Day 4

# *Going Live*

The Coding Bootcamp | April 4, 2016

# Check-Up Session

## *How's it going?*

*After 1 week of Bootcamp, how are you holding up?*

*What feedback do you have so far?*

# Instructor Feedback

Seriously, mind-blown.

# Instructor Feedback

**Things I’ve noticed people doing *incredibly* well:**

- All of you are handling an enormous volume of information.

- All of you are asking the right questions.

- You notice the right details.

- You all help each other out.

- And, most importantly, you are **figuring out things on your own.**

# *A Few Admin Things…*

# Instructor Feedback

- **Remember, Homework #1 is due on Wednesday / Thursday.**

- **Homework Link:**
- <<<<<INSERT LINK HERE>>>>>

- **Remember to submit Homework via GitHub (and Heroku):**
- <<<<<INSERT LINK HERE>>>>>

**And seriously!**

**Submit whatever you have! Don't get a 0.**

(Even if you don't like what you've made.)

# Office Hours + Additional Help

**Also, remember….**

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

•**Review In Class Material (Exercises and Slides):**
•<<<<INSERT LINK>>>

•**Re-Watch Class Videos:**
<<<INSERT LINK>>>>

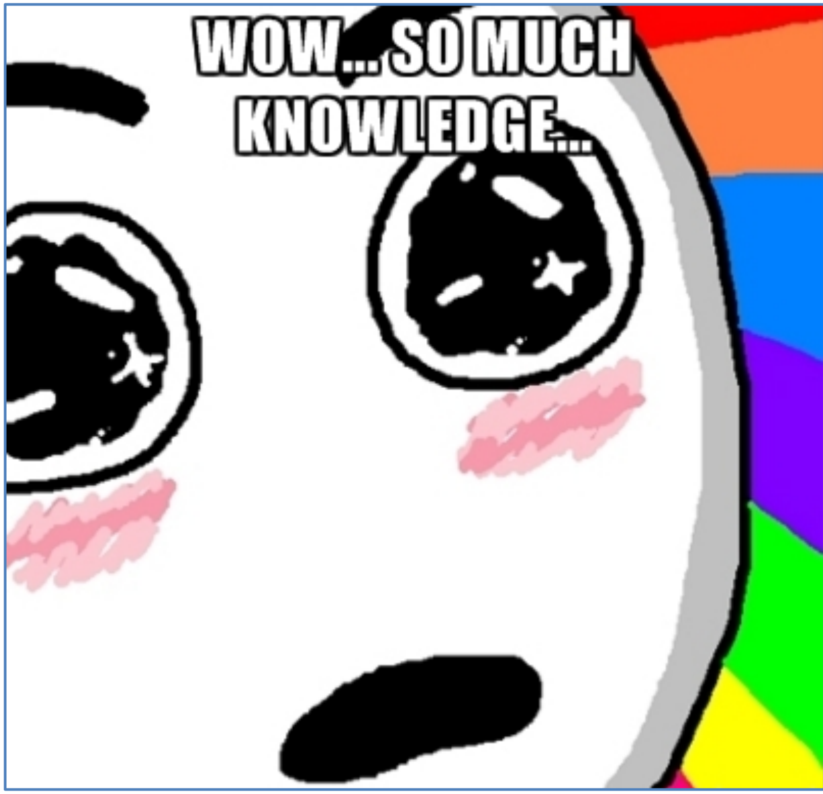
# *Recapping*

# Recap

**In just one whirlwind week we've covered:**

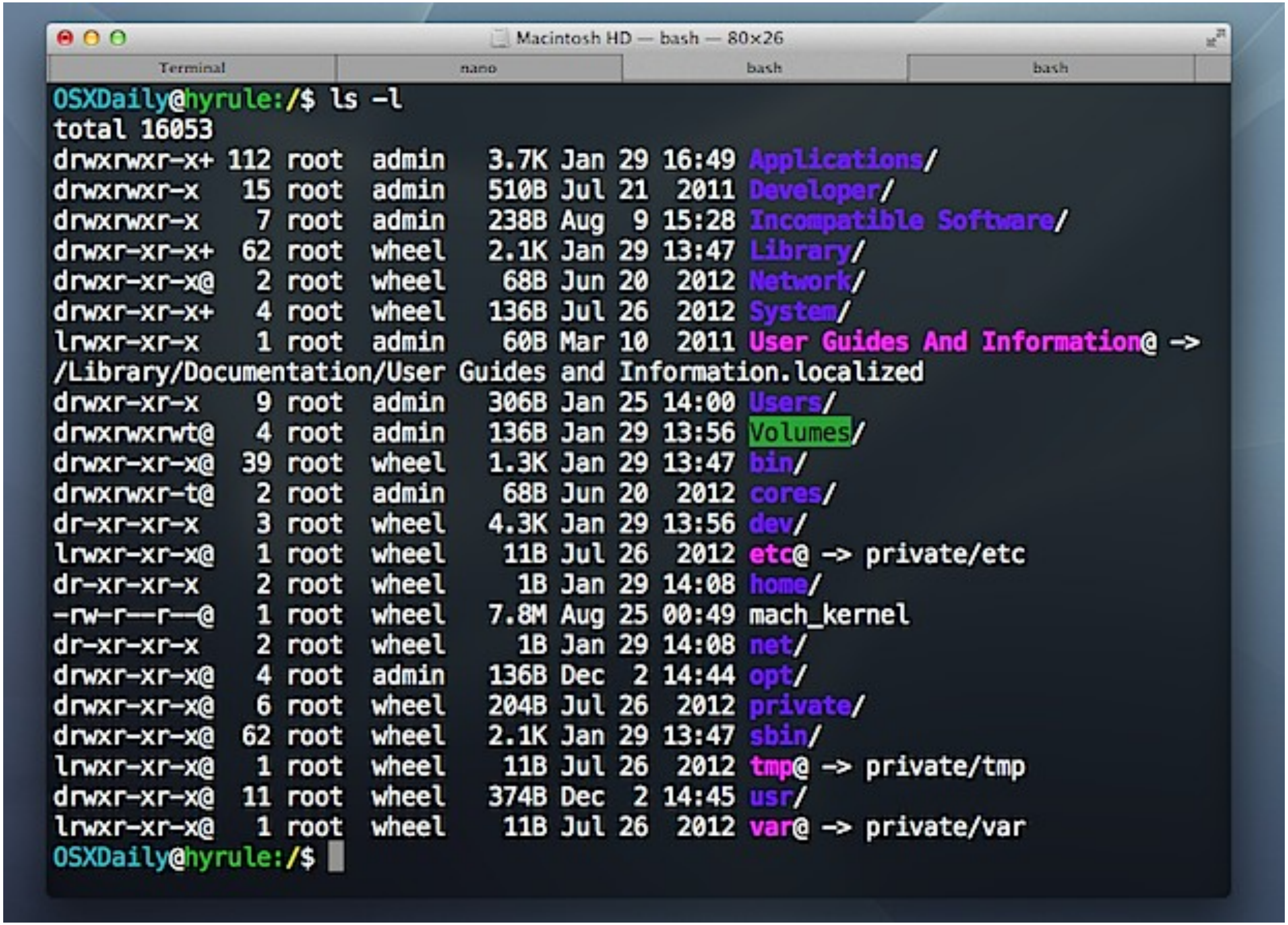
- Full-Stack Development Conceptually
- Terminal / Git Bash
- HTML Syntax
- Git Concepts and Commands
- CSS Purpose, Syntax, and Styles
- Floating
- Positioning
- Box Model
- Google Dev Tools

- **<u>How to learn on Your own!!</u>**



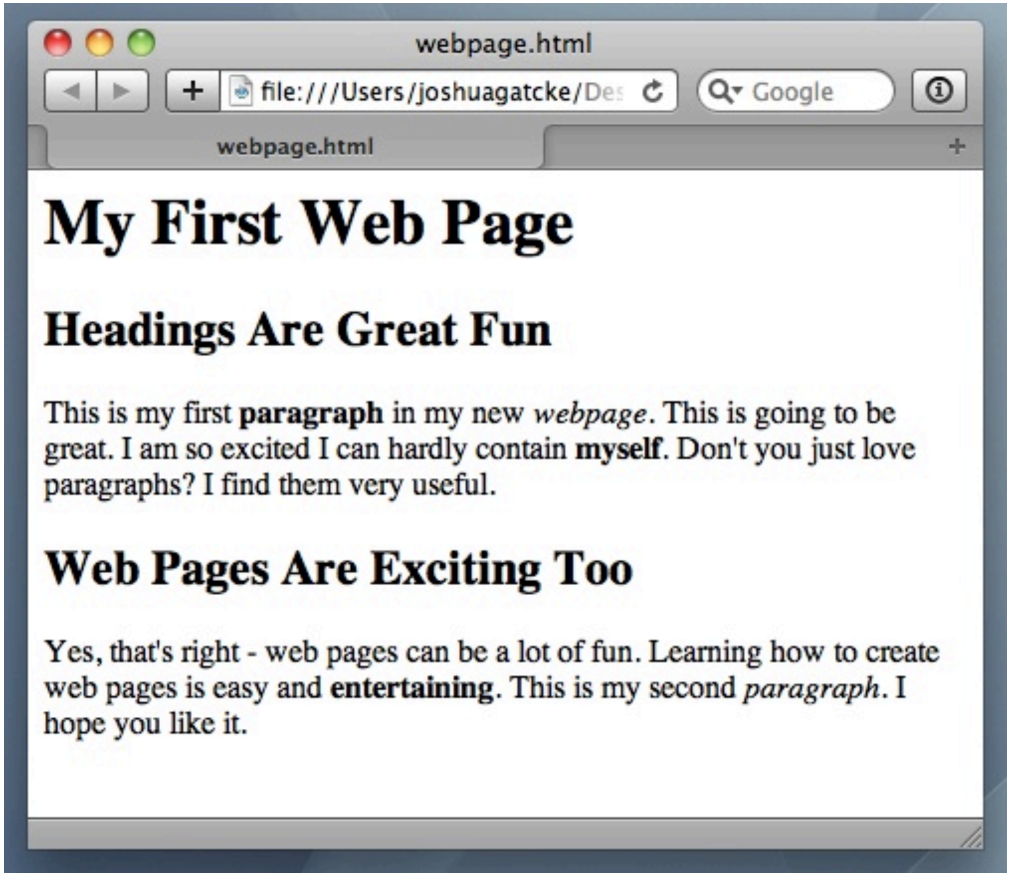
# Full-Stack Development?

