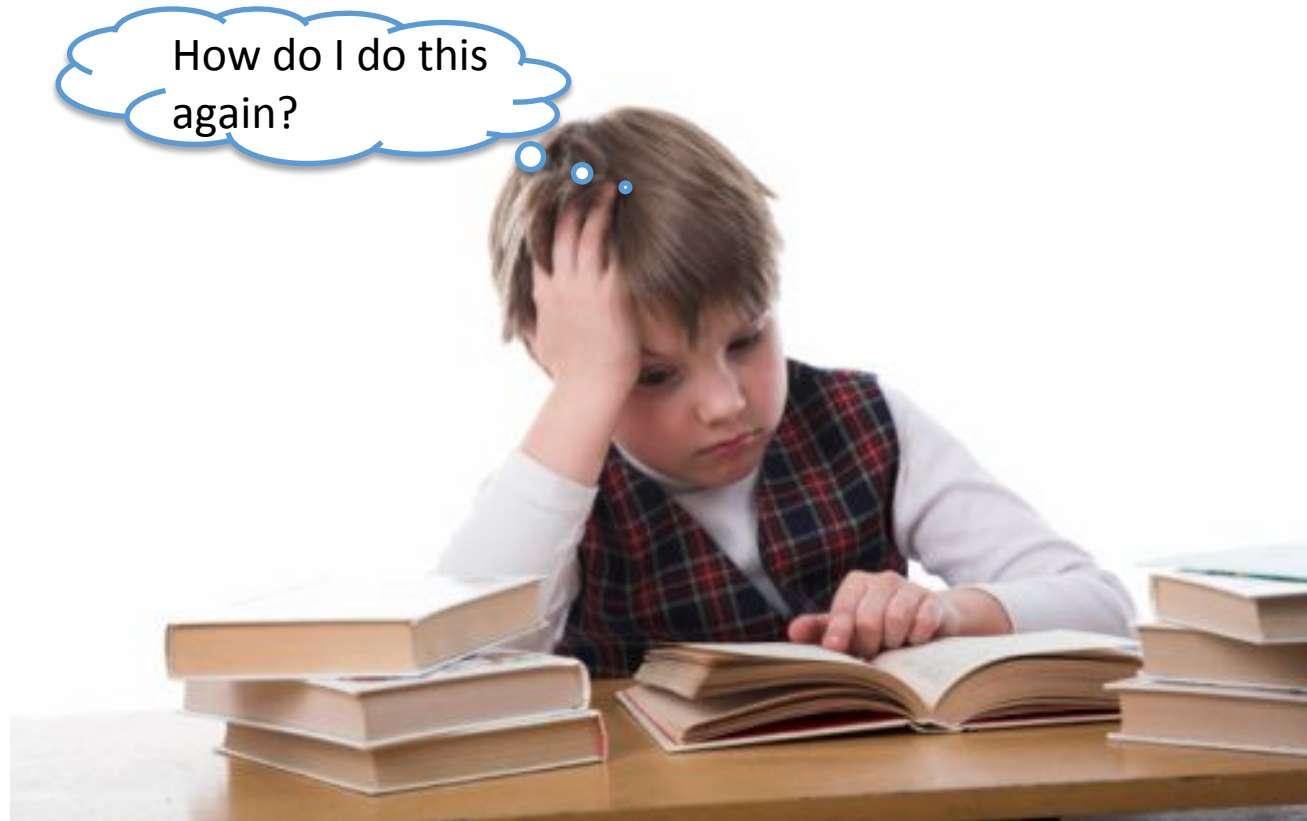


Going Pro with HTML/CSS

The Coding Bootcamp

It's Okay!



Admin Items

Where to Get Help

- **Review In Class Material (Exercises and Slides):**

<https://smu.bootcampcontent.com/SMU-Coding-Bootcamp/SMDA201804FSF4>

- **Re-Watch Class Videos:**

<https://codingbootcamp.hosted.panopto.com/Panopto/Pages/Sessions/List.aspx?folderID=f89ccab8-12b0-40b9-a65d-a8d30014a7ee>

- **In Class Office Hours:** 45 minutes before class, 30 minutes after

Homework #1 - Assignment

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

<https://smu.bootcampcontent.com/SMU-Coding-Bootcamp/SMDA201804FSF4/tree/master/01-html-git-css/02-Homework/Instructions>

- **Homework Assignment #1 this week**
 - **Due:** Wednesday, June 6th 2018 by 11:59 PM

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:

