



# Git'n Pro with HTML/CSS

Web Development Boot Camp  
Lesson 1.2



**How do I do this again?**



# **Admin Items**

# How to Get Help

---

01

Practice, practice, practice: work individually or in groups.

02

Review in-class material (activities and slides).

03

Watch the class videos again.

04

Attend office hours, which are held 45 minutes before and 30 minutes after class.

05

Attend one-on-one sessions (to be announced by your Student Success Manager (SSM)).

06

Contact your Student Success Manager anytime!

# Today's Class!

# Today's Objectives

---

Today we will:

01

Understand the importance of Git version control and how to use it.

02

Create GitHub repositories, push up code, and share with the class.

03

Create more HTML documents.

04

Learn how to properly use basic HTML tags.

05

Apply basic CSS styling to HTML documents.

# Know Thyself

---

**If you are a beginner to HTML/CSS and coding, your objectives are to:**

- Continue to get comfortable with HTML.
- Be able to write a complete, basic HTML document (like in the last class).
- Understand the function of CSS and how it works with HTML.
- Be able to use Git and GitHub to upload code.

**If you've had past exposure to HTML/CSS and coding and felt comfortable with the last lesson, your objectives are to:**

- Aim to build up your skills.
- Clear up any questions or confusion you have 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)

---



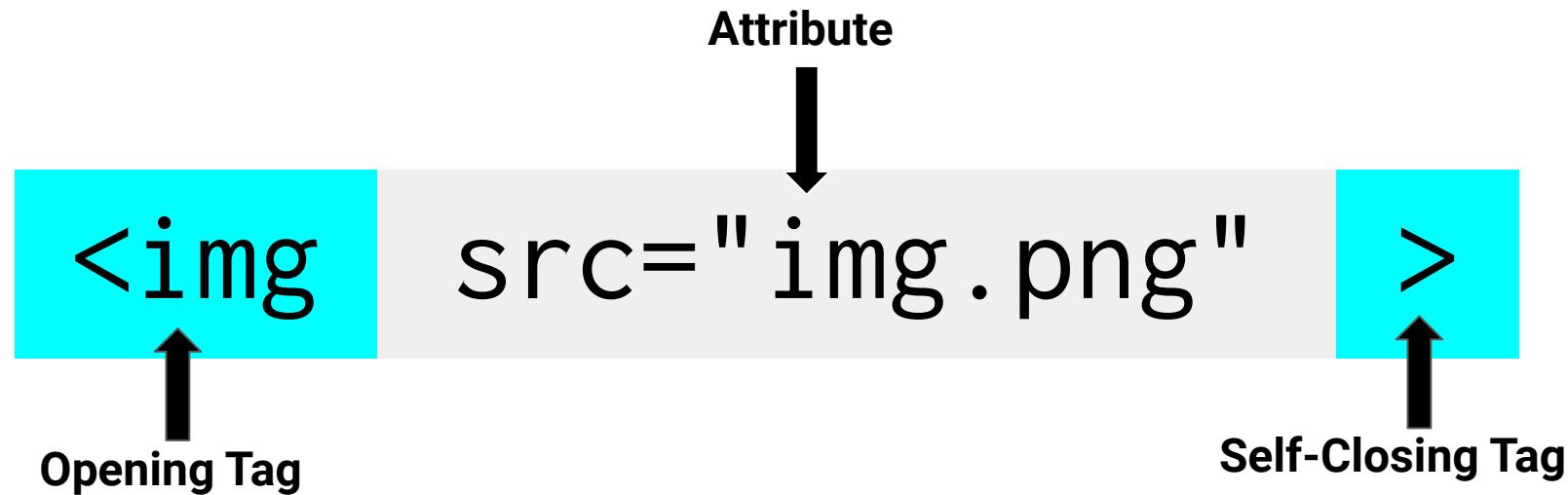
# HTML Syntax (with Attribute)

---



# Tricky Tags (Self-Closing)

---



# Important Common Tags

Headings:	Containers:	Others:	
<h1> </h1>	Heading 1 (Largest heading)	<html> </html>	Wraps the entire page
<h2> </h2>	Heading 2 (Next largest heading)	<head> </head>	Wraps the header of the page
<h3> </h3>	Heading 3	<body> </body>	Wraps the main content
		<div> </div>	Logical container
		<p> </p>	Wraps individual paragraphs
		<title>	title
		 	line break
		<table>	tables
		<!-- -->	comments

# Less Common Tags

---

All HTML tags are listed here: [w3schools.com/tags/](https://www.w3schools.com/tags/)

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

<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 text box

# 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:

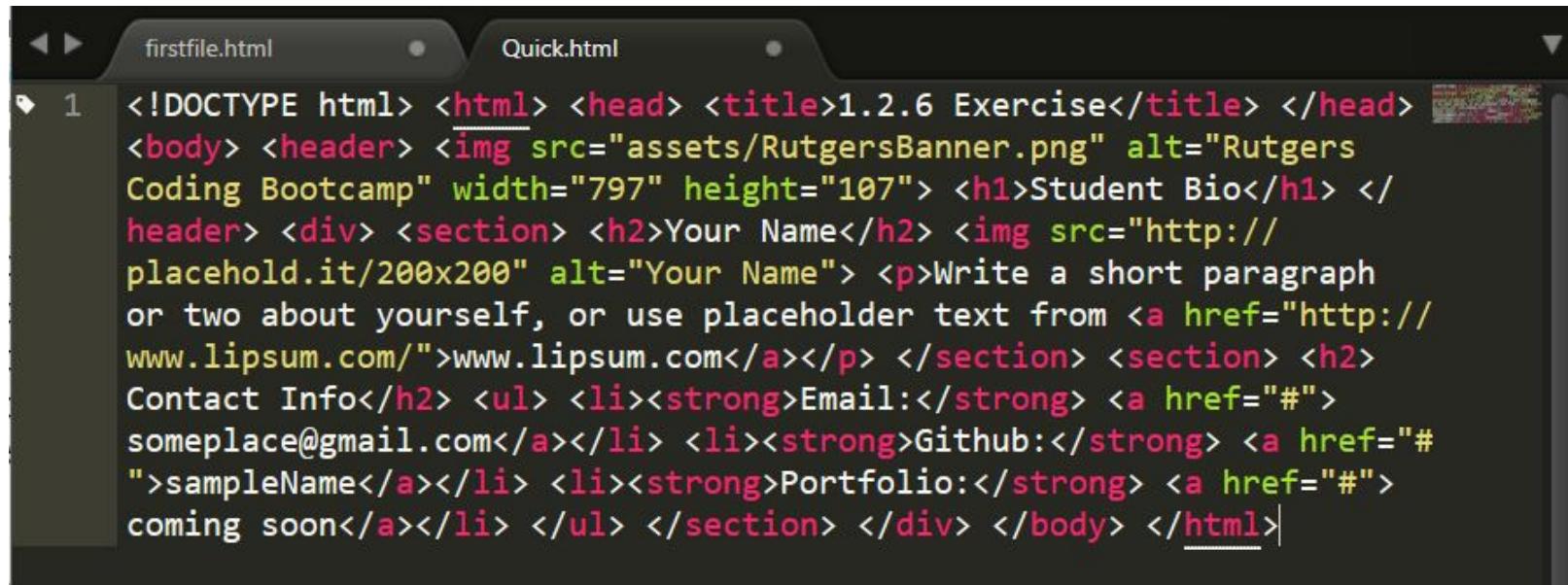
Note that the form itself is not visible.