# > Intro to Console



```
OSXDaily@hyrule:/$ ls -l
total 16053
drwxrwxr-x+ 112 root  admin   3.7K Jan 29 16:49 Applications/
drwxrwxr-x   15 root  admin   510B Jul 21  2011 Developer/
drwxrwxr-x    7 root  admin   238B Aug  9 15:28 Incompatible Software/
drwxr-xr-x+  62 root  wheel   2.1K Jan 29 13:47 Library/
drwxr-xr-x@   2 root  wheel    68B Jun 20  2012 Network/
drwxr-xr-x+   4 root  wheel   136B Jul 26  2012 System/
lrwxr-xr-x    1 root  admin    60B Mar 10  2011 User Guides And Information@ ->
/Library/Documentation/User Guides and Information.localized
drwxr-xr-x    9 root  admin   306B Jan 25 14:00 Users/
drwxrwxrwt@   4 root  admin   136B Jan 29 13:56 Volumes/
drwxr-xr-x@  39 root  wheel   1.3K Jan 29 13:47 bin/
drwxrwxr-t@   2 root  admin    68B Jun 20  2012 cores/
dr-xr-xr-x    3 root  wheel   4.3K Jan 29 13:56 dev/
lrwxr-xr-x@   1 root  wheel    11B Jul 26  2012 etc@ -> private/etc
dr-xr-xr-x    2 root  wheel     1B Jan 29 14:08 home/
-rw-r--r--@   1 root  wheel   7.8M Aug 25 00:49 mach_kernel
dr-xr-xr-x    2 root  wheel     1B Jan 29 14:08 net/
drwxr-xr-x@   4 root  admin   136B Dec  2 14:44 opt/
drwxr-xr-x@   6 root  wheel   204B Jul 26  2012 private/
drwxr-xr-x@  62 root  wheel   2.1K Jan 29 13:47 sbin/
lrwxr-xr-x@   1 root  wheel    11B Jul 26  2012 tmp@ -> private/tmp
drwxr-xr-x@  11 root  wheel   374B Dec  2 14:45 usr/
lrwxr-xr-x@   1 root  wheel    11B Jul 26  2012 var@ -> private/var
OSXDaily@hyrule:/$
```

# <title> Intro to HTML </title>



# My First Web Page

## Headings Are Great Fun

This is my first **paragraph** in my new *webpage*. This is going to be great. I am so excited I can hardly contain **myself**. Don't you just love paragraphs? I find them very useful.

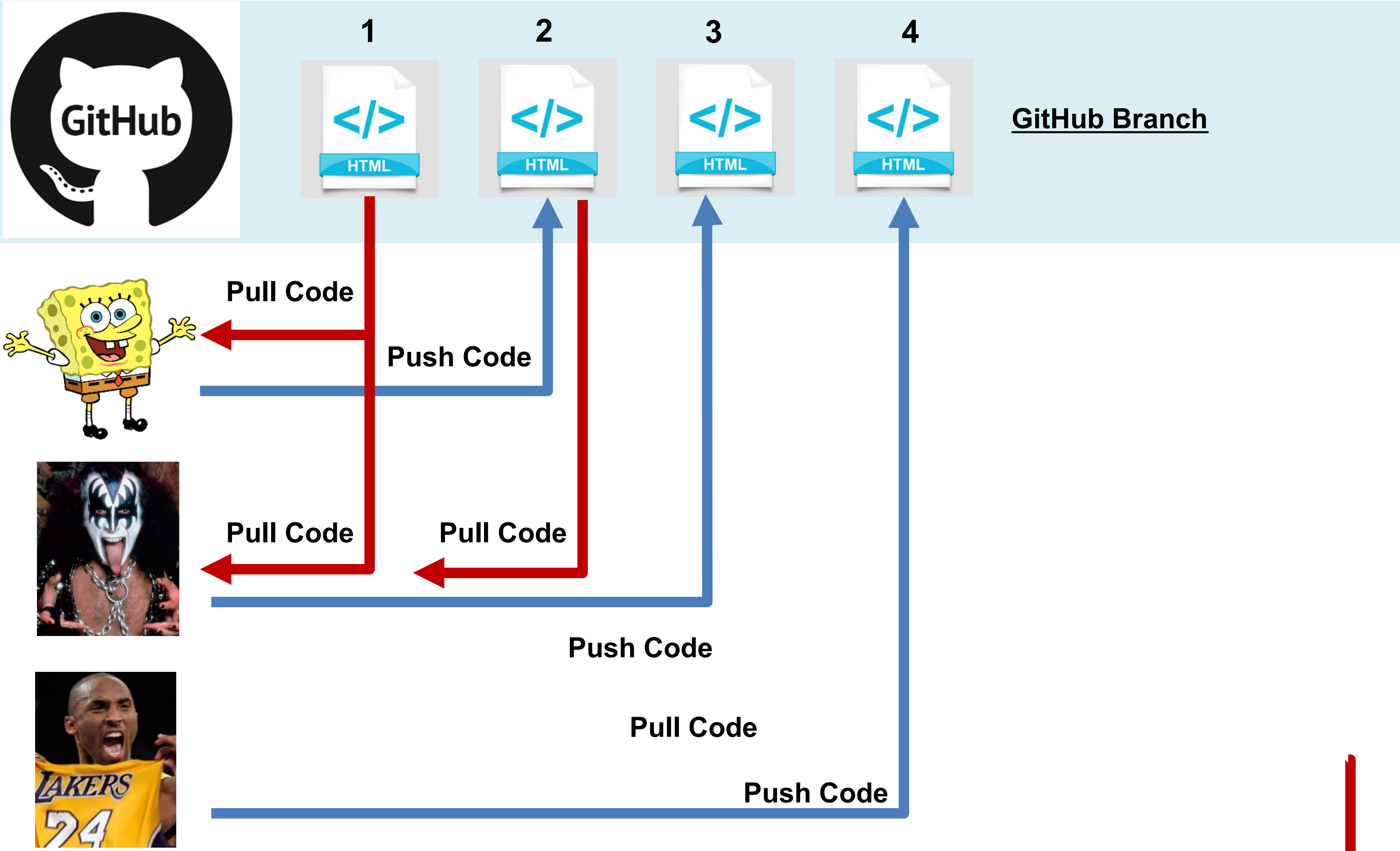
## Web Pages Are Exciting Too

Yes, that's right - web pages can be a lot of fun. Learning how to create web pages is easy and **entertaining**. This is my second *paragraph*. I hope you like it.

- **HTML** is one of the three base languages behind every single website.

- It defines all of the basic content and a *bit* of formatting.

# Pushing and Pulling to GitHub



GitHub Branch

Pull Code

Push Code

Pull Code

Pull Code

Push Code

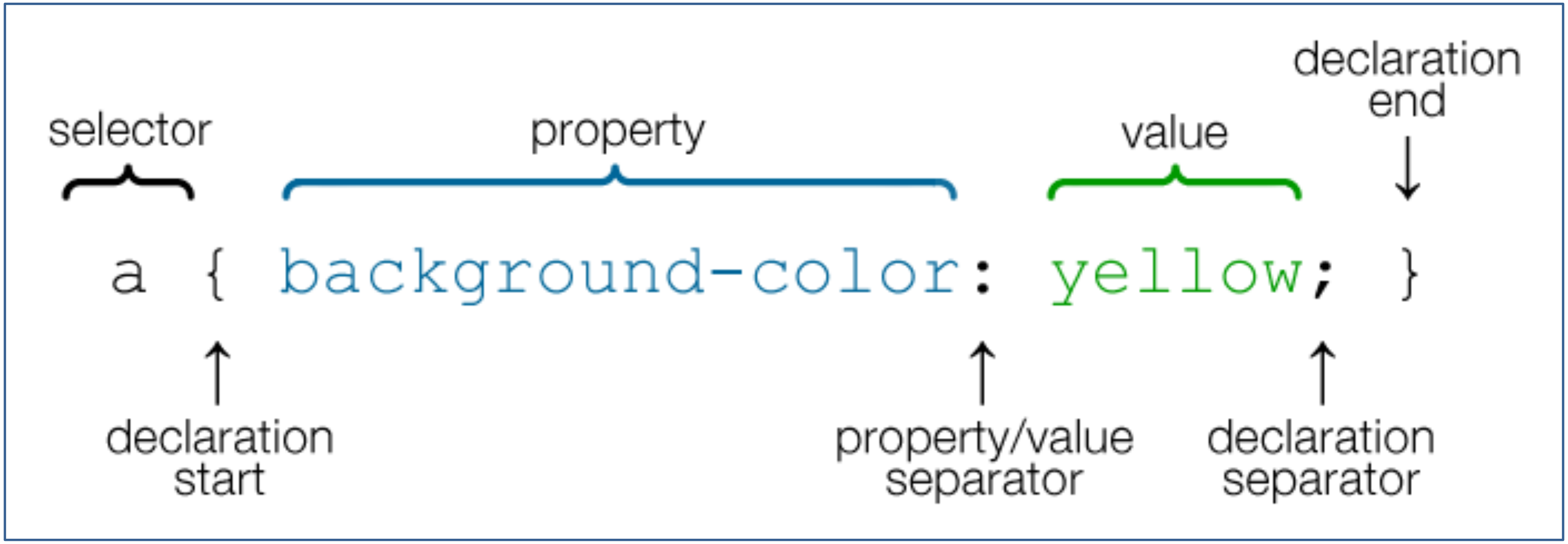
Pull Code

Push Code

# CSS Syntax

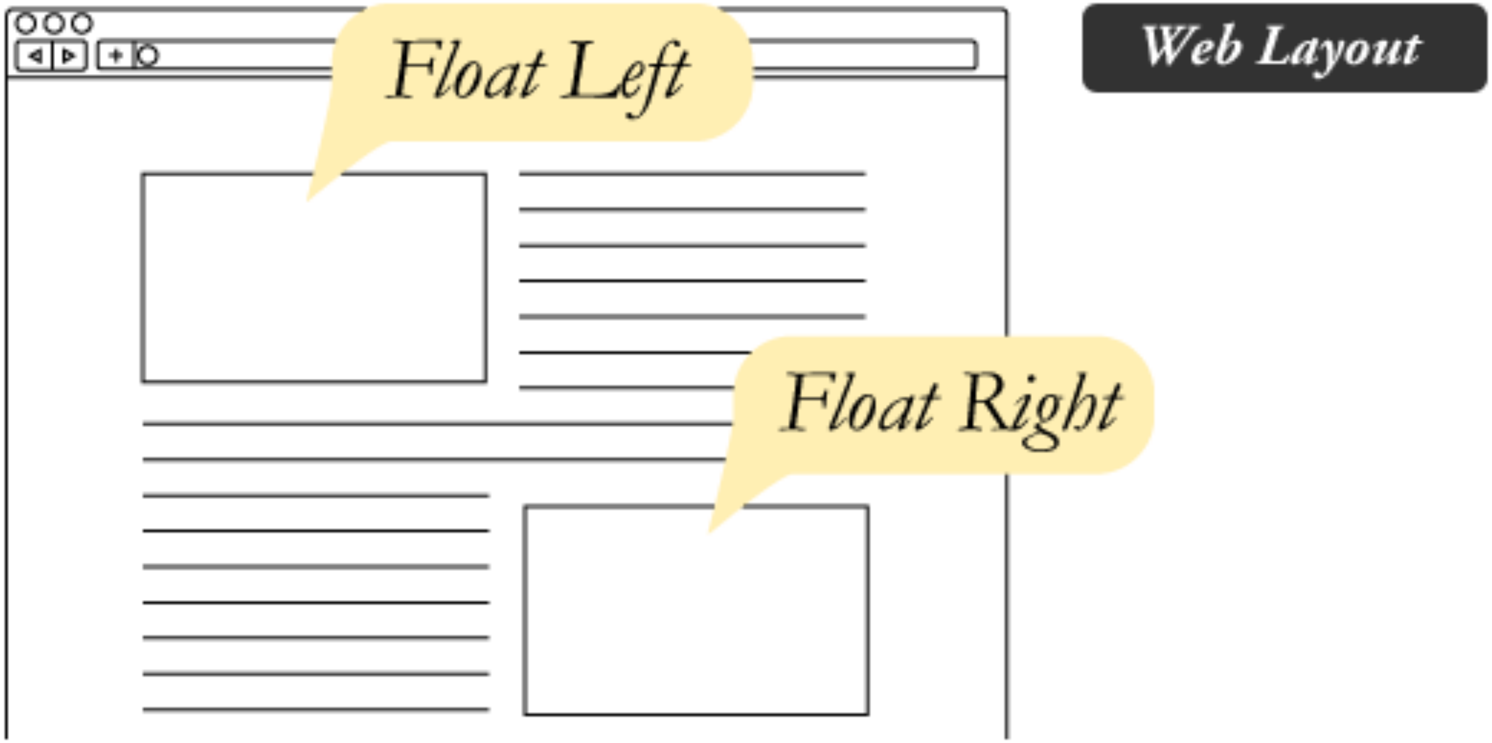
- CSS works by hooking onto **selectors** added into HTML using "**classes** and **identifiers**".

