Heroes of CSS

The Coding Bootcamp

Admin Work

Homework Assignment

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

  https://upenn.bootcampcontent.com/upenn-bootcamp/UPENN201901FSF5

- **Homework Assignment #1 is** Due:

  2/9, Next Saturday

Homework Assignment

- **Really, work hard on this assignment!** This assignment introduces you to fundamental concepts that we'll be building the entire course-long.

- **Review In Class Material, *especially* Exercises:** https://upenn.bootcampcontent.com/upenn-bootcamp/UPENN201901FSF5

- **Work with your peers!** It's much better than screaming at your computer alone.

- **Ask Questions on Slack!** Your peers, TAs, and Instructors are all here to help when they can.

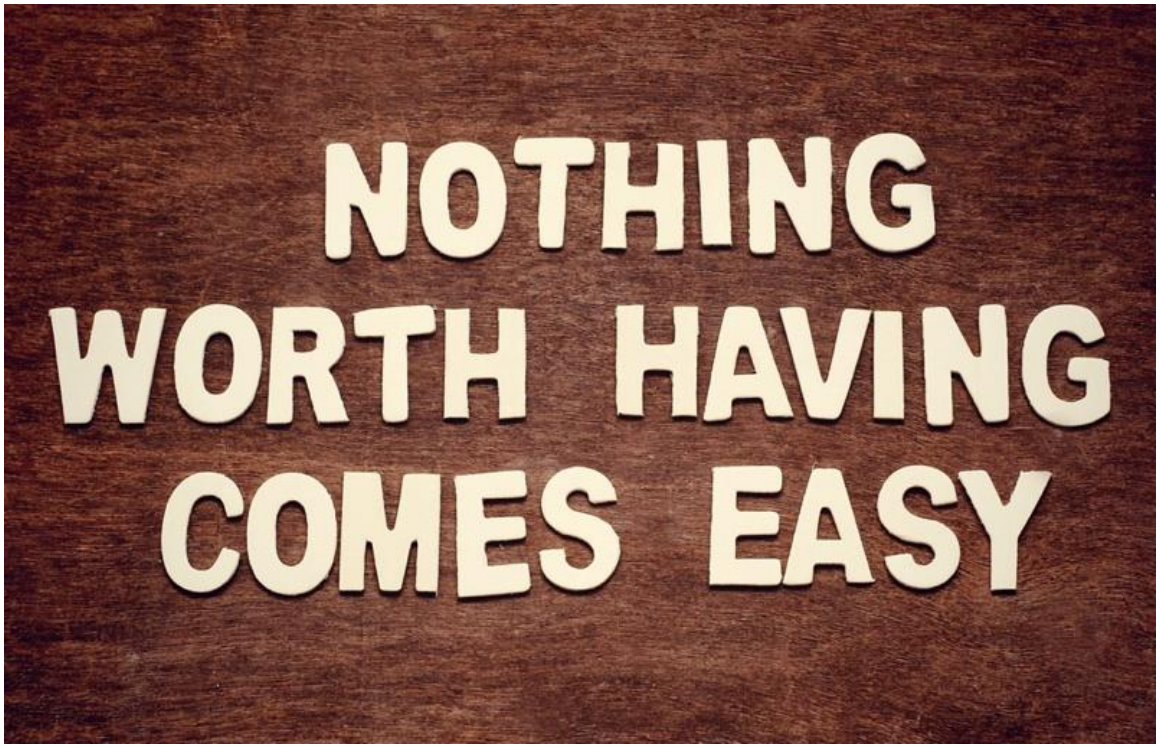
Most Important of All….

Just Submit SOMETHING (even if it seems pretty crummy)!

Warning!

Brace Yourselves

Today is going to be a bit tough.



But trust us!

It will all look easy a few weeks from now.

Brace Yourselves

Don't expect to understand EVERYTHING at once.

Today is all about getting immersed.

CSS Recap

Critical Question

What is "CSS?"

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 - 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

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

Critical Question

How do we style HTML…

Elements?

Classes?

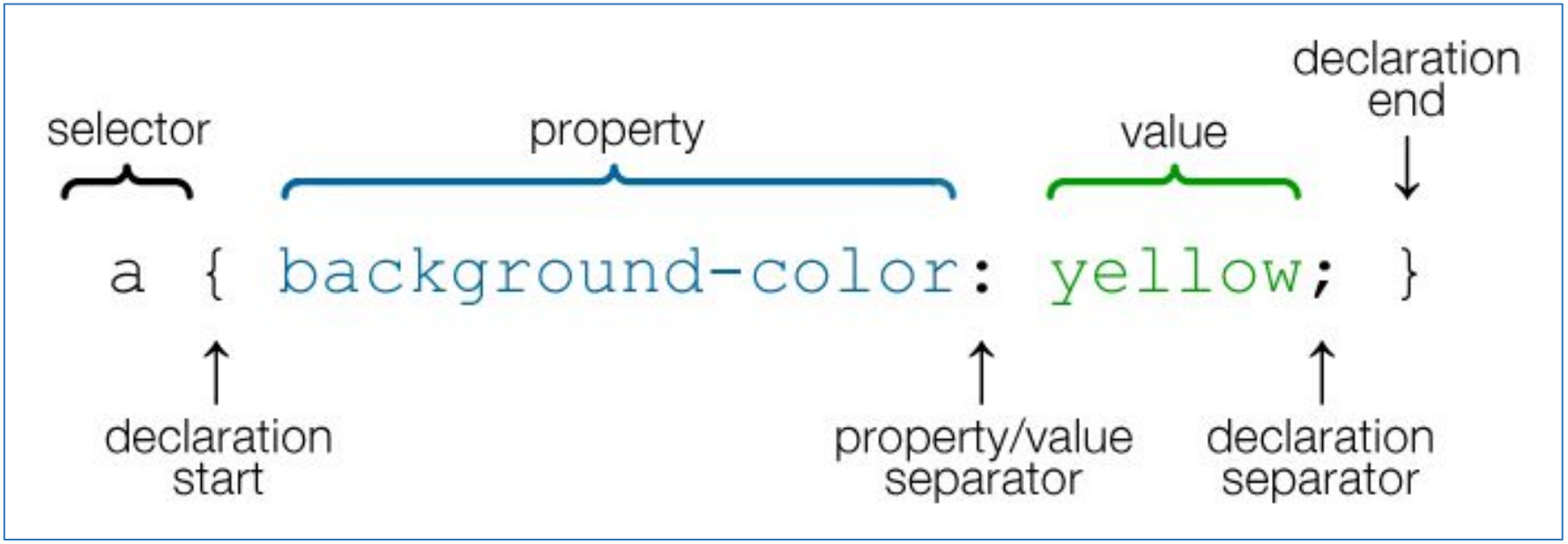
IDs?

CSS Syntax

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

- Classes use **.classname,** IDs use **#idname**, and elements use just their name.

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



Selectors

Element selector

Element name
(p, a, div, span, etc)

Applies to all <p> elements

```
p {
   background-color: blue;
}
```

Class Selector

Period (.) + variable name
(.myDiv, .phoneNumber, etc.)

Applies to all elements with class="classItem"

```
.classItem {
   background-color: orange;
}
```

ID Selector

Hash (#) + variable name
(#myDiv, #phoneNumber)

Applies to all elements with id="idItem"

```
#idItem {
   background-color: green;
}
```

CSS Selectors

```
p {
   background-color: blue;
}

.classItem {
   background-color: orange;
}

#idItem {
   background-color: green;
}
```

+

```
<p>
   A paragraph with a blue background.
</p>
<div class=“classItem">
   A div with an orange background.
</div>
<div id=“idItem">
   A div with a green background.
</div>
```

Lingering Questions

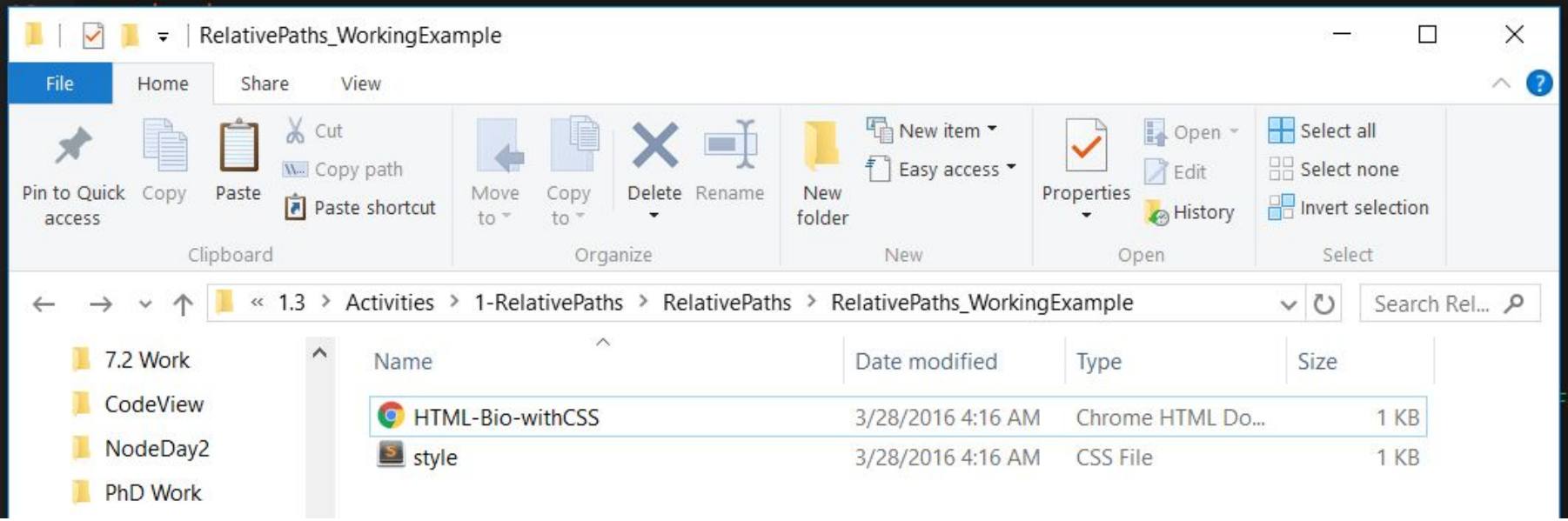
Questions so far?

