

Day 3

Heroes of CSS

June 25th, 2016



UCF CODING BOOT CAMP

Admin Work

Homework Assignment

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

<https://github.com/UCF-Coding-Boot-Camp/06-16-VW-Class-Content>

- Homework Assignment #1 is Due:

MW Class – Wednesday, June 29th, 2016

TTH Class – Thursday, June 30th, 2016



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://github.com/UCF-Coding-Boot-Camp/06-16-VW-Class-Content> Subdirectories ending with --class-content
- **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!

Today is going to be a bit tough.



But trust us!
It will all look easy a few weeks from now.

***Don't expect to understand
EVERYTHING at once.***

Today is all about getting immersed



CSS Recap

Tell your neighbor all about it...

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.



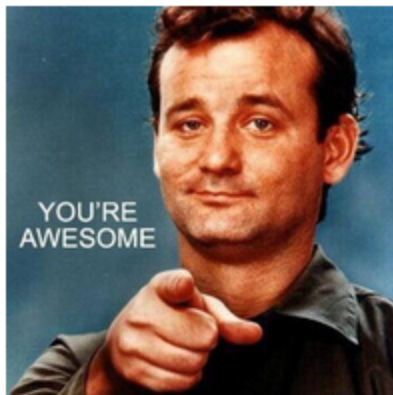
Basic HTML Page - Result

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](#)

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

How do we style HTML...

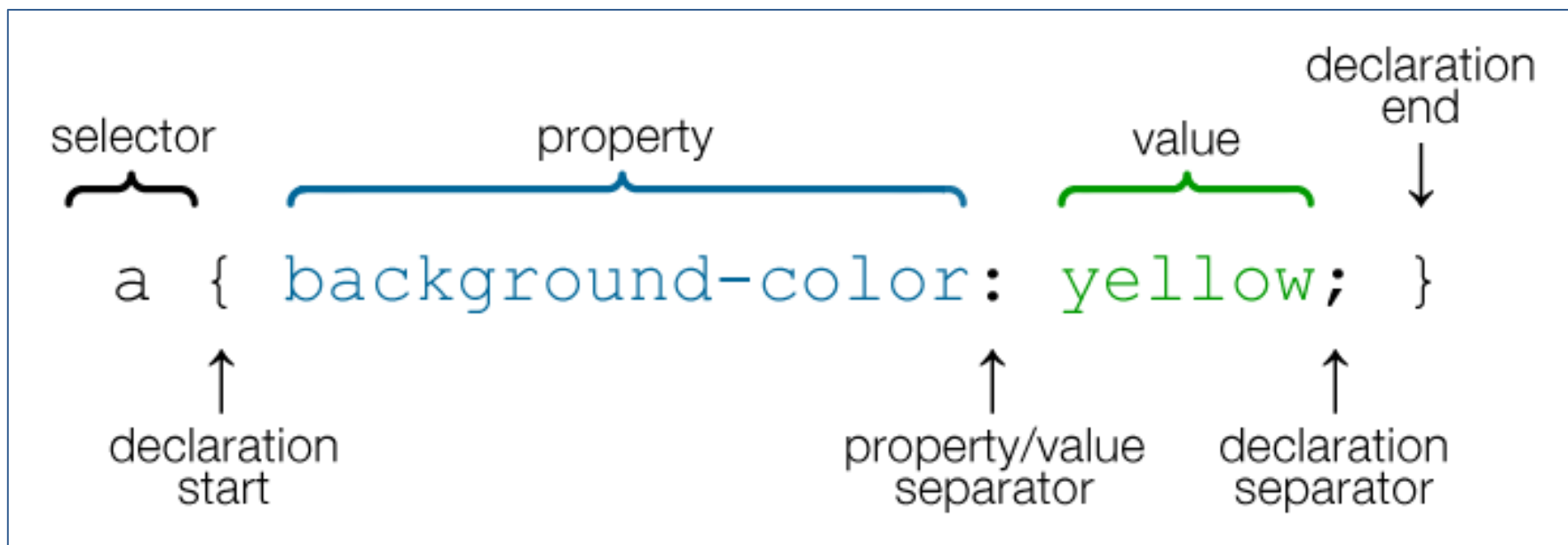
Elements?

Classes?

IDs?

CSS Syntax

- CSS works by hooking onto **selectors** added into HTML using “**classes** and **identifiers**”.
- Classes use **.classname**, IDs **#idname**, and elements their name
- Once hooked, we apply **styles** to those HTML elements using css .

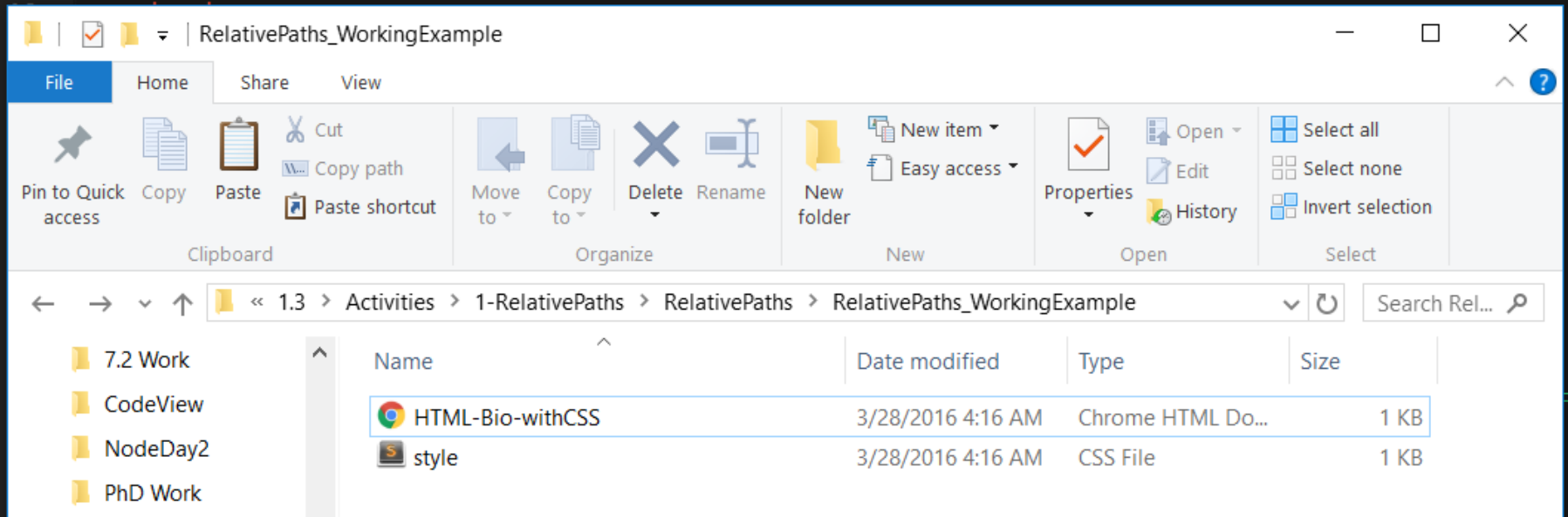


Questions so far?