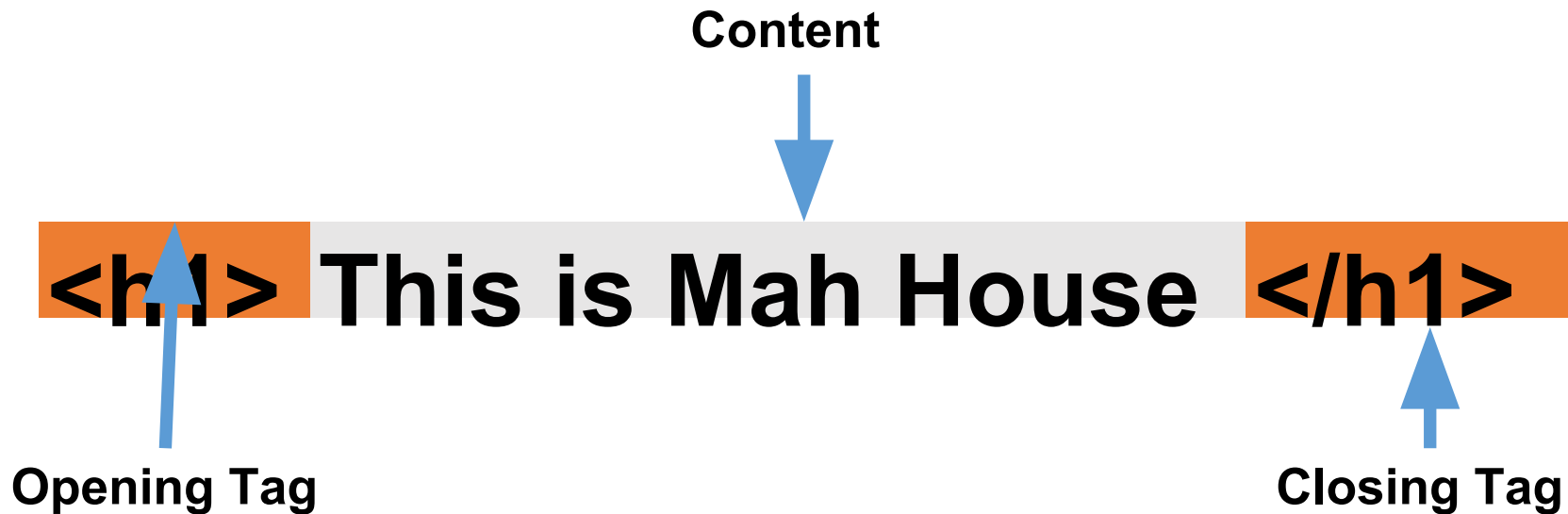
- Continue getting comfortable with HTML.
- Be able to completely write a basic HTML document (like in last class).
- Understand what CSS is, what it's for, and how it works with HTML.
- *Be able to use Git and GitHub to upload code.*

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

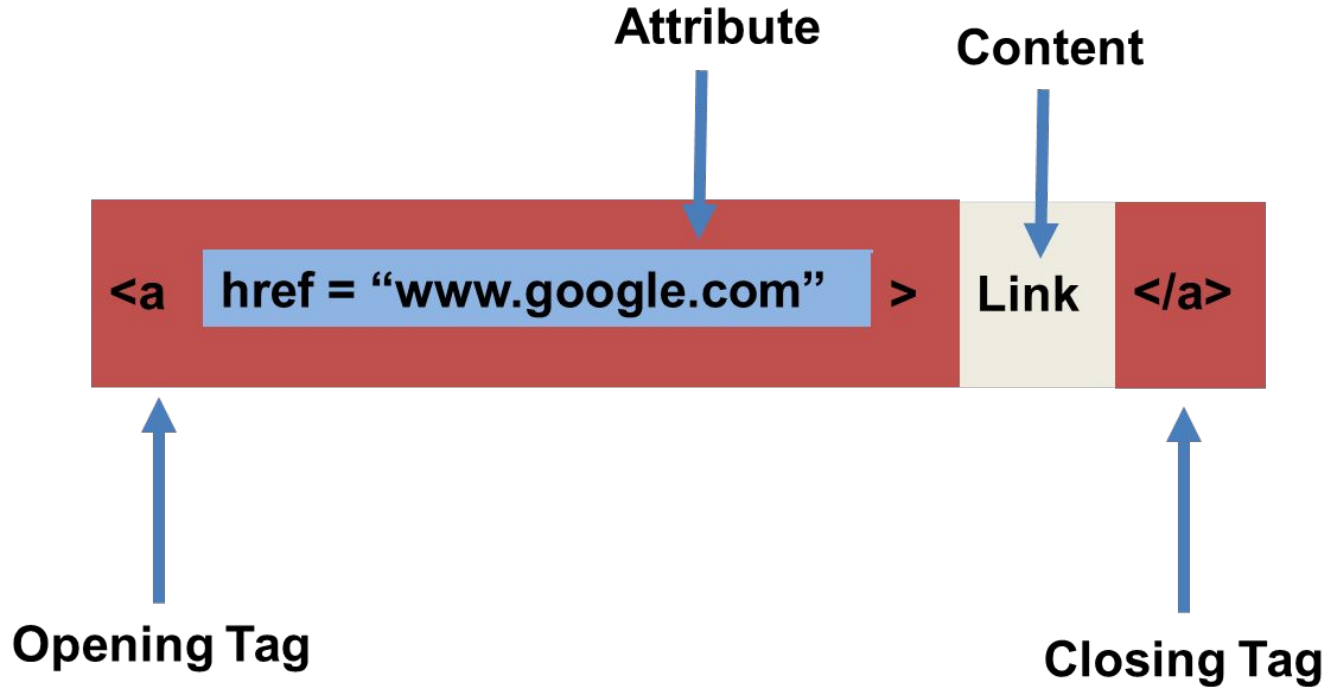
- Aim to build up your skills. Clear up any questions or confusions about HTML.
- Become knowledgeable about a wider range of HTML and CSS tags.
- Be able to selectively apply CSS to specific HTML elements.
- *Be able to use Git and GitHub to upload code.*