Relative File Paths

Relative File Paths

```
<!DOCTYPE html>
<html>
<head>
    <title>1.2.9 Exercise</title>

    <!-- This critical line points your HTML to the CSS file. Notice the "relative" pathway  -->
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<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.

Quick Demo

Instructor: Demo

(RelativePaths_DEMO | 1-RelativePaths)

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.

Assignment

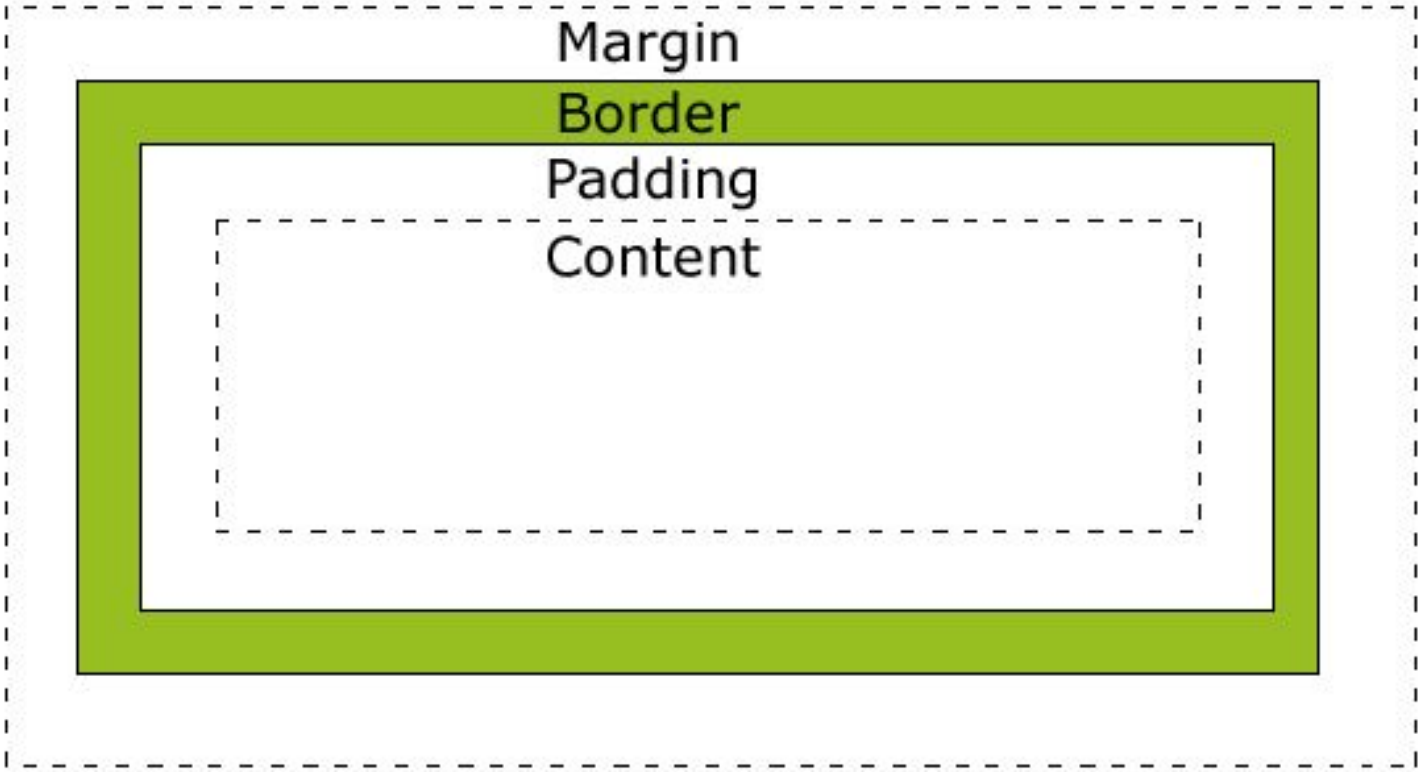
1. Unzip the folder sent to you via Slack.

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.

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?

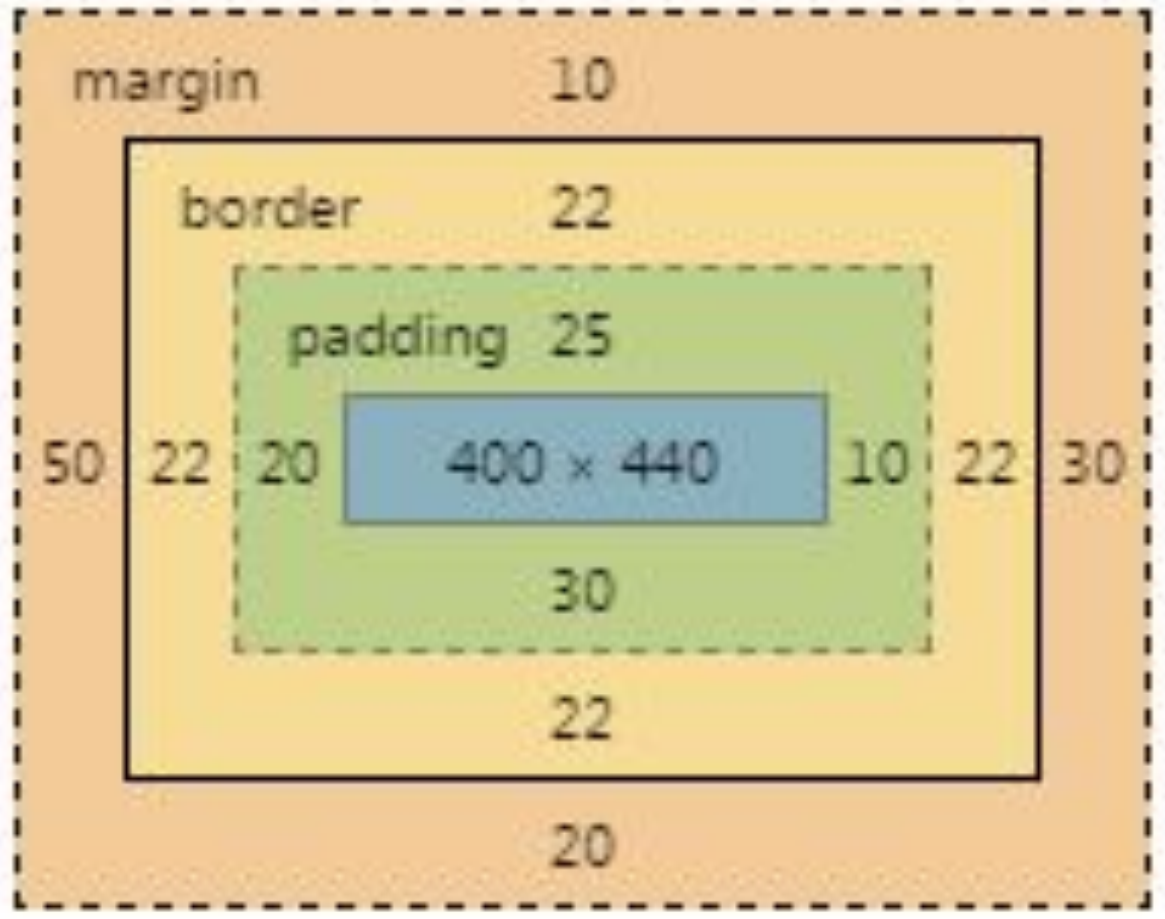
Total element width = content width + left padding + right padding + left border + right border + left margin + right margin

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)

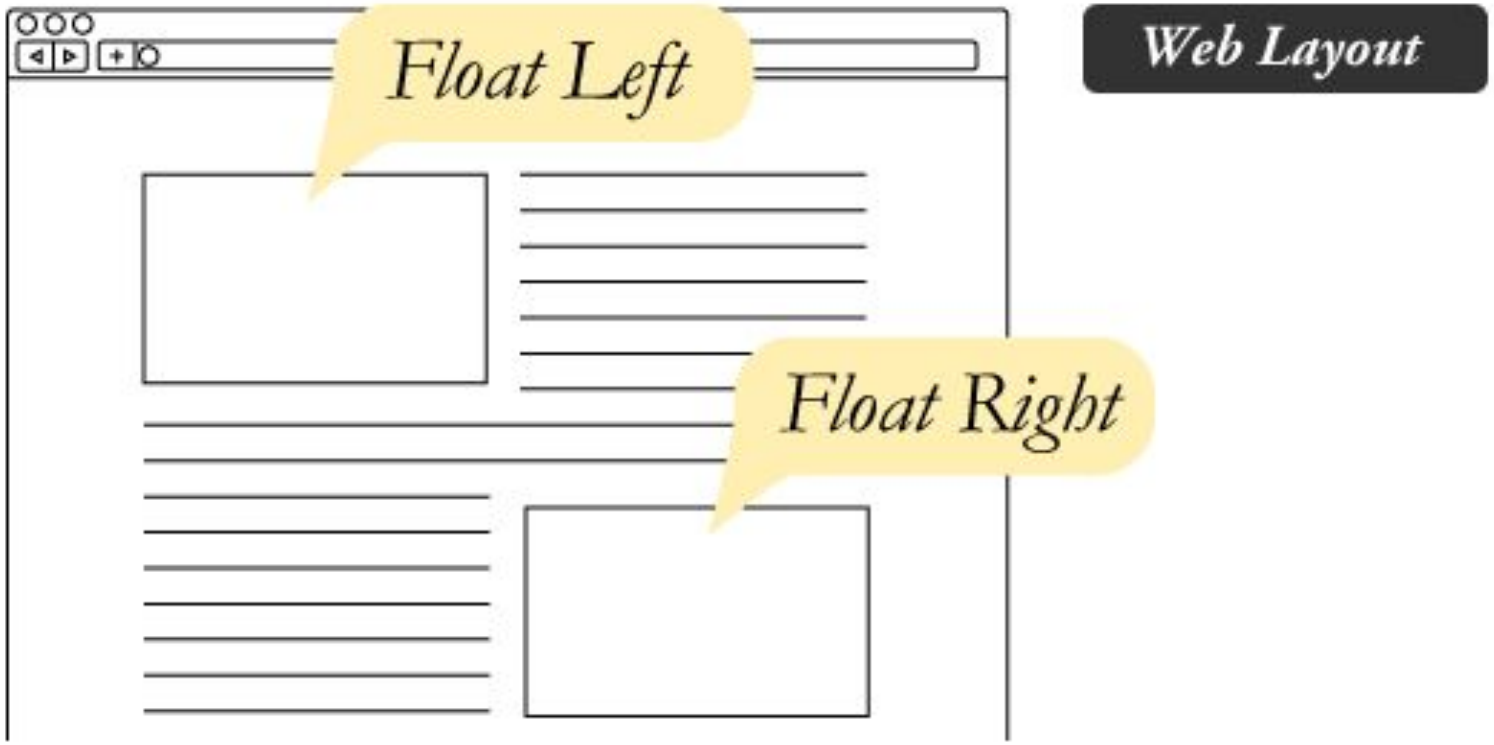
We Be Floatin’

Take a Facebook Break…

Warning!

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

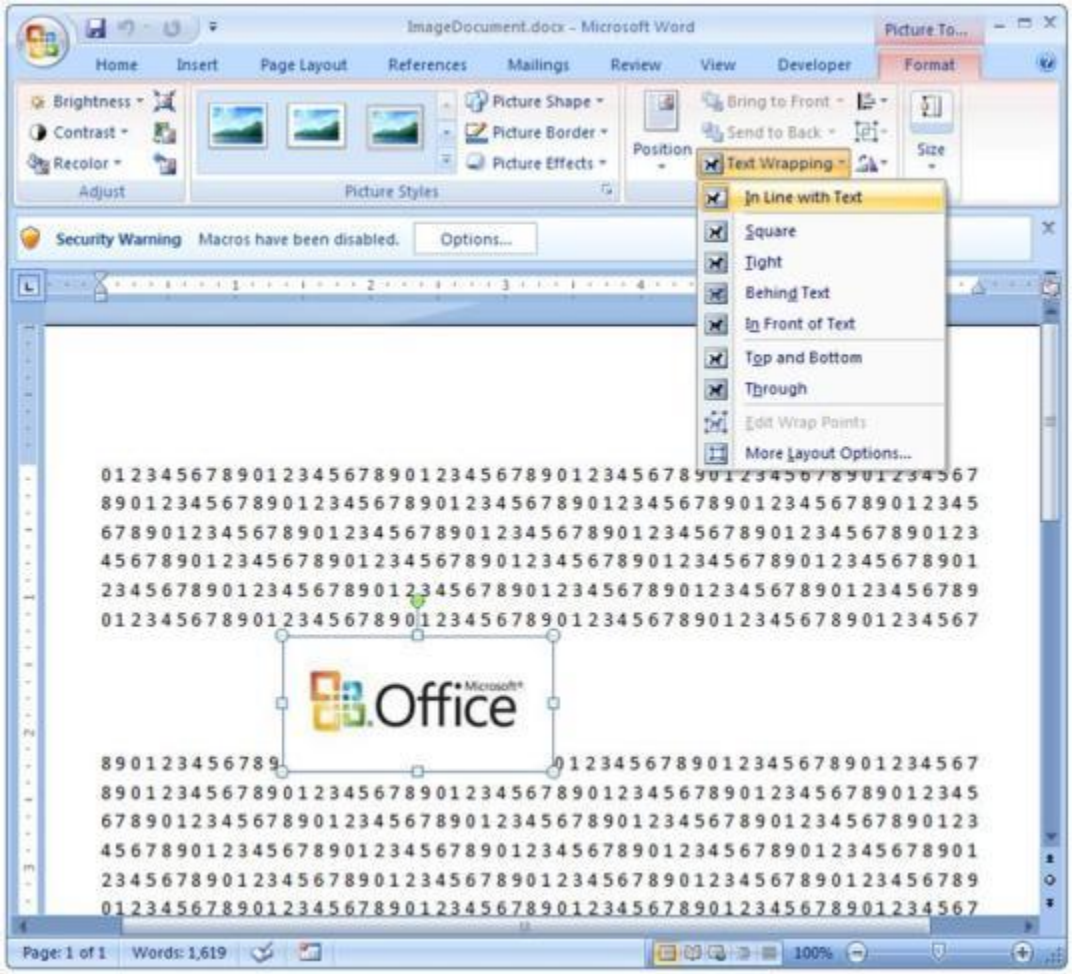
The Concept of “Flow”



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

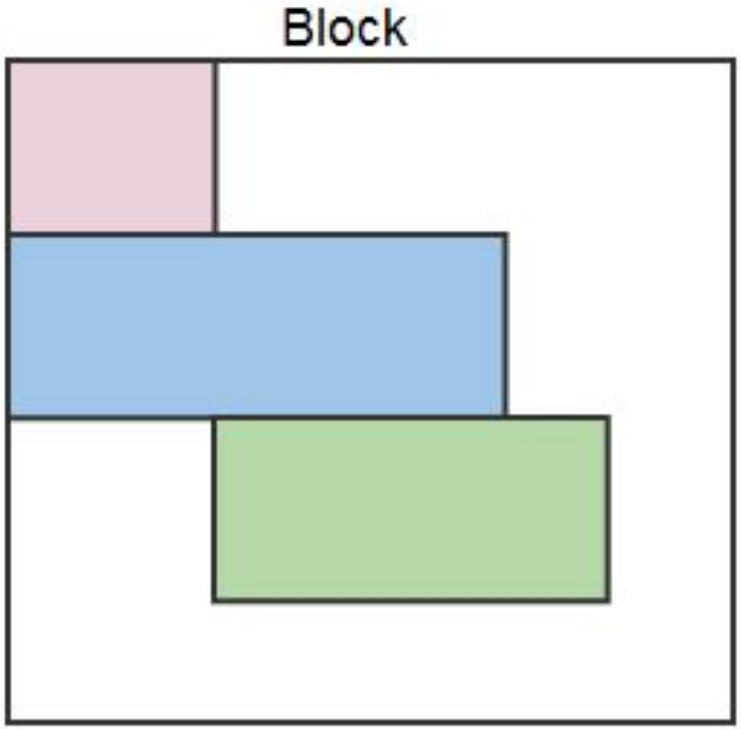
- This means that HTML elements force their adjacent elements to **flow around them**.

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.

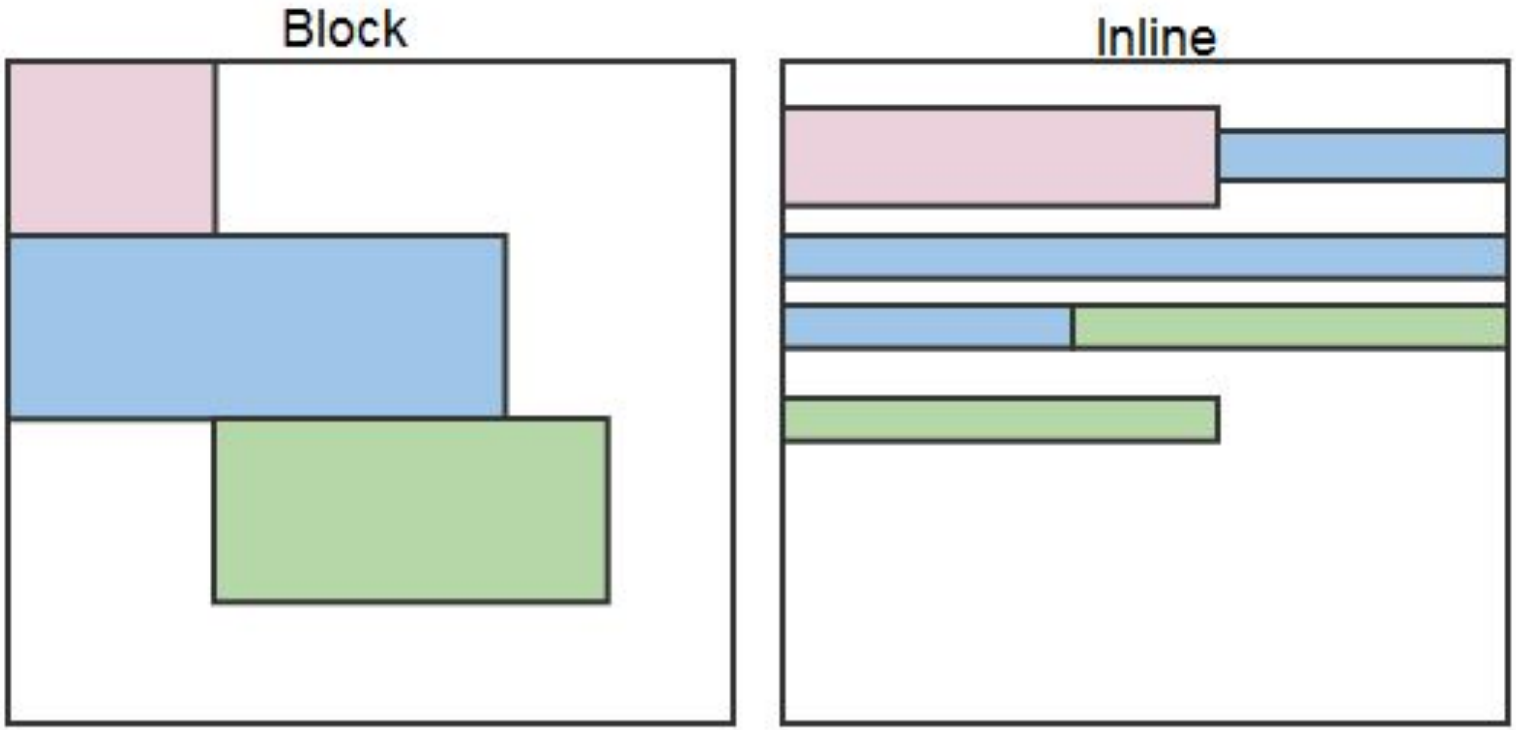
Block Elements



- 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.

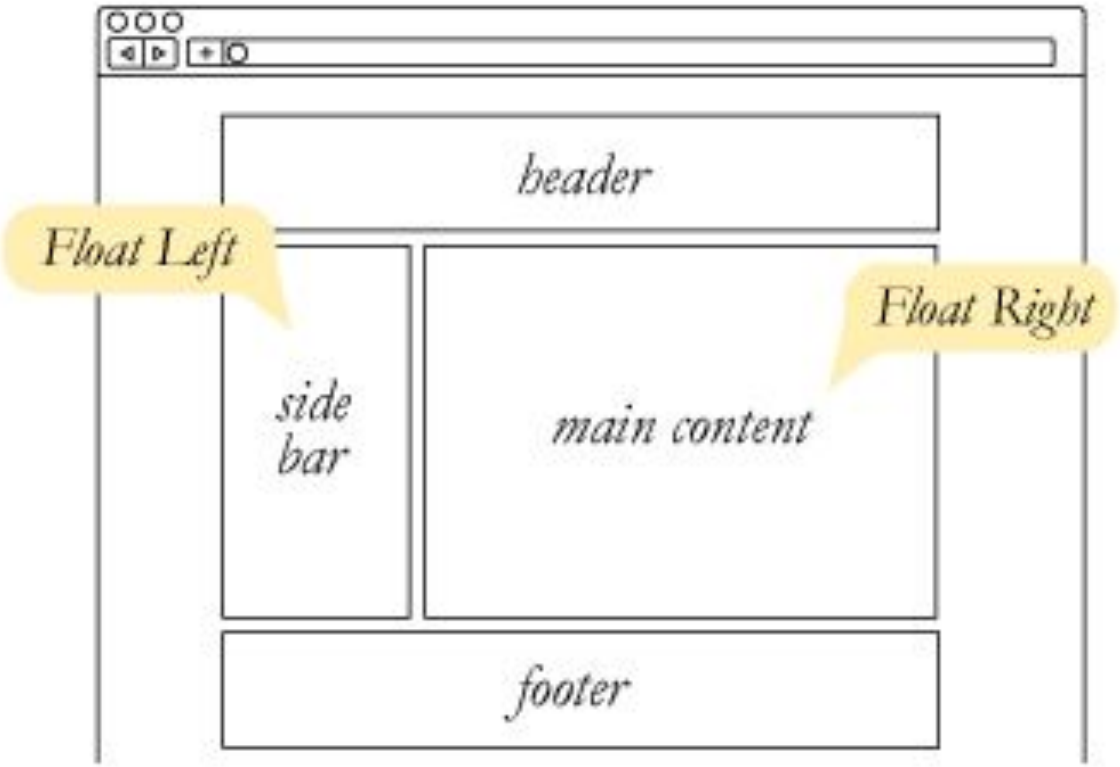
Block Elements vs. Inline Elements



- 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;
}
```