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** allow us to access other files (like CSS or image files) relative to the file we are looking at.

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.

- Otherwise, when moving your web page or deploying your site – all of your links will fail to load correctly.
- Remember there is no such thing as a “C:” drive on the internet.



Instructor: Demo
(RelativePaths_DEMO | 1-RelativePaths)

Assignment

Unzip the folder sent to you via slack.

Then edit the HTML files inside of each “RelativePaths” folder such that each HTML file can access the CSS file using relative paths.

An example is given to you in the “RelativePaths_WorkingExample” folder.

Box Model

Boxes Upon Boxes



In CSS, every element can be considered to fit within a series of boxes. Each box can be individually adjusted to provide spacing between elements or to fill in elements with colors.

Styles can applied as follows: 20px 10px 10px 20px (top, right, bottom, left)


```
#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

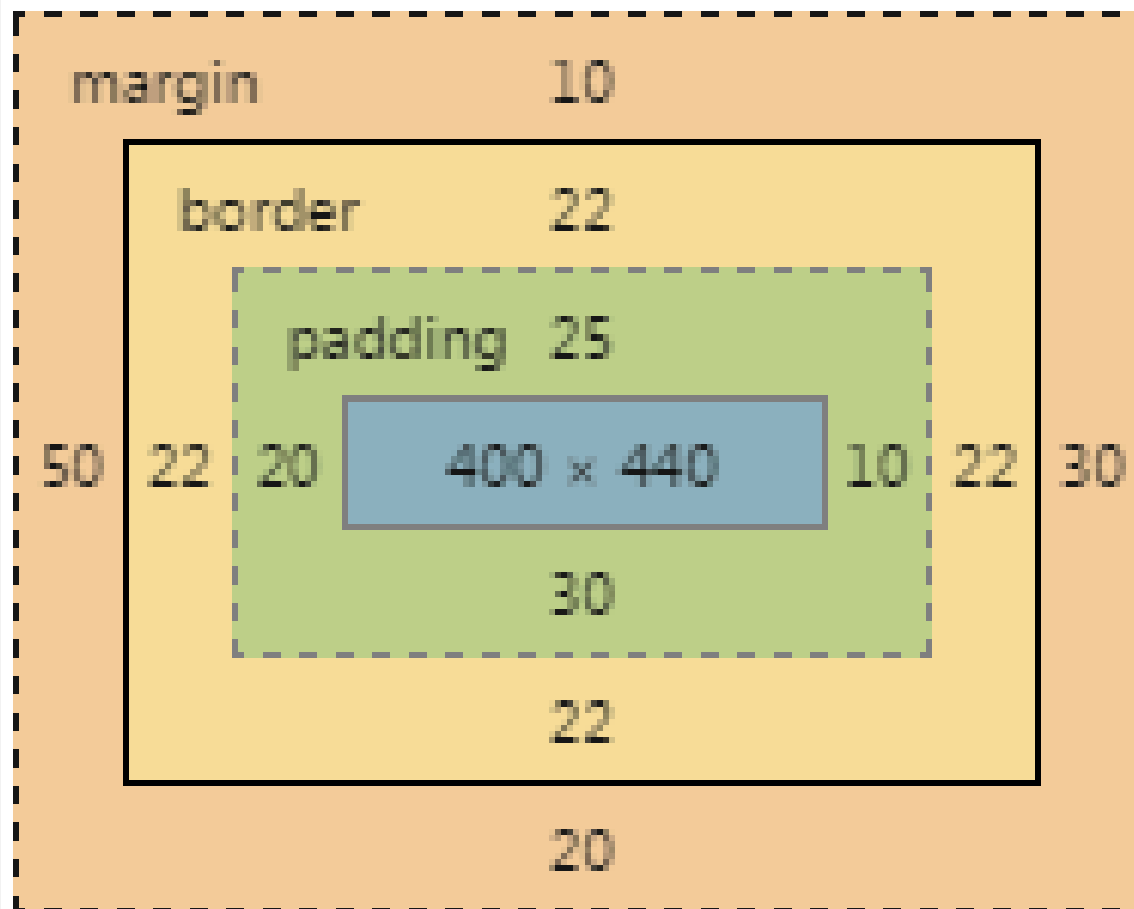
Total element height = content height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

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)

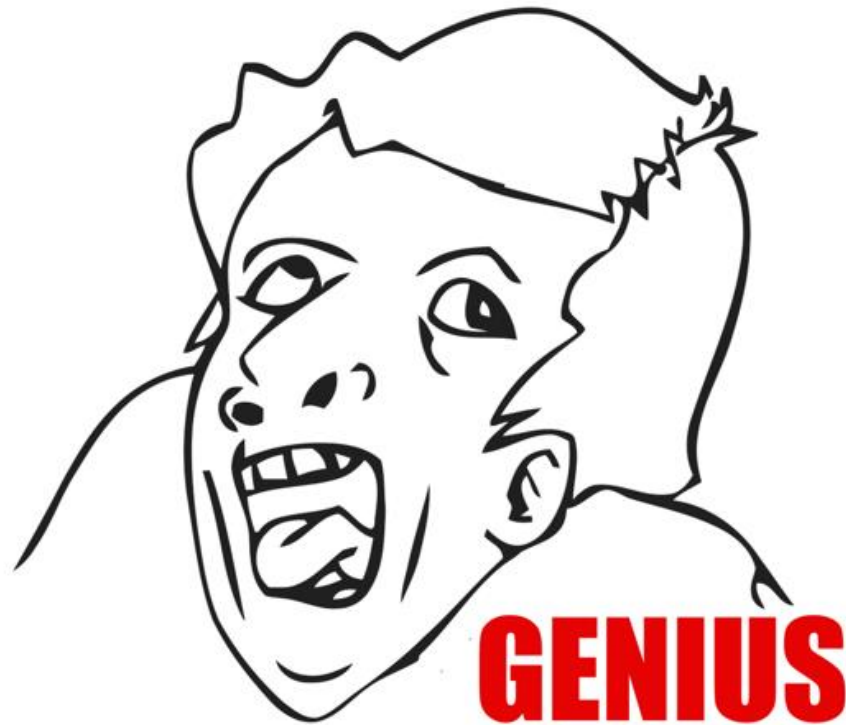
Height: 539 px (no margin), 569 px (with margin)

We Be Floatin'

Take a Facebook Break...

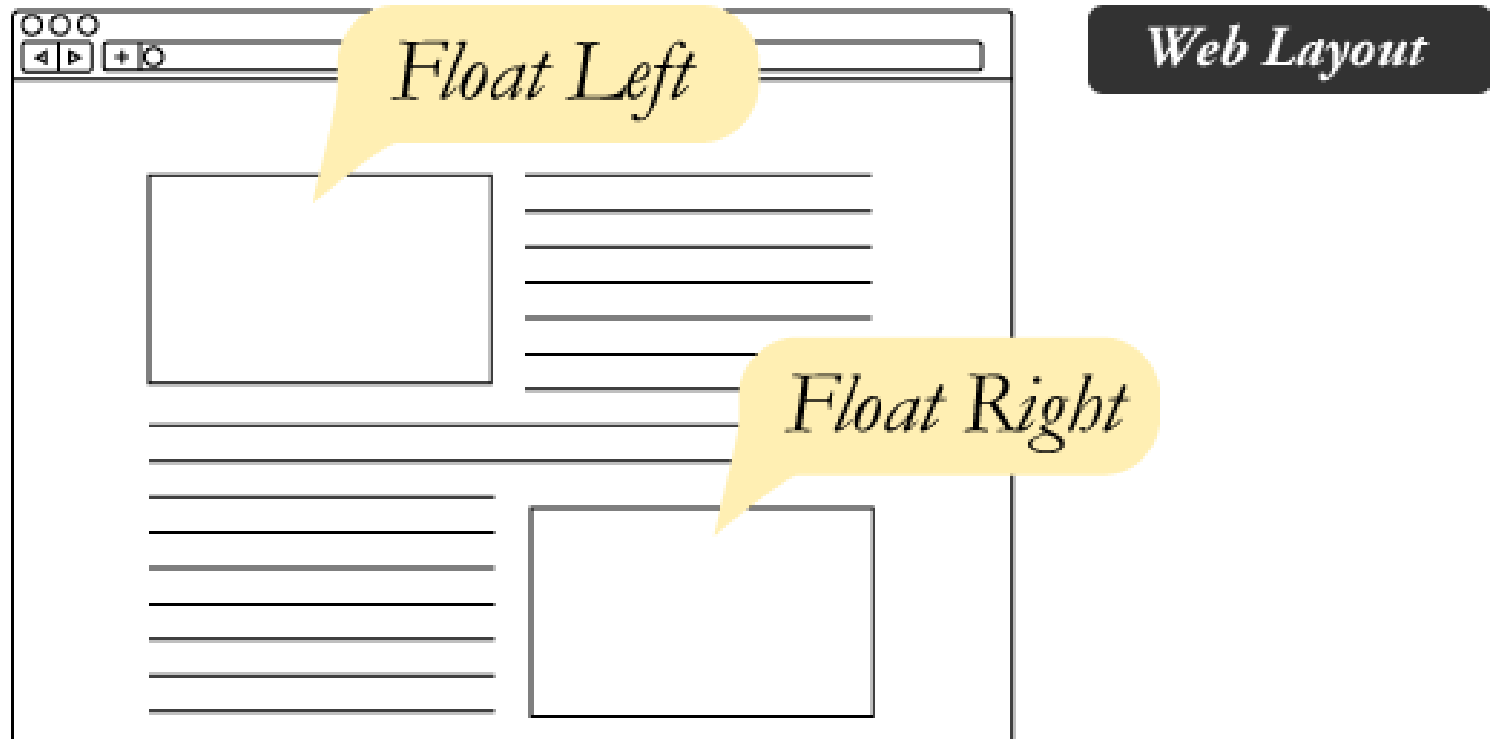
Warning!

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



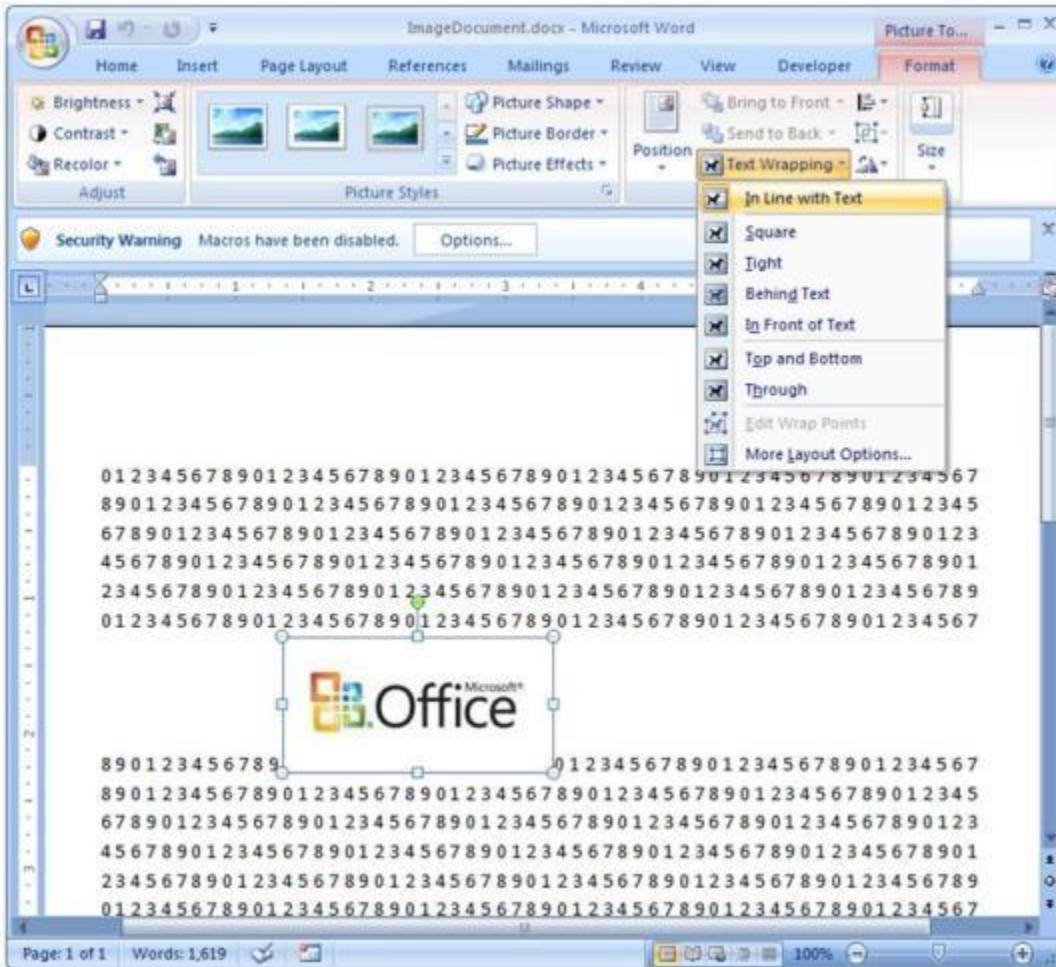
