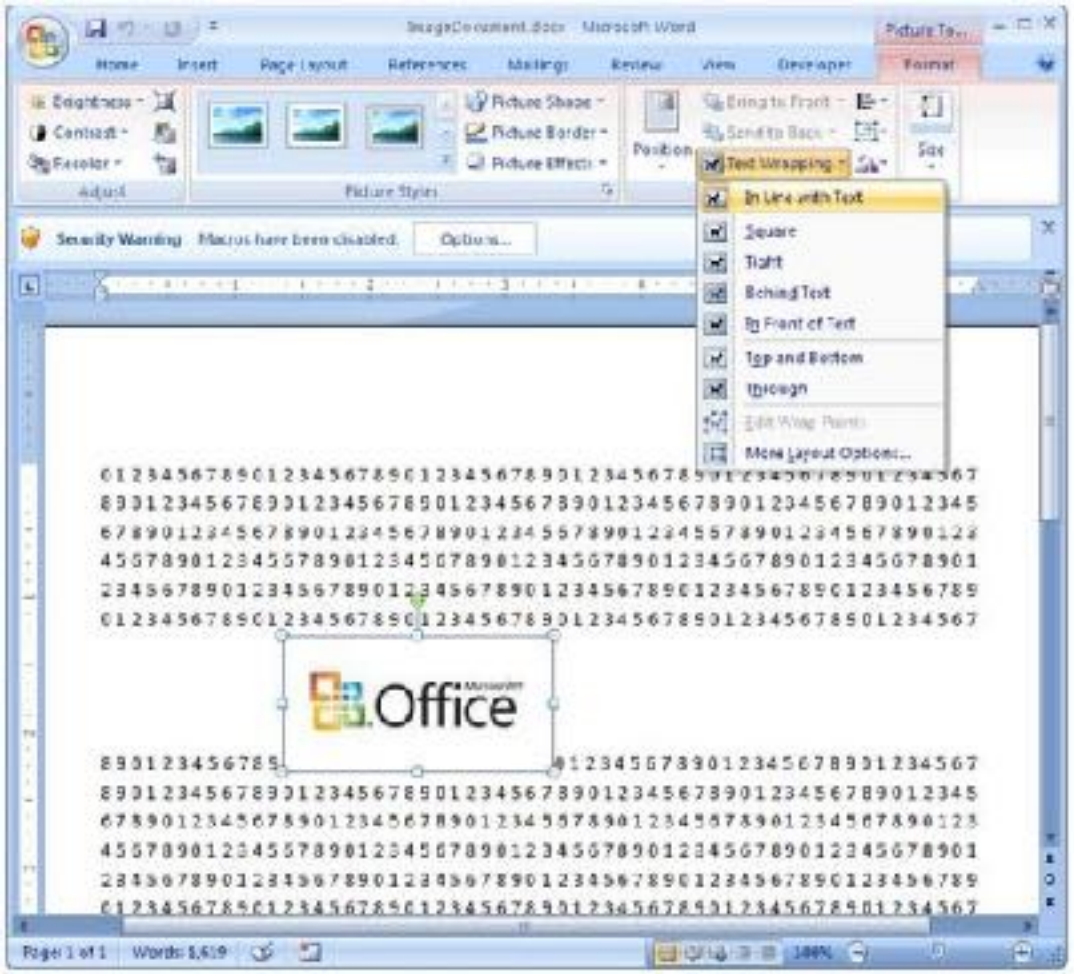
# The Concept of “Flow”