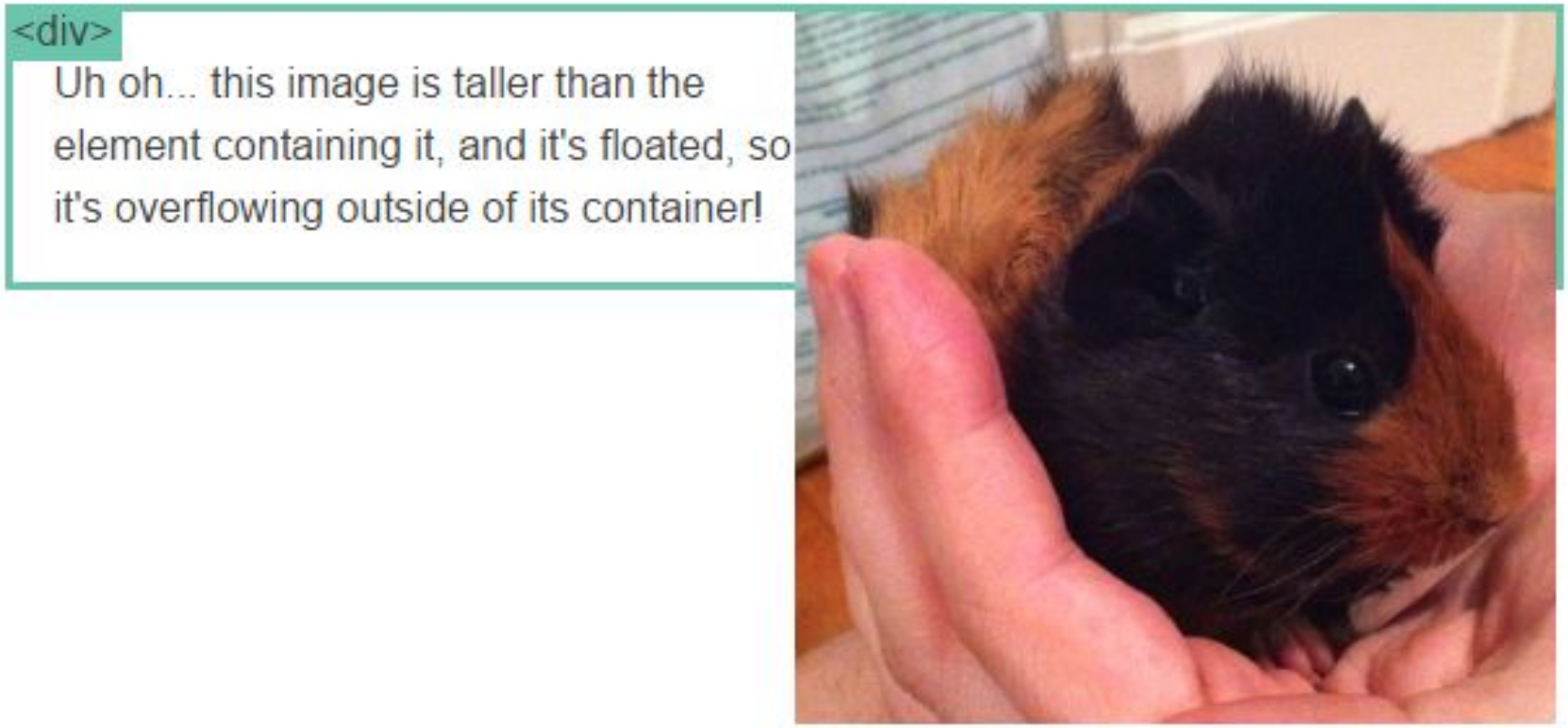
- To transform these block elements into inline elements, we use a CSS property called **float.**