Time to channel that inner genius.

The Concept of “Flow”



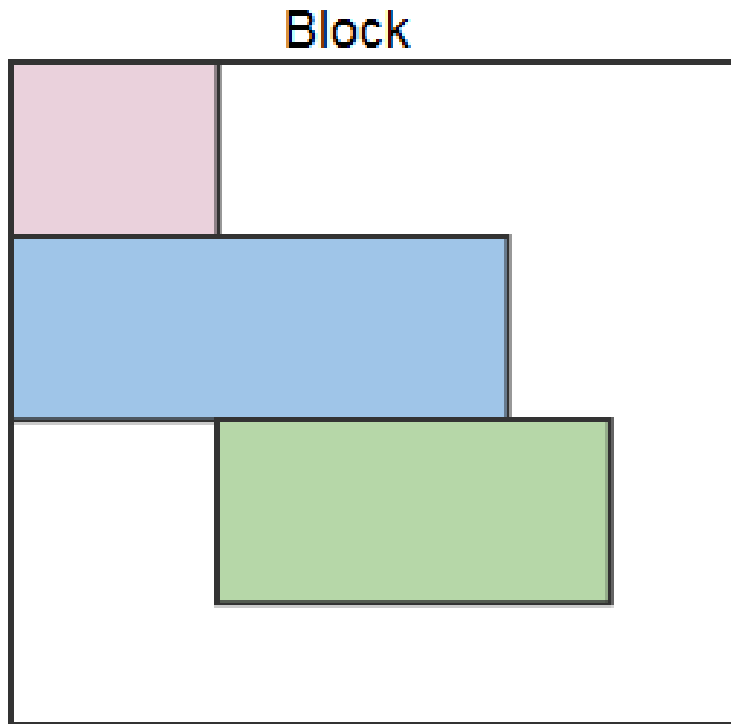
- In HTML/CSS, (by default) every element displayed is governed by a concept called “**flow**”.
- This means that HTML elements force their adjacent elements to **flow around** them.

Flow Analogy to MS Word



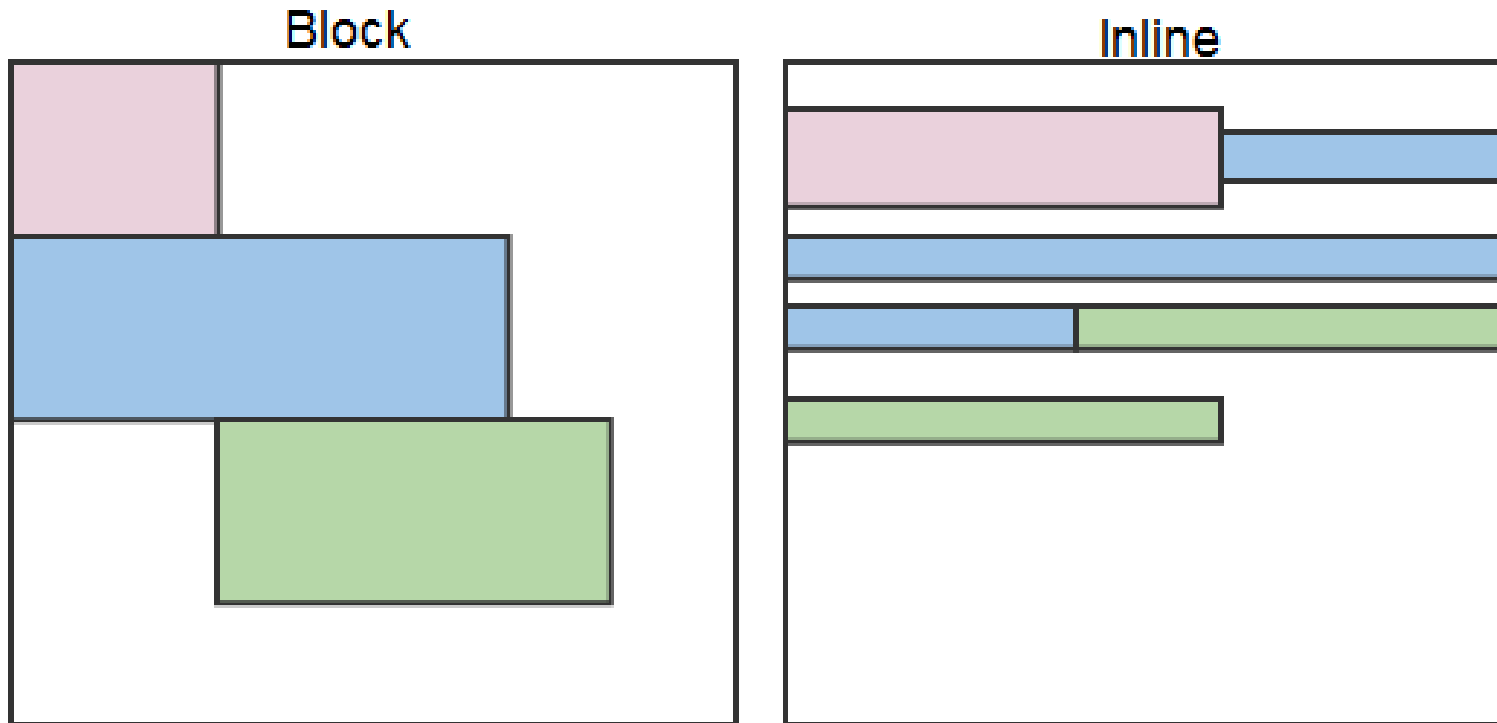
- This concept of “flow” is very similar to the **wrap-text options** you may be familiar with in Microsoft Word.
- Just as in MS Word, you can have images in-line with text, on-top of text, etc.

Block Elements



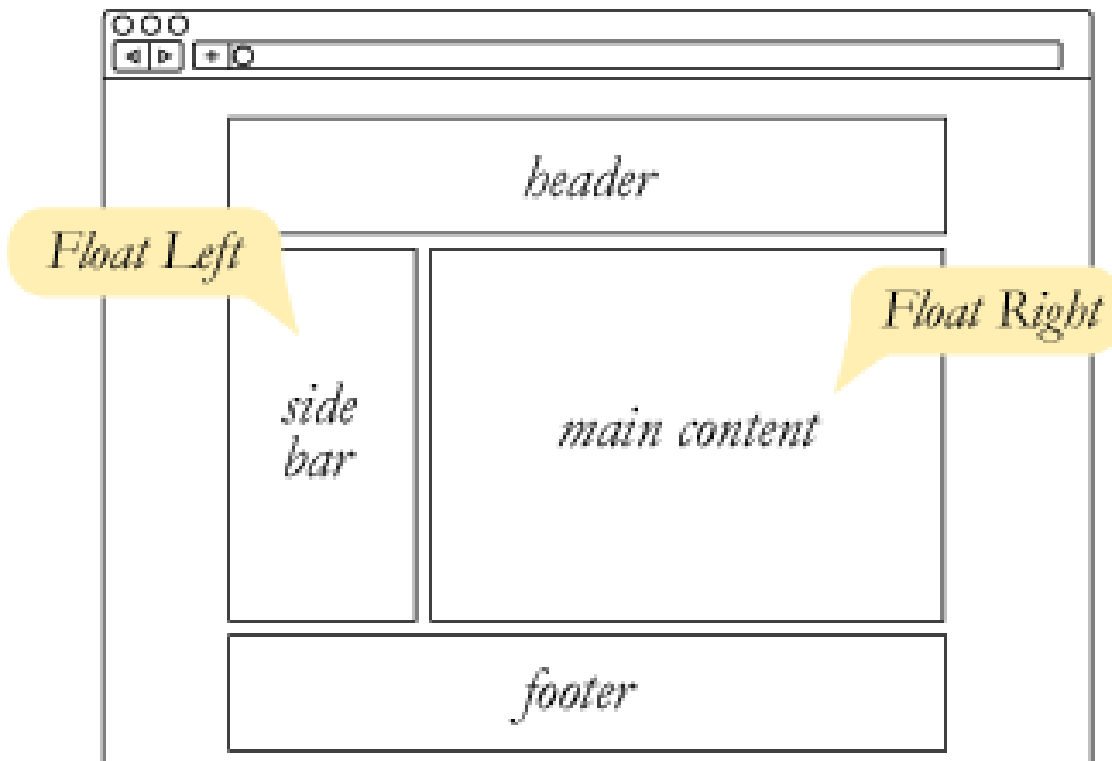
- By default, many HTML elements (paragraphs, headers, div, etc..) are treated as **block elements**.