HTML Round 1

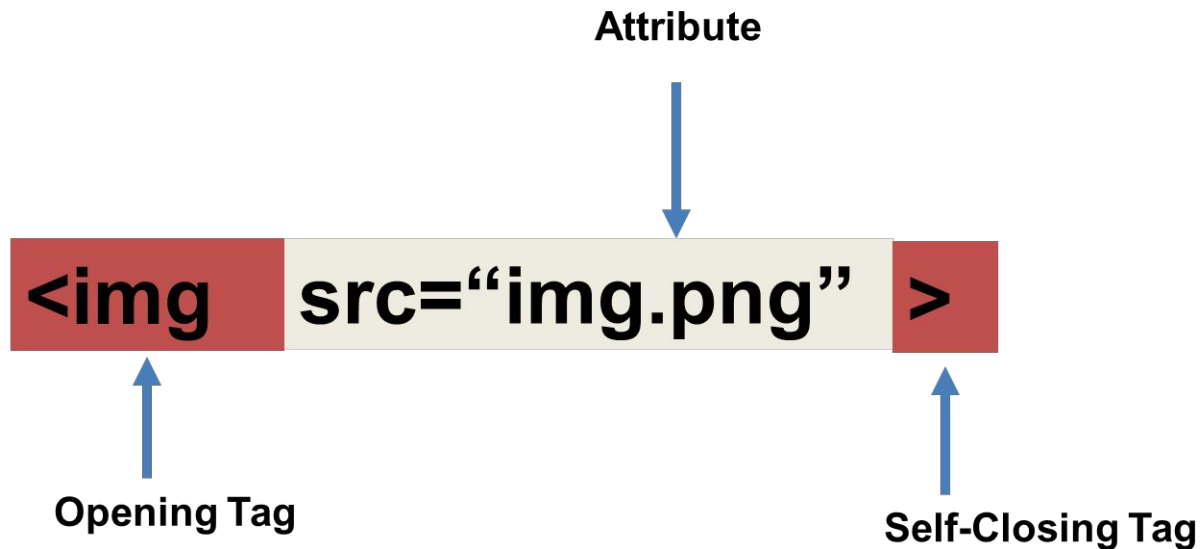
HTML Syntax (Basic)



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
- `<div> </div>` - Logical Container ***
- `<p> </p>` - Wraps individual Paragraphs

Others:

- `` (bold), `` (emphasis)
- `` (images), `<a href>` (links), `` (list items) , `<title>` (title), `
` (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>
```

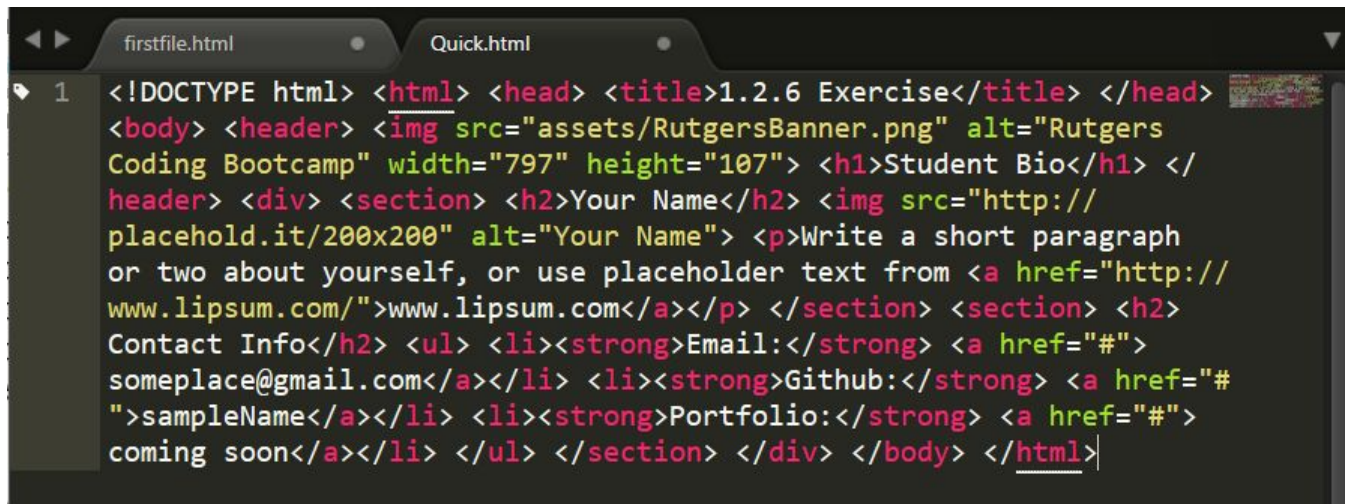
First name:

Last name:

Note that the form itself is not visible.

Also note that the default width of a text input field is 20 characters.

On Ugly HTML

A screenshot of a code editor with two tabs: 'firstfile.html' and 'Quick.html'. The 'Quick.html' tab is active, showing HTML code with poor indentation and inconsistent use of tags. The code is as follows:

```
1 <!DOCTYPE html> <html> <head> <title>1.2.6 Exercise</title> </head>
  <body> <header>  <h1>Student Bio</h1> </
header> <div> <section> <h2>Your Name</h2>  <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!

> YOUR TURN!

Activity: 1-HTML_Git | **Suggested Time:** 20 min

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

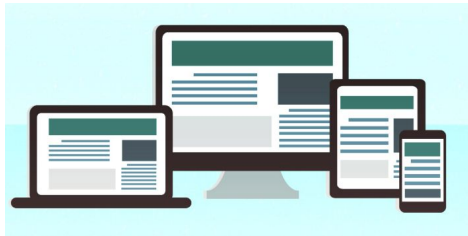
Contact Info

- 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>
  

  <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?



Menu Links

- [Google](#)
- [Facebook](#)
- [Twitter](#)

Hella Boring...

