Going Live

The Coding Bootcamp

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.**

Remember to submit Homework via GitHub (& GitHub Pages)

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):**
GitLab

Re-Watch Class Videos:
https://bootcampspot-v2.com/

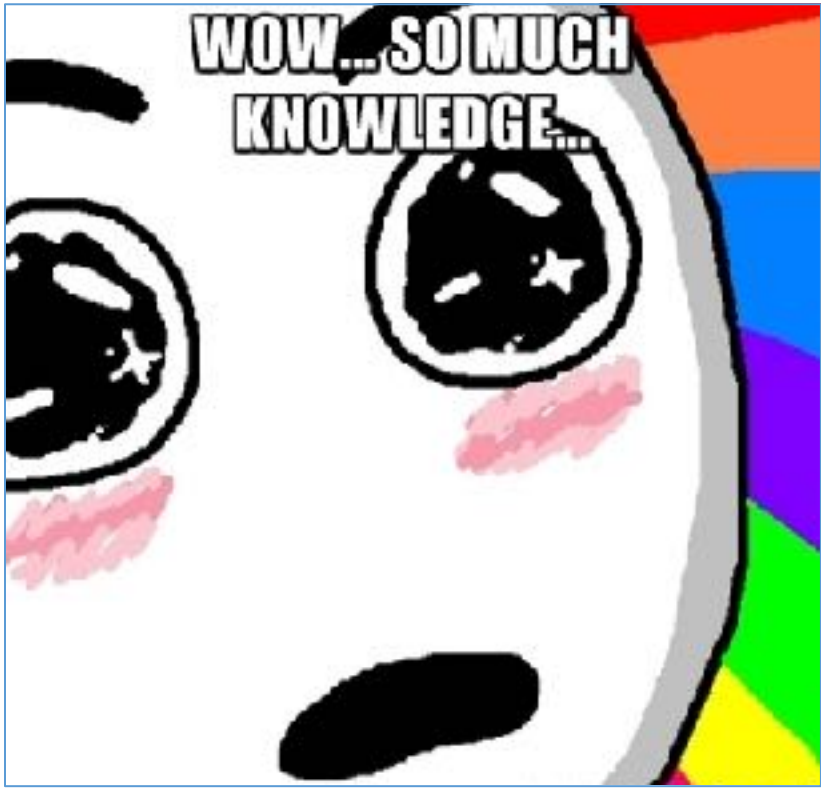
Recapping

Recap

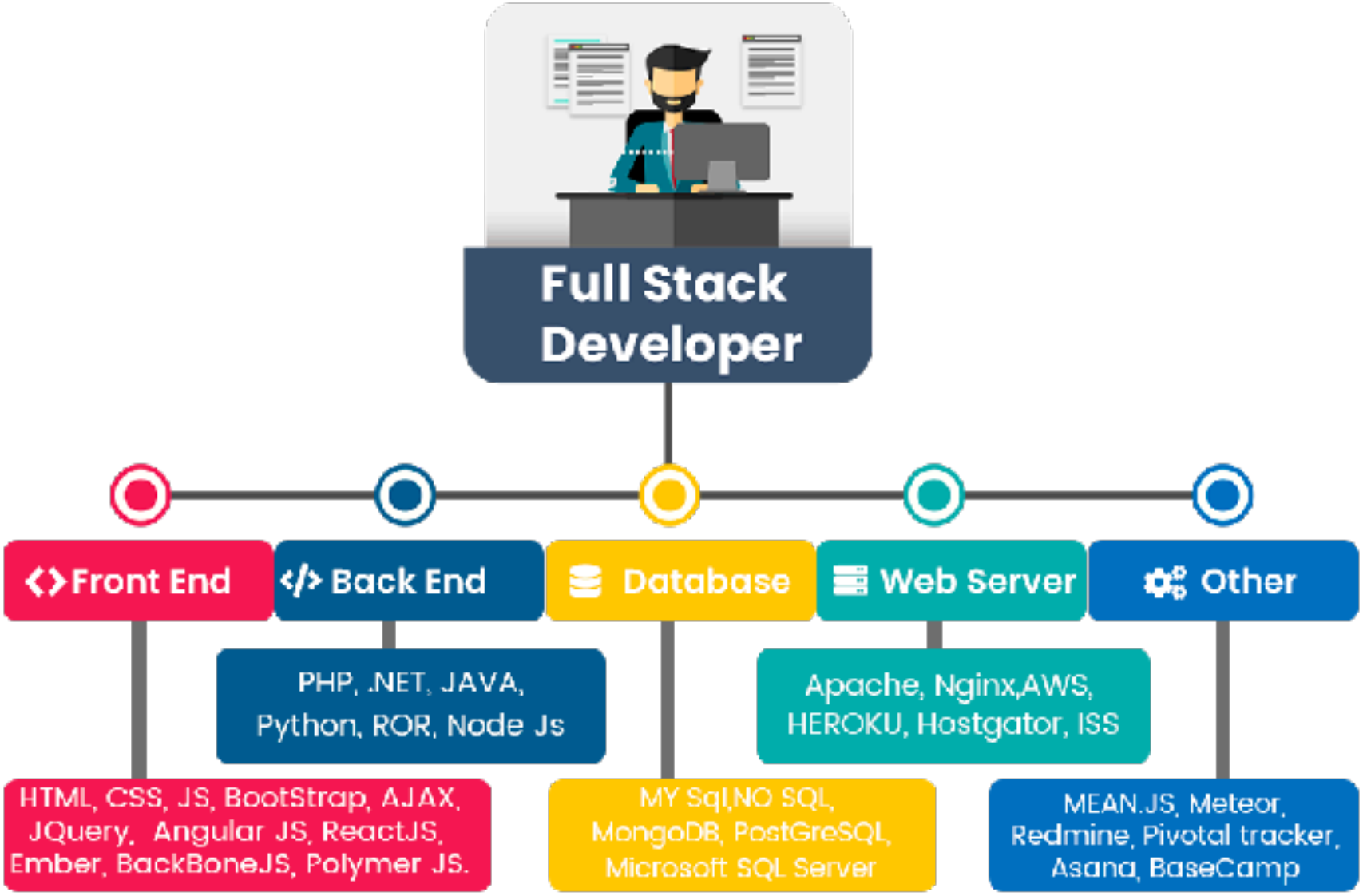
In just one whirlwind week we've covered:

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

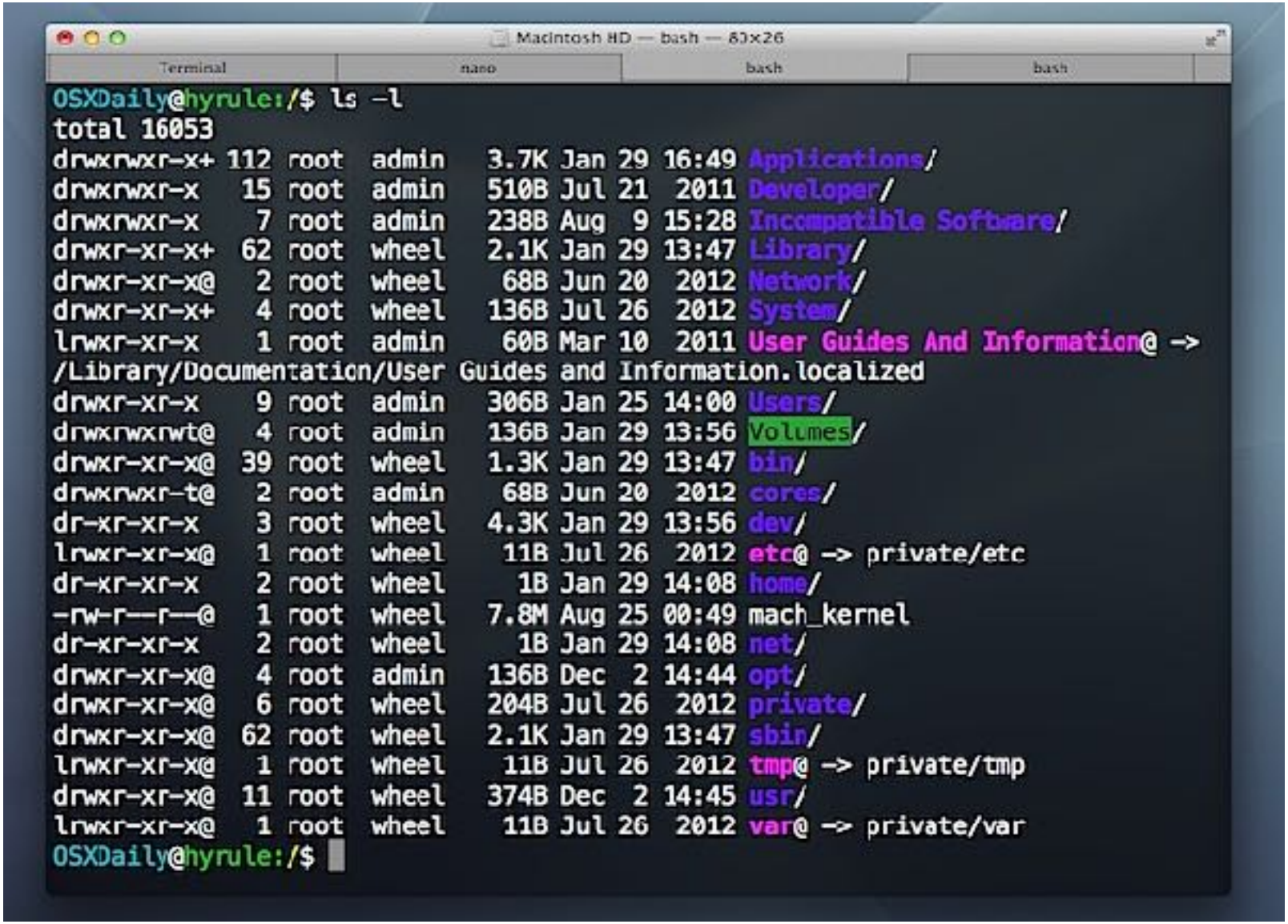
- **How to Learn on Your Own!!**



Full-Stack Development?