- 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



- 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

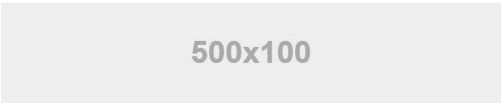
```
.clearfix::after {
    content: "";
    display: block;
    clear: both;
}
```

- **::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!



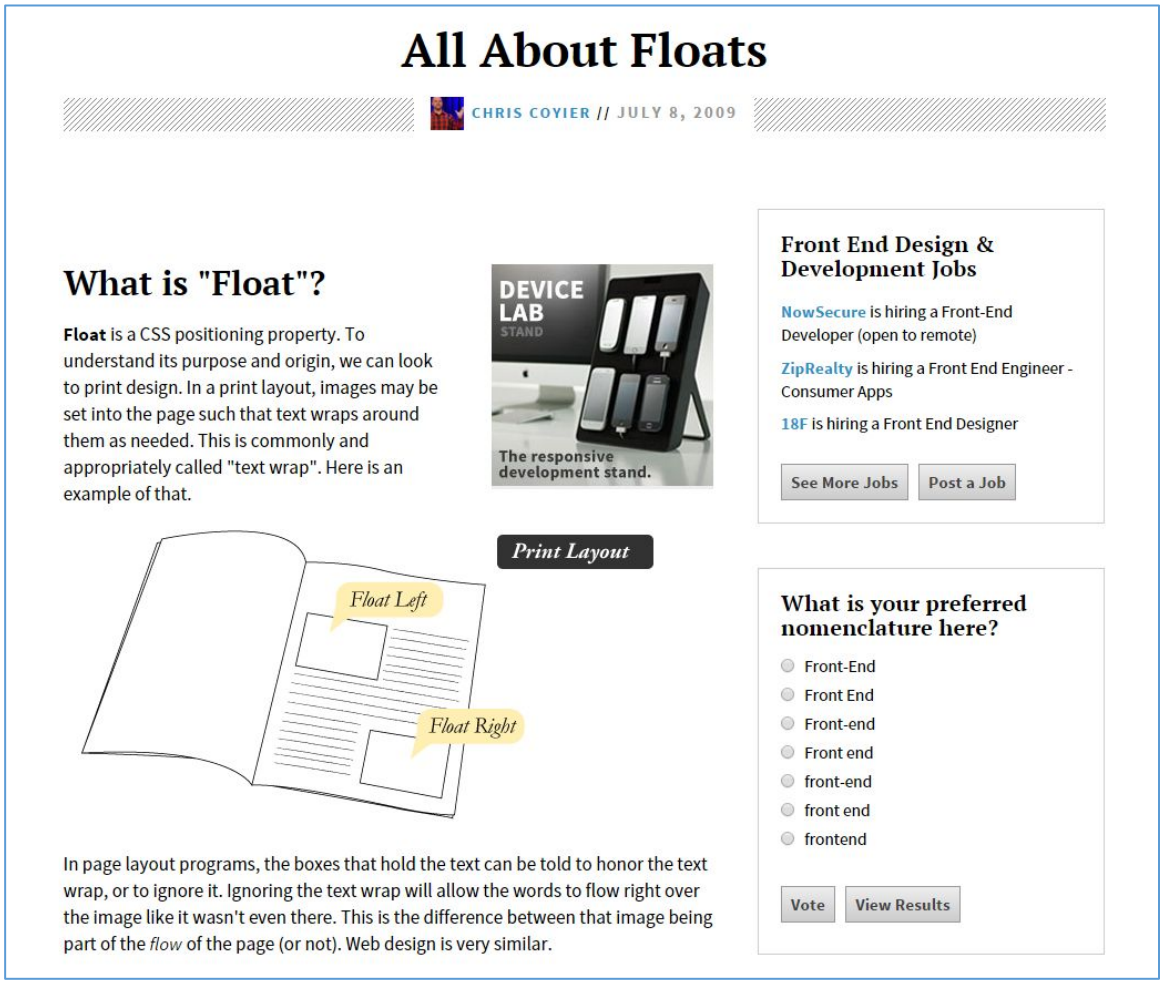
Quick Demo

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.

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/

Assignment

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

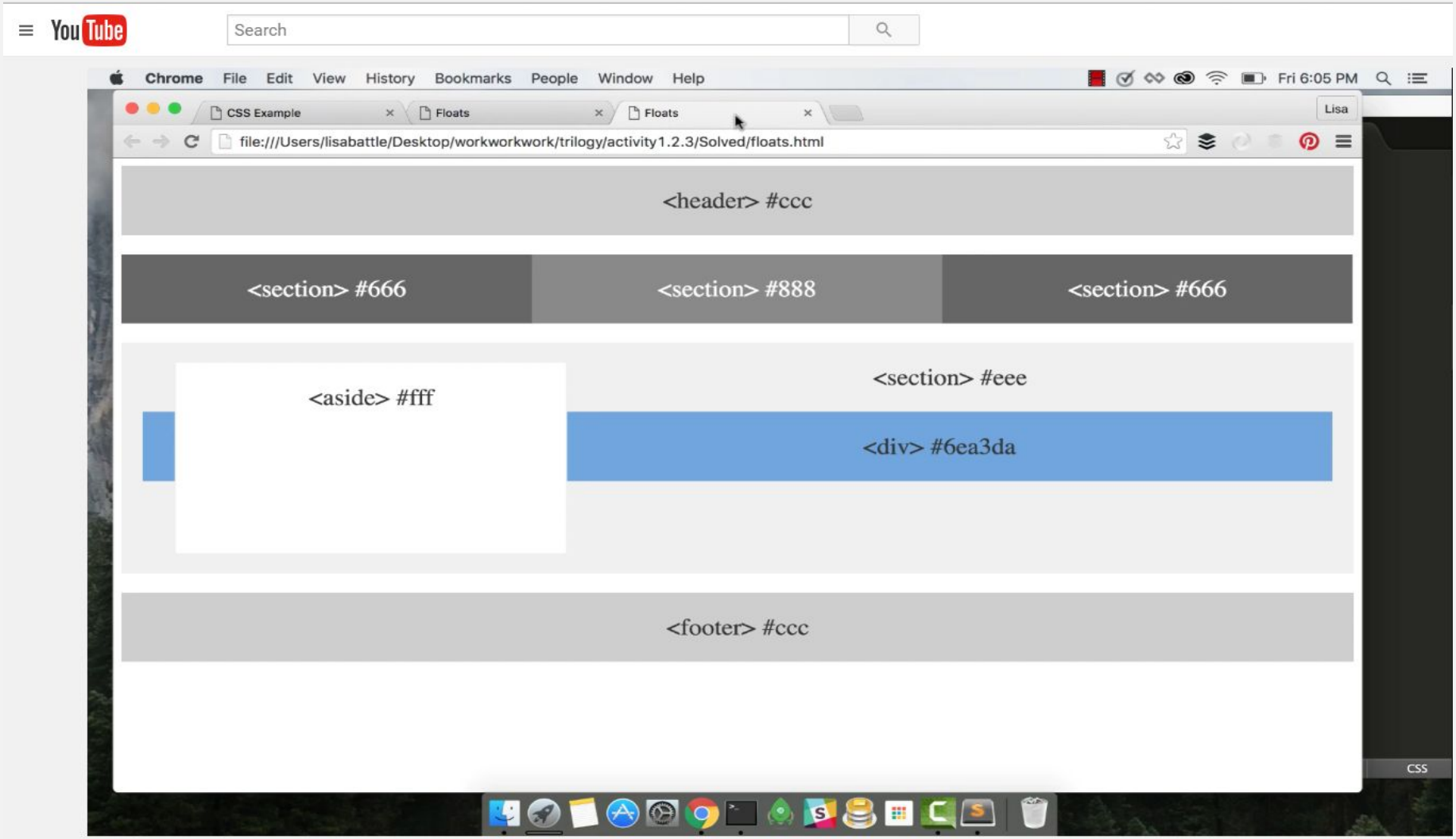
Check your Slack for more instructions.

