

Day 3

# *Heroes of CSS*

---

6/25/2016



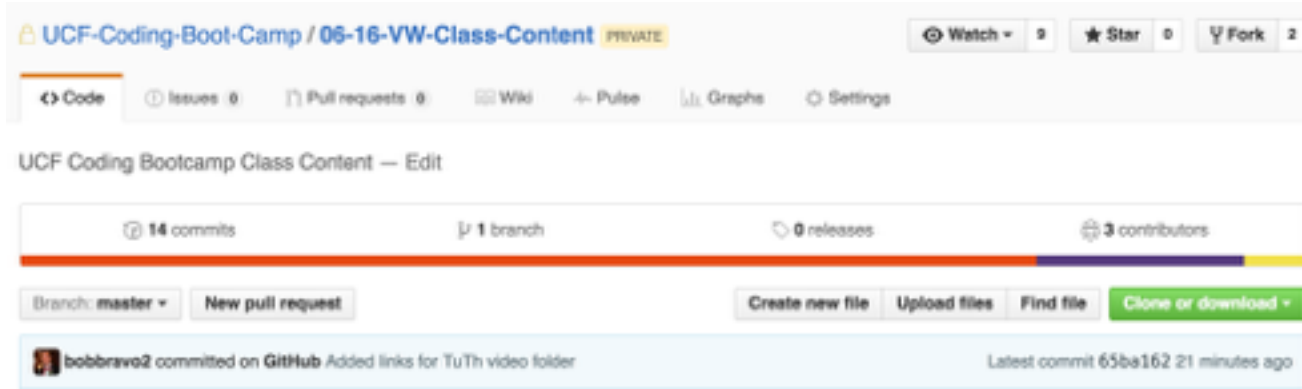
UCF CODING BOOT CAMP

# *Admin Work*

---

# Homework Assignment

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



- Homework Assignment #1 is Due:

**MW Class – Wednesday (6/29/2016)**

**TTH Class – Thursday (6/30/2016)**

# 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* Activities:**  
<https://github.com/UCF-Coding-Boot-Camp/06-16-VW-Class-Content/tree/master/0621-tth-class-content/01-html-css-three-days/1.2/Activities>
- **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.***

***Today is going to be a bit tough.***





***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***

---

***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.



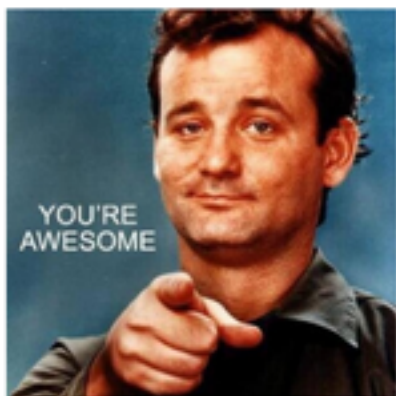
# Basic HTML Page - Result

## 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

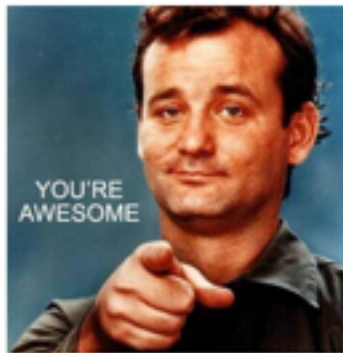
- [Google](#)
- [Facebook](#)
- [Twitter](#)