Enter CSS

```
26 <style>
27   h1 {
28     font-size: 60px;
29     text-align: center;
30     margin-bottom: 15px;
31     text-decoration: underline;
32     background-color: black;
33     color: white;
34   }
35
36   h2 {
37     font-size: 40px;
38     text-align: center;
39     margin-top: 15px;
40     margin-bottom: 15px;
41   }
42
43   h3 {
44     font-size: 20px;
45     text-align: center;
46     margin-top: 15px;
47   }
48
```

```
49   img {
50     display: block;
51     margin-left: auto;
52     margin-right: auto;
53   }
54
55   p {
56     text-align: center;
57     font-size: 20px;
58     font-weight: bold;
59   }
60
61   ul {
62     text-align: center;
63     font-size: 35px;
64     list-style-position: inside;
65     border-style: solid;
66     border-width: 5px;
67   }
68 </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?

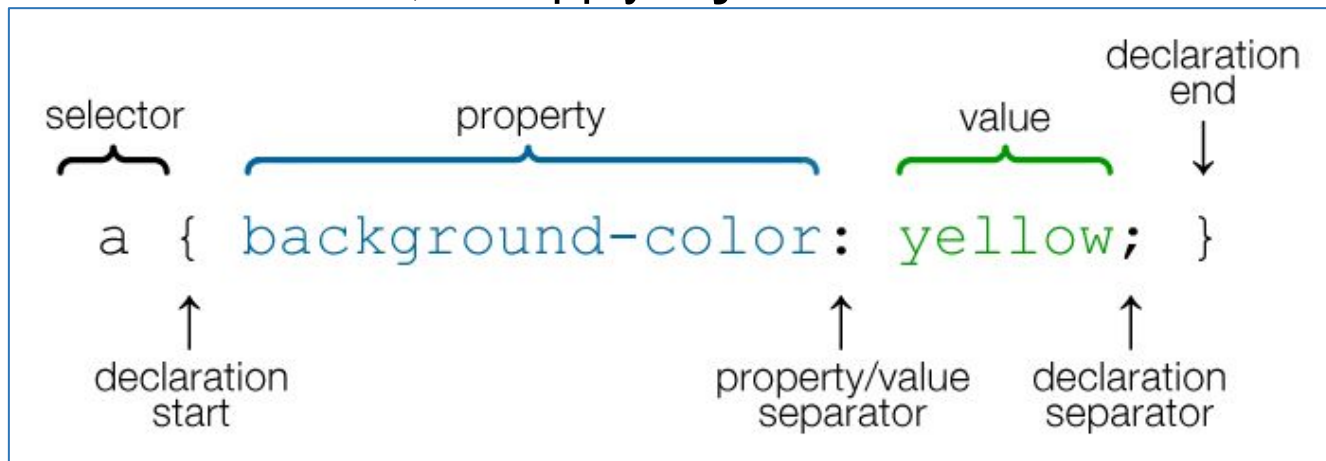


Menu Links

- 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;
```

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

Powerful Duo

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

Instructor: Demo

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

> YOUR TURN!

Activity: 3-HTML_CSS_Layout | **Suggested Time:** 20 min

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

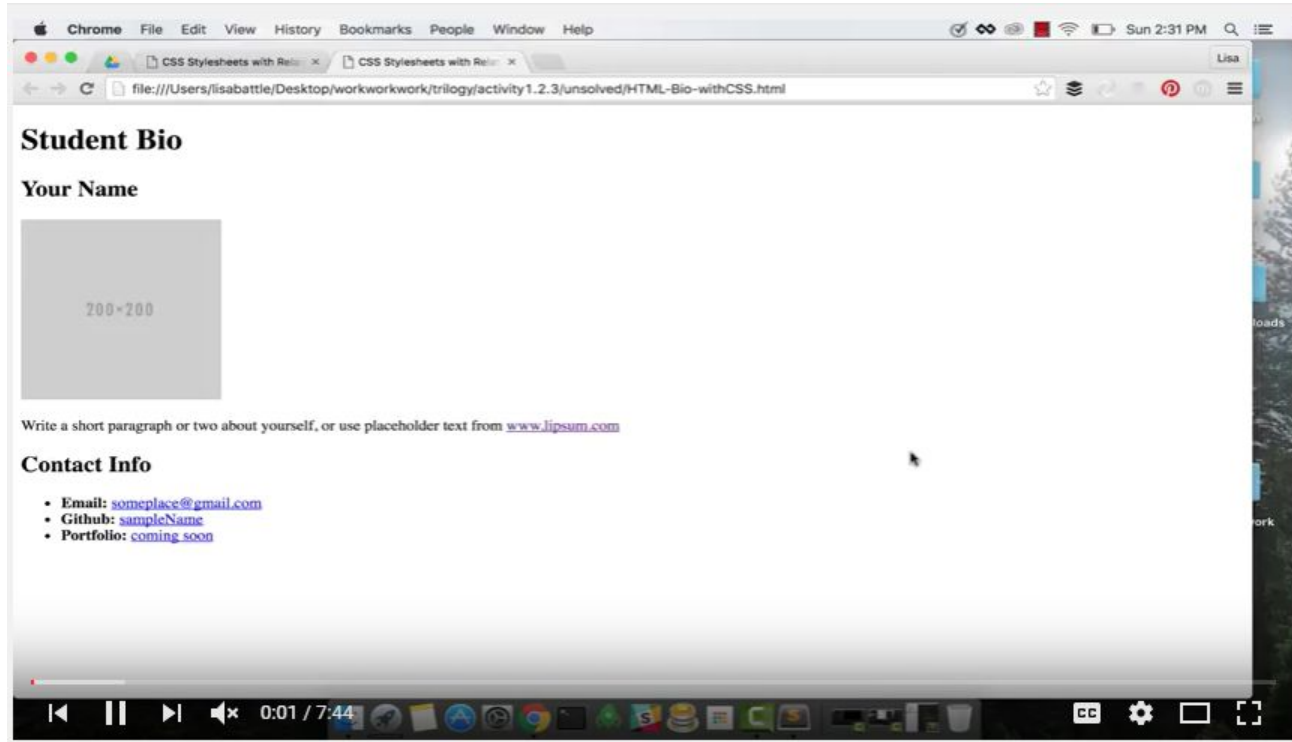


Write a short paragraph or two about yourself, or use placeholder text from www.lipsum.com

Contact Info

- Email: someplace@gmail.com
- Github: [sampleName](#)
- Portfolio: [coming soon](#)

Video Walkthrough!!