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



```
a { background-color: yellow; }
```

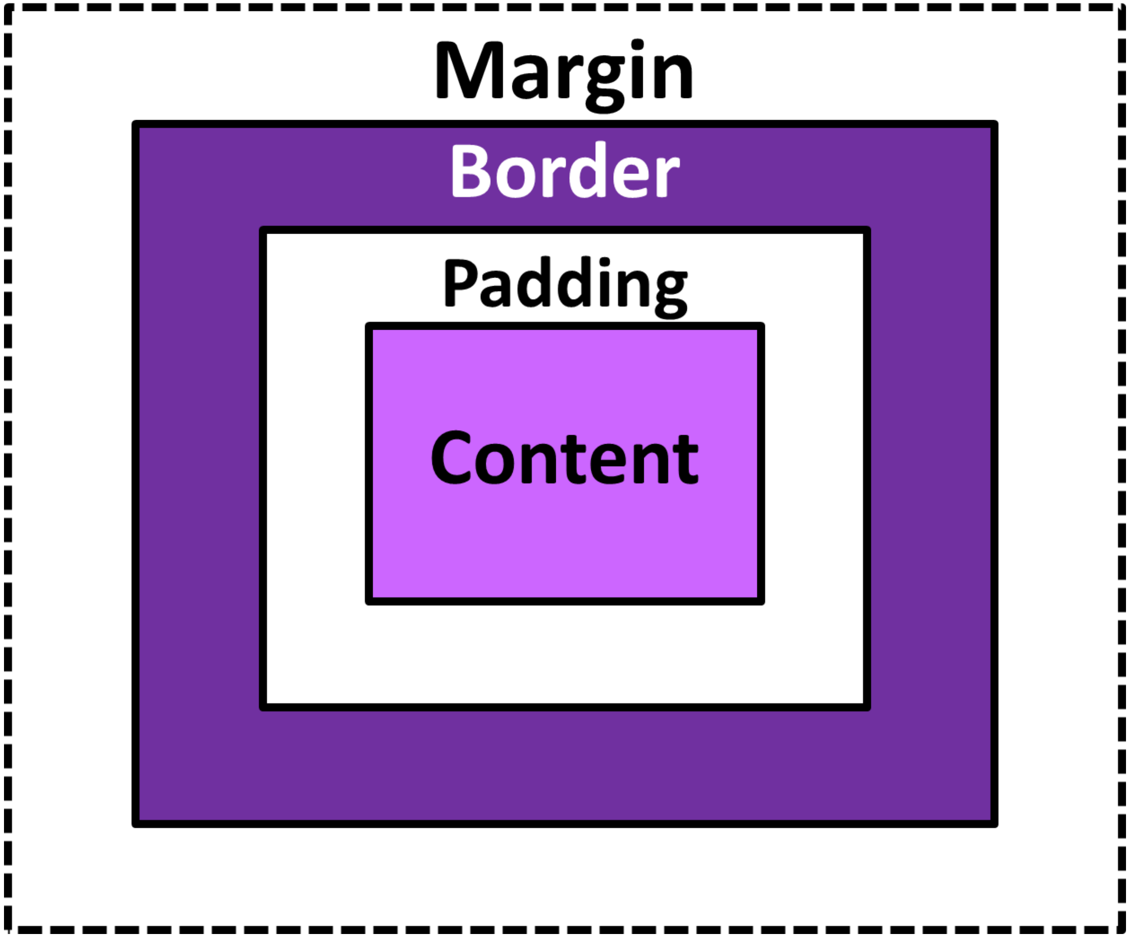
# The Concept of “Flow”



•In HTML/CSS, (by default) every element displayed is governed by a concept called “**flow.**”

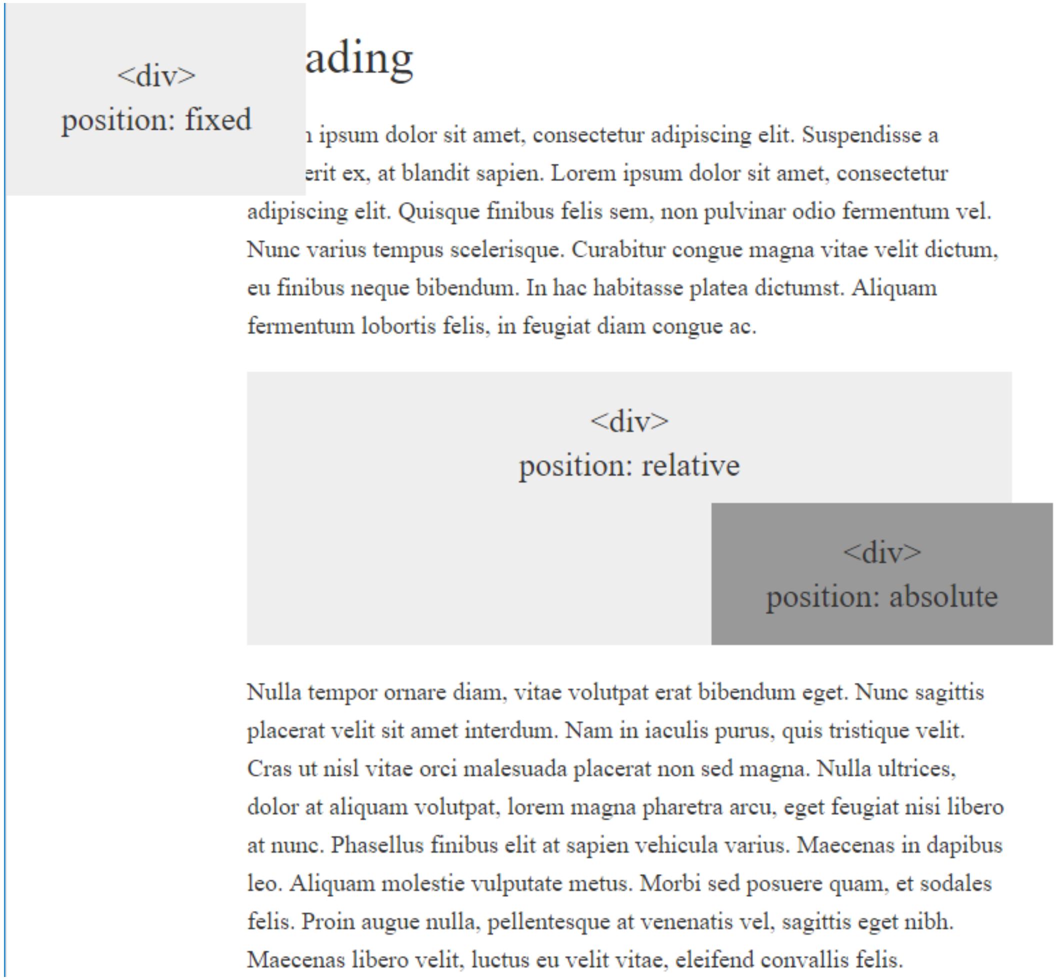
•This means that HTML elements force their adjacent elements to **flow around** them.

# The Box Model



•The Box Model wraps every CSS element in **padding, border and margin** – allowing developers to modify spacing styles.

# CSS Positioning



ading

n 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.

Nulla tempor ornare diam, vitae volutpat erat bibendum eget. Nunc sagittis placerat 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.

- We can orient our HTML elements in relation to space with CSS positioning **(static, relative, fixed, absolute)**.

# How to Learn…

w3schools.com

design shack

CSS newbie

cssreset

stackoverflow

CSS-TRICKS

sitepoint FORUMS

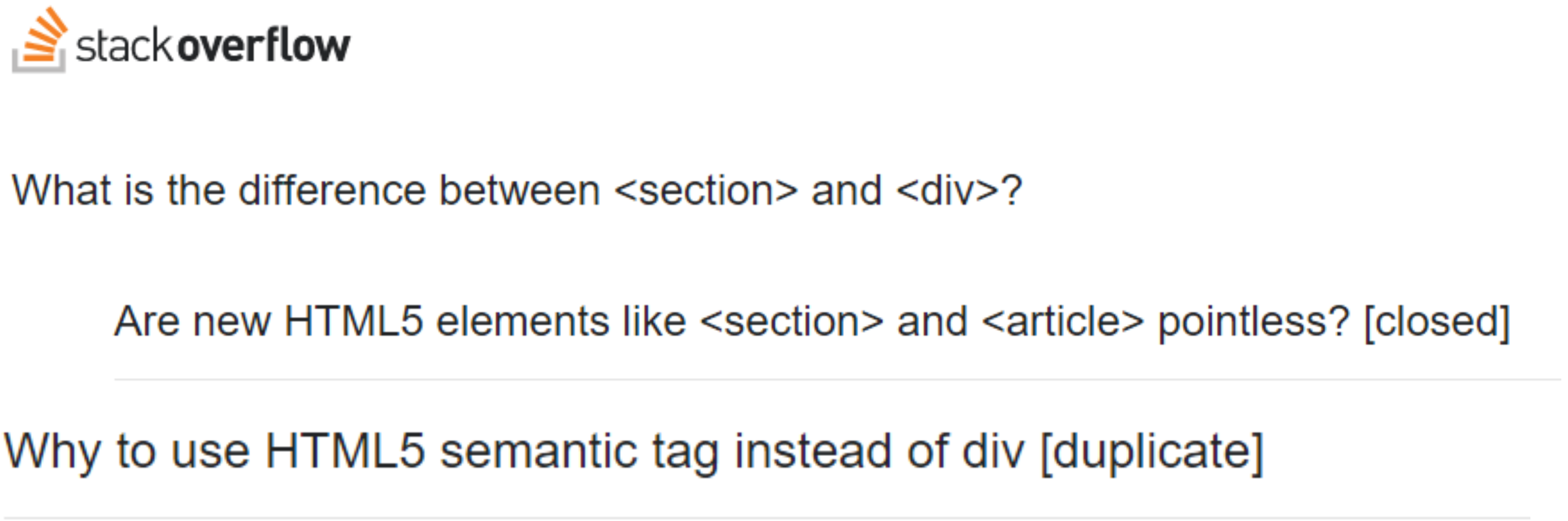
SMASHING MAGAZINE

MDN MOZILLA DEVELOPER NETWORK

# General Questions / Issues?

# *Double Take*

# Divs, Sections, Navs, Etc….



stackoverflow

What is the difference between <section> and <div>?

Are new HTML5 elements like <section> and <article> pointless? [closed]

Why to use HTML5 semantic tag instead of div [duplicate]

10 months ago by Noel Hale

USE DIV OR SECTION ELEMENT

Section (Layout) vs. Div Block (Basic): Which one to use for Semantic Markup and Page Layout?