## 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...*

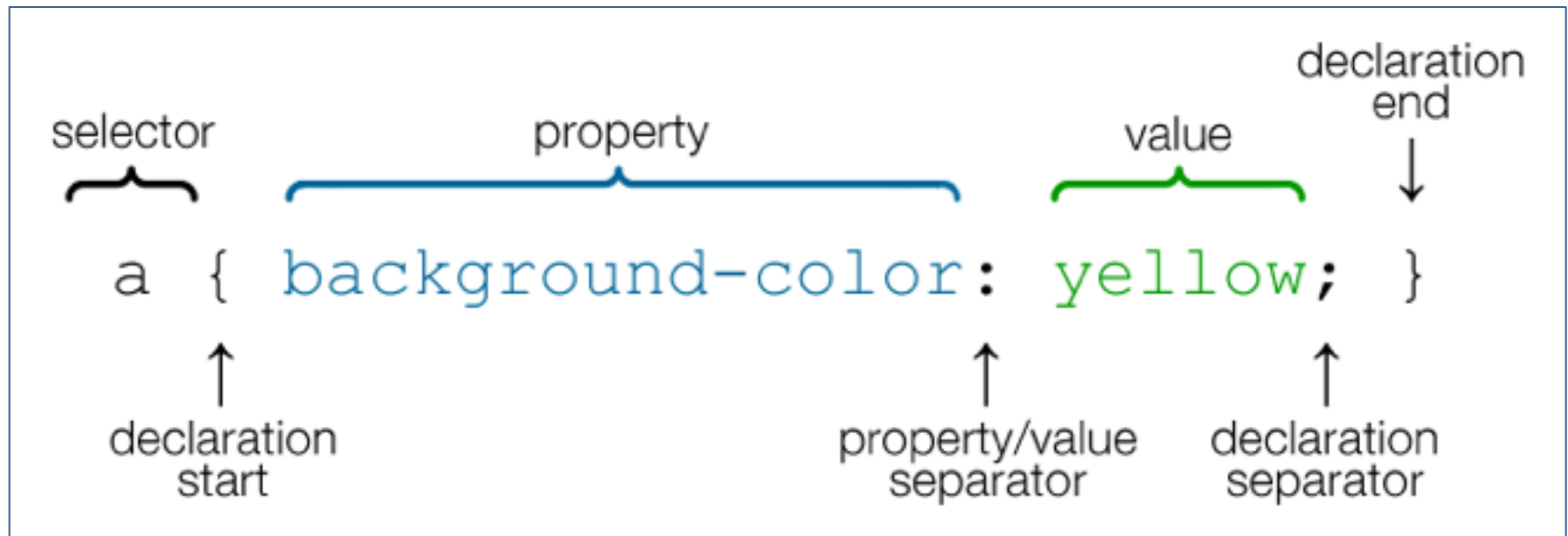
*Elements?*

*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 .



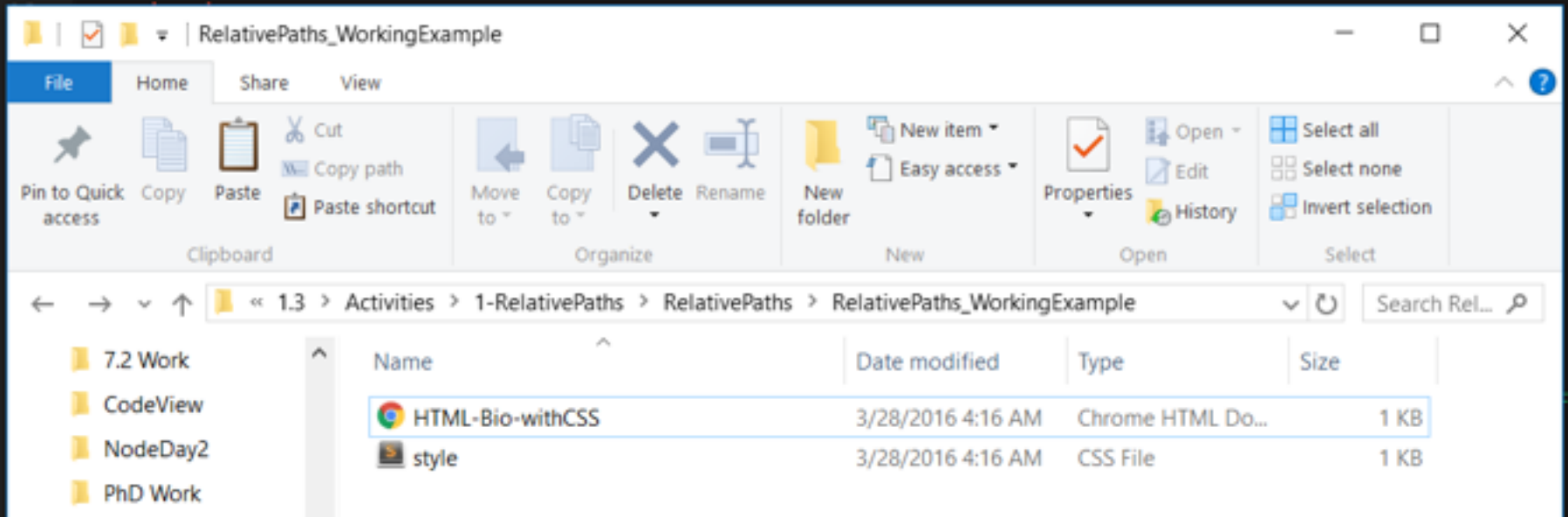
***Questions so far?***

# *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

```
<!-- 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">
```