- This means, they take up an entire line of space, unless you intervene with CSS properties.

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 CSS elements adjacently.

Floating



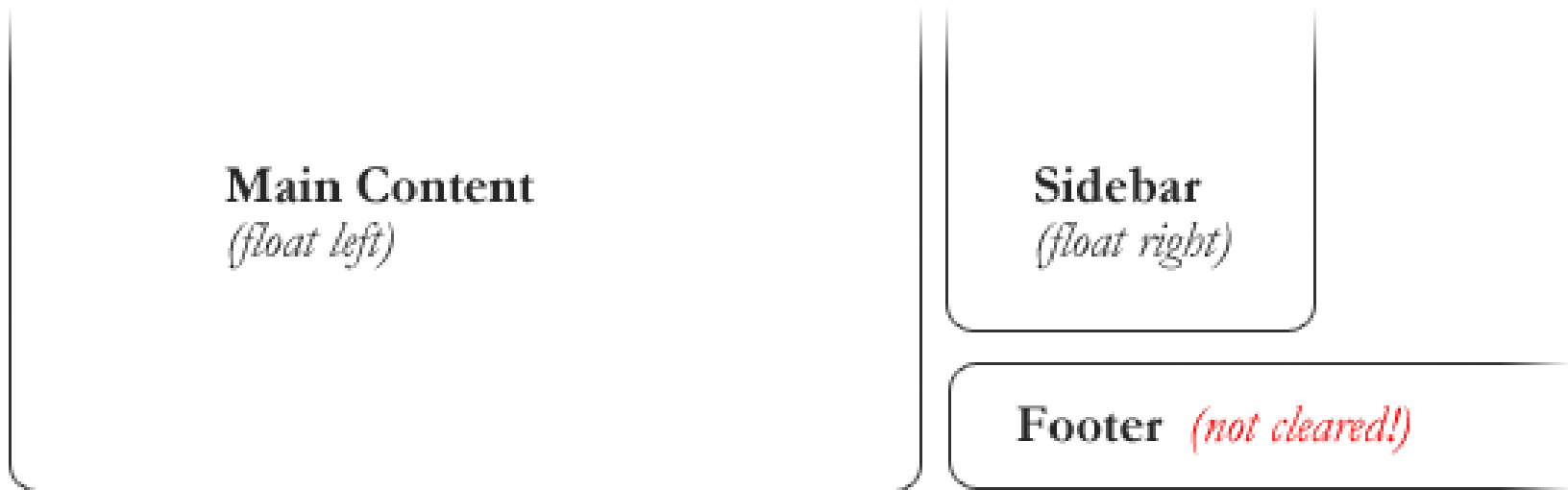
CSS

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

- In order to convert these block elements into inline elements, we utilize a CSS property called **float**.
- Floats are **necessary** for building web layouts



Clearing the Float



- Sometimes... **floats** get in the way, and elements that we don't want to be "in-line" are treated as such.

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 it by using what's called the “clearfix hack”. By using the CSS property **overflow: auto**, the first element will fill up the empty parts of the flow.