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

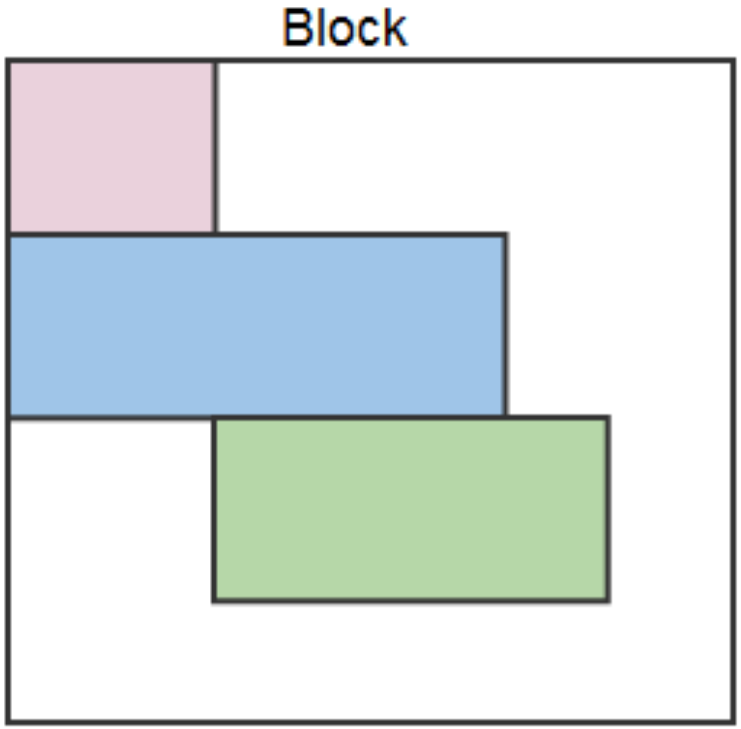
- This means that HTML elements force their adjacent elements to **flow around them**.

# Flow Analogy to MS Word




- This concept of “flow” is very similar to the **wrap-text options** you may be familiar with in Microsoft Word.
- Just as in MS Word, you can have images in-line with text, on-top of text, etc.