# 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">
```



# 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">
```





# 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.**



# 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.**



# 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.



# 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.





# 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

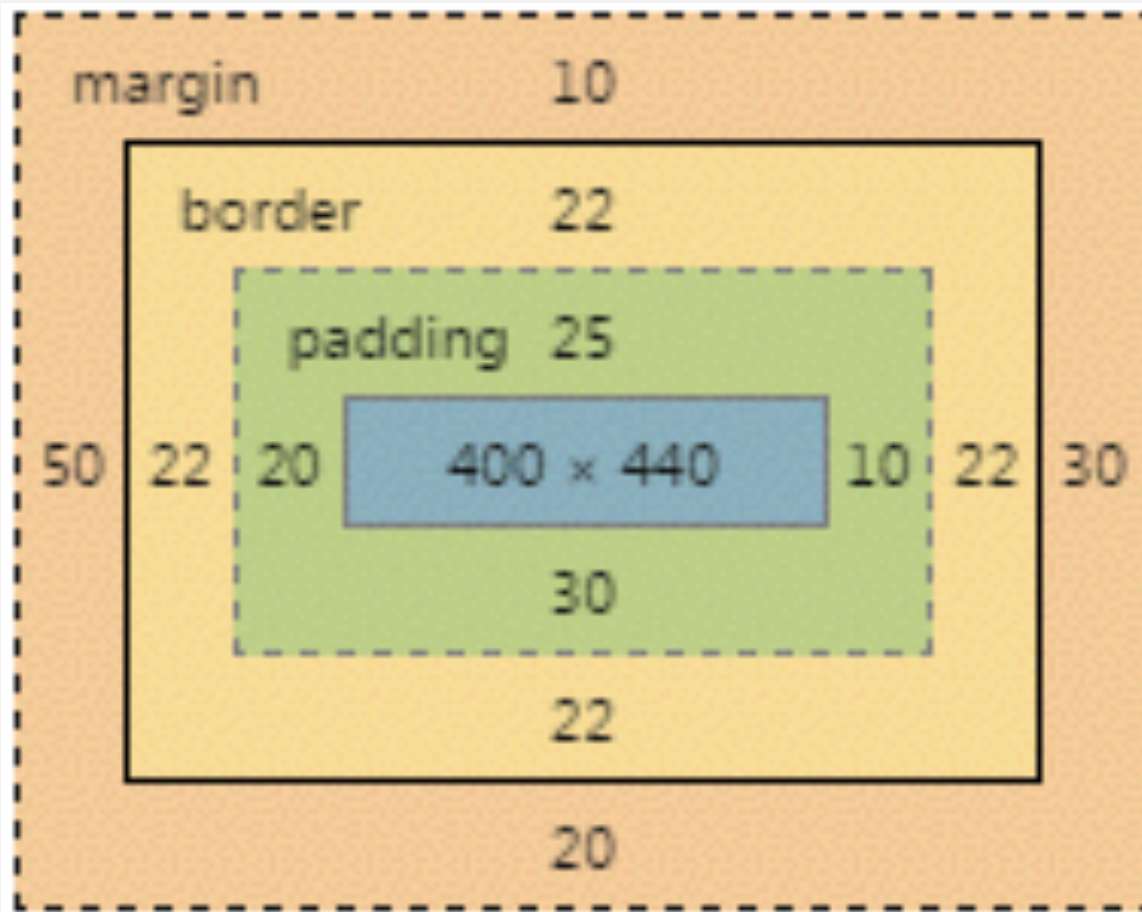
**Total element height** = content height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