> YOUR TURN!!

Good work!

Your brain may rest now…

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

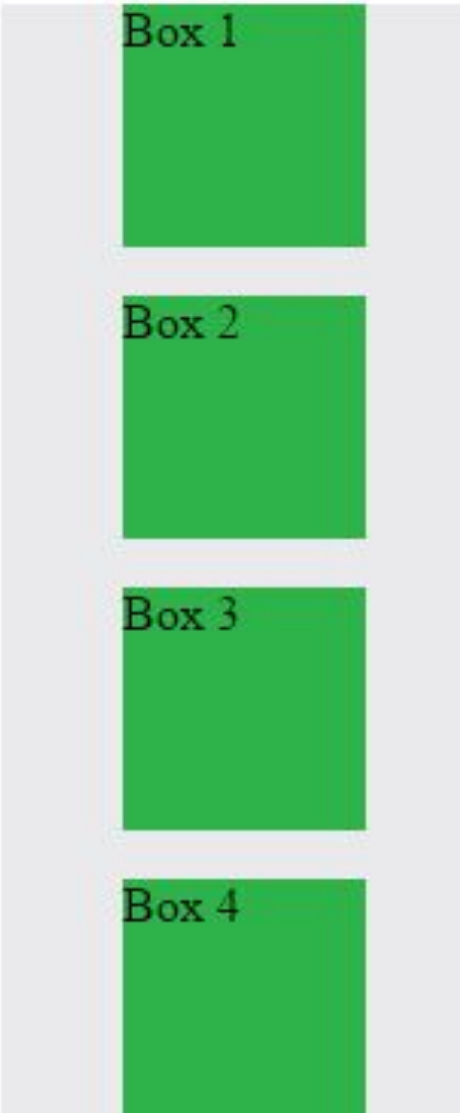


Video Link: https://youtu.be/0lpxKw6E90Y

BREAK!

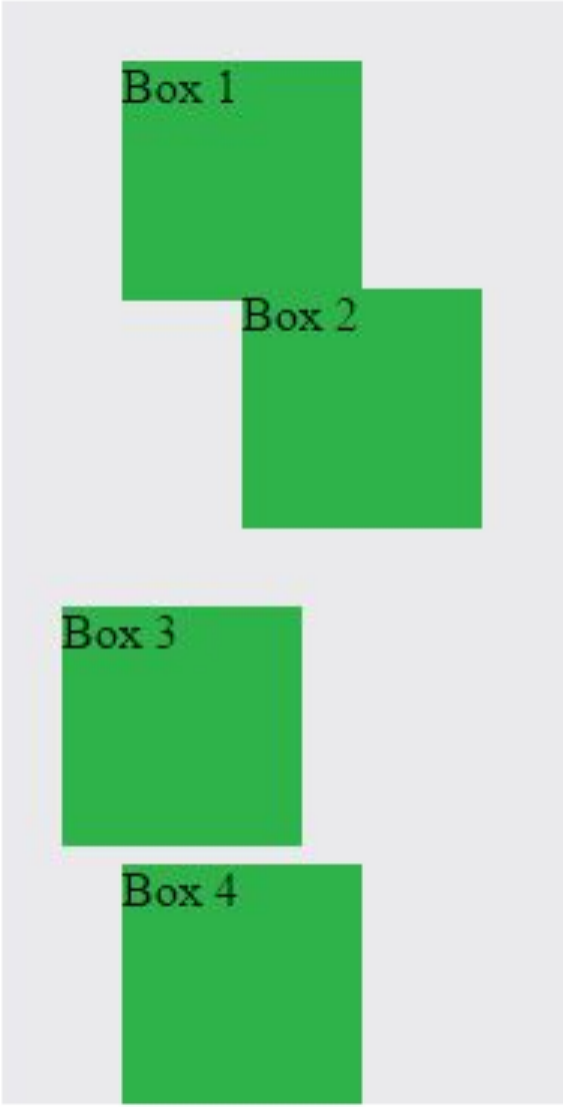
CSS Positioning

Position: Static (Default)



- **Four boxes placed statically (default)**

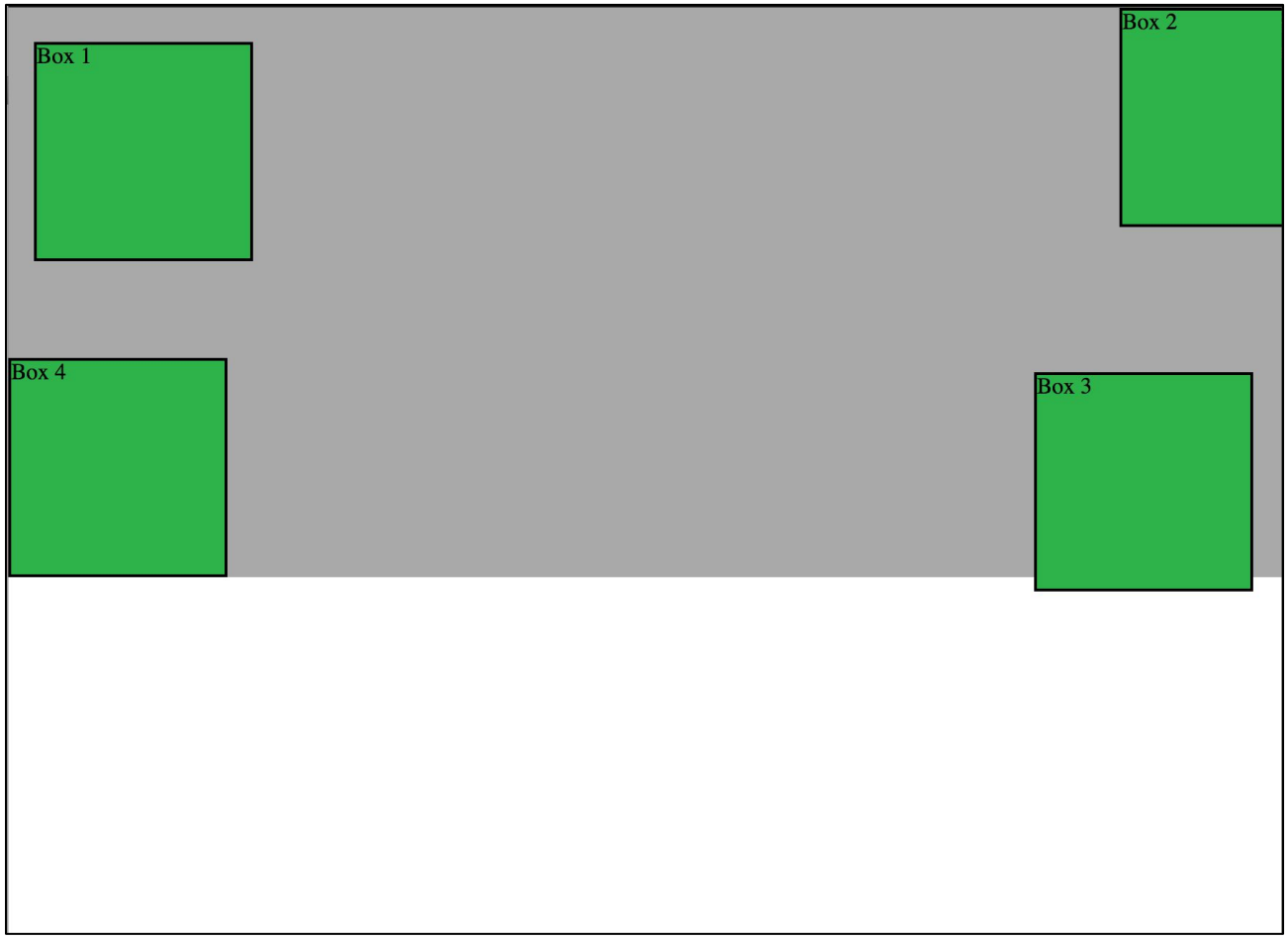
Position: Relative



- **Switching the boxes to relative will nudge the boxes in relation to their "original" location.**

```css
.box {
  background: #2db34a;
  height: 80px;
  position: relative;
  width: 80px;
}
.box-1 {
  top: 20px;
}
.box-2 {
  left: 40px;
}
.box-3 {
  bottom: -10px;
  right: 20px;
}
```