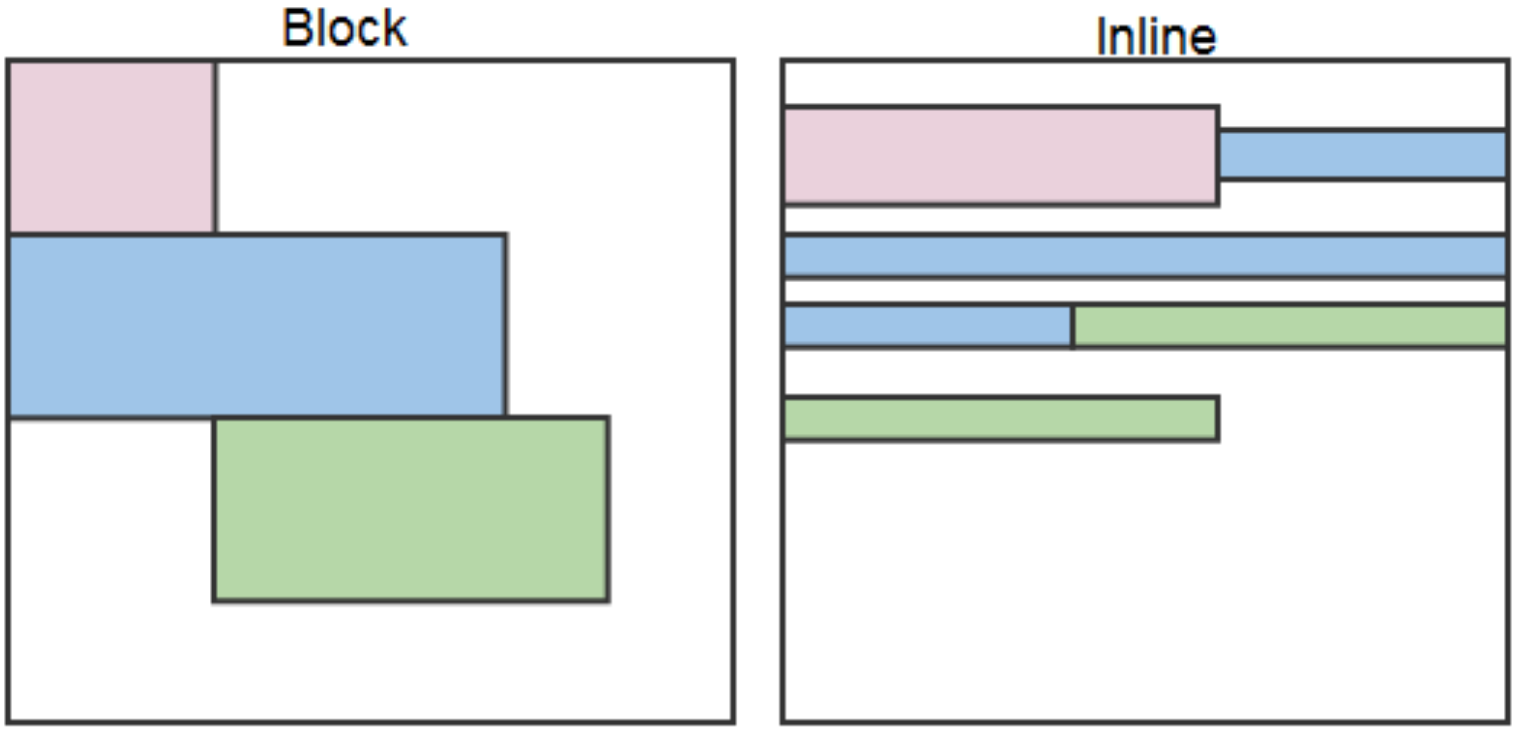
# Block Elements




- By default, web clients render many HTML elements as **block elements.** Paragraphs, headers, divs and more receive this treatment.

- A block element will take up an entire line of space—unless you intervene with CSS properties.

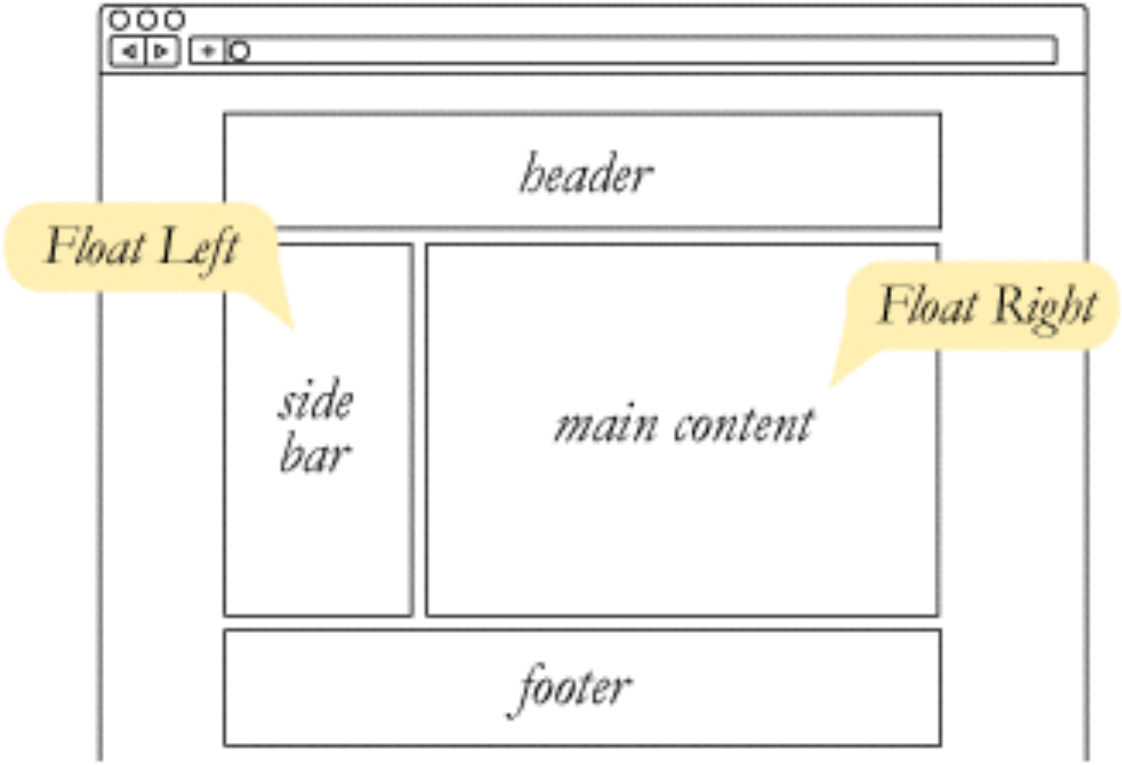
# Block Elements vs. Inline Elements




- Now contrast the block elements with **inline elements**.

- By using **float CSS** properties, we can command our website to display multiple HTML elements adjacently.