## 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'***

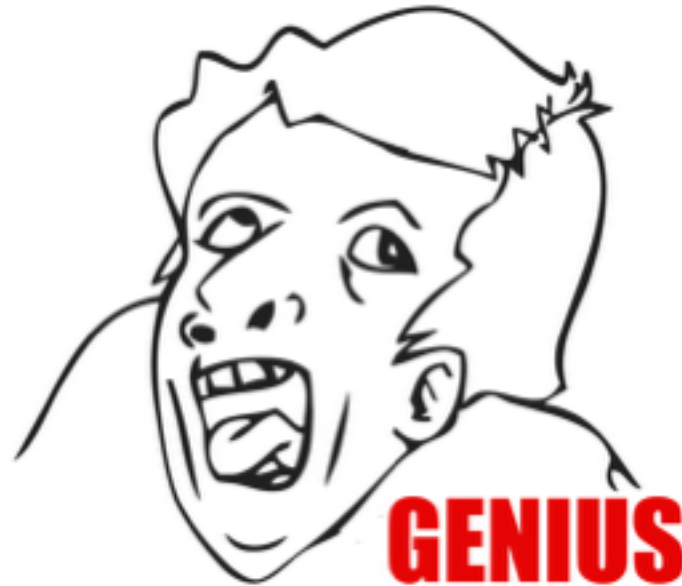
---

# Take a Facebook Break...

---

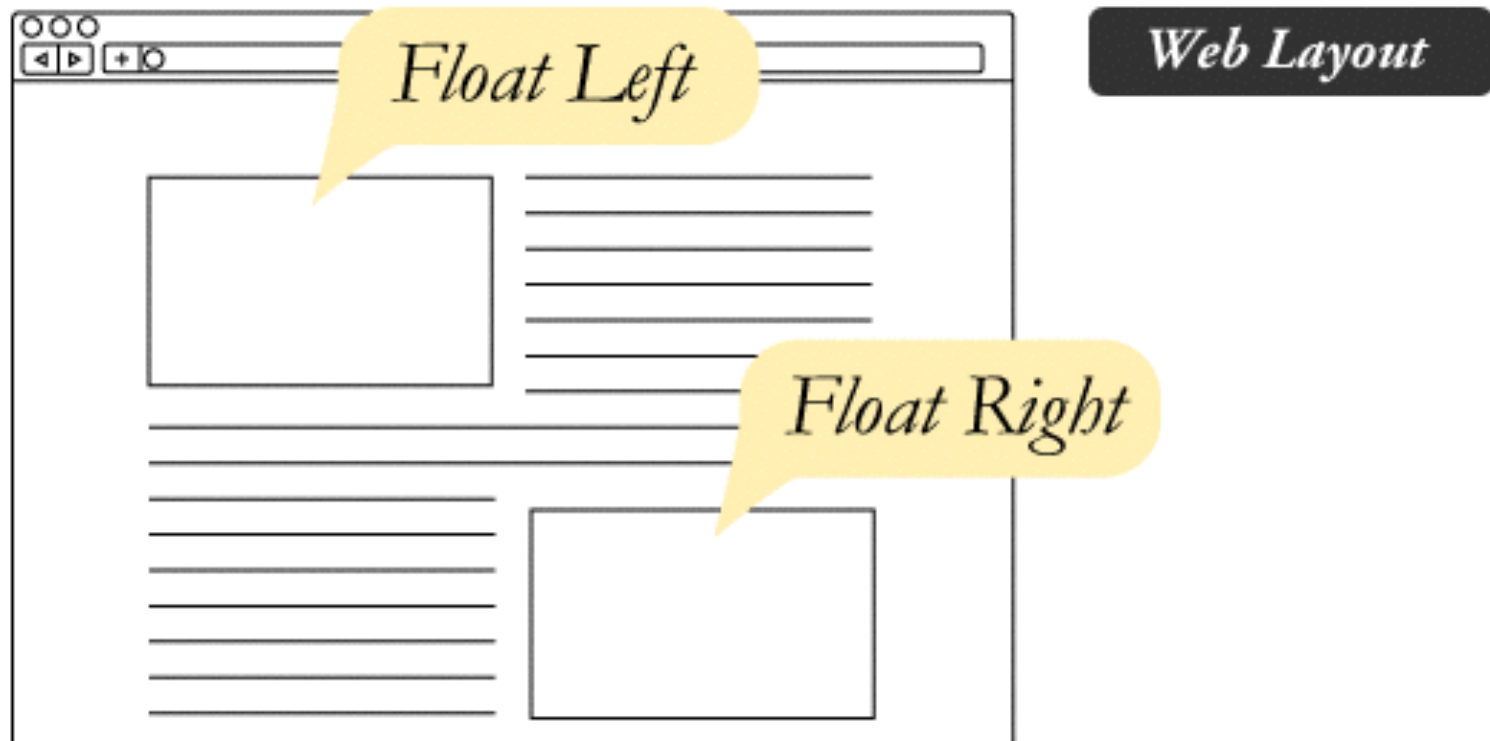
## Warning!