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

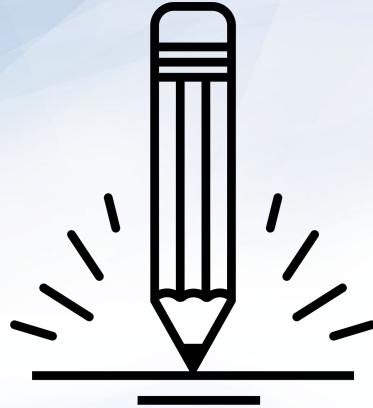
# Ugly HTML

---

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



```
<!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>
```



## Activity: Basic Student Bio

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 will be sent via Slack)**

Suggested Time:  
20 minutes

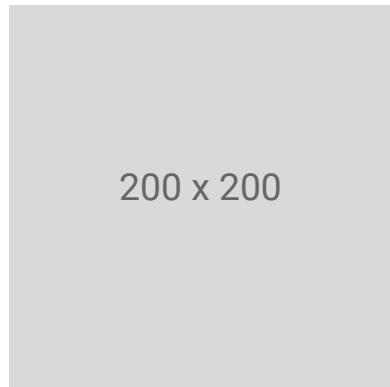


# Activity: Basic Student Bio

---

## Student Bio

### Your Name



Write a short paragraph or two about yourself, or use placeholder text from [www.lipsum.com](http://www.lipsum.com).

### Contact Info

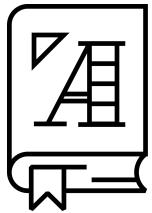
- Email: [someplace@gmail.com](mailto:someplace@gmail.com)
- GitHub: [Sample Name](#)
- Portfolio: [Coming Soon](#)



# CSS Stylin'

# HTML and CSS Definitions

---



**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 aesthetics of a webpage, including layouts, colors, fonts, and content placement. (JavaScript is the language that deals with logic, animation, etc.)

# HTML/CSS Analogy

---

HTML Alone	HTML and CSS
Like writing papers in Notepad	Like writing papers in Microsoft Word
Can only write unformatted text	Can format text, page layout, alignment, and more 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 Awesome Header

  Lorem ipsum dolor sit amet, consectetur adipiscing 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.



#### Menu Links

- Google
- Facebook
- Twitter

# Basic HTML Page: Result

---

## Awesome Header

### Smaller Awesome Header

#### Even Smaller Awesome Header

  Lorem ipsum dolor sit amet, consectetur adipiscing 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.



**Boring**

#### Menu Links

- Google
- Facebook
- Twitter

# 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 Awesome Header

  Lorem ipsum dolor sit amet, consectetur adipiscing 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.



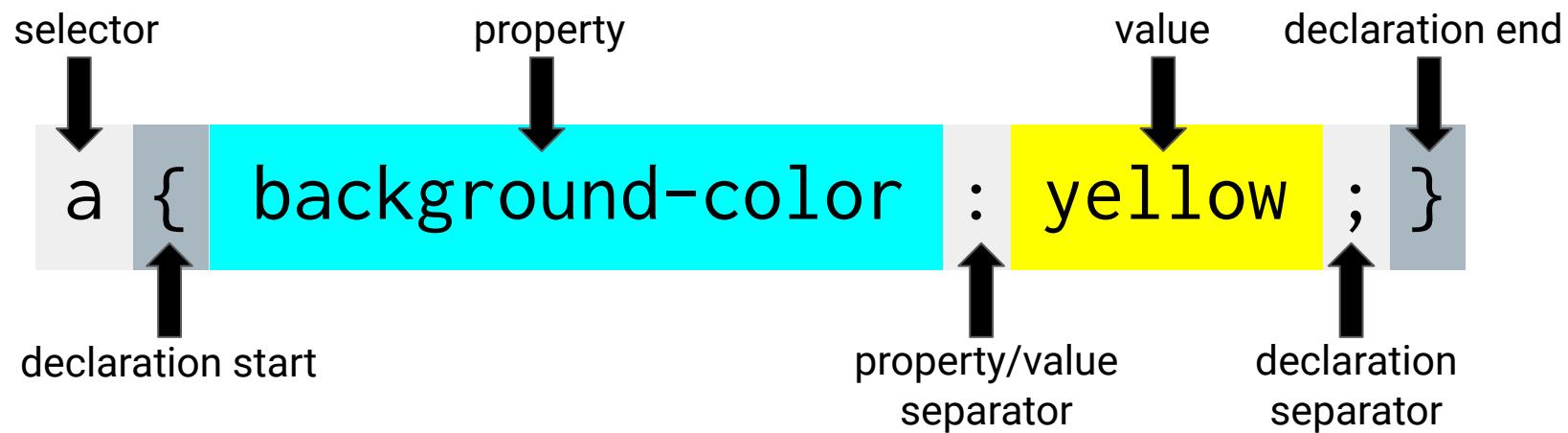
#### Menu Links

- 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 following example, the header would become 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)	Example (CSS)
< p   class="bigBlue" > Header </ p >	. bigBlue { font-size: 100px; color: blue; }

# Key CSS Attributes

---

Font and Color:	
color:	sets color of text
font-size:	sets size of the font
font-style:	sets italics
font-weight:	sets bold
Alignment and 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

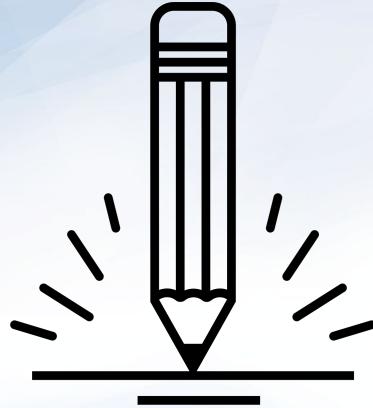


## **Powerful Duo**

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



Instructor Demonstration  
CSS Basics



# Activity: CSS-Styled Bio Page

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

**(Additional instructions will be sent via Slack)**

Suggested Time:  
20 minutes



# Activity: CSS-Styled Bio Page

---

## Student Bio

**Your Name**



200x200

Write a short paragraph or two about yourself, or use placeholder text from [www.lipsum.com](http://www.lipsum.com)