# Floating




CSS

```
#sidebar {
    float: left;
}

#main-content {
    float: right;
}
```

- To transform these block elements into inline elements, we use a CSS property called **float.**

- Floats are **necessary** for building web layouts.

# Clearing the Float

- **Floats often get in the way of our layouts**.
- Sometimes we don’t want to give each element the “inline” treatment.

# Clearfix Hack

Uh oh... this image is taller than the element containing it, and it's floated, so it's overflowing outside of its container!

- Sometimes when elements don’t match up in size, we get situations like the above…

# Clearfix Hack

```
<div class="clearfix">
  Much better!
```

- We can get around this by using “the clearfix hack.”

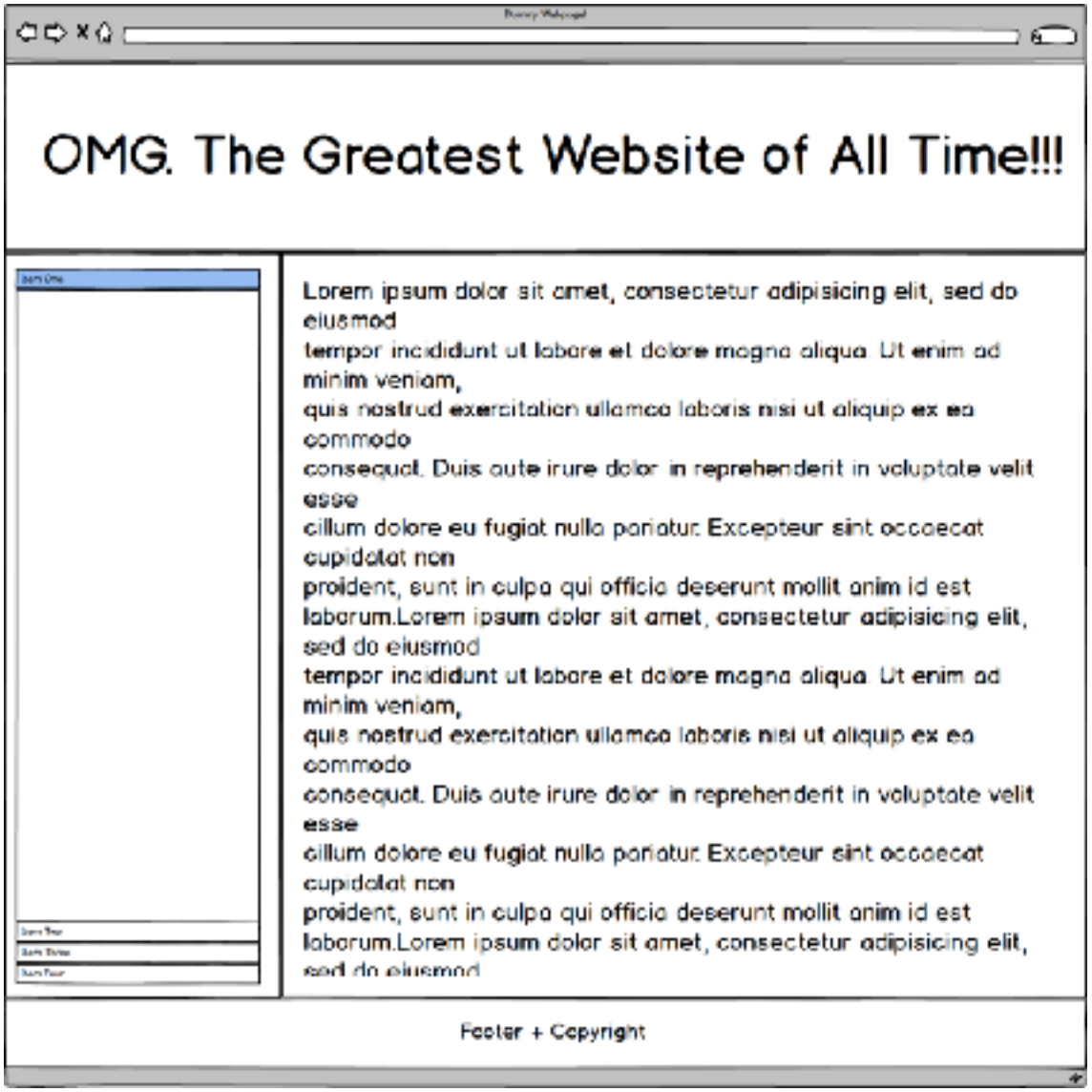
# Clearfix Hack

```
.clearfix::after {
    content: "";
    display: block;
    clear: both;
}
```