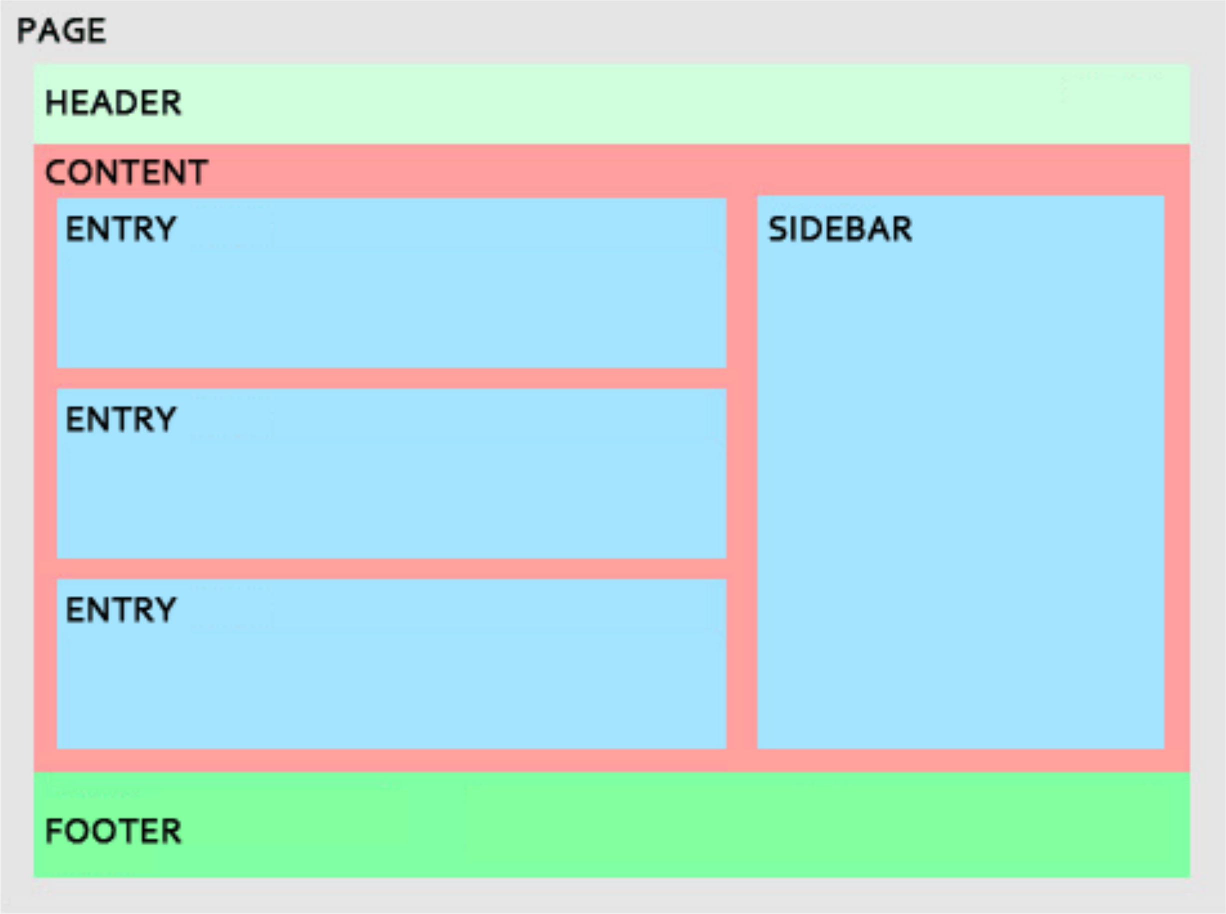
🟩 Need Help

About.com › About Tech › Web Design & HTML › . . . › HTML 5 Tags

What is the Difference Between DIV and SECTION?

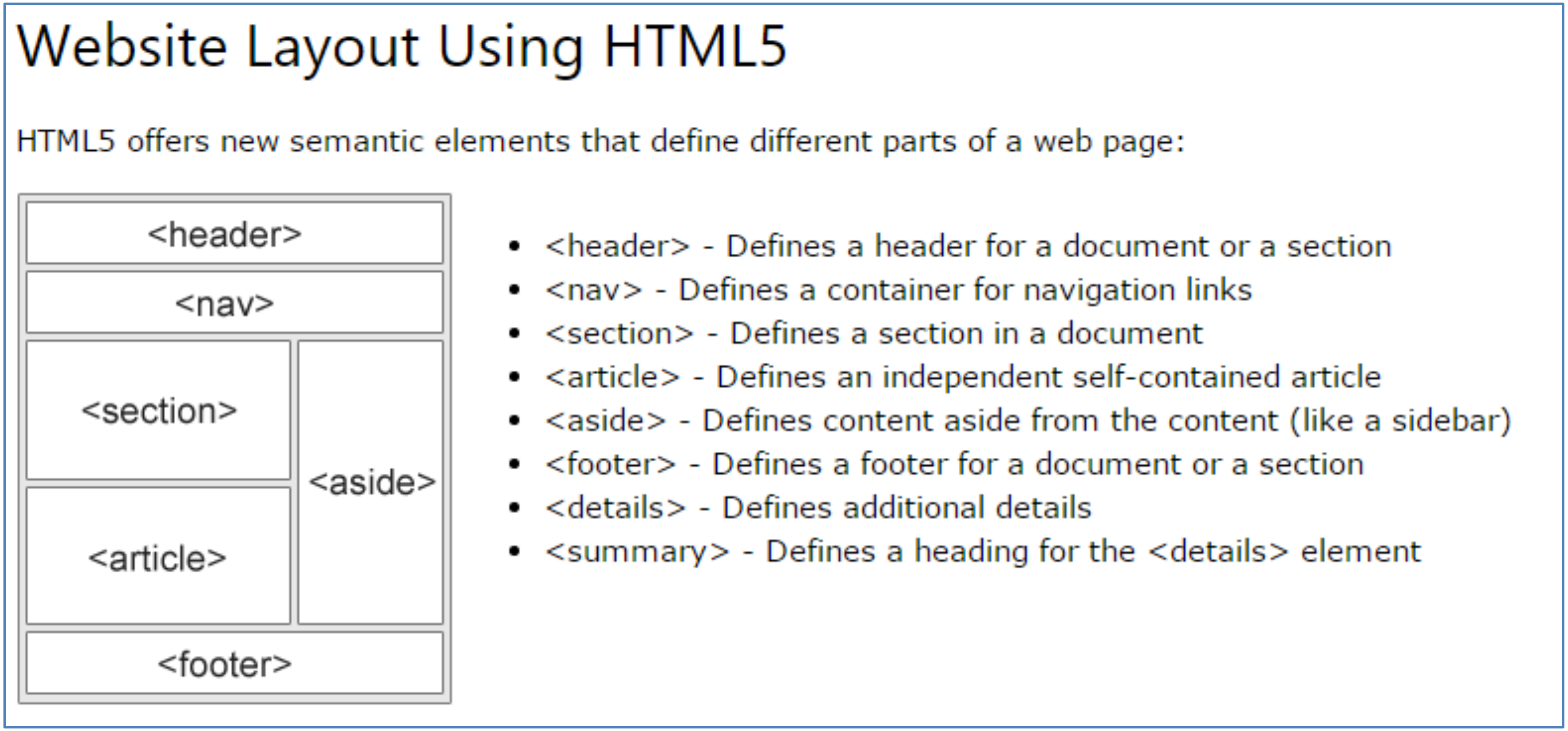
Understanding the HTML5 SECTION Element

# Divs, Sections, Navs, Etc….



- All web layouts are inherently composed of containers, traditionally called “**divs.**”

# Divs, Sections, Navs, Etc….



## Website Layout Using HTML5

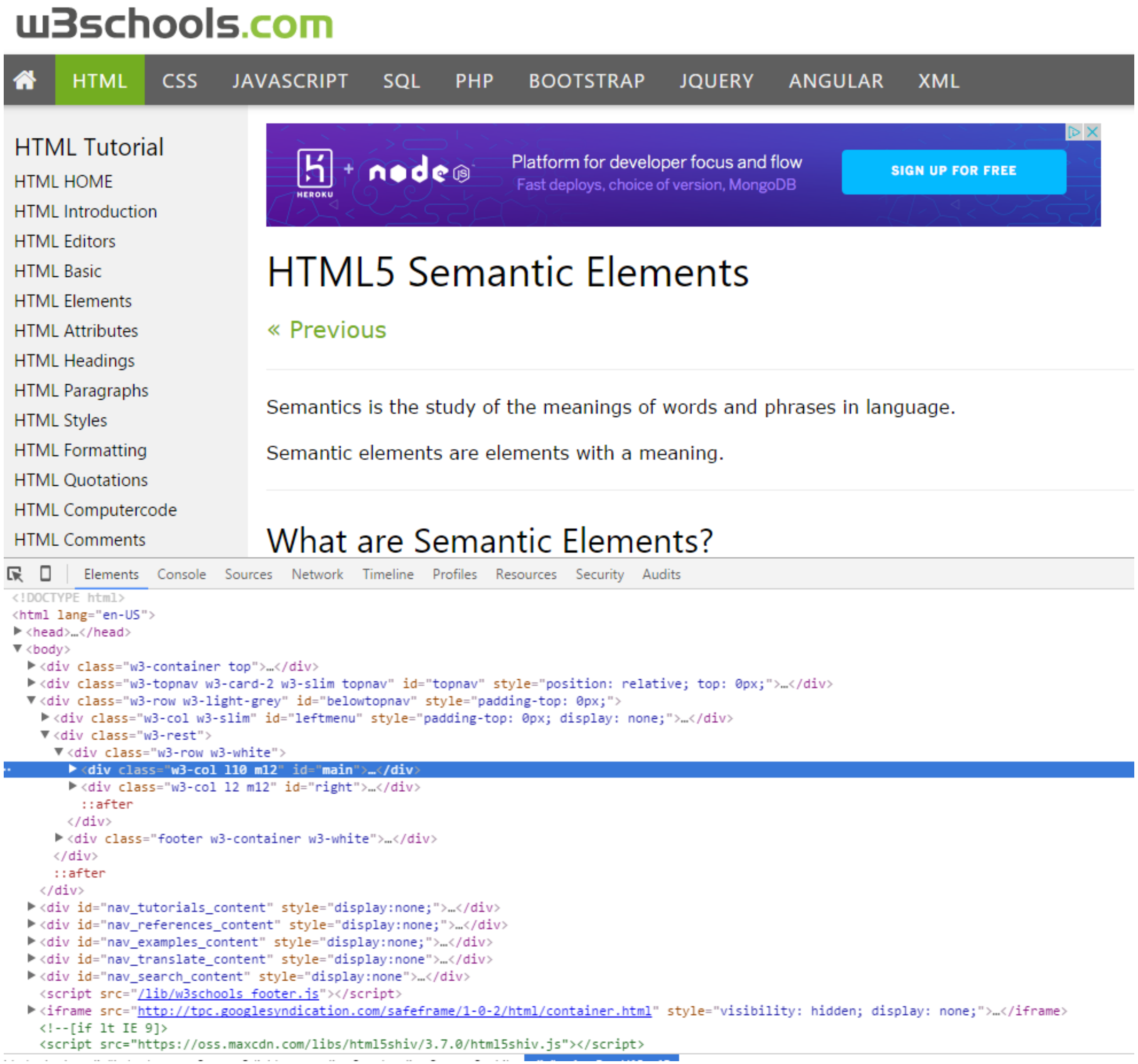
HTML5 offers new semantic elements that define different parts of a web page:

- <header> - Defines a header for a document or a section
- <nav> - Defines a container for navigation links
- <section> - Defines a section in a document
- <article> - Defines an independent self-contained article
- <aside> - Defines content aside from the content (like a sidebar)
- <footer> - Defines a footer for a document or a section
- <details> - Defines additional details
- <summary> - Defines a heading for the <details> element

•HTML5 introduced the concept of **“semantic layouts,”** meaning “divs” could be given more meaningful names.

•In theory, this helps with organization and search engine optimization.

# Divs, Sections, Navs, Etc….



HTML5 Semantic Elements

Semantics is the study of the meanings of words and phrases in language.

Semantic elements are elements with a meaning.