```
.clearfix {  
  overflow: auto;  
}
```

Quick Demo!



Quick Demo!

2000x200

300x400

900x400

500x100

Instructor: Demo
(2-FloatExamples)

Fantastic Guide on Floats ****

CSS-TRICKS

All About Floats

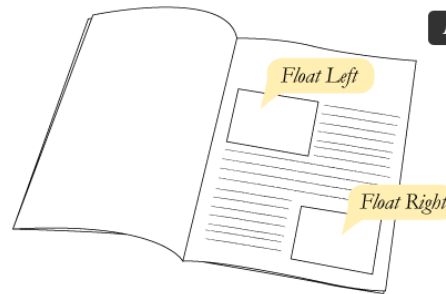
CHRIS COYIER // JULY 8, 2009

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/>

Assignment

In this activity, you will be flexing your newfound floating skills to create a conceptual layout. Eyeball the design to your best ability.

Additional instructions, sent via Slack.

> YOUR TURN!!

<header> #ccc

<section> #666

<section> #888

<section> #666

<aside> #fff

<section> #eee

<div> #6ea3da

<footer> #ccc



Good work!

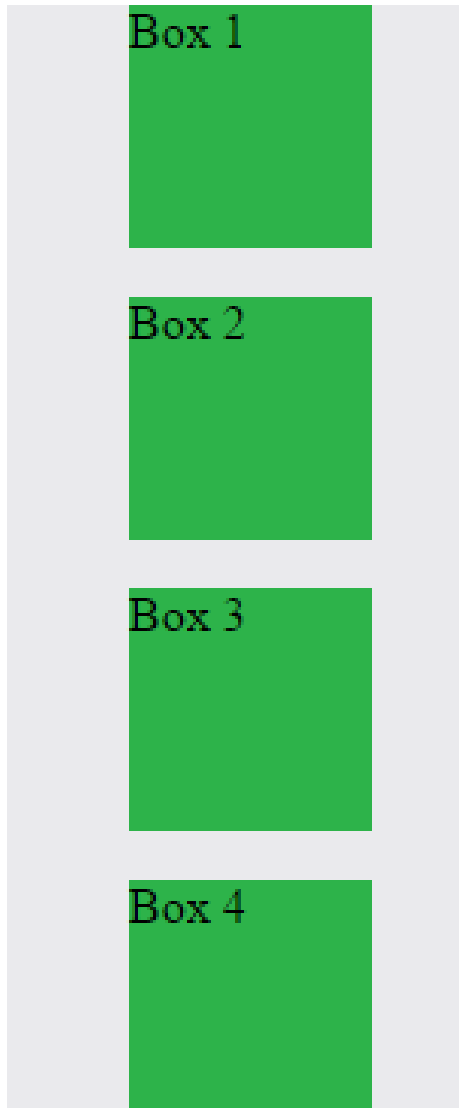


Your Brain may rest now

BREAK!

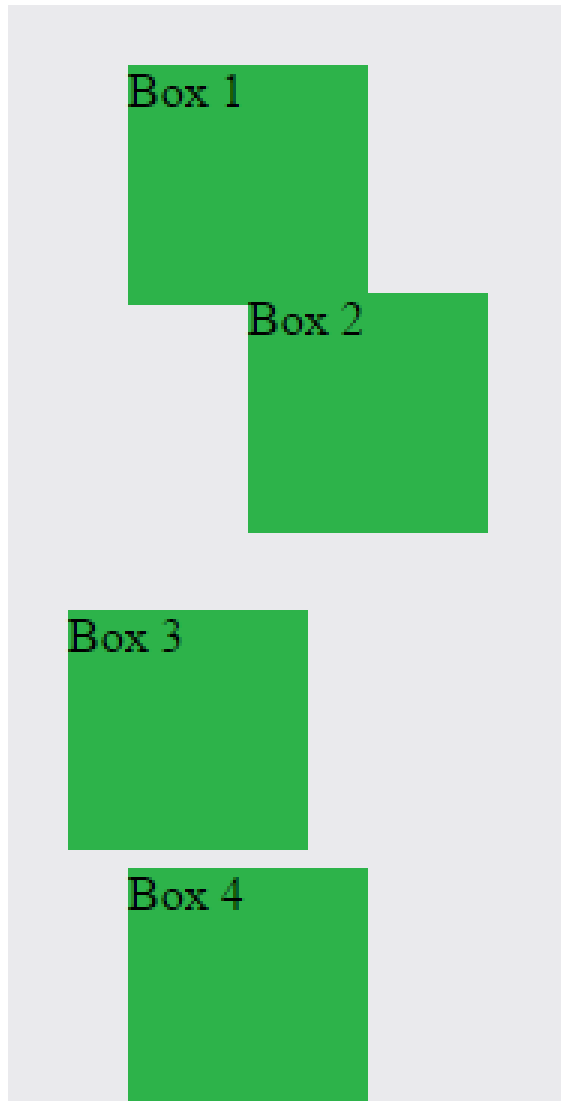
CSS Positioning

Position: Static (Default)



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

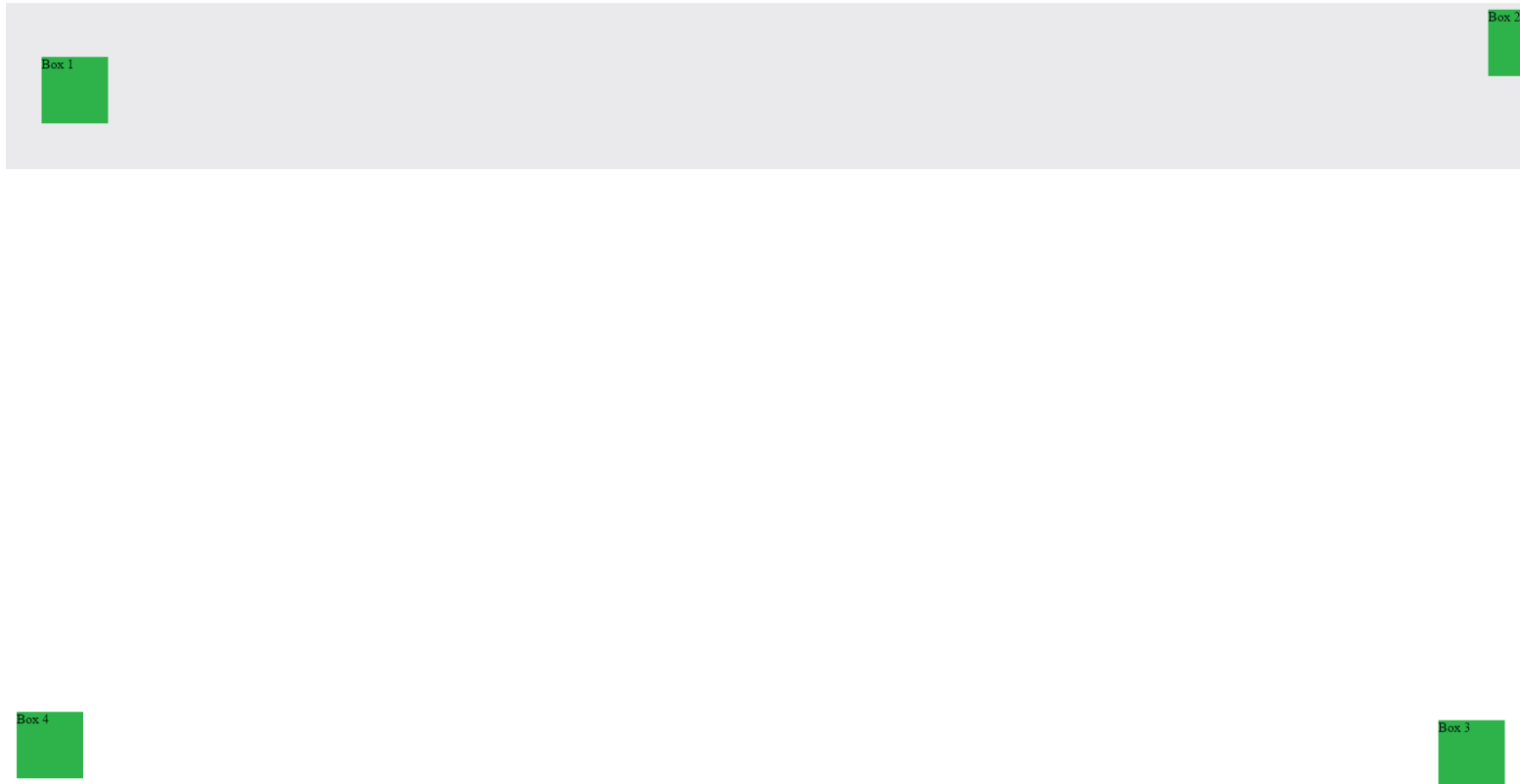
Position: Relative



- Switching the boxes to relative nudges the boxes in relation to their “original” location.