**Contact Info**

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





**Time For a Quick Video**

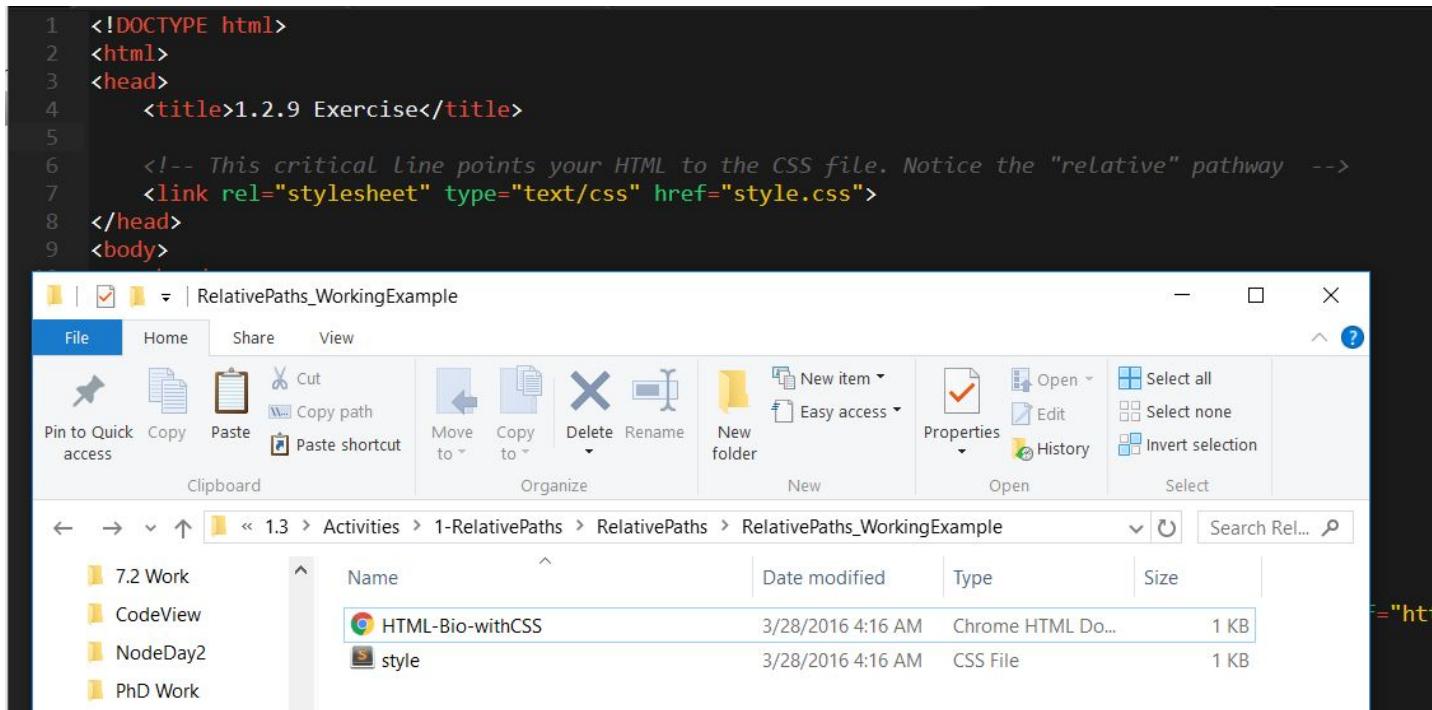
---

[Student Bio Layout](#)

# 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.



The screenshot shows a Windows File Explorer window titled "RelativePaths\_WorkingExample". The address bar indicates the path: "1.3 > Activities > 1-RelativePaths > RelativePaths > RelativePaths\_WorkingExample". The main pane displays two files: "HTML-Bio-withCSS" (Chrome HTML Document, 1 KB) and "style.css" (CSS File, 1 KB). The "style.css" file is highlighted with a blue selection bar. The toolbar at the top includes standard file operations like Pin to Quick access, Copy, Paste, Cut, Copy path, Move to, Copy to, Delete, Rename, New folder, Properties, Open, Select all, Select none, Edit, History, and Invert selection.

```
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 --&gt;
7     &lt;link rel="stylesheet" type="text/css" href="style.css"&gt;
8 &lt;/head&gt;
9 &lt;body&gt;</pre>
```

# Absolutely No Absolute Paths

---

Always use relative file paths!



If you deploy websites 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.

**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">
```



## Instructor Demonstration Relative File Paths



# Activity:

## Relative File Paths

Suggested Time:  
12 minutes



# Activity: Relative File Paths

---

01

Unzip the folder sent to you via Slack.

02

Edit the HTML files in all of the `RelativePaths` folders. Write relative paths that link the HTML documents with CSS stylesheets.

**HINT:** Check out the `RelativePaths_WorkingExample` folder.

Suggested Time: 12 Minutes





**Time's Up! Let's Review.**

# Take a Break!

---



# Box Model

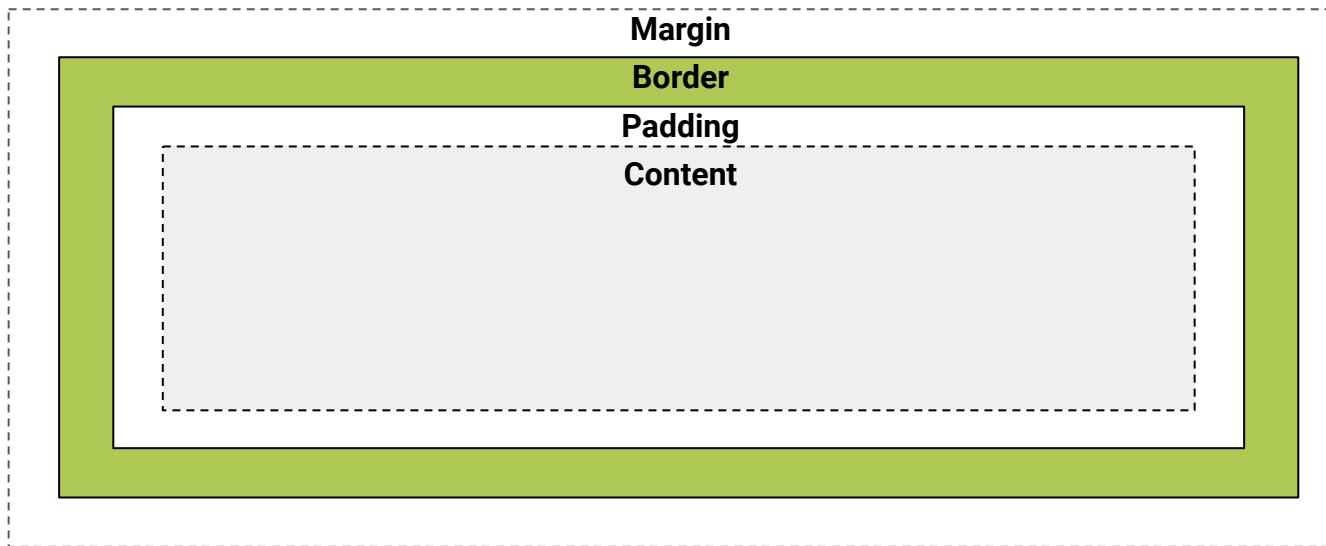
# Boxes Upon Boxes

---

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)





# Partner Activity: Box Model

Suggested Time:  
10 minutes



# Activity: Box Model