What are Semantic Elements?

```
<!DOCTYPE html>
<html lang="en-US">
▶<head>…</head>
▼<body>
  ▶<div class="w3-container top">…</div>
  ▶<div class="w3-topnav w3-card-2 w3-slim topnav" id="topnav" style="position: relative; top: 0px;">…</div>
  ▼<div class="w3-row w3-light-grey" id="belowtopnav" style="padding-top: 0px;">
    ▶<div class="w3-col w3-slim" id="leftmenu" style="padding-top: 0px; display: none;">…</div>
    ▼<div class="w3-rest">
      ▼<div class="w3-row w3-white">
        ▶<div class="w3-col l10 m12" id="main">…</div>
        ▶<div class="w3-col l2 m12" id="right">…</div>
          ::after
        </div>
      ▶<div class="footer w3-container w3-white">…</div>
      </div>
      ::after
    </div>
  ▶<div id="nav_tutorials_content" style="display:none;">…</div>
  ▶<div id="nav_references_content" style="display:none;">…</div>
  ▶<div id="nav_examples_content" style="display:none;">…</div>
  ▶<div id="nav_translate_content" style="display:none">…</div>
  ▶<div id="nav_search_content" style="display:none">…</div>
  <script src="/lib/w3schools_footer.js"></script>
  ▶<iframe src="http://tpc.googlesyndication.com/safeframe/1-0-2/html/container.html" style="visibility: hidden; display: none;">…</iframe>
  <!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
```

- That said… many (if not most) websites, seem to still be using basic **divs.**

- There are reasons for this that we'll showcase in later sections.

- Additionally, it's possible to include "semantics" by using id names and classes.

# Divs, Sections, Navs, Etc….

div?

Section?

- **Bottom line:**
- Follow your homework’s instructions. But when you get out in the “real world,” follow the convention of where you work!

# Classes vs. IDs

**Classes = Barcode (all iPod)**

**IDs = Serial Number (unique iPod)**

When choosing between a CSS ID and a CSS Class follow the convention:

•**Classes (.classname)** are to be used if the same style will be used on multiple HTML elements.

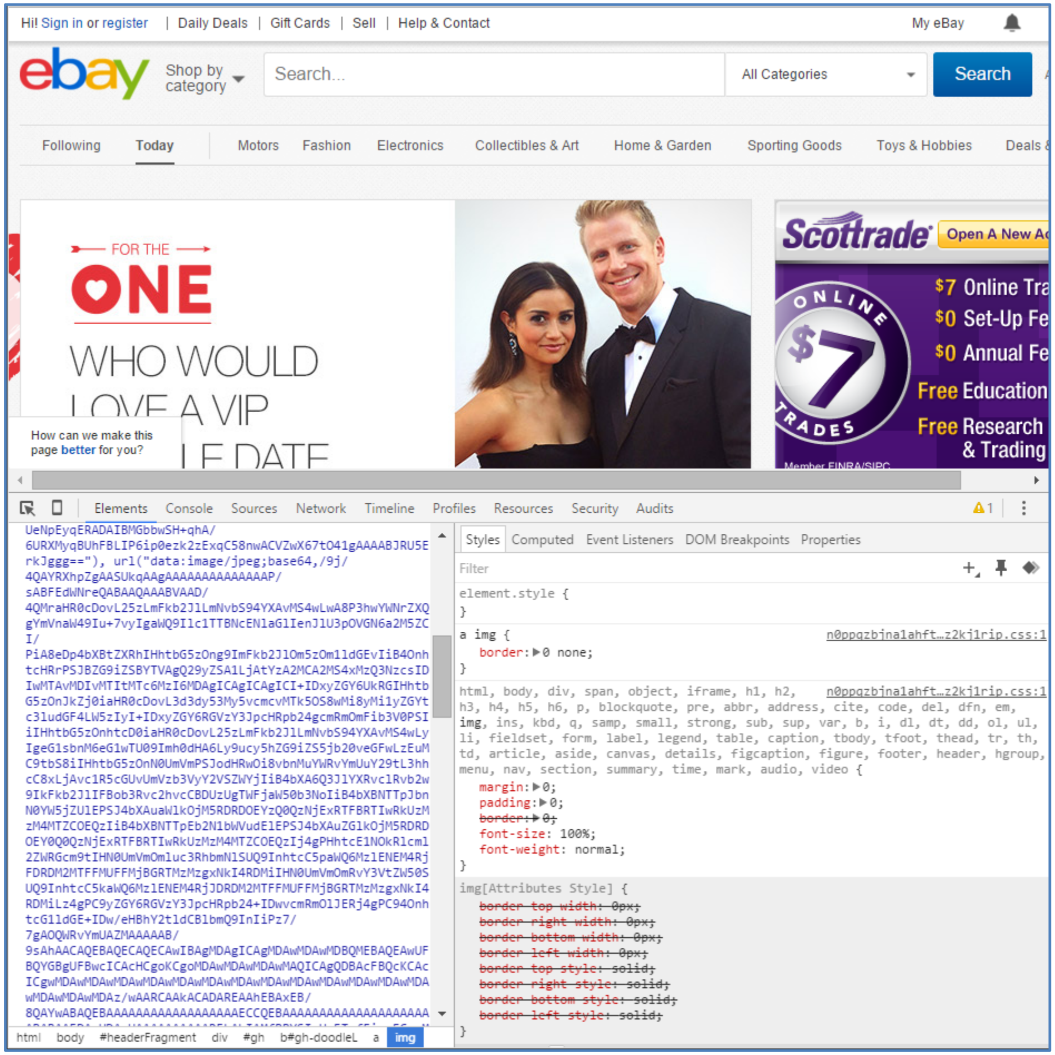
•**IDs (#idname)** are to be used if a style is *unique* to that HTML element.

# Google Developer Tools (Inspector)

- **GDT** is one of the most frequent tools you will use in web dev.

- It allows you to truly, debug your web designs.

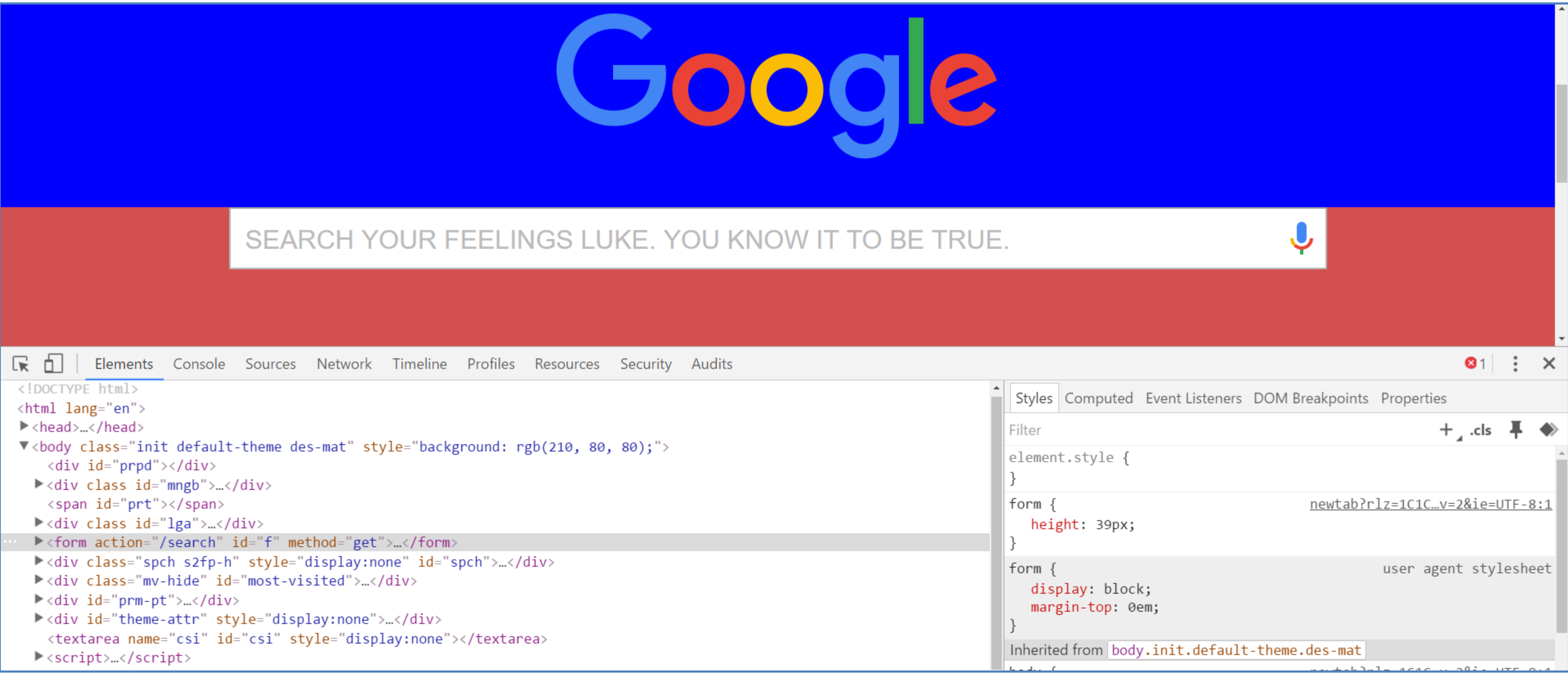
- **<u>Start using it!</u>**



```
UeNpEyqERADAIBMGbbwSH+qhA/
6URXMyqBUhFBLIP6ip0ezk2zExqC58nwACVZwX67tO41gAAAABJRU5E
rkJggg=="), url("data:image/jpeg;base64,/9j/
4QAYRXhpZgAASUkqAAgAAAAAAAAAAAAAAP/
sABFEdWNreQABAAQAAABVAAD/
4QMraHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLwA8P3hwYWNrZXQ
gYmVnaW49Iu+7vyIgaWQ9Ilc1TTBNcENlaGlIenJlU3pOVGN6a2M5ZC
I/
PiA8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIiB4Onh
tcHRrPSJBZG9iZSBYTVAgQ29yZSA1LjAtYzA2MCA2MS4xMzQ3NzcsID
IwMTAvMDIvMTItMTc6MzI6MDAgICAgICAgICI+IDxyZGY6UkRGIHhtb
G5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYt
c3ludGF4LW5zIyI+IDxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSI
iIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLy
IgeG1sbnM6eG1wTU09Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuM
C9tbS8iIHhtbG5zOnN0UmVmPSJodHRwOi8vbnMuYWRvYmUuY29tL3hh
cC8xLjAvc1R5cGUvUmVzb3VyY2VSZWYjIiB4bXA6Q3JlYXRvclRvb2w
9IkFkb2JlIFBob3Rvc2hvcCBDUzUgTWFjaW50b3NoIiB4bXBNTTpJbn
N0YW5jZUlEPSJ4bXAuaWlkOjM5RDRDOEYzQ0QzNjExRTFBRTIwRkUzM
zM4MTZCOEQzIiB4bXBNTTpEb2N1bWVudElEPSJ4bXAuZGlkOjM5RDRD
OEY0Q0QzNjExRTFBRTIwRkUzMzM4MTZCOEQzIj4gPHhtcE1NOkRlcml
2ZWRGcm9tIHN0UmVmOmluc3RhbmNlSUQ9InhtcC5paWQ6MzlENEM4Rj
FDRDM2MTFFMUFFMjBGRTMzMzgxNkI4RDMiIHN0UmVmOmRvY3VtZW50S
UQ9InhtcC5kaWQ6MzlENEM4RjJDRDM2MTFFMUFFMjBGRTMzMzgxNkI4
RDMiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmRmOlJERj4gPC94Onh
tcG1ldGE+IDw/eHBhY2tldCBlbmQ9InIiPz7/
7gAOQWRvYmUAZMAAAAAB/
9sAhAACAQEBAQECAQECAwIBAgMDAgICAgMDAwMDAwMDBQMEBAQEAwUF
BQYGBgUFBwcICAcHCgoKCgoMDAwMDAwMDAwMAQICAgQDBAcFBQcKCAc
ICgwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDA
wMDAwMDAwMDAz/wAARCAAkACADAREAAhEBAxEB/
8QAYwABAQEBAAAAAAAAAAAAAAAAAAECCQEBAAAAAAAAAAAAAAAAAAAA
```

```
element.style {
}
a img {                                                n0ppqzbjna1ahft…z2kj1rip.css:1
   border:▶0 none;
}
html, body, div, span, object, iframe, h1, h2,         n0ppqzbjna1ahft…z2kj1rip.css:1
h3, h4, h5, h6, p, blockquote, pre, abbr, address, cite, code, del, dfn, em,
img, ins, kbd, q, samp, small, strong, sub, sup, var, b, i, dl, dt, dd, ol, ul,
li, fieldset, form, label, legend, table, caption, tbody, tfoot, thead, tr, th,
td, article, aside, canvas, details, figcaption, figure, footer, header, hgroup,
menu, nav, section, summary, time, mark, audio, video {
   margin:▶0;
   padding:▶0;
   ~~border:▶0;~~
   font-size: 100%;
   font-weight: normal;
}
img[Attributes Style] {
   ~~border-top-width: 0px;~~
   ~~border-right-width: 0px;~~
   ~~border-bottom-width: 0px;~~
   ~~border-left-width: 0px;~~
   ~~border-top-style: solid;~~
   ~~border-right-style: solid;~~
   ~~border-bottom-style: solid;~~
   ~~border-left-style: solid;~~
}
```