```
.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



- **Position with location relevant to first “placed” element**

Position: Fixed

Box 1



Box 2



Box 4

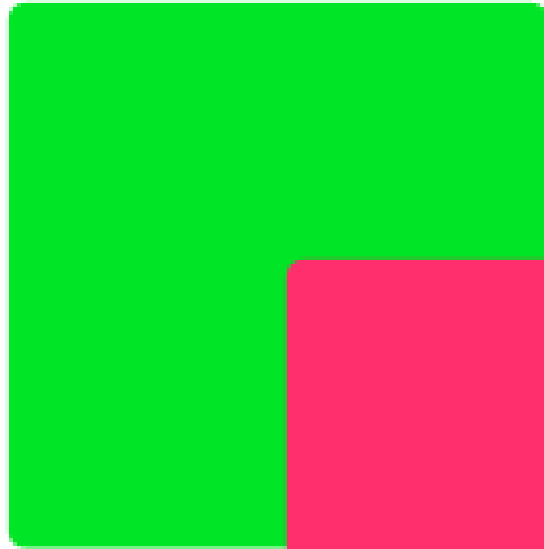


Box 3



- **Position with exact coordinates to the browser window**

Layering with Z-Index



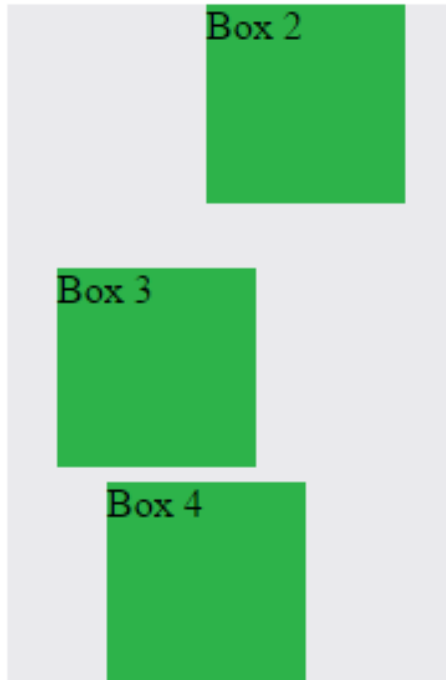
**position: absolute;
z-index: 1;**



**position: absolute;
z-index: 2;**

- **Z-Index allows you to layer elements on top of each other when placed in absolute coordinates**

Hiding Things



- Display none allows elements to be completely hidden from view.
- This becomes useful because in later sections, we will be selectively activating and de-activating.

Instructor: Demo
(5-CSS_PositionedLayout)

Great Resource

Learn to Code

HTML & CSS

Lesson 1

Building Your First Web Page

Lesson 2

Getting to Know HTML

Lesson 3

Getting to Know CSS **New**

Lesson 4

Opening the Box Model

Lesson 5

Positioning Content **New**

Lesson 6

Lesson 5

Positioning Content

One of the best things about CSS is that it gives us the ability to position content and elements on a page in nearly any imaginable way, bringing structure to our designs and helping make content more digestible.

There are a few different types of positioning within CSS, and each has its own application. In this chapter we're going to take a look at a few different use cases—creating reusable layouts and uniquely positioning one-off elements—and describe a few ways to go about each.

Positioning with Floats

One way to position elements on a page is with the `float` property. The `float` property is pretty versatile and can be used in a number of different ways.

Essentially, the `float` property allows us to take an element, remove it from the normal flow of a page, and [position it](#) to the left or right of its parent element. All other

In this Lesson

5

CSS

- [Positioning with Floats](#)
- [Positioning with Inline-Block](#)
- [Creating Reusable Layouts](#)
- [Uniquely Positioning Elements](#)

SHARE



- **Another great read for front-end developers (and game designers):**
<http://learn.shayhowe.com/html-css/positioning-content/>

Assignment

In this activity, you will be flexing your newfound positioning skills to create another conceptual layout. Eyeball the design to your best ability.

Additional instructions, sent via Slack.

> YOUR TURN!!

`<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
placemat 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.



Advice



Re-do this at home.
*It's a good one to firm up
your HTML/CSS skills.*

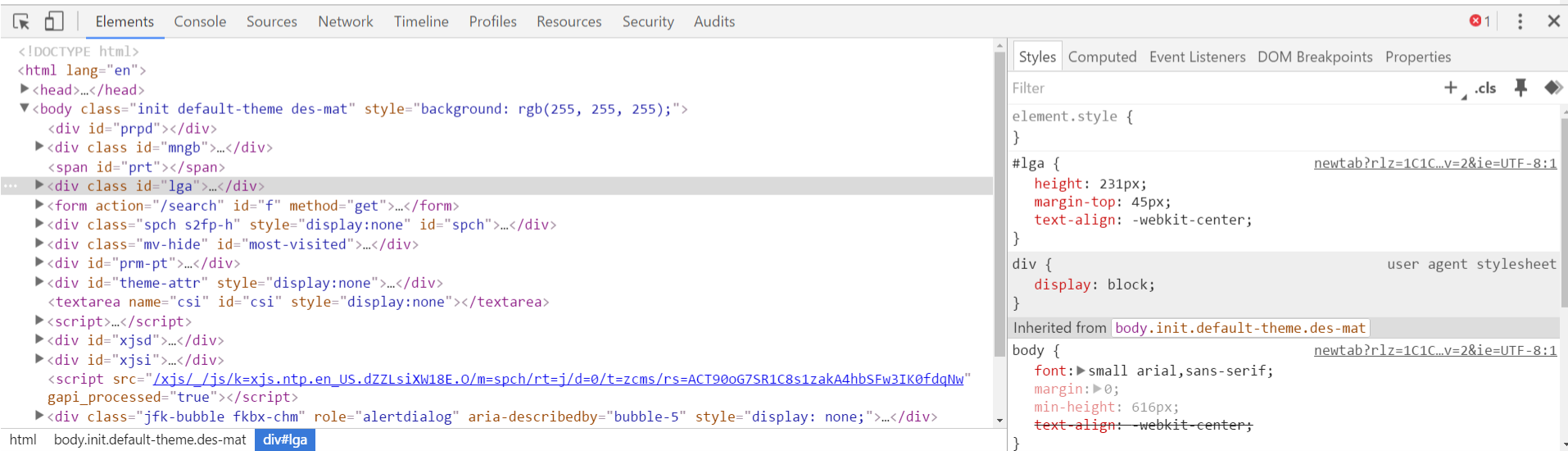
REMEMBER:
*The best way to get good
is to PRACTICE*