```
#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

$$\begin{aligned} w &= 554 \\ + &= 569 \end{aligned}$$

Suggested Time: 10 Minutes



# Activity: Boxes Upon Boxes

## Answer:

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

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

```
#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

Suggested Time: 10 Minutes

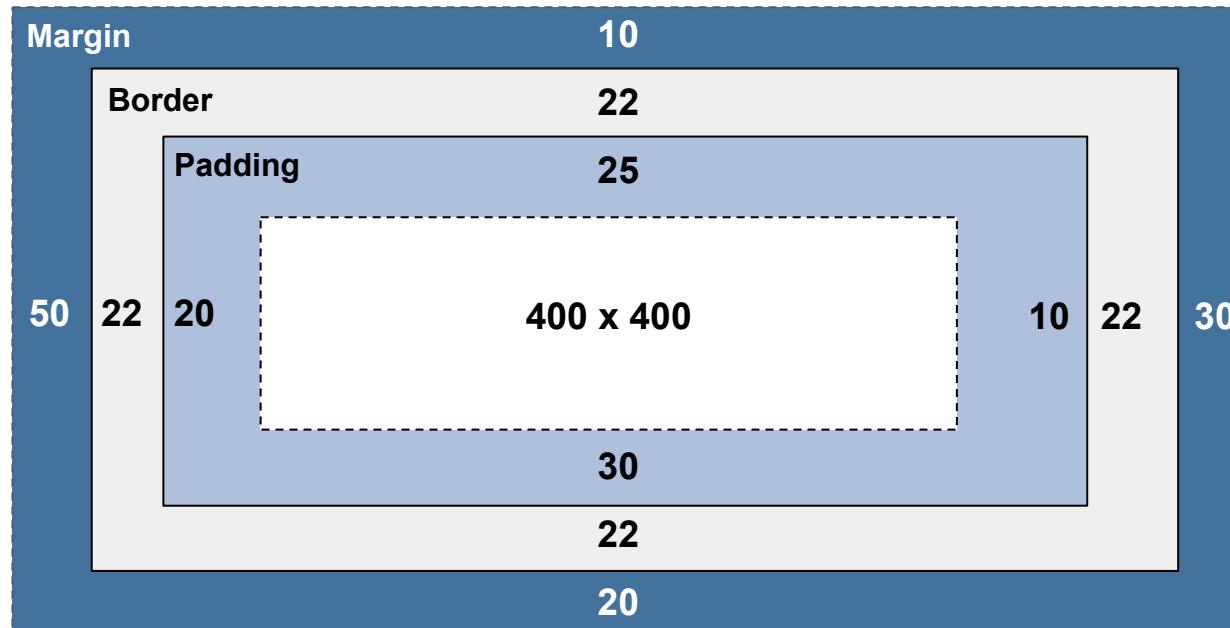


# Activity: Box Model

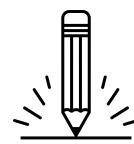
## Answer:

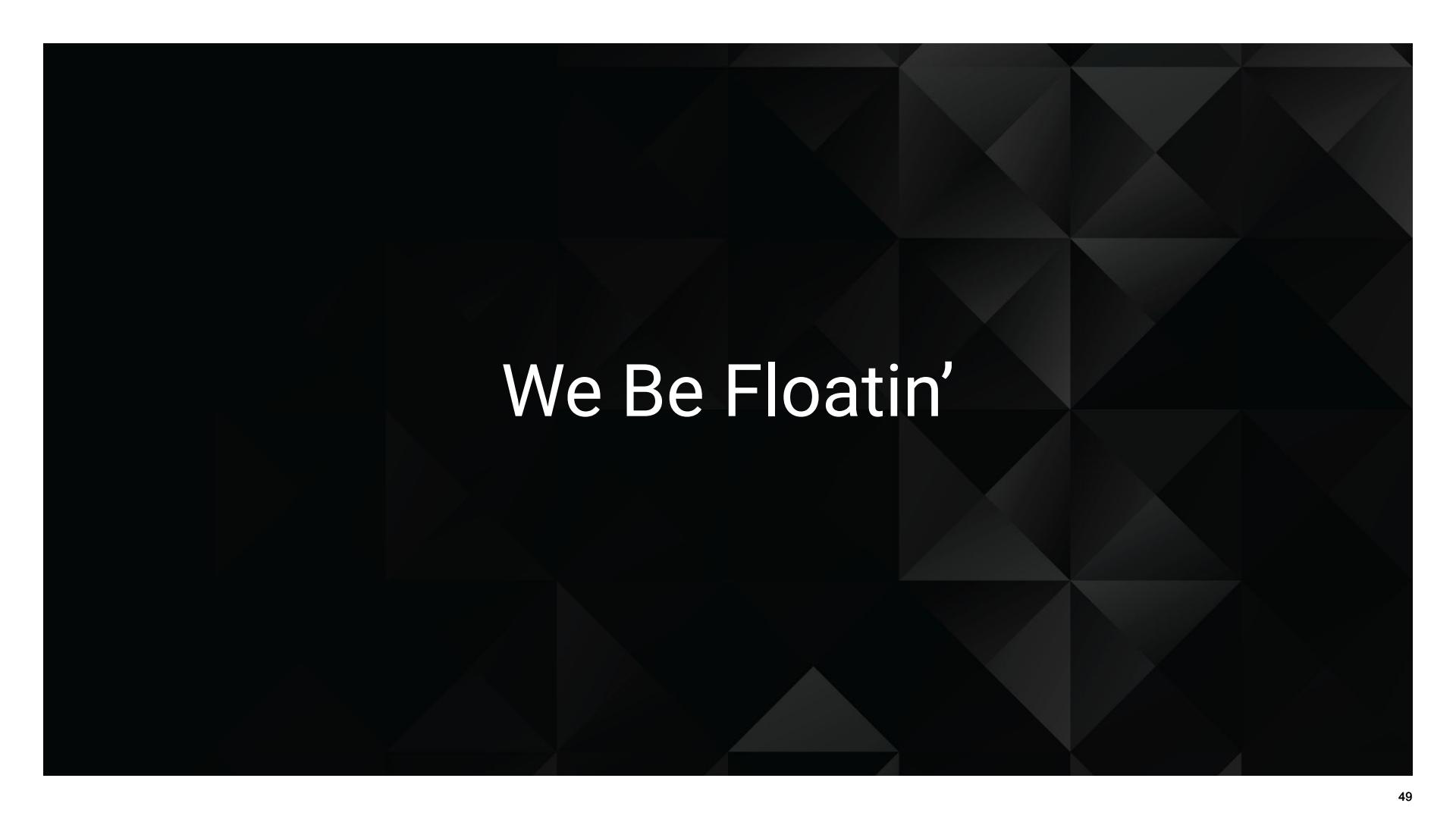
Width: 474 px (no margin), 554 px (with margin)

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



Suggested Time: 10 Minutes





We Be Floatin'



## Warning

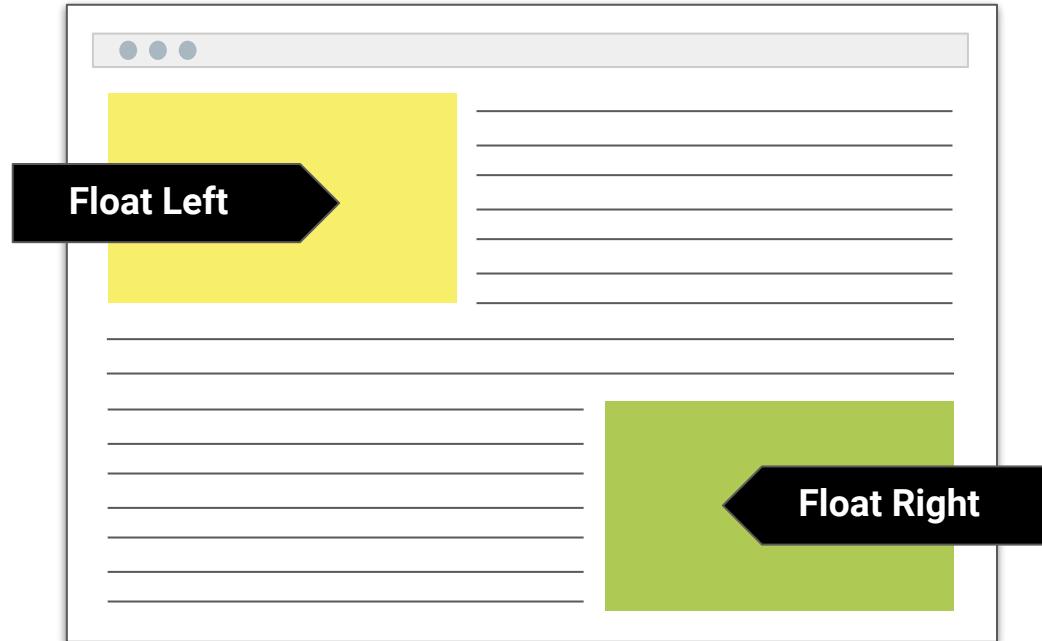