# Modifying Sites



```
<!DOCTYPE html>
<html lang="en">
▶<head>…</head>
▼<body class="init default-theme des-mat" style="background: rgb(210, 80, 80);">
   <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>
```

```
element.style {
}
form {                                  newtab?rlz=1C1C…v=2&ie=UTF-8:1
   height: 39px;
}
form {                                  user agent stylesheet
   display: block;
   margin-top: 0em;
}
Inherited from body.init.default-theme.des-mat
```

- **You can edit any web page's HTML and CSS with Chrome Inspector.**
- Plus, you'll see your results instantly.

# INSTRUCTOR DEMO!

***Instructor: Demo***

*(Google Developer Tools)*

# > YOUR TURN!!

**Suggested Time:** 15 min

## Assignment

For the next 15 minutes, take a website you commonly use (Amazon, Google, Huff Po, etc.) and heavily modify it using the Google Developer Tools.

Be sure to at least modify:

- Content (Change words)
- Colors
- Spacing

Send a screenshot to the class's slack profile when you're done.

# > YOUR TURN!!

**Suggested Time:** 10 min

## Assignment

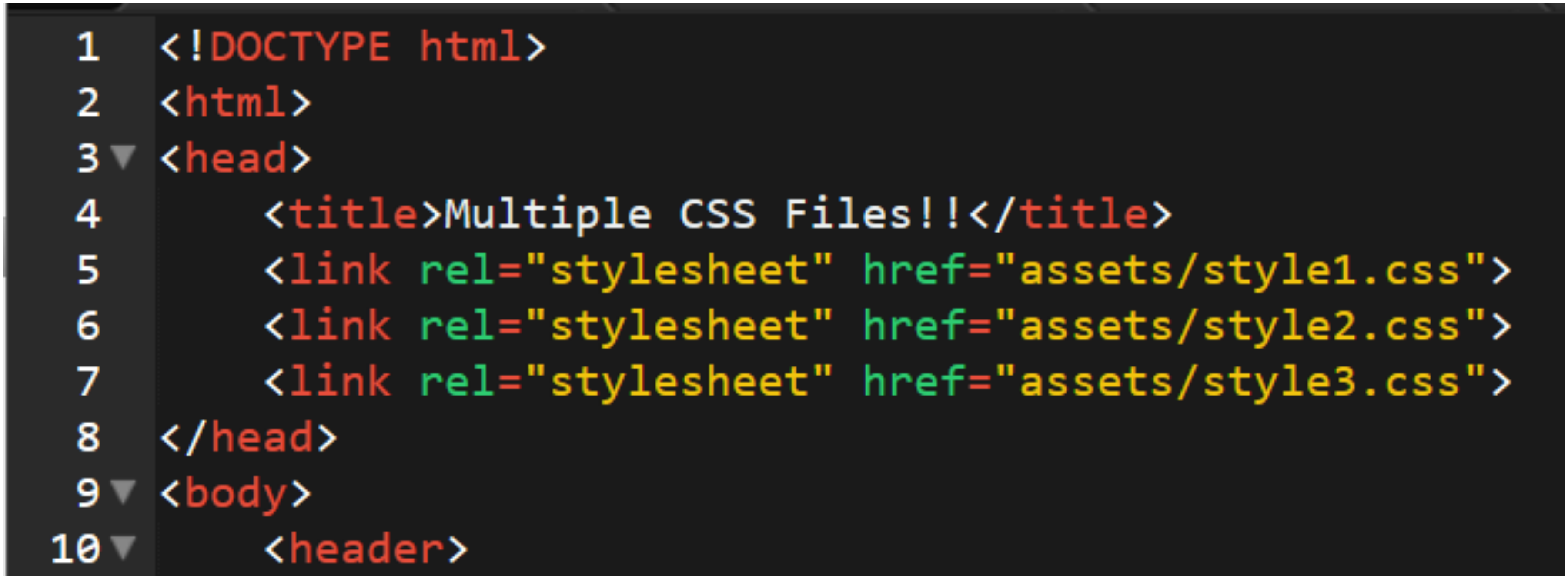
For the next 10 minutes, edit any site that you've been working on in-class or for homework with Google's dev tools.

Be sure to at least modify:

- Content (Change words)
- Colors
- Spacing

# *CSS Resets*

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



```html
<!DOCTYPE html>
<html>
<head>
    <title>Multiple CSS Files!!</title>
    <link rel="stylesheet" href="assets/style1.css">
    <link rel="stylesheet" href="assets/style2.css">
    <link rel="stylesheet" href="assets/style3.css">
</head>
<body>
    <header>
```

•An incredibly powerful technique: deploying multiple CSS files simultaneously.

•This lets developers to create complex designs made up of abounding design elements.

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

# INSTRUCTOR DEMO!

***Instructor: Demo***

*(1-3_CSSFiles.html | 1-MultipleCSS)*

# What Browser?

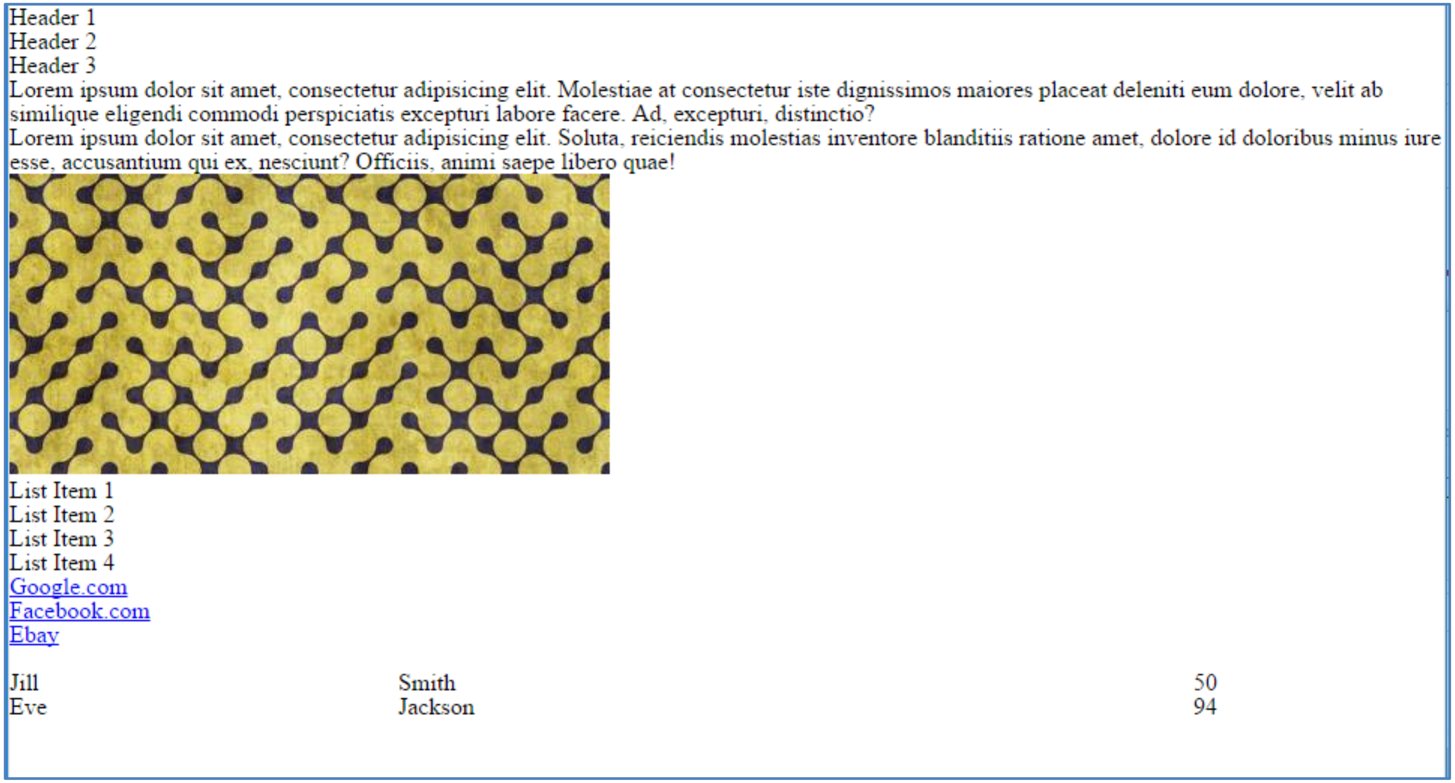
*By a show of hands…*

***Which browser do you use?***

# Battle of the Browsers

- Under the hood, web browsers often **render web pages 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)



Header 1
Header 2
Header 3
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Molestiae at consectetur iste dignissimos maiores placeat deleniti eum dolore, velit ab similique eligendi commodi perspiciatis excepturi labore facere. Ad, excepturi, distinctio?
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Soluta, reiciendis molestias inventore blanditiis ratione amet, dolore id doloribus minus iure esse, accusantium qui ex, nesciunt? Officiis, animi saepe libero quae!