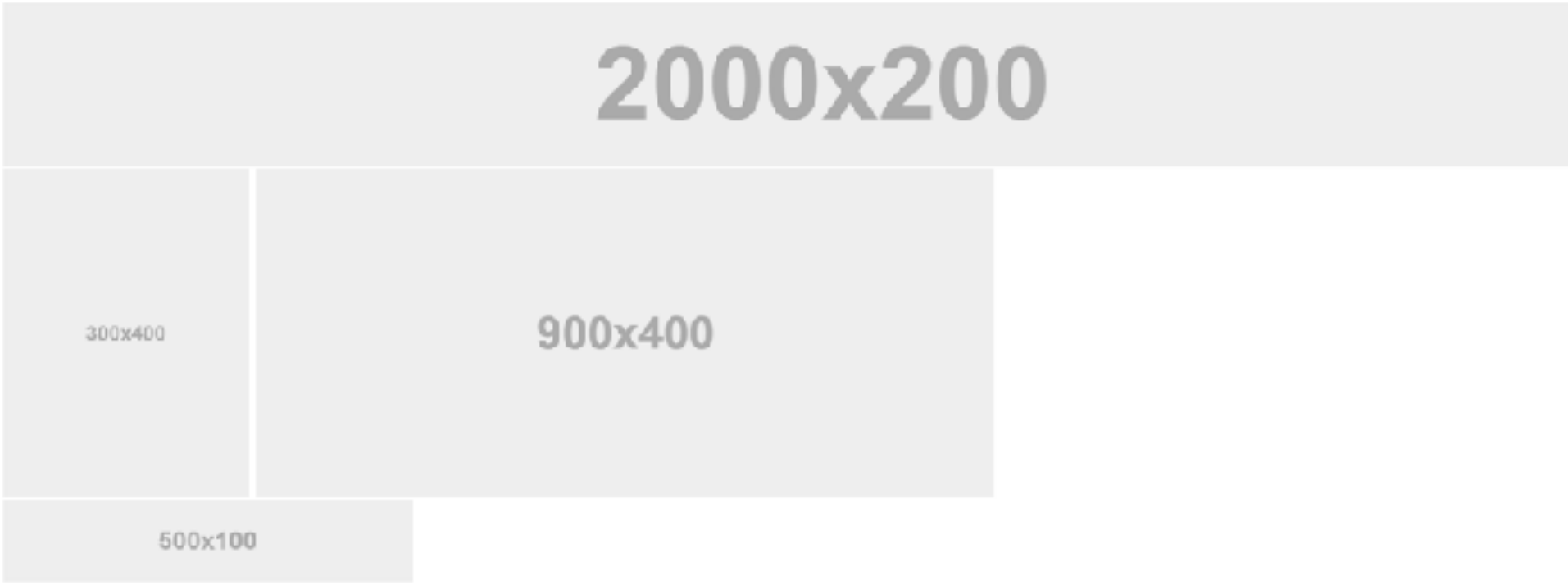
- **::after** is what we call a pseudo-element. We use it to style specific parts of an element.

- This will add an HTML element, hidden from view, after the content of the **“.clearfix”** element. This clears the float.