The next few topics are fairly tricky but also *very important*. Time to channel that inner genius!

# 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 adjacent elements to flow around them.



# Analogy: Flow and MS Word

---



The concept of flow is similar to wrap-text in Microsoft Word.



Just as in MS Word, with CSS you can position images to be in-line with text, on top of text, and so on.



# Block Elements

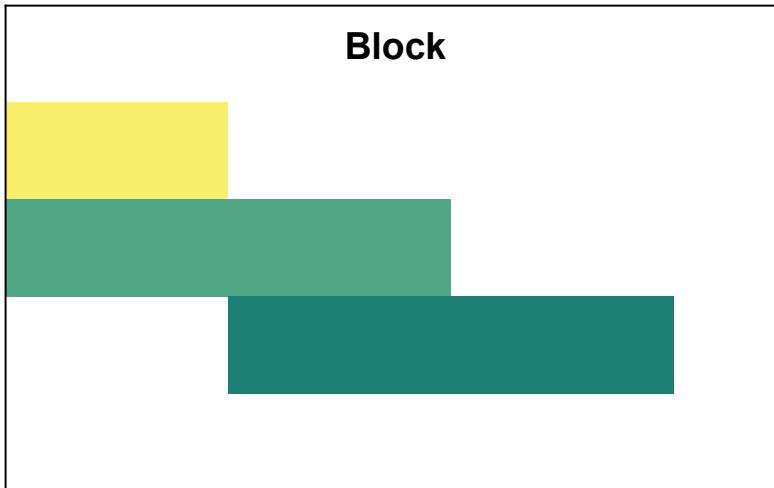
---



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. In-line Elements

---

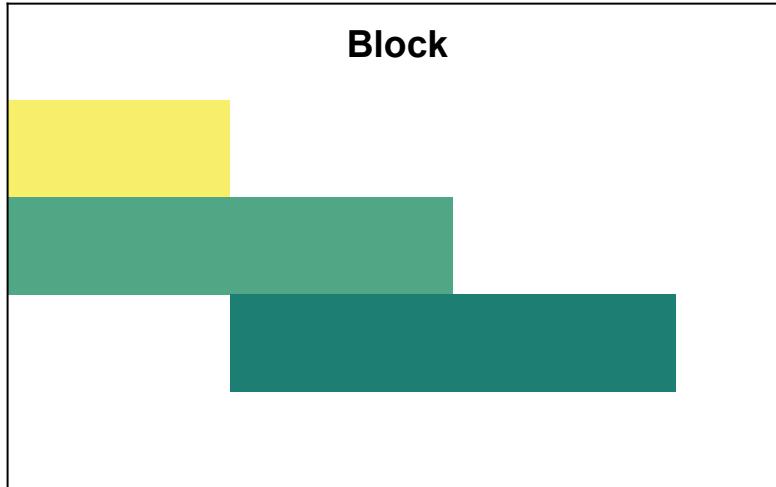


Now, contrast block elements with **in-line elements**.

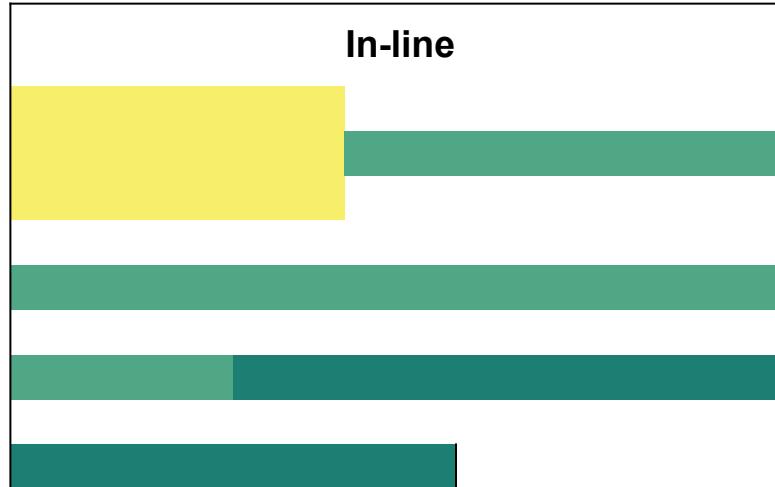


By using **float CSS properties**, we can command our website to display multiple HTML elements adjacently.