Chrome Inspector

Chrome Inspector is Your Friend



Search Google or type URL



The screenshot displays the Chrome DevTools interface. The top bar shows tabs for Elements, Console, Sources, Network, Timeline, Profiles, Resources, Security, and Audits. The Elements panel on the left shows the DOM tree with the following structure:

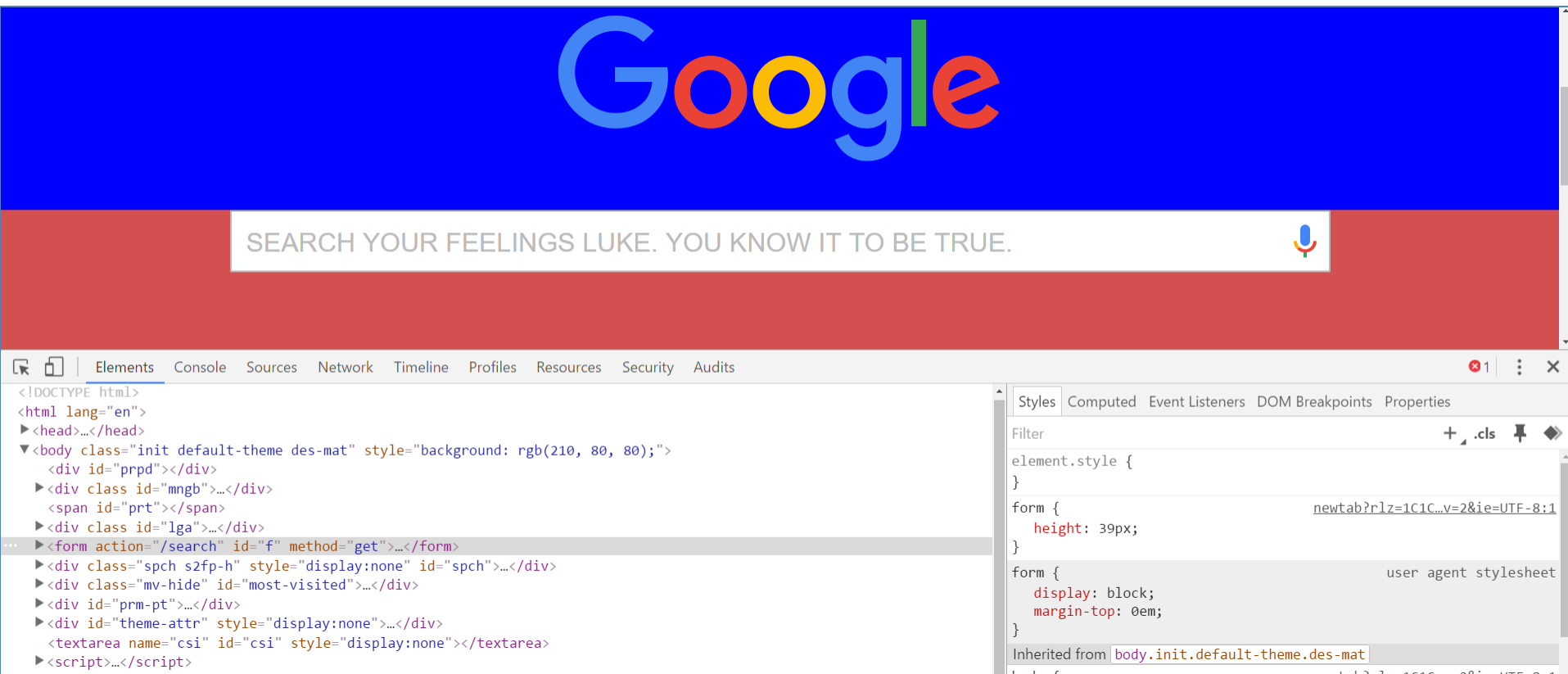
```
<!DOCTYPE html>
<html lang="en">
<head>...</head>
<body class="init default-theme des-mat" style="background: rgb(255, 255, 255);">
  <div id="prpd"></div>
  <div class id="mngb">...</div>
  <span id="prt"></span>
  <div class id="lga">...</div>
  <form action="/search" id="f" method="get">...</form>
  <div class="spch s2fp-h" style="display:none" id="spch">...</div>
  <div class="mv-hide" id="most-visited">...</div>
  <div id="prm-pt">...</div>
  <div id="theme-attr" style="display:none">...</div>
  <textarea name="csi" id="csi" style="display:none"></textarea>
  <script>...</script>
  <div id="xjsd">...</div>
  <div id="xjsi">...</div>
  <script src="/xjs/_/js/k=xjs.ntp.en_US.dZ7LsiXW18E.O/m=spch/rt=j/d=0/t=zcms/rs=ACT90oG7SR1C8s1zakA4hbSFw3IK0fdqNw"
    gapi_processed="true"></script>
  <div class="jfk-bubble fkbx-chm" role="alertdialog" aria-describedby="bubble-5" style="display: none;">...</div>
```

The Styles panel on the right shows the CSS rules for the selected element:

```
element.style {
}
#lga {
  height: 231px;
  margin-top: 45px;
  text-align: -webkit-center;
}
div {
  display: block;
}
Inherited from body.init.default-theme.des-mat
body {
  font: small arial,sans-serif;
  margin: 0;
  min-height: 616px;
  text-align: -webkit-center;
```

- 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



- **Using Chrome Inspector you can readily edit any webpage** (your own or otherwise) to see the immediate impact of HTML or CSS changes

Next Class!

***We'll be coming back to this next
class 😊***



Recap + Reassurances



You
got
this

Questions?