# Quick Demo!


OMG. The Greatest Website of All Time!!!
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id est laborum.Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id est laborum.Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
Footer + Copyright

# Quick Demo!



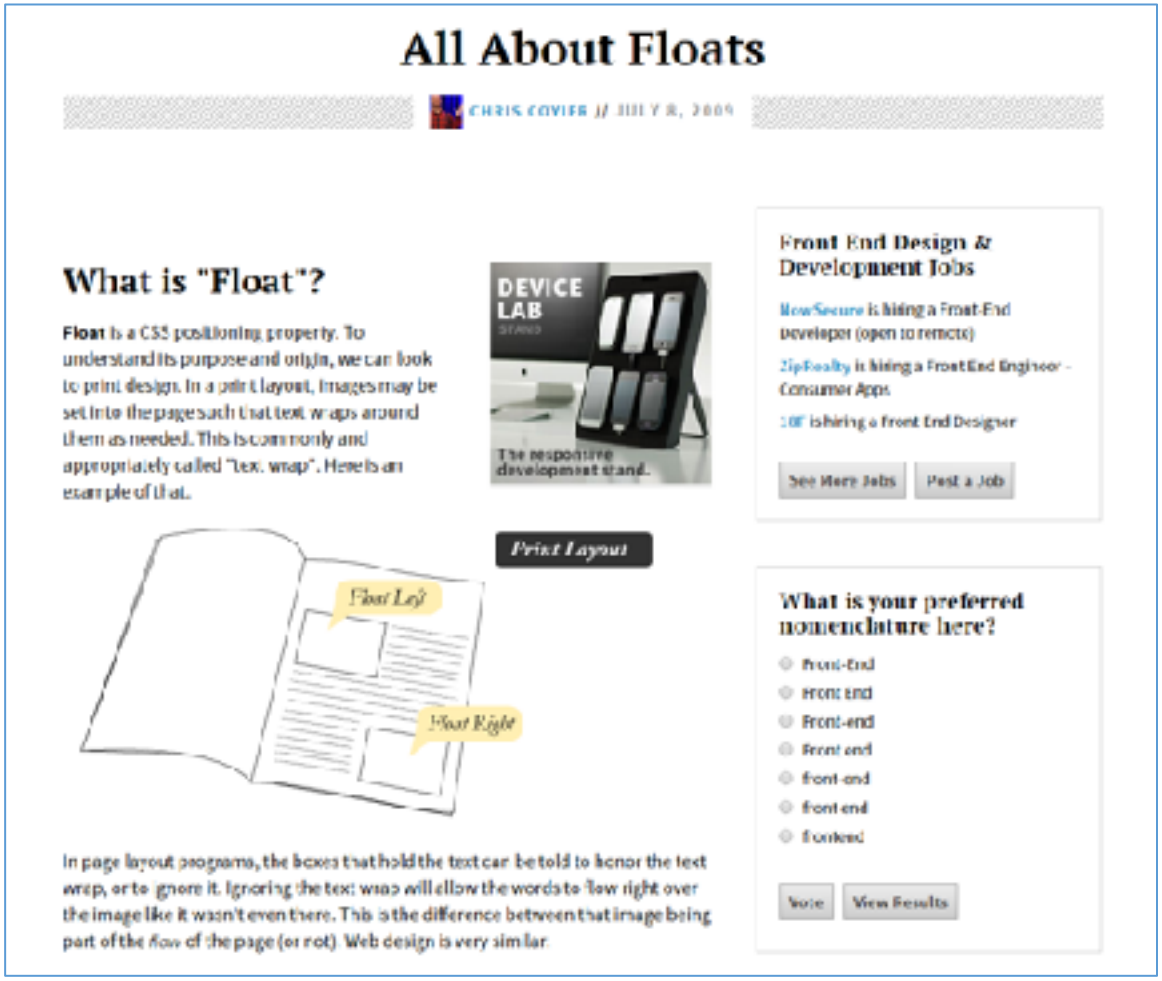
# Quick Demo

***Instructor: Demo***

*(9-FloatExamples)*

# Fantastic Guide on Floats ****

CSS-TRICKS




- **To all serious front-end developers (this is a <u>necessary</u> read):** https://css-tricks.com/all-about-floats/

## Assignment

In this activity, you'll flex your newfound floating skills by creating a conceptual layout. Eyeball the design to your best ability.

Check your Slack for more instructions.