**Block**



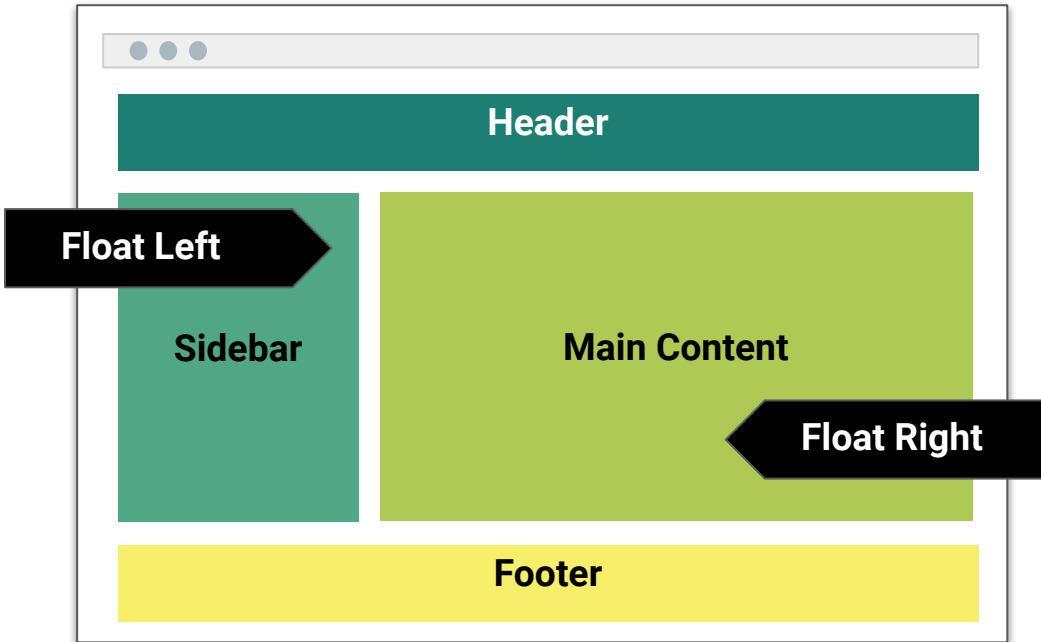
**In-line**



# Floats

---

To transform these block elements into in-line elements, we use a CSS property called **float**. Floats are necessary when building web layouts.



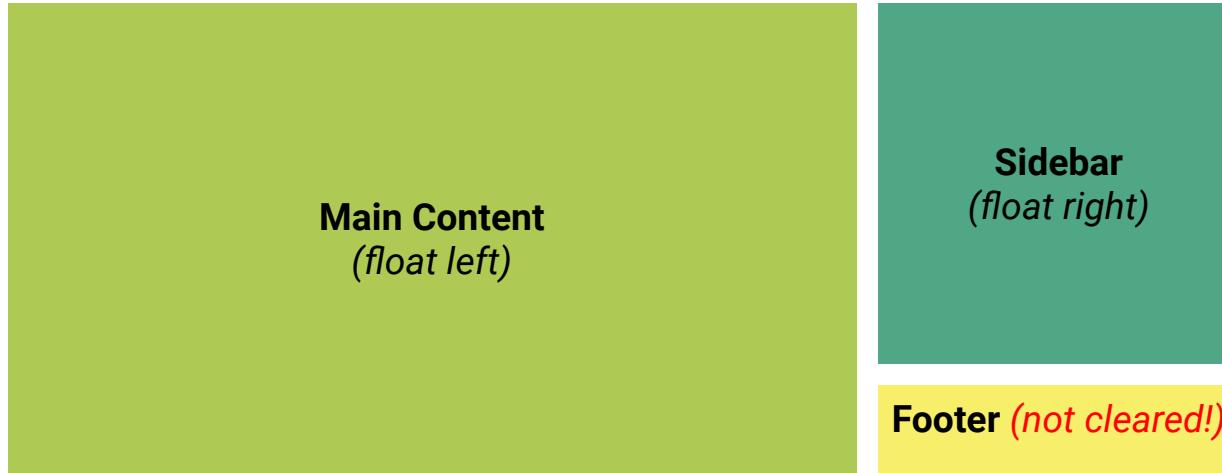
## CSS

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

# Clearing the Float

---

However, floats often get in the way of layouts. Sometimes we don't want to give each element the “in-line” treatment.



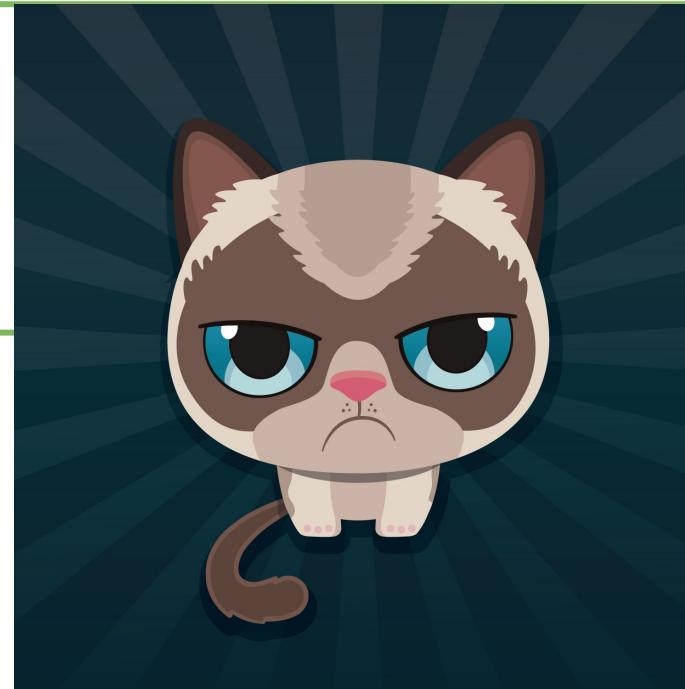
# clearfix Hack

---

Sometimes when elements don't match up in size, we get situations like this:

<div>

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



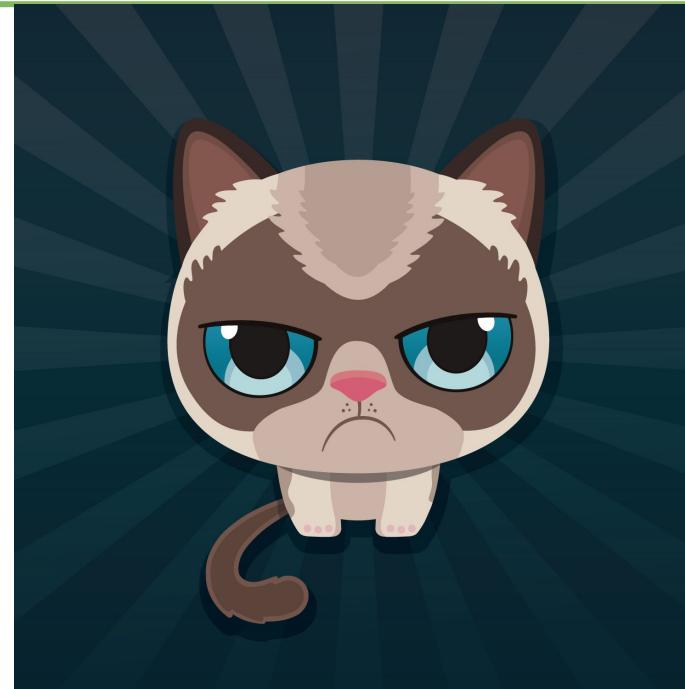
# clearfix Hack

---

We can get around this by using the **clearfix hack**.