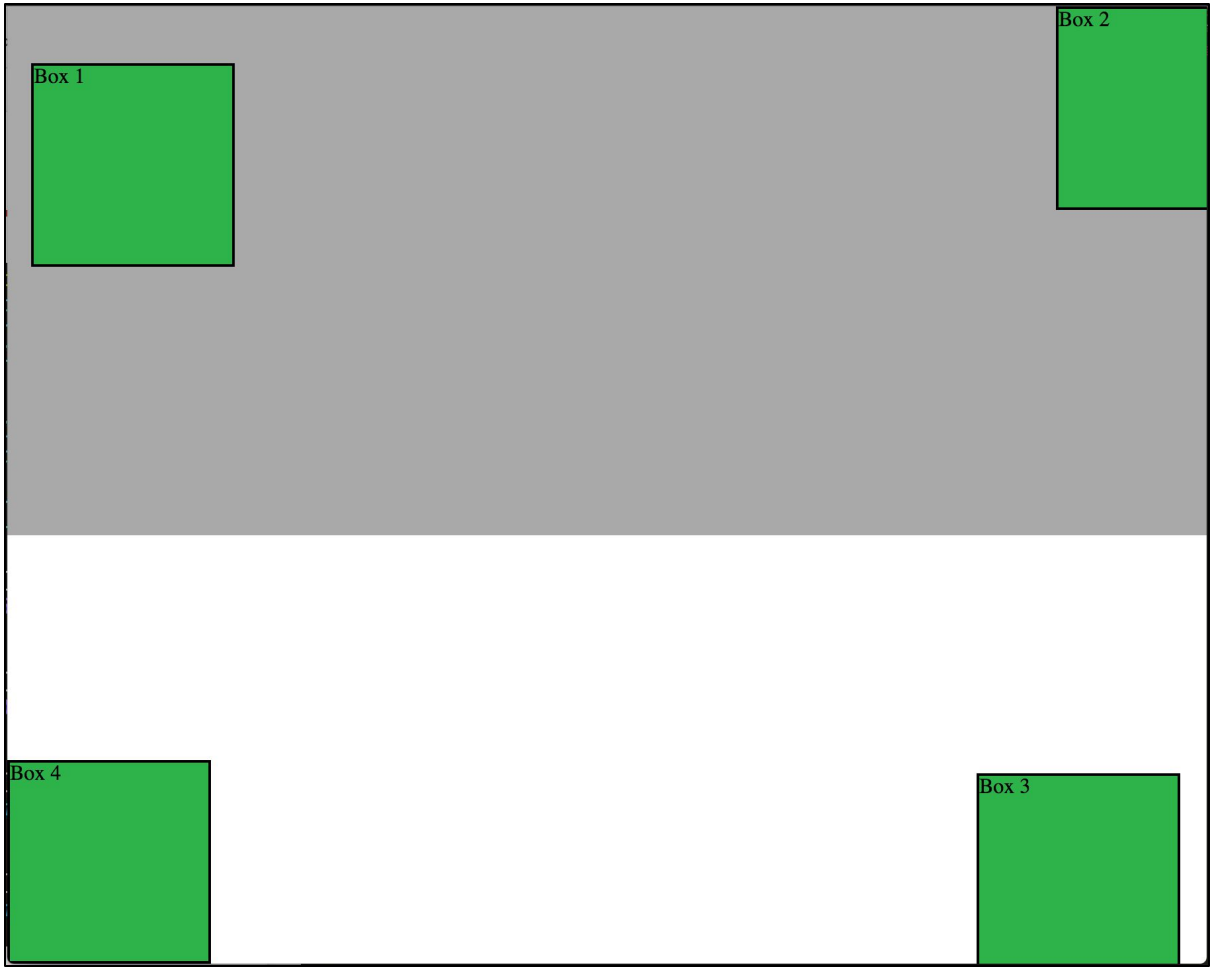
Position: Absolute




```css
.box-set {
  height: 400px;
  background: darkgray;
  position: relative;
}
.box {
  position: absolute;
  height: 150px;
  width: 150px;
  background: #2db34a;
  border: 2px solid black;
}
.box-1 {
  top: 6%;
  left: 2%;
}
.box-2 {
  top: 0;
  right: -40px;
}
.box-3 {
  bottom: -10px;
  right: 20px;
}
.box-4 {
  bottom: 0;
}
```

Positioned relative to nearest positioned ancestor

Position: Fixed



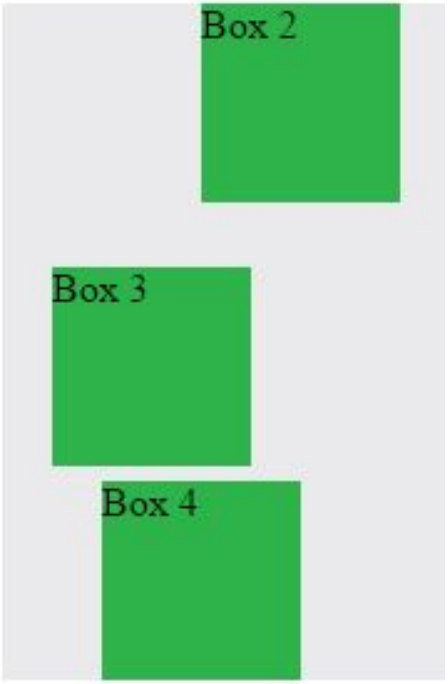

```
.box-set {
  height: 400px;
  background: darkgray;
}
.box {
  position: fixed;
  height: 150px;
  width: 150px;
  background: #2db34a;
  border: 2px solid black;
}
.box-1 {
  top: 6%;
  left: 2%;
}
.box-2 {
  top: 0;
  right: -40px;
}
.box-3 {
  bottom: -10px;
  right: 20px;
}
.box-4 {
  bottom: 0;
}
```

Position with exact coordinates to the browser window

Layering with Z-Index

- **Z-Index allows you to layer elements on top of each other when they're positioned.**

Hiding Things



- “Display: none” allows us to hide elements from view.

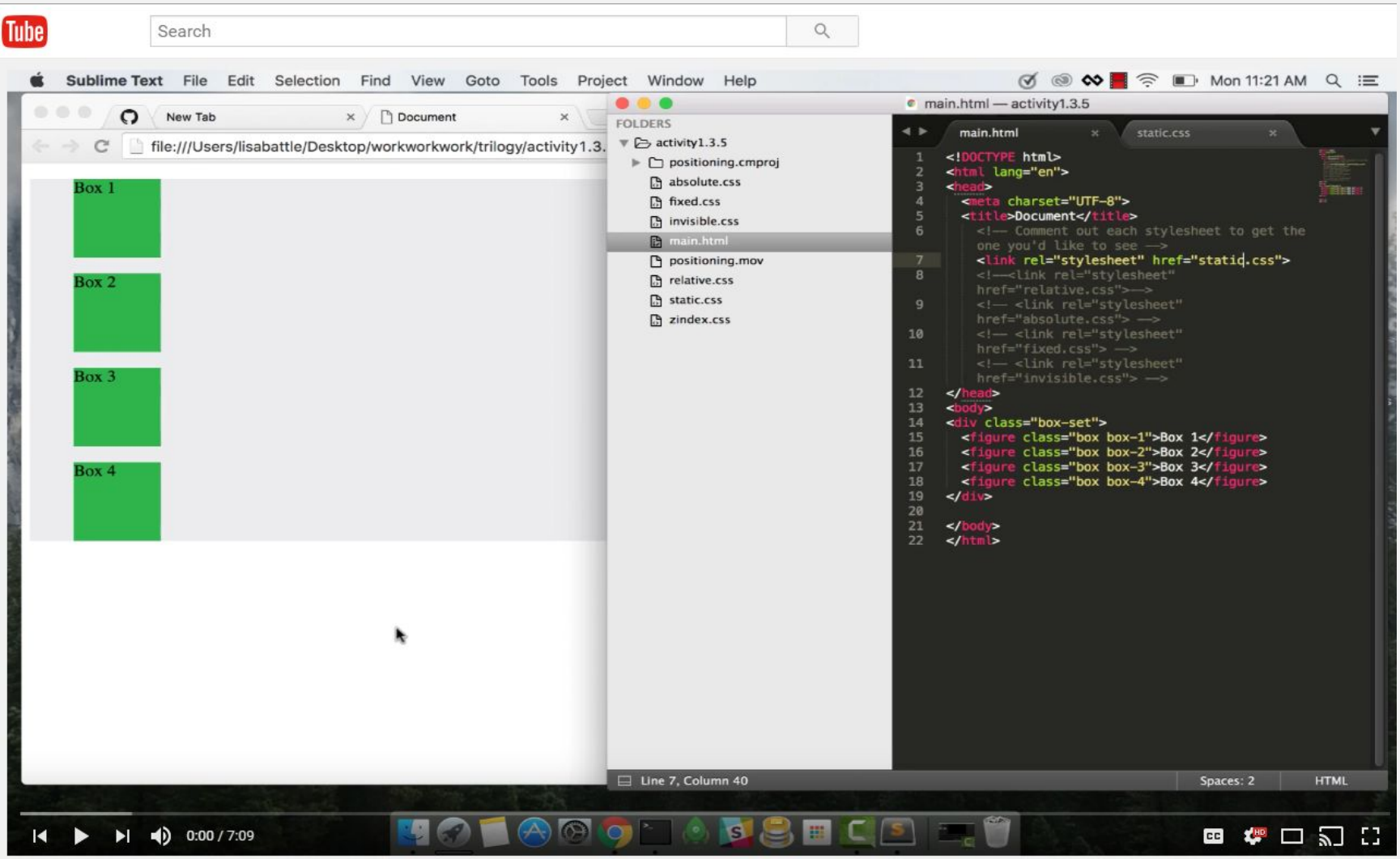
- This will become useful in later sections, when we’ll be hiding and revealing specific HTML elements of our choice.

Quick Demo

Instructor: Demo

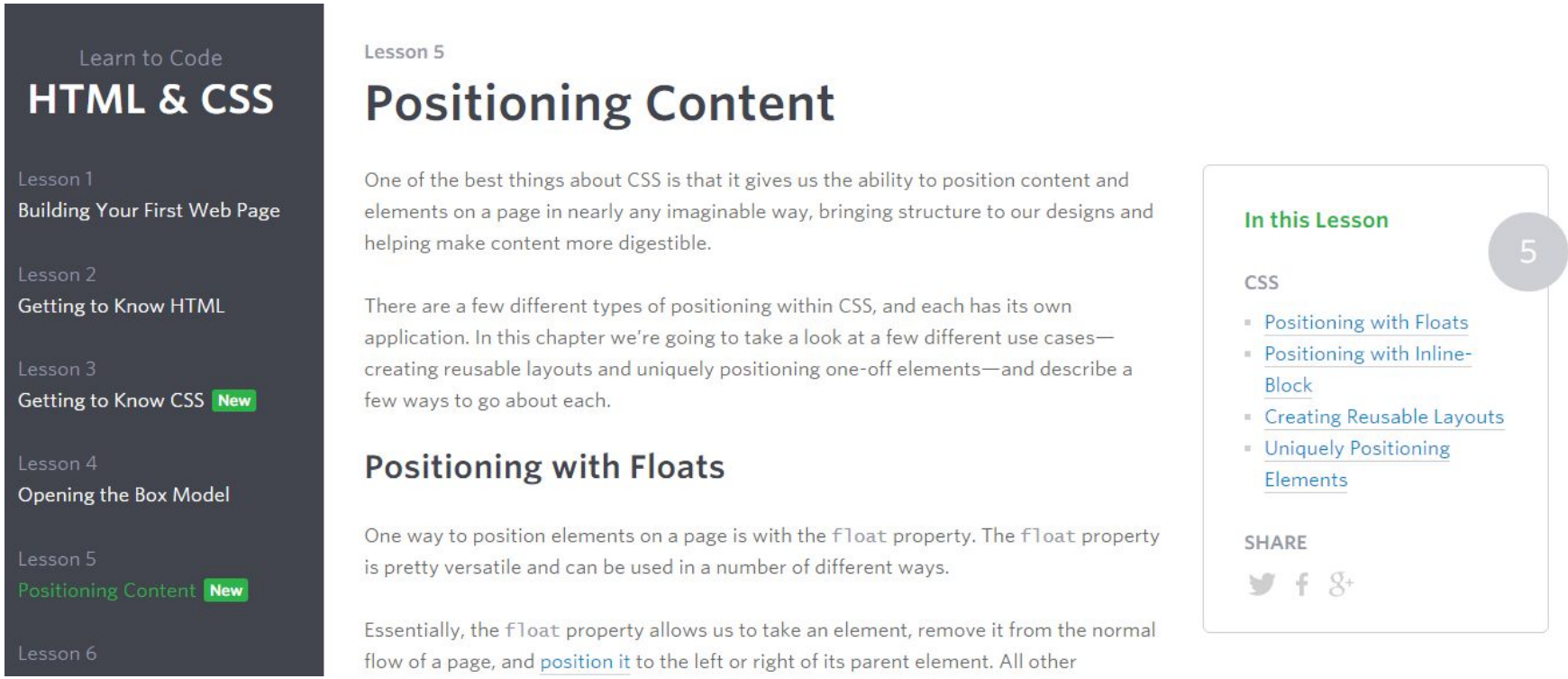
(5-CSS_PositionedLayout)

Video Walkthrough!