List Item 1
List Item 2
List Item 3
List Item 4
Google.com
Facebook.com
Ebay

| | | |
|---|---|---|
| Jill | Smith | 50 |
| Eve | Jackson | 94 |

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

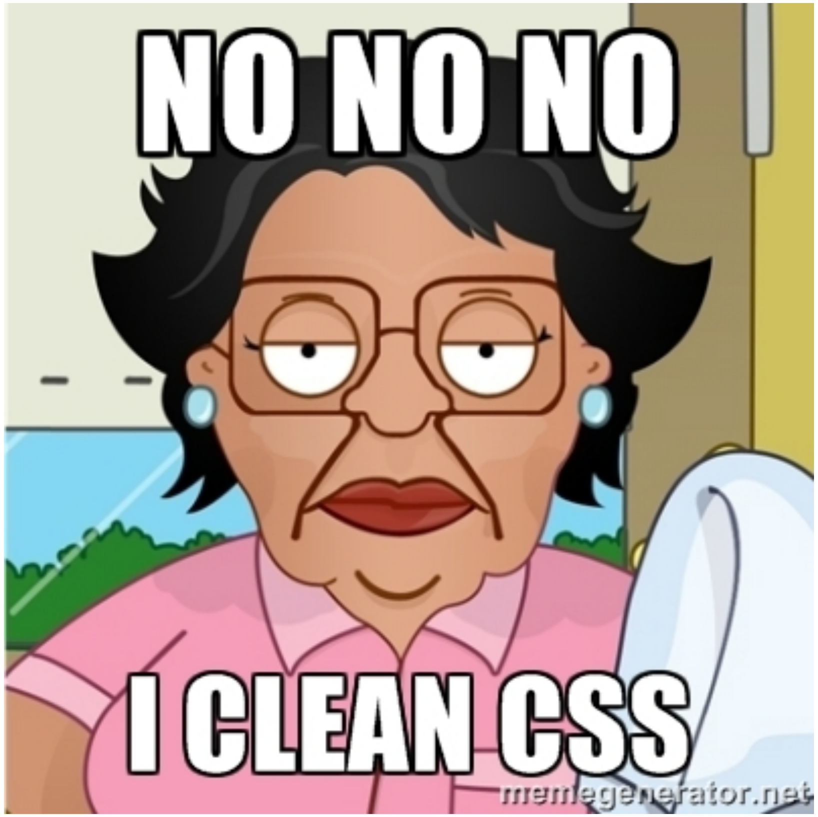
- However, you will have to re-style everything yourself.

# INSTRUCTOR DEMO!

***Instructor: Demo***

*(Example.html | 2-ResetCSS)*

# Why CSS Resets Matter

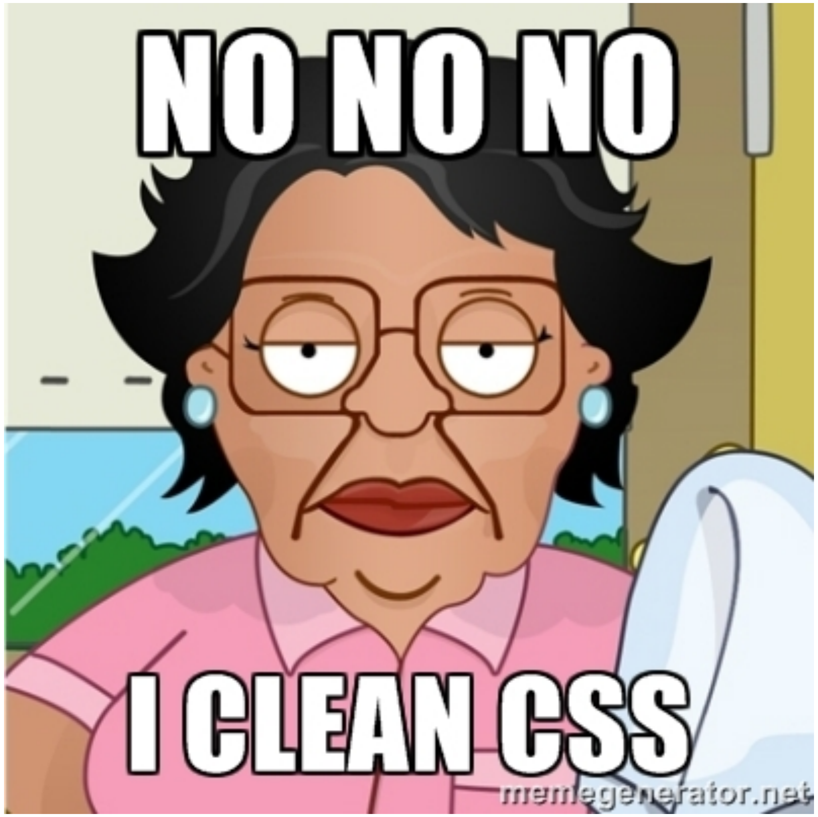


1. It's important for creating browser-compatible websites

1. It's an example of using someone else's CSS in *your* website!!!

1. It's a common Front-End Developer Interview question.

# Why CSS Resets Matter



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

1. It's an example of using someone else's CSS in *your* website!!!

1. It's a common Front-End Developer Interview question.

# > YOUR TURN!!

**Suggested Time:** 10 min

## Assignment

Follow the instructions given via slack to incorporate a reset.css file into a basic HTML file.

Note the impact the reset file makes after its inclusion.

# *To the Web with Heroku!*

# The Internet

A deep and complex diagram above on how the internet works.

# The World Will See Our Greatness!

- **Heroku provides a cloud application hosting platform –** which means we can deploy our websites and applications onto their servers for the world to see.

# Together Now…

***Let’s all login to Heroku***

# INSTRUCTOR DEMO!

***Instructor: Demo***

*(Heroku Deployment)*

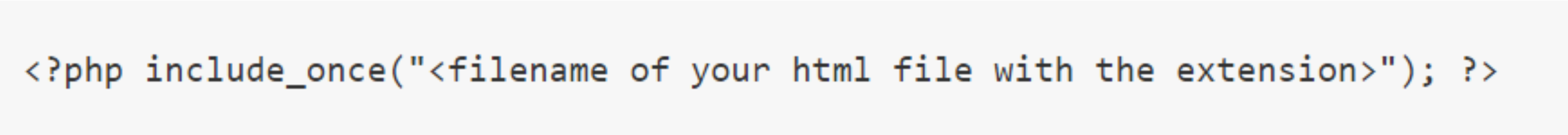
# Deploying Static Websites to Heroku

**Basic Steps:**

1.Go to folder you want to host (must be .git enabled).

1.Add a file called composer.json and include an empty bracket {}.

1.Add a file called index.php with the following inside:



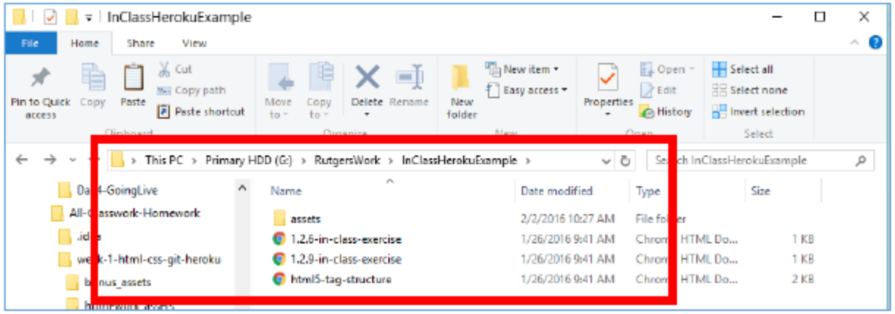
```
<?php include_once("<filename of your html file with the extension>"); ?>
```

2.Run git remote –v.
3.Run heroku create.
4.Run git remote –v.
5.Run git push heroku master.

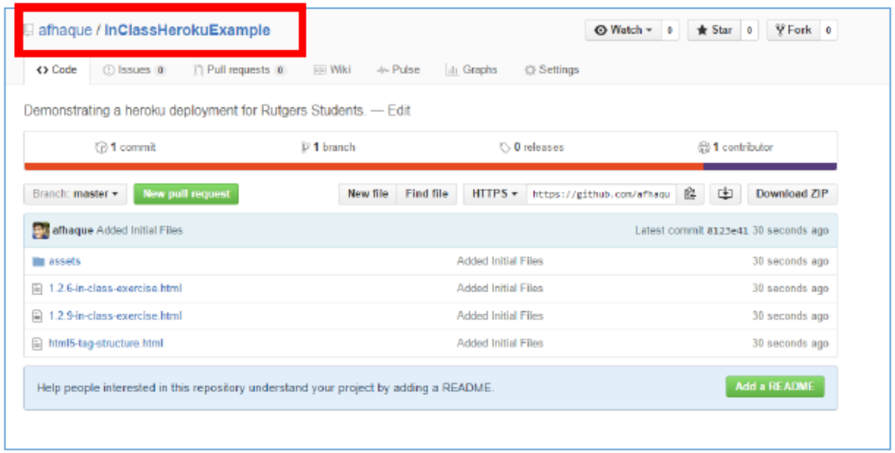
# Follow our Guide!

## Deploying Your Static Website to Heroku

1. Choose a local folder of code that is tied to GitHub. (If you do not have one available, clone a repository you have in GitHub.)



| Name | Date modified | Type | Size |
|---|---|---|---|
| assets | 2/2/2016 10:27 AM | File folder | |
| 1.2.6-in-class-exercise | 1/26/2016 9:41 AM | Chrome HTML Do... | 1 KB |
| 1.2.9-in-class-exercise | 1/26/2016 9:41 AM | Chrome HTML Do... | 1 KB |
| html5-tag-structure | 1/26/2016 9:41 AM | Chrome HTML Do... | 2 KB |



Demonstrating a heroku deployment for Rutgers Students. — Edit

https://github.com/afhaqu

afhaque Added Initial Files — Latest commit 8123e41 30 seconds ago

| | | |
|---|---|---|
| assets | Added Initial Files | 30 seconds ago |
| 1.2.6-in-class-exercise.html | Added Initial Files | 30 seconds ago |
| 1.2.9-in-class-exercise.html | Added Initial Files | 30 seconds ago |
| html5-tag-structure.html | Added Initial Files | 30 seconds ago |

Help people interested in this repository understand your project by adding a README.

You can see that in my example, I'm working with a folder called InClassHerokuExample. I have this folder both locally (on my machine) and in GitHub.

## Step-by-Step Guide on Creating Heroku Deployments

# > YOUR TURN!!

**Suggested Time:** 30 min

## Assignment

Time to take your newfangled website and deploy it to the cloud. Setup your own instance of Heroku and deploy one of your HTML creations to Heroku.

Additional instructions to be sent via Slack.

# Practice Through Frustration

***Keep Practicing!***

*It gets better.*

# *Questions?*

# *Homework 1 - Help?*

# EXTRA MATERIAL

# And Back to Git…

# The Group Project

OMG. I HAZ THE GREATEST HTML IDEA!!!!!