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



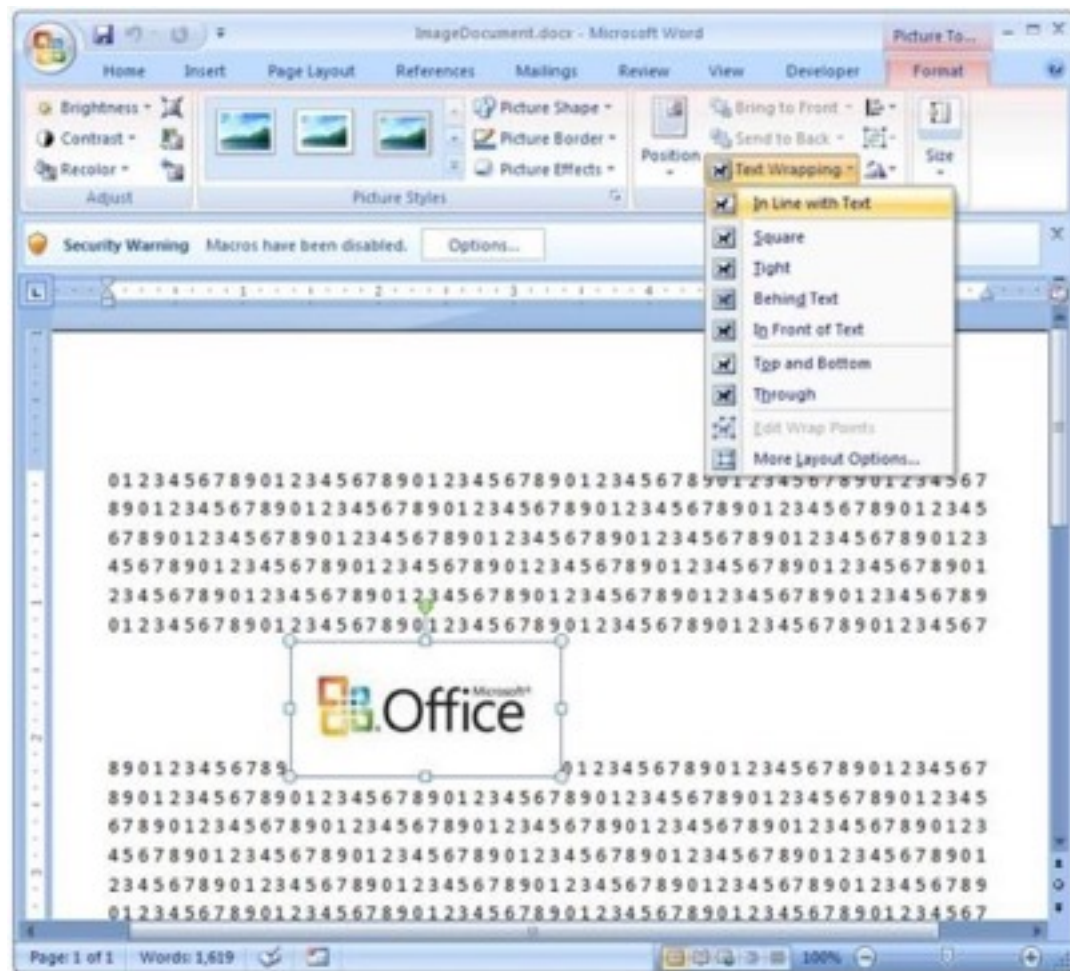
Time to channel that inner genius.