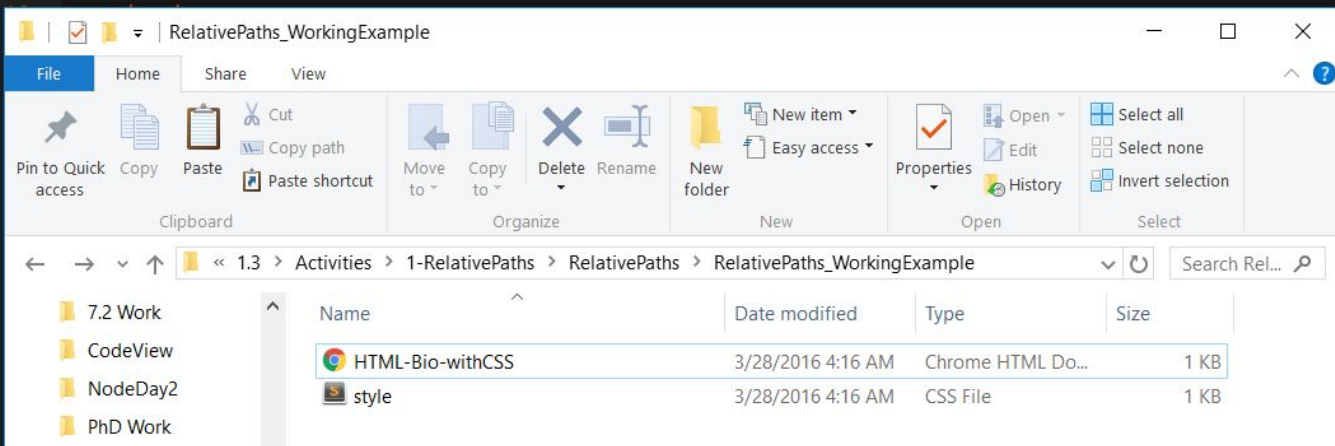


<https://www.youtube.com/watch?v=kMBinXTCrXI&list=PLgJ8UgkiorCnMLsUevoQRxH8t9bt7ne14&index=2>