Video Link: https://youtu.be/sHfJn0jqBro

Great Resource



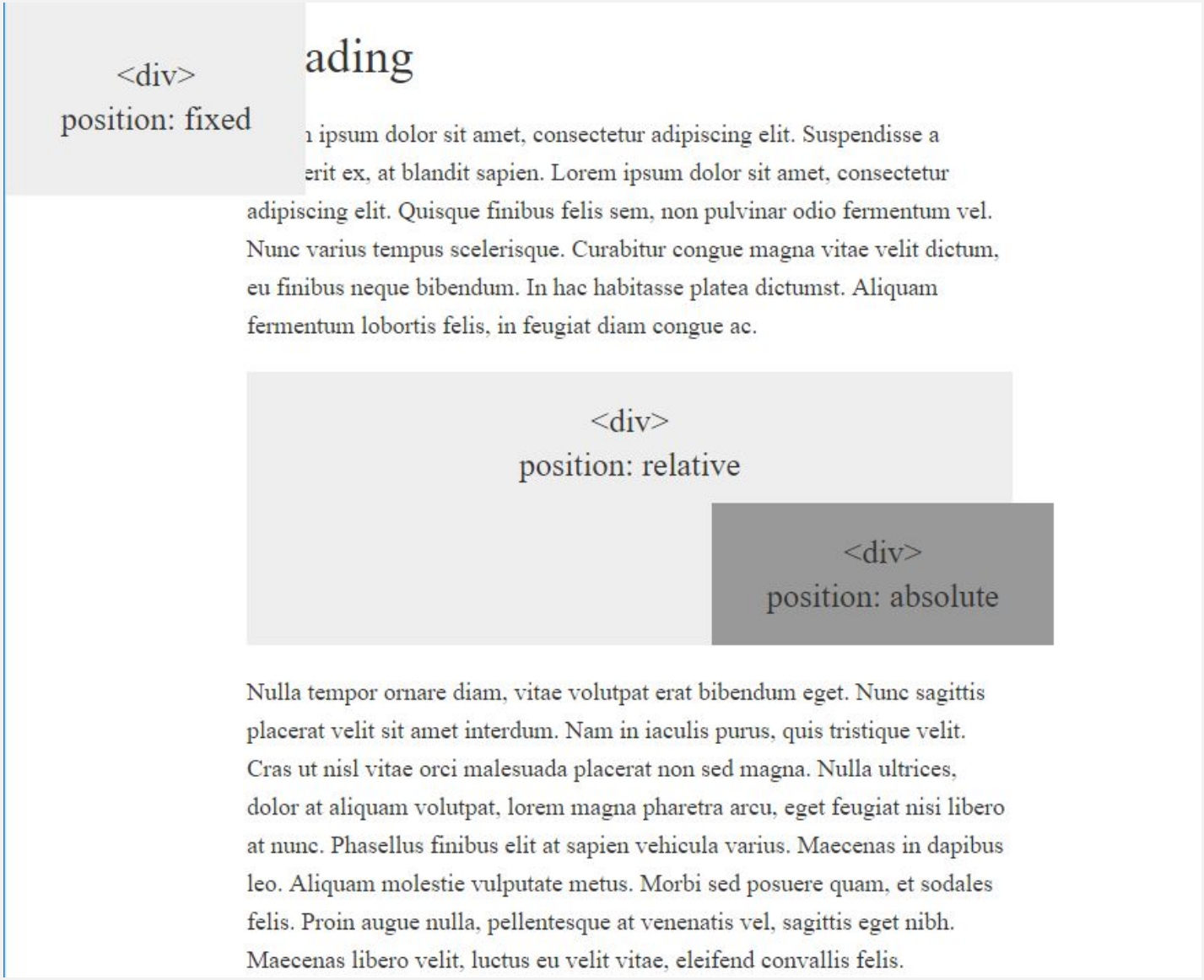
- **Another great read for front-end developers:** http://learn.shayhowe.com/html-css/positioning-content/

Assignment

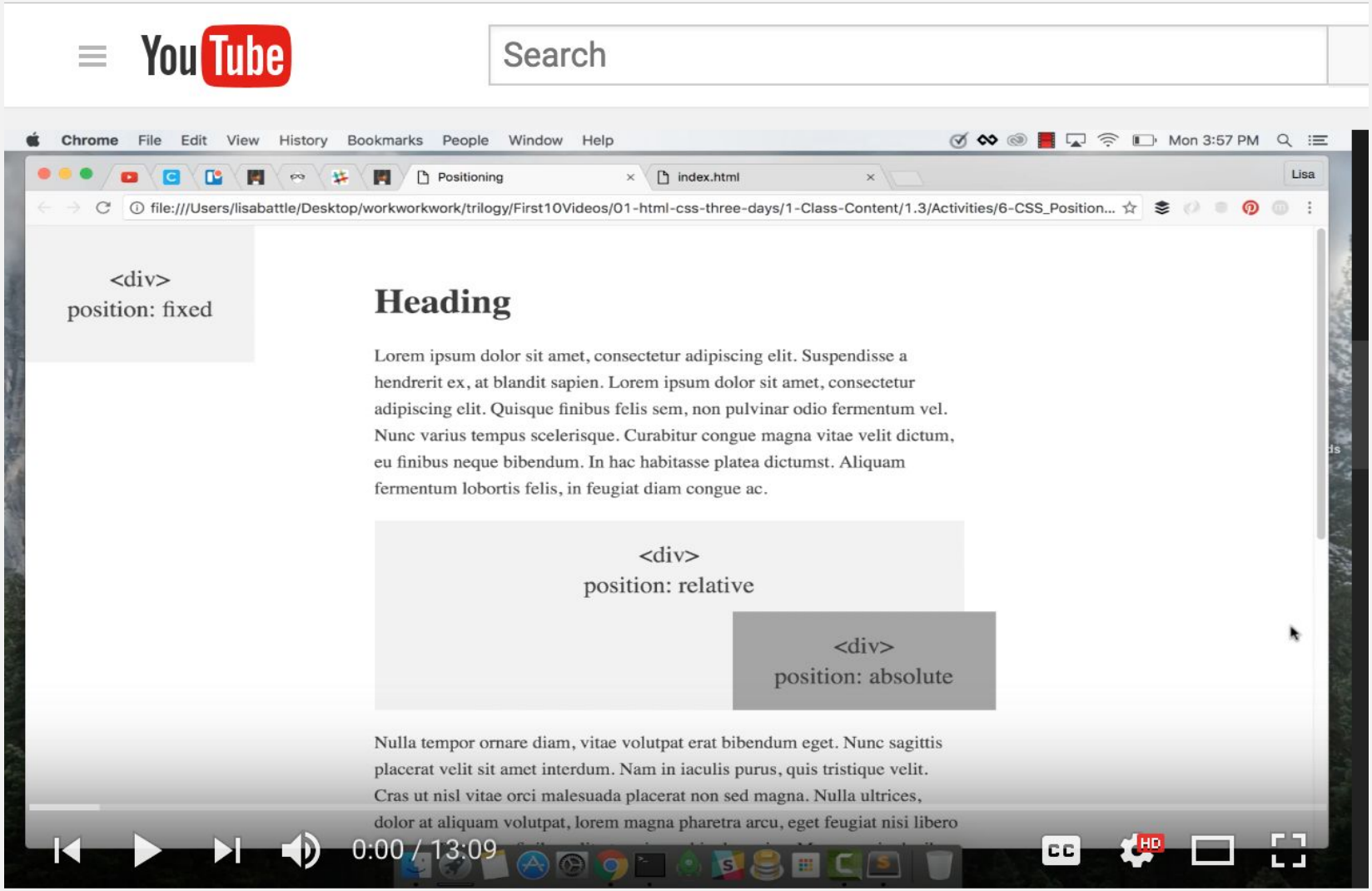
In this activity, you'll flex your newfound positioning skills by creating another conceptual layout. Eyeball the design to your best ability.

Check your Slack for additional instructions.

> YOUR TURN!!

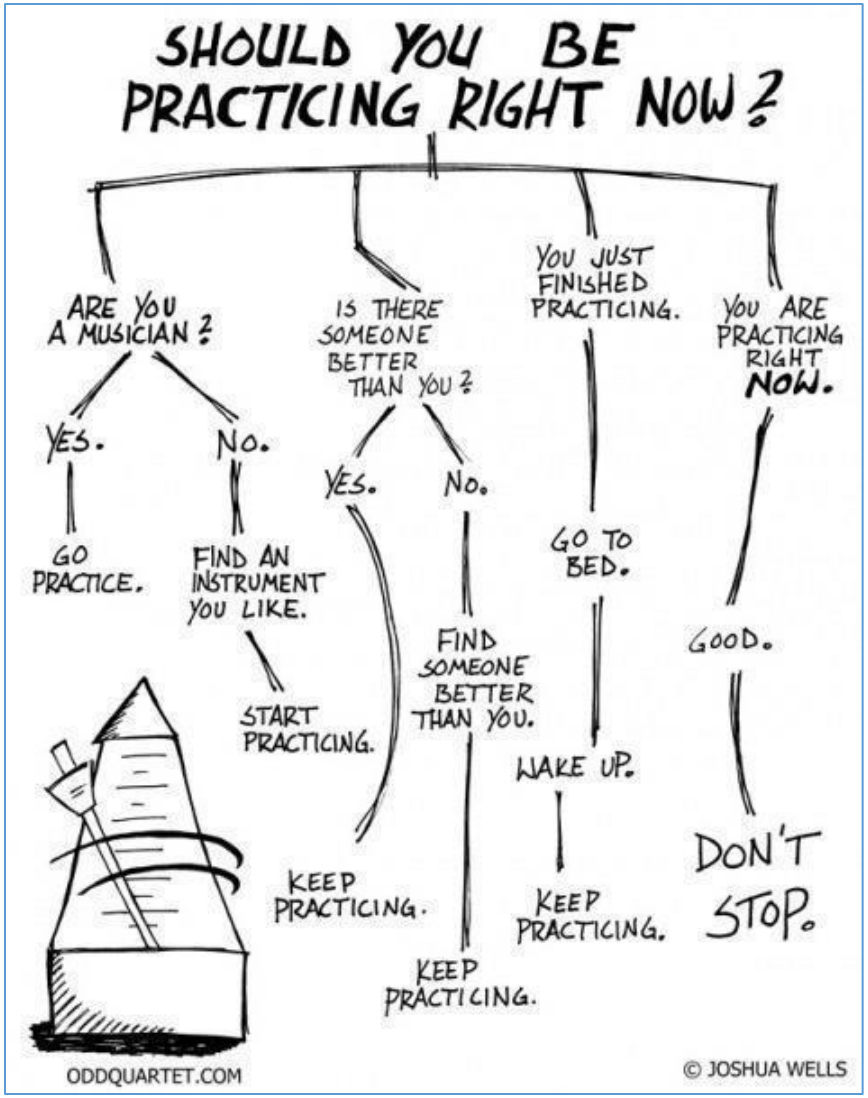


Video Walkthrough!