# 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.

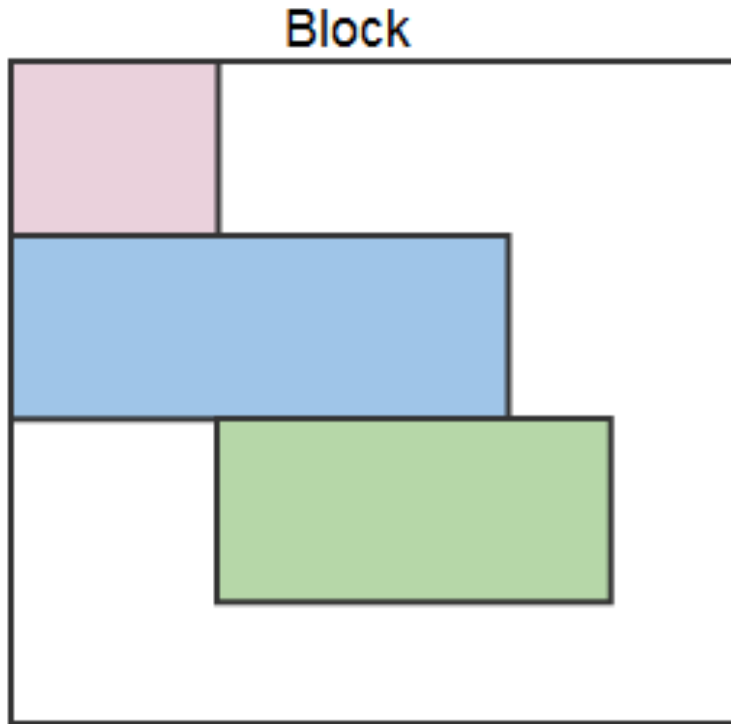
# Flow Analogy to MS Word



- 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.