Relative File Paths

Relative File Paths

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>1.2.9 Exercise</title>
5
6   <!-- This critical line points your HTML to the CSS file. Notice the "relative" pathway -->
7   <link rel="stylesheet" type="text/css" href="style.css">
8 </head>
9 <body>
```



- **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.



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: #1E5792;  
    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: #1E5792;  
    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

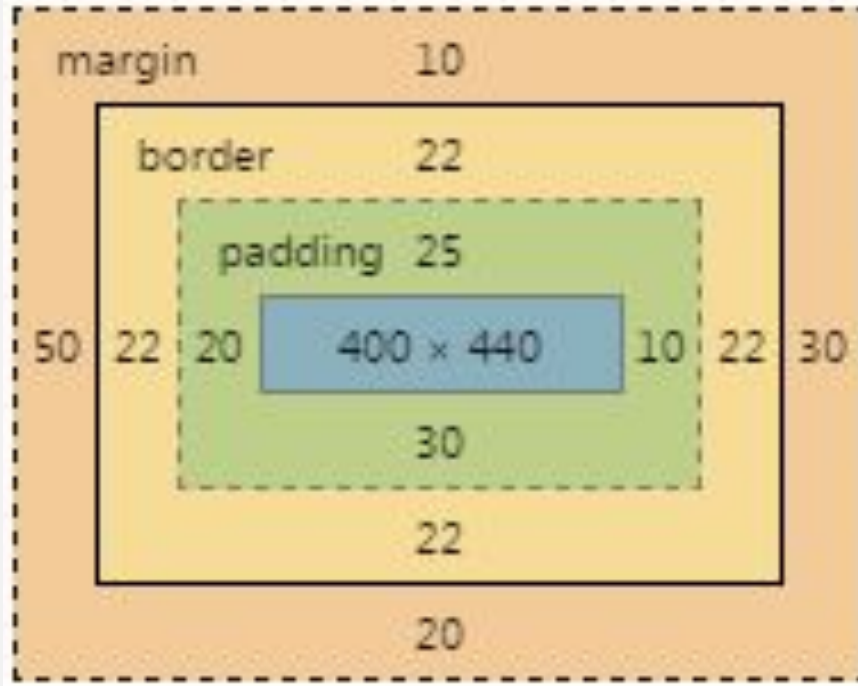
Total element height = content height + top padding +
bottom padding + top border + bottom border + top
margin + bottom margin

Answer

Width: 474 px (no margin), 554 px (with margin)

Height: 539 px (no margin), 569 px (with margin)

> YOUR TURN!!



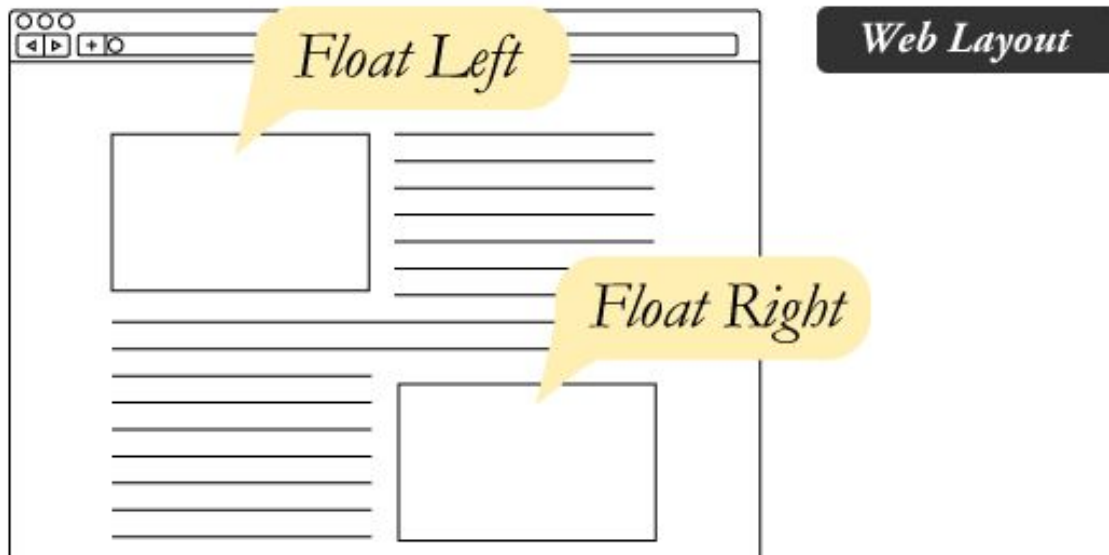
Answer

Width: 474 px (no margin), 554 px (with margin)

Height: 539 px (no margin), 569 px (with margin)