***SpongeSite.com***

# The Group Project

Programming Away…

**Spongebob's idea is dumb. We should call it…**

***PrincezzzSite.com***

# The Group Project – Tragedy #1

Programming Away…

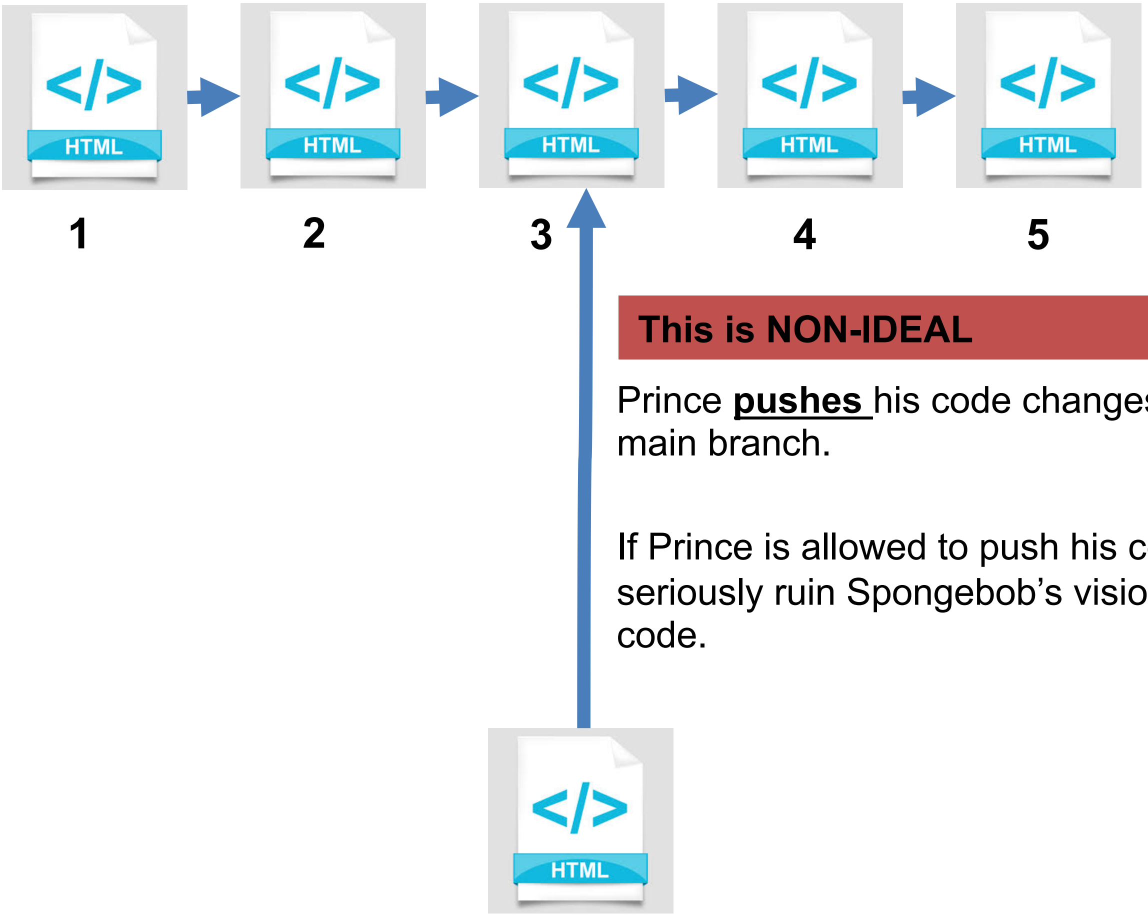
Now they have two completely ***__different__*** versions.

Programming Away…

# The Group Project – Push vs Pull

**Main Branch (Spongebob's)**

Spongebob continues programming



**This is NON-IDEAL**

Prince **pushes** his code changes into the main branch.

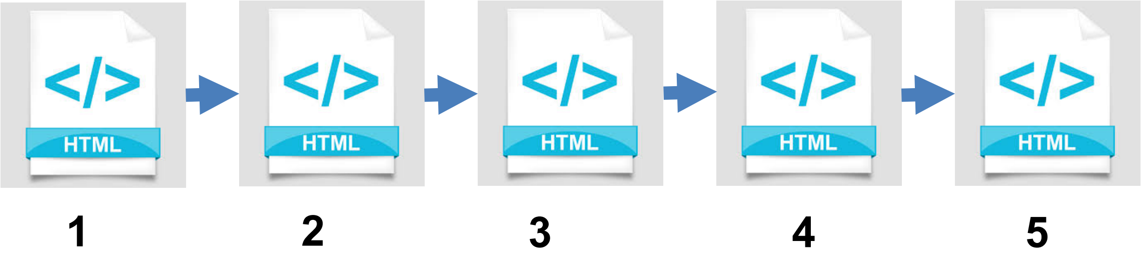
If Prince is allowed to push his code, it could seriously ruin Spongebob's vision and working code.

**Prince's Branch**

# The Group Project – Push vs Pull

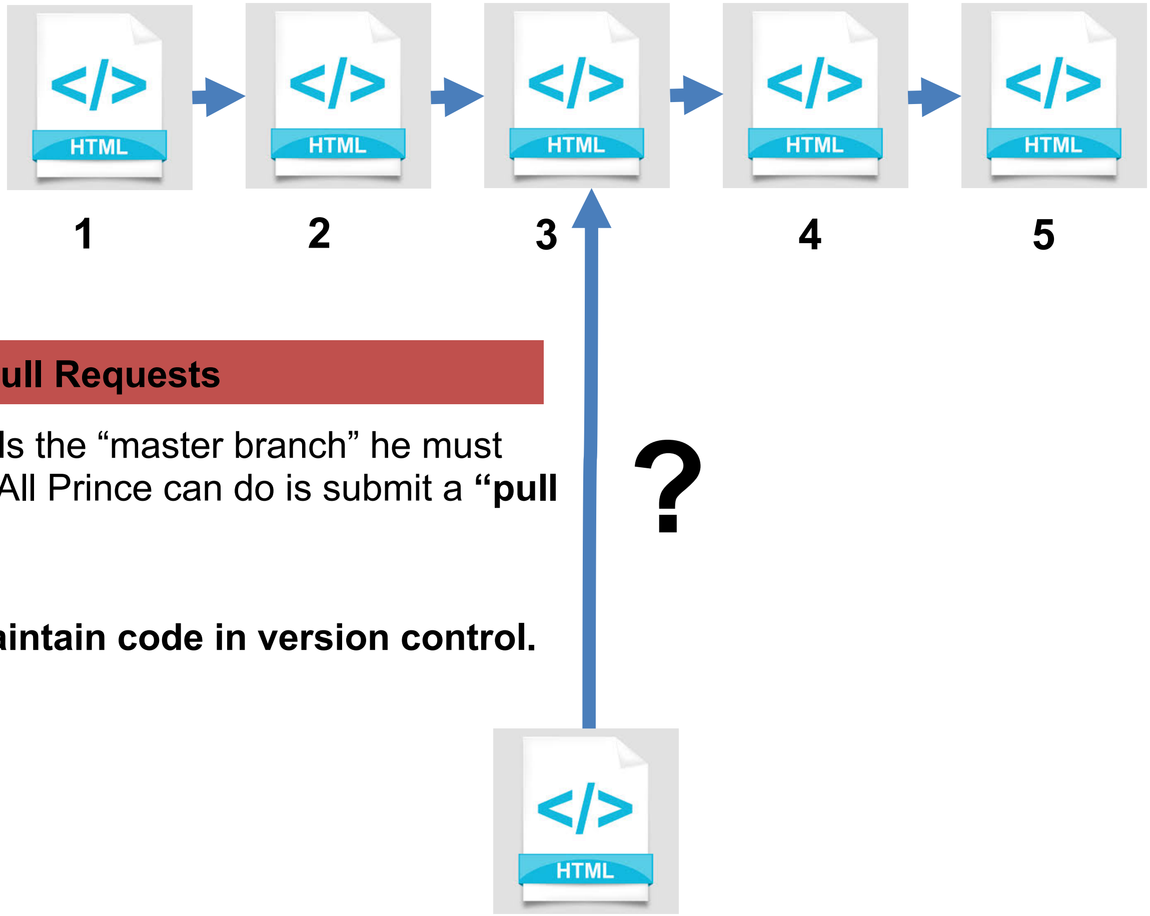
**Main Branch (Spongebob's)**

Spongebob continues programming



**Prince's Branch**

# The Group Project – Push vs Pull



**Main Branch (Spongebob's)**

Spongebob continues programming

1 2 3 4 5

?

**Prince's Branch**

## Ideal Approach – Using Pull Requests

Because Spongebob controls the "master branch" he must elect to **pull** Prince's Code. All Prince can do is submit a **"pull request"**

**This is the ideal way to maintain code in version control.**

# General Steps for Git Pull Requests

1.Create a new branch of on your local computer

2. *git branch <BRANCH NAME>*

1.Checkout that branch (locally) on your machine

2. *git checkout <BRANCH NAME>*

1.Add / Commit your changes (will automatically save to this branch)

2. *git add –A*

*3. git commit –m “Comment”*

1.Push your branch to GitHub

*git push origin <BRANCH NAME>*

1.Submit a Pull Request on GitHub

1.Other user must accept these changes on GitHub

# INSTRUCTOR DEMO!

***Git Pull Request***

# > YOUR TURN!!

## Assignment

Time to take your newfound collaborative git skills to the real-world. Find a partner and follow the steps sent via slack to

•Share each other’s code

•Make modifications

•Submit a Pull Request

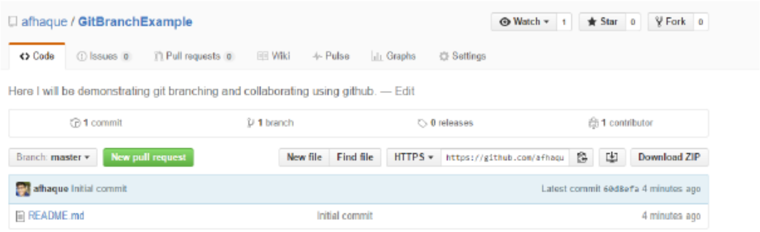
•Accept the Pull Changes

# Follow our Guide!

## Create, Checkout, and Pull Git Merges

This guide will walk you through the process of creating branches on a single code repository. While this guide is intended for a single developer who would like to use pull requests on his/her own repository, it can be adapted to cases where multiple developers are working together. (See step for "Adding Collaborator").

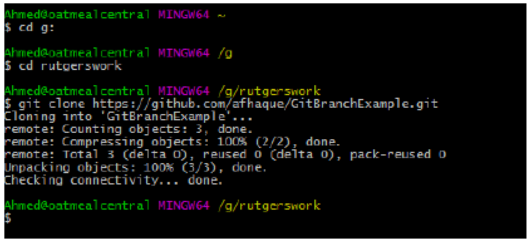
1. Find or create a code repository in GitHub. Clone this repository to your local directory if it hasn't been cloned already. (In my case, I have a completely empty repository).



Here I will be demonstrating git branching and collaborating using github. — Edit

afhaque Initial commit — Latest commit 60d8efa 4 minutes ago

README.md — Initial commit — 4 minutes ago

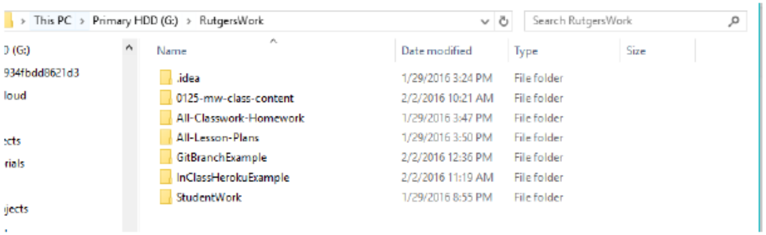


```
Ahmed@oatmealcentral MINGW64 ~
$ cd g:

Ahmed@oatmealcentral MINGW64 /g
$ cd rutgerswork

Ahmed@oatmealcentral MINGW64 /g/rutgerswork
$ git clone https://github.com/afhaque/GitBranchExample.git
Cloning into 'GitBranchExample'...
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.

Ahmed@oatmealcentral MINGW64 /g/rutgerswork
$
```



| Name | Date modified | Type | Size |
|---|---|---|---|
| .idea | 1/29/2016 3:24 PM | File folder | |
| 0125-mw-class-content | 2/2/2016 10:21 AM | File folder | |
| All-Classwork-Homework | 1/29/2016 3:47 PM | File folder | |
| All-Lesson-Plans | 1/29/2016 3:50 PM | File folder | |
| GitBranchExample | 2/2/2016 12:36 PM | File folder | |
| InClassHerokuExample | 2/2/2016 11:19 AM | File folder | |
| StudentWork | 1/29/2016 8:55 PM | File folder | |

## Step-by-step guide on creating Git Pull Requests

# Don’t Worry!

***We’ll be coming back to this.***

*You won’t need this fully until Week 8.*

# Practice At Home

***But practice when you can!***

*You don’t need a partner to submit pull requests!*