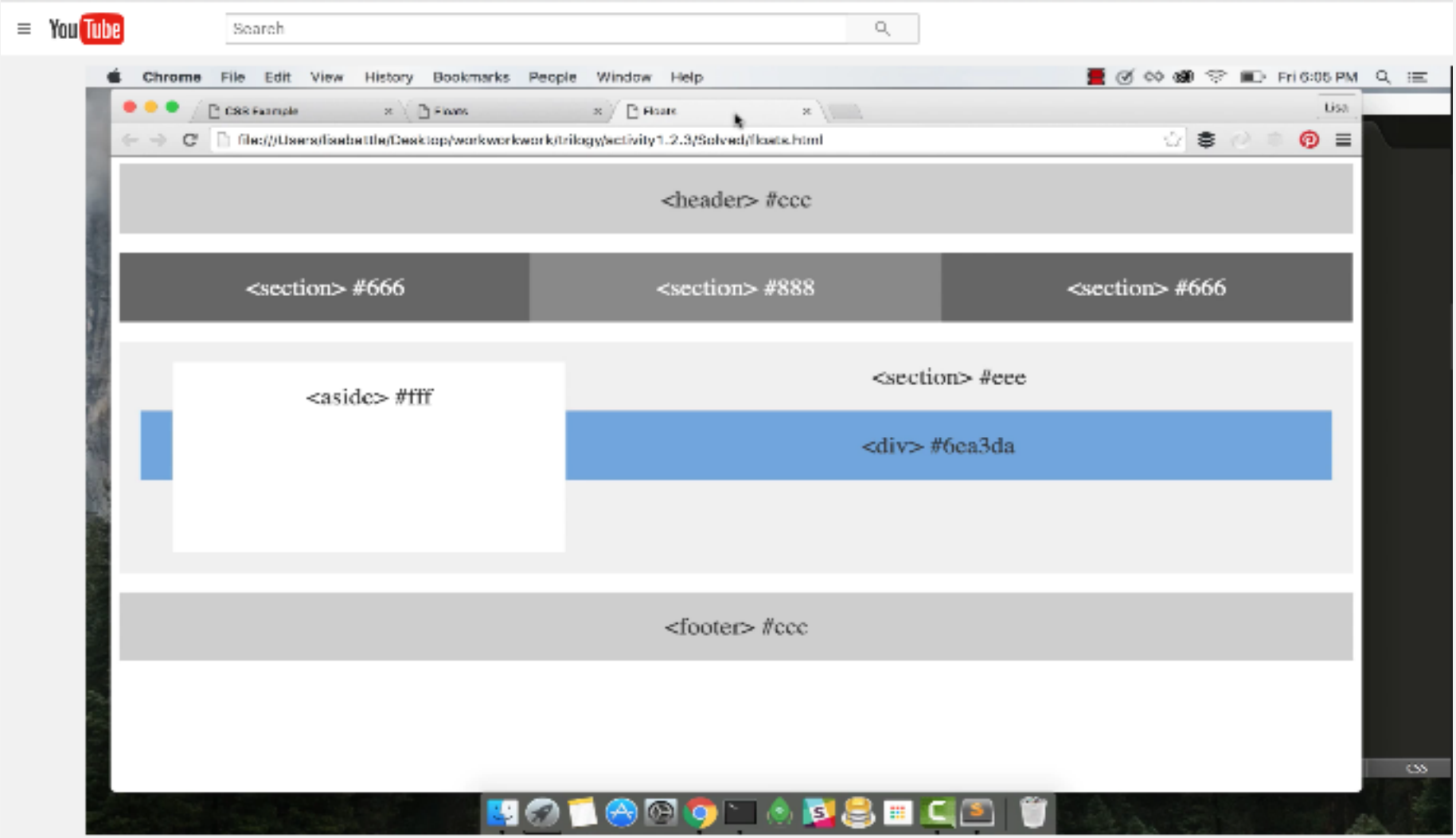
# > YOUR TURN!!

<header> #ccc
<section> #666
<section> #888
<section> #666
<aside> #fff
<section> #ccc
 #6ca3da
<footer> #ccc

# Good work!

**Your brain may rest now…**

# Video Walkthrough! (Highly, HIGHLY Recommend!!!)




**Video Link:** https://youtu.be/0lpxKw6E90Y

# *Recap + Questions*

# Still a Bit Confused?

Remember! We've got video guides for key activities like that last one.

If you feel like you are EVER falling behind, use those online walkthroughs to help catch back up. They are made to be easy to understand.

Still having trouble? Shoot your instructor or one of your TAs a message!
We are here to help you out in whatever way we can!