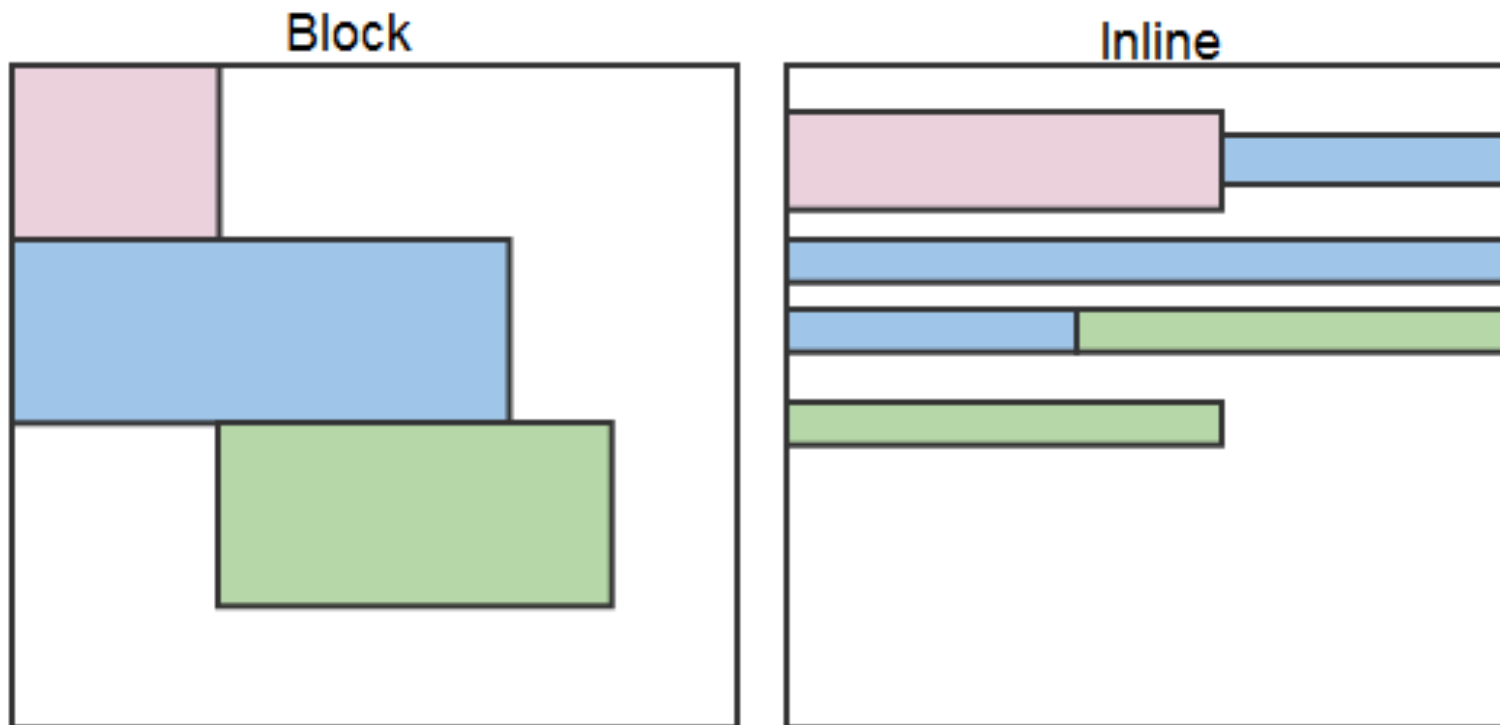


# Block Elements



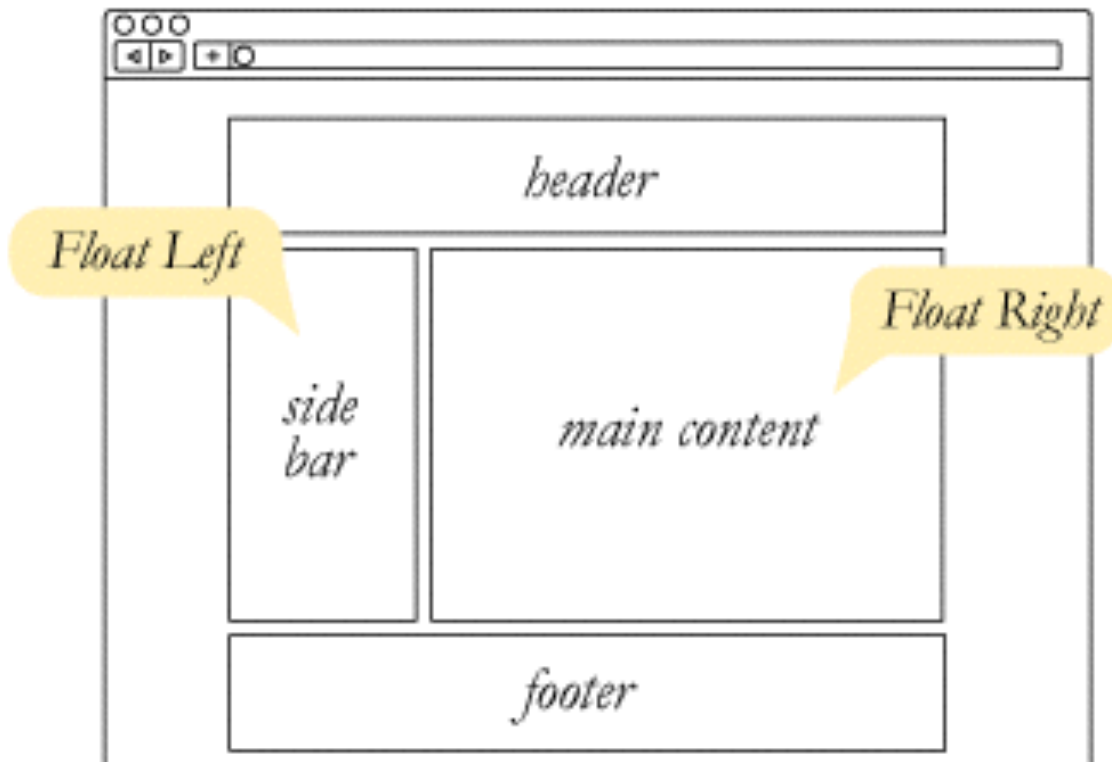
- By default, many HTML elements (paragraphs, headers, div, etc..) are treated as **block elements**.
- This means, they take up an entire line of space, unless you intervene with CSS properties.