```
<div class="clearfix">
```

Much better!



# clearfix Hack

---



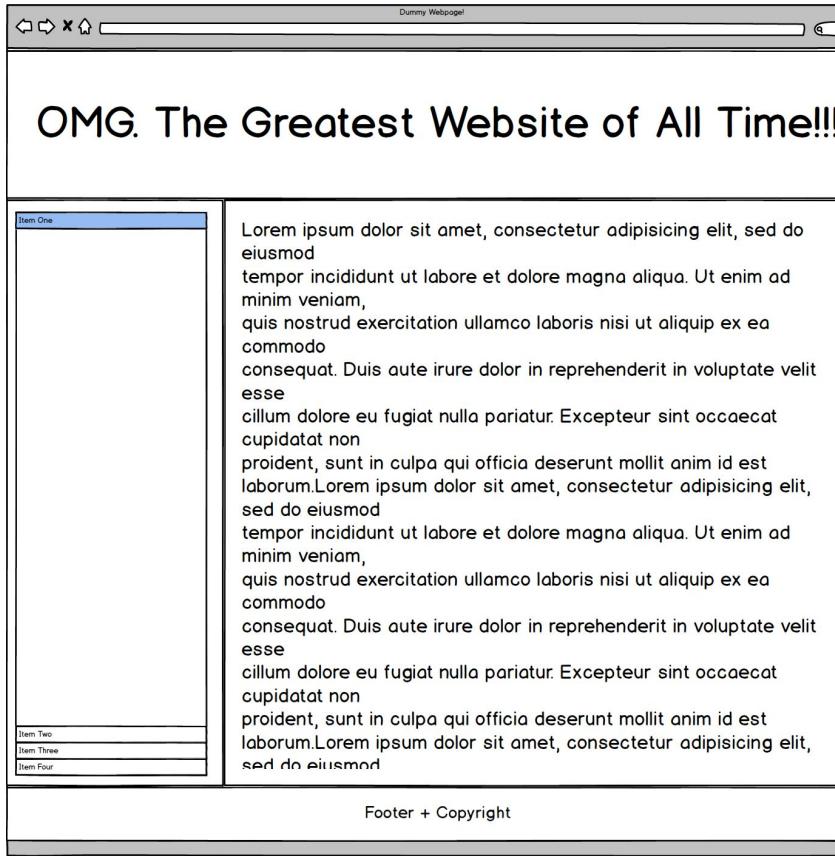
`::after` is what we call a pseudo-element. We use it to style specific parts of an element.



This will add an HTML element, hidden from view, after the content of the `.clearfix` element.  
This clears the float.

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

# Quick Demo



# Quick Demo

---



2000 x 200



300 x 400



900 x 400



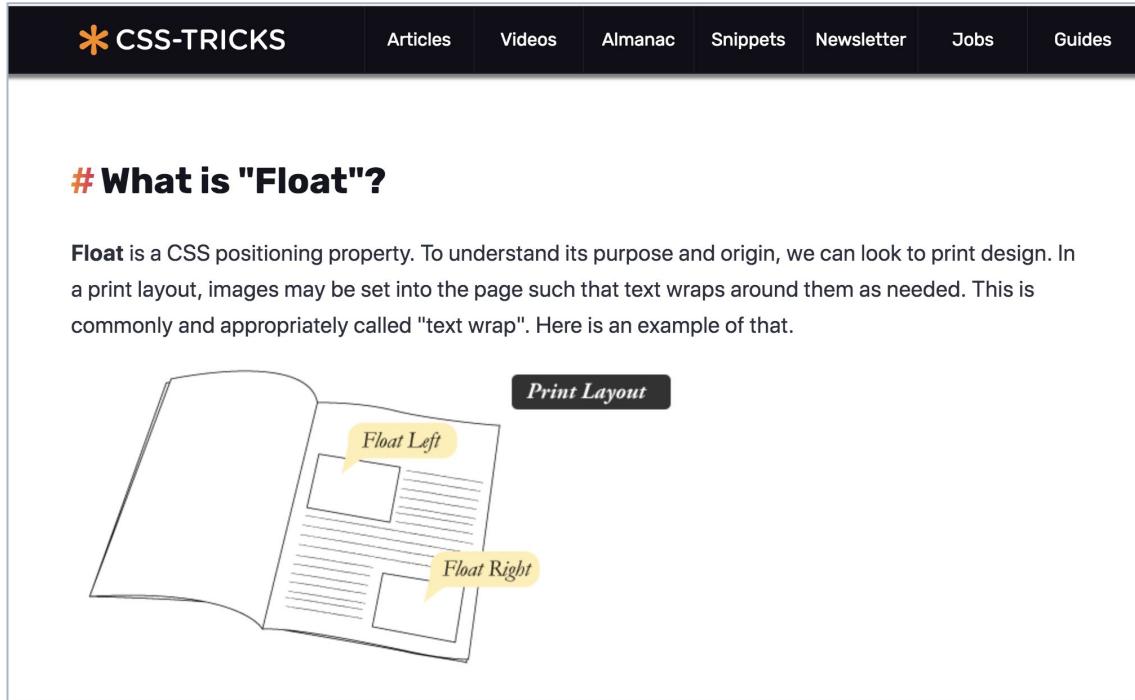
500 x 100



Instructor Demonstration  
Floats

# A Fantastic Guide to Floats

To all serious frontend developers, this is a necessary read:  
[css-tricks.com/all-about-floats](https://css-tricks.com/all-about-floats)

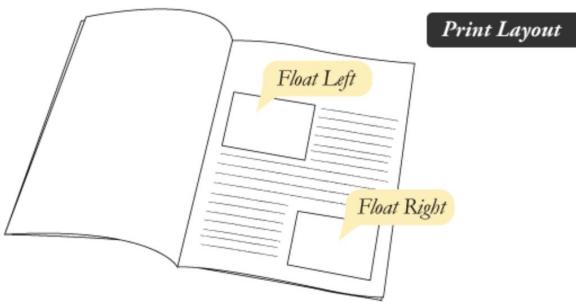


The screenshot shows the CSS-Tricks homepage with a navigation bar at the top containing links for Articles, Videos, Almanac, Snippets, Newsletter, Jobs, and Guides. Below the navigation, there's a large search bar. The main content area features a section titled "# What is "Float"?" with a sub-section titled "Print Layout". An illustration of an open book is shown, with text blocks labeled "Float Left" and "Float Right" indicating how content wraps around floated elements.

## # 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*





# **Partner Challenge:**

## **Float Layout**

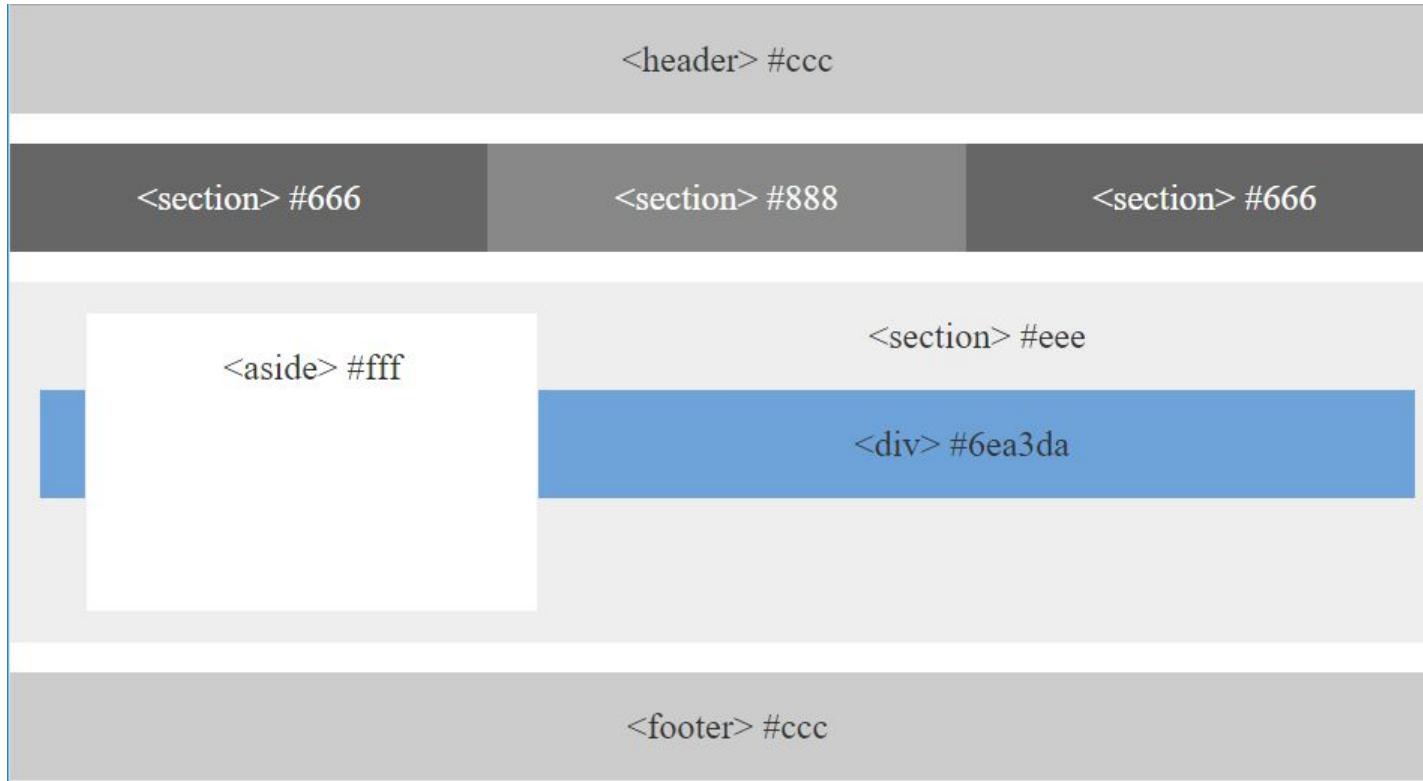
In this activity, you'll flex your newfound floating skills by creating a conceptual layout. Eyeball the design to your best ability.

**Instructions sent via Slack.**

**Suggested Time:**  
**30 minutes**



# Partner Challenge: Float Layout



**Suggested Time:** 30 minutes

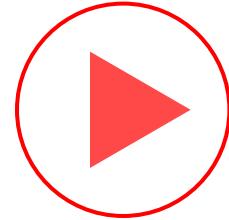


# Good work!

---

Your brain may rest now.





## Time For a Quick Video

---

[Floats in CSS](#)

# Recap

# Still a Bit Confused?

---

- Remember the video guides for key activities (such as the last one).
- If you EVER feel like you are falling behind, use the video walk-throughs to catch 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 however we can.