Video Link: https://youtu.be/yWXgnQaWSW0

Advice



Re-do this at home.

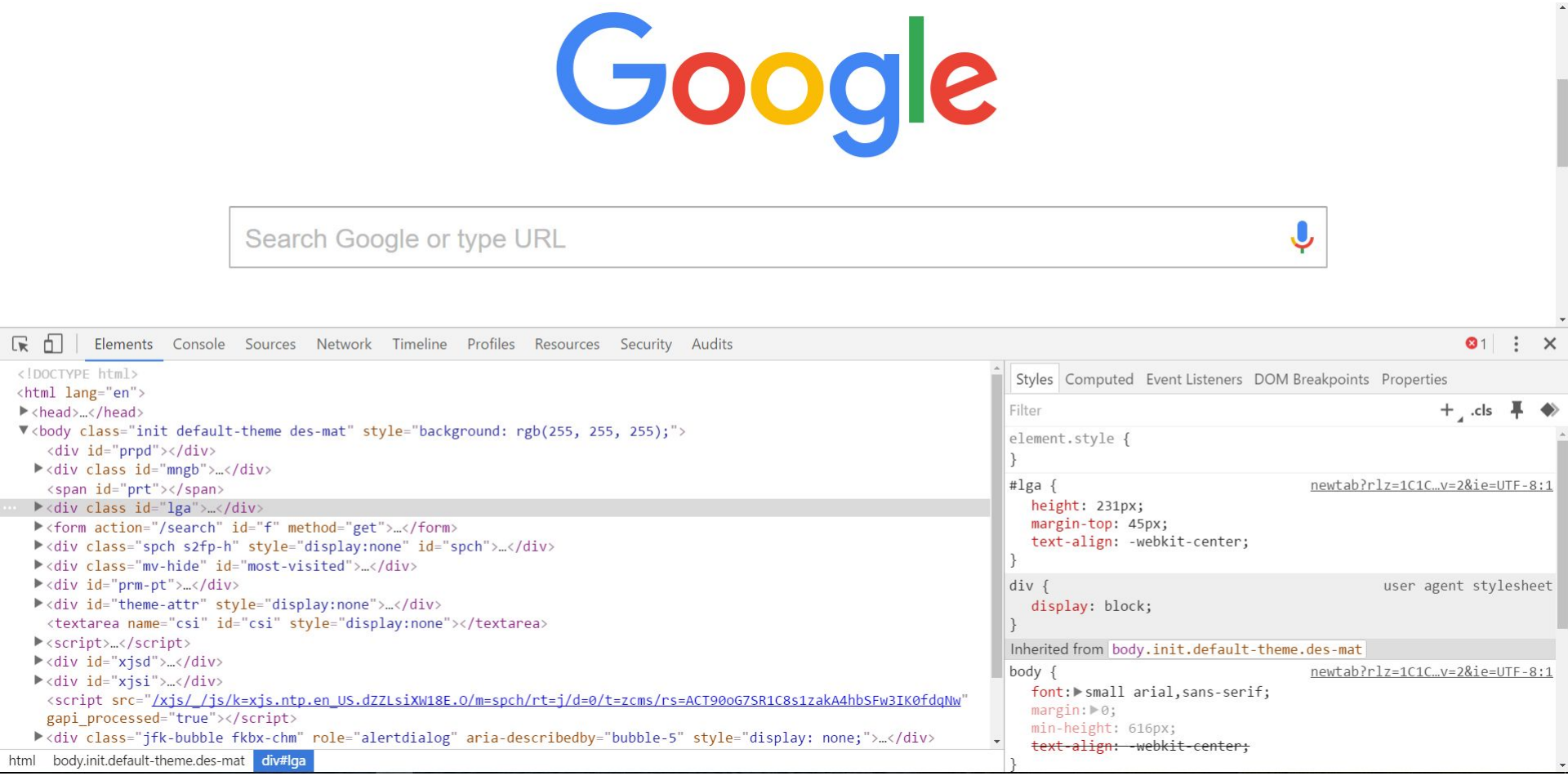
We designed this exercise to firm up your HTML/CSS skills.

REMEMBER:

The best way to learn web development is to PRACTICE!

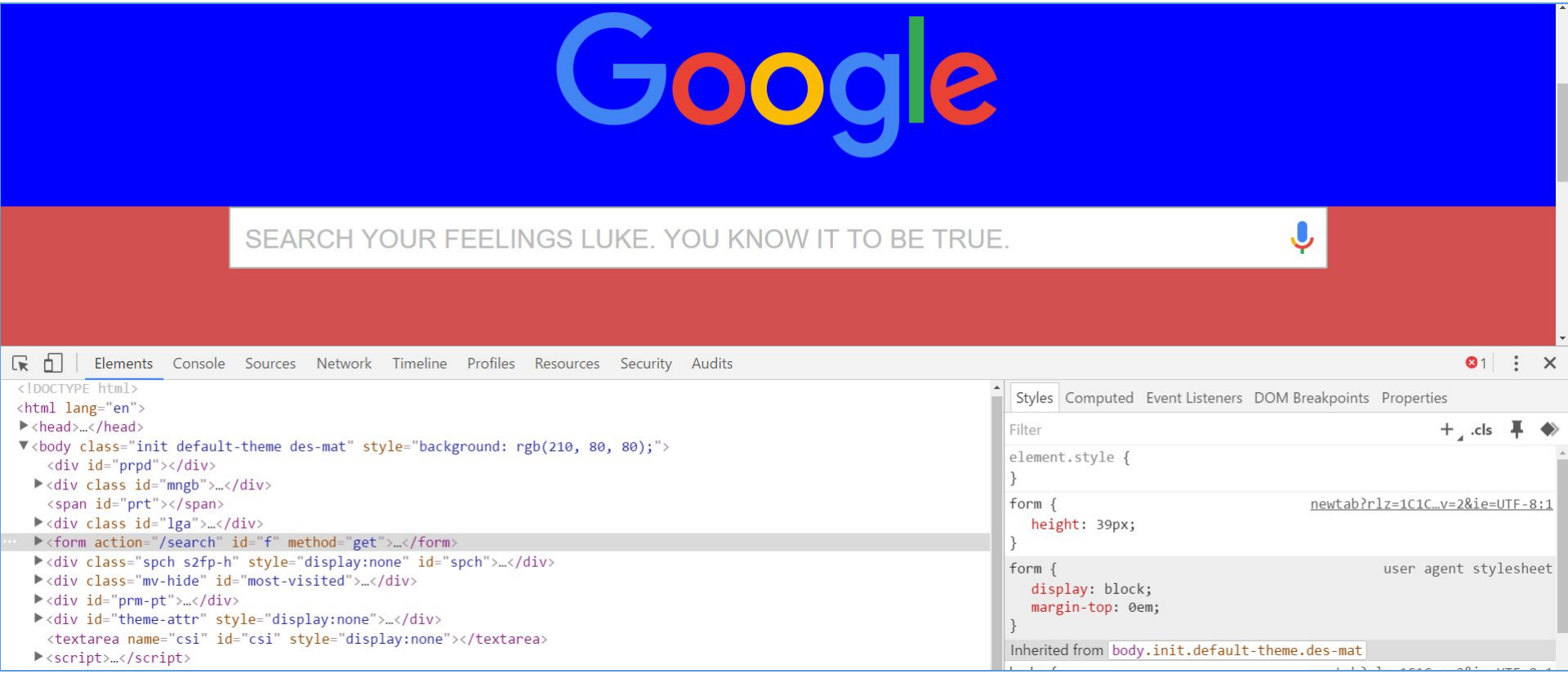
Chrome Inspector

Chrome Inspector is Your Friend



- **To access chrome inspector, right click on a page. Then hit “Inspect.”**
- It will allow you to inspect the HTML, CSS and more!

Chrome Inspector is Your Friend



- **You can even edit the HTML/CSS of a webpage and instantly view your changes in the browser!**
- This works on any website, whether yours or not.

Next Class!

We'll be coming back to this in our next class.

Recap + Reassurances


You got this

Questions?

EXTRA MATERIAL

> YOUR TURN!!

Assignment

Take a website you commonly visit (Amazon, Google, Huff Po, etc.) and heavily modify it with Google Developer Tools.

You have 15 minutes to test Chrome Inspector. Try modifying:

- Content (change words).
- Colors.
- Spacing.
- Any other CSS style rules.

When you're done, send a screenshot to your class's Slack.

CSS Resets

Loading Multiple CSS Files ***(Very Important!!!)***

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Multiple CSS Files!!</title>
5      <link rel="stylesheet" href="assets/style1.css">
6      <link rel="stylesheet" href="assets/style2.css">
7      <link rel="stylesheet" href="assets/style3.css">
8  </head>
9  <body>
10     <header>
```

- We can link our documents to more than one stylesheet at a time—one of the most powerful features of CSS/HTML.

- By tapping into different stylesheets simultaneously, we can create complex layouts with plenty of design rules.

- Just remember: the loading ***order matters!!!***

What Browser?

By a show of hands…

Which browser do you use?

Battle of the Browsers

- Under the hood, web browsers often **render webpages differently** than their competition.
- These disparities could mean HTML/CSS displaying differently in each web client.
- Because of these potential divergences, web developers need to make their websites **cross-browser compatible**.

Reset.css (or Normalize.css)

- Reset.css will “reset” all browser-specific CSS. This means your site will appear the same in all browsers.

- However, you will have to re-style everything yourself.

Why CSS Resets Matter

1. It’s important for creating browser-compatible websites.

2. It’s an example of using someone else’s CSS in *your* website!!!

3. It’s a common Front-End Developer Interview question.

> YOUR TURN!!

Assignment

Follow the instructions given via Slack to incorporate a reset.css file in a basic HTML file.

Note the impact of the reset file makes after you include it.