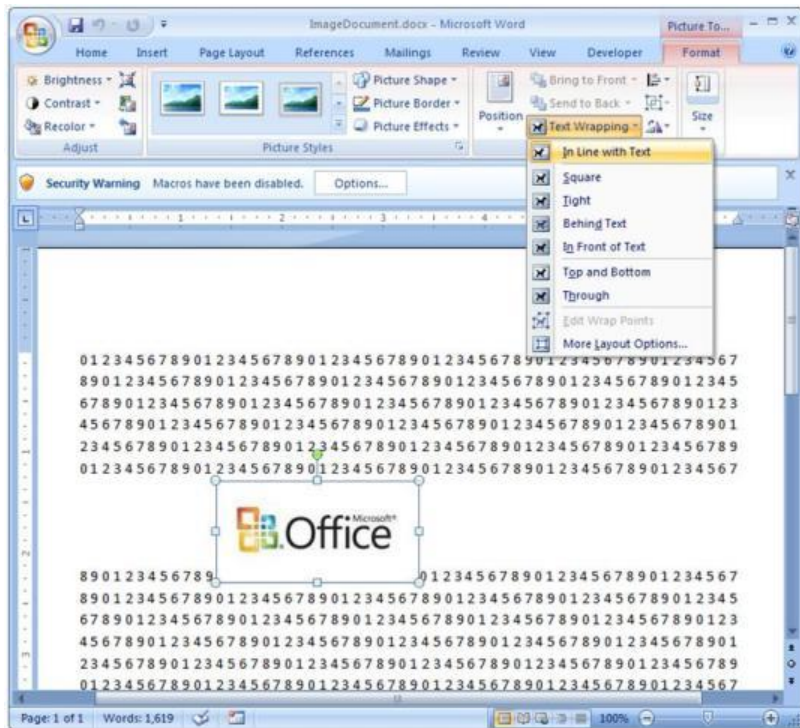
We Be Floatin'

The Concept of “Flow”



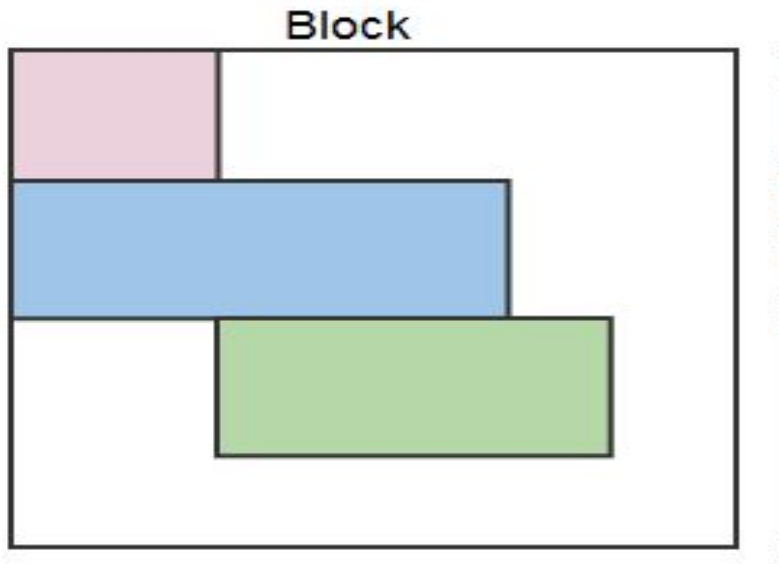
- By default, every HTML element displayed in the browser is governed by a concept called **flow**.
- This means that HTML elements force their adjacent elements to **flow around them**.

Flow Analogy to MS Word



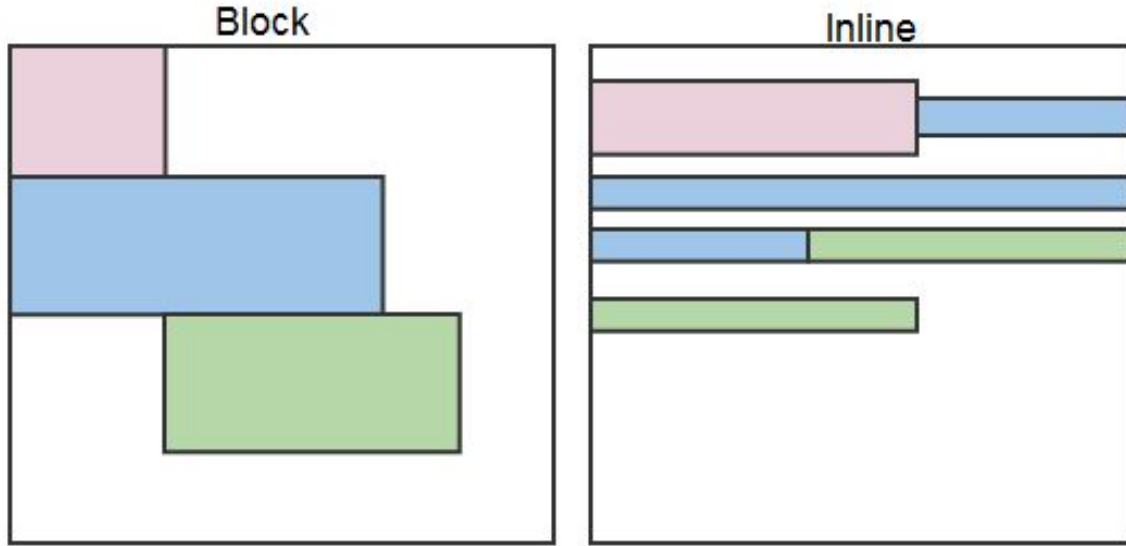
- This concept of “flow” is very similar to the **wrap-text options** you may be familiar with in Microsoft Word.
- Just as in MS Word, you can have images in-line with text, on-top of text, etc.

Block Elements



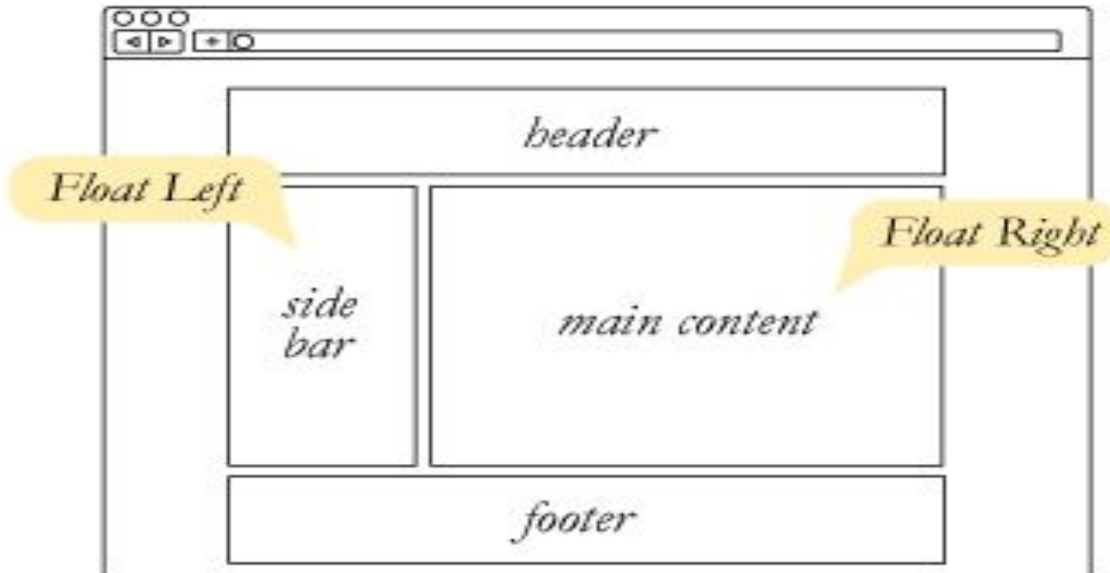
- By default, web clients render many HTML elements as **block elements**. Paragraphs, headers, divs and more receive this treatment.
- A block element will take up an entire line of space—unless you intervene with CSS properties.

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

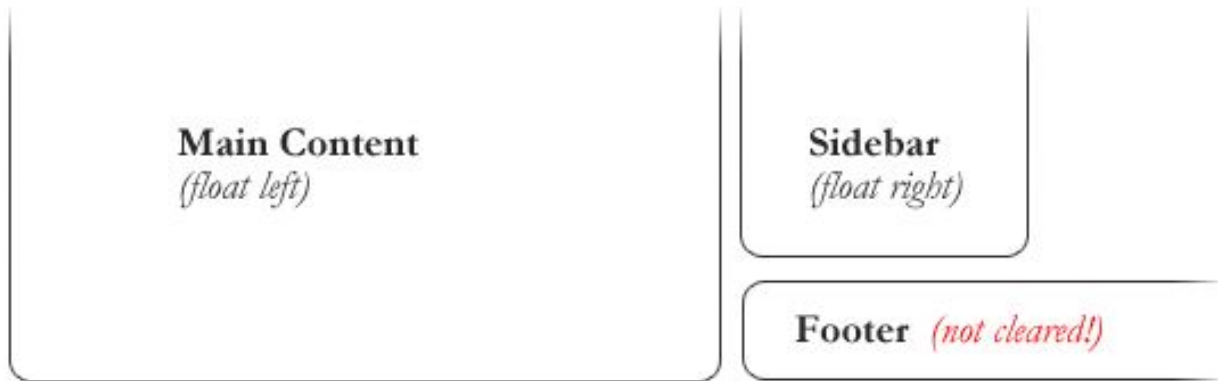


CSS