# 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



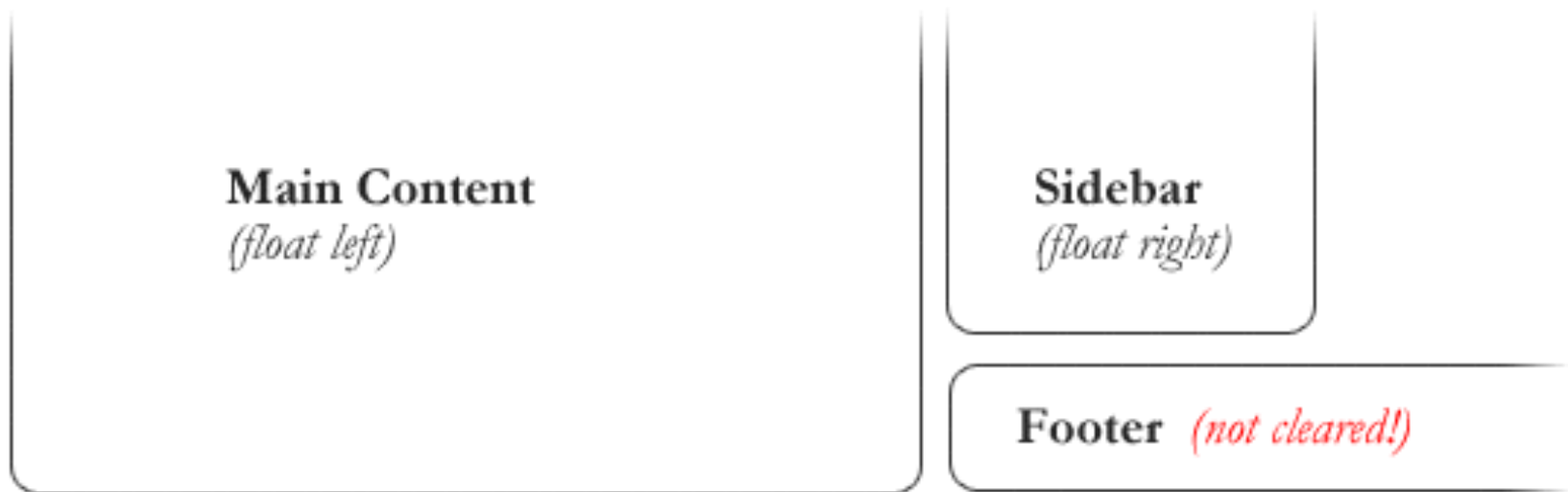
## CSS

```
#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

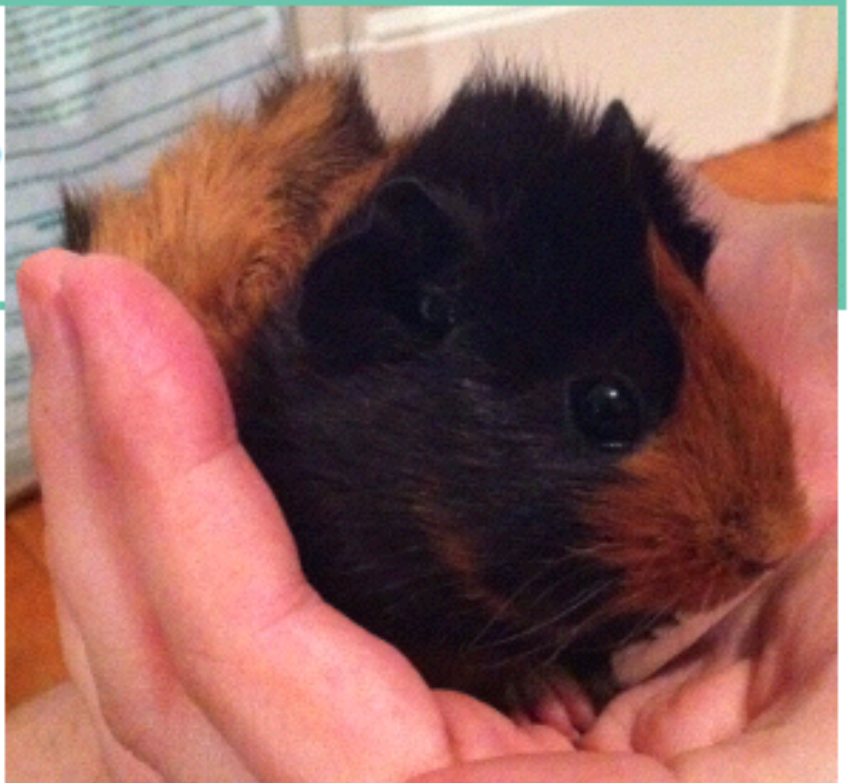


- Sometimes... **floats** get in the way, and elements that we don't want to be "in-line" are treated as such.

# 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.



The responsive development stand.

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**

***000:00:0:40:00 BREAK!***

---