```
#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

`<div>`

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!



Quick Demo!

2000x200

300x400

900x400

500x100

Instructor: Demo
(2-FloatExamples)

Fantastic Guide on Floats ****

CSS-TRICKS

All About Floats

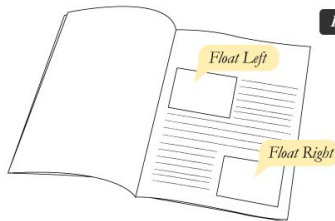
CHRIS COYIER // JULY 8, 2009

What is "Float"?

Float is a CSS positioning property. To understand its purpose and origin, we can look to print design. In a print layout, images may be set into the page such that text wraps around them as needed. This is commonly and appropriately called "text wrap". Here is an example of that.



Print Layout



In page layout programs, the boxes that hold the text can be told to honor the text wrap, or to ignore it. Ignoring the text wrap will allow the words to flow right over the image like it wasn't even there. This is the difference between that image being part of the *flow* of the page (or not). Web design is very similar.

Front End Design & Development Jobs

[NowSecure](#) is hiring a Front-End Developer (open to remote)

[ZipRealty](#) is hiring a Front End Engineer - Consumer Apps

[18F](#) is hiring a Front End Designer

See More Jobs

Post a Job

What is your preferred nomenclature here?

- ☐ Front-End
- ☐ Front End
- ☐ Front-end
- ☐ Front end
- ☐ front-end
- ☐ front end
- ☐ frontend

Vote

View Results

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

> YOUR TURN!!

Activity: 3-FloatLayout-Activity | **Suggested Time:** 30 min

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> #eee

<div> #6ea3da

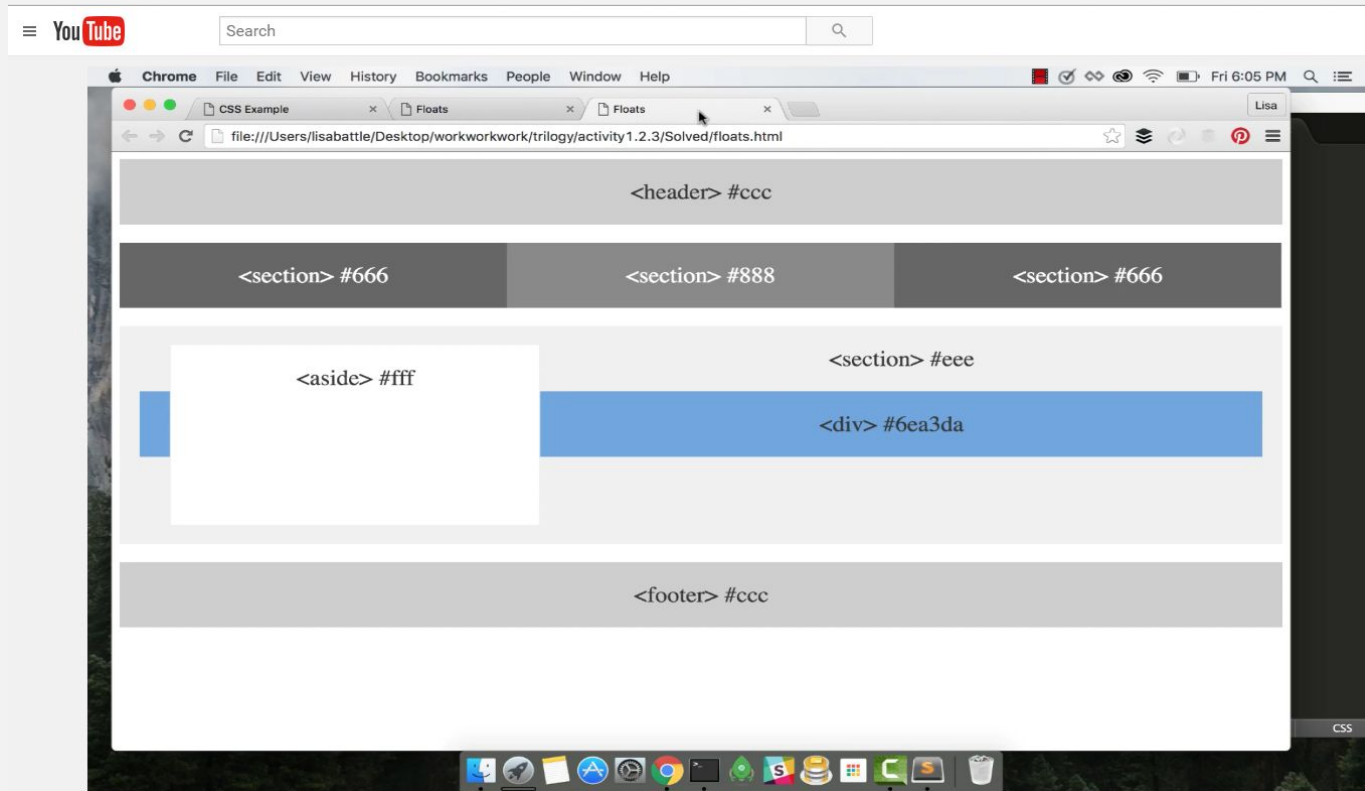
<footer> #ccc

Good work!



Your brain may rest now...

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

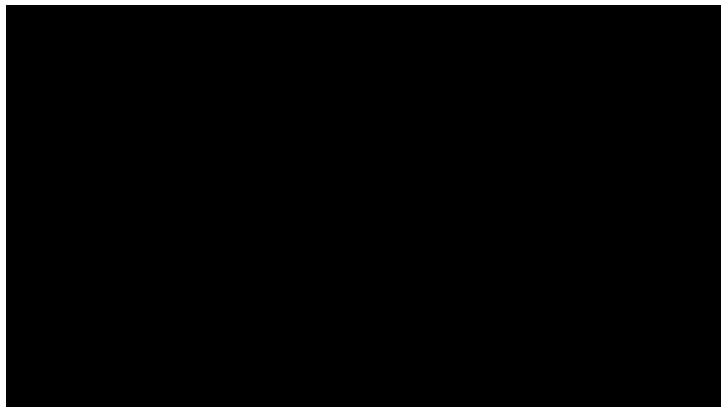


Video Link: <https://youtu.be/0lpxKw6E90Y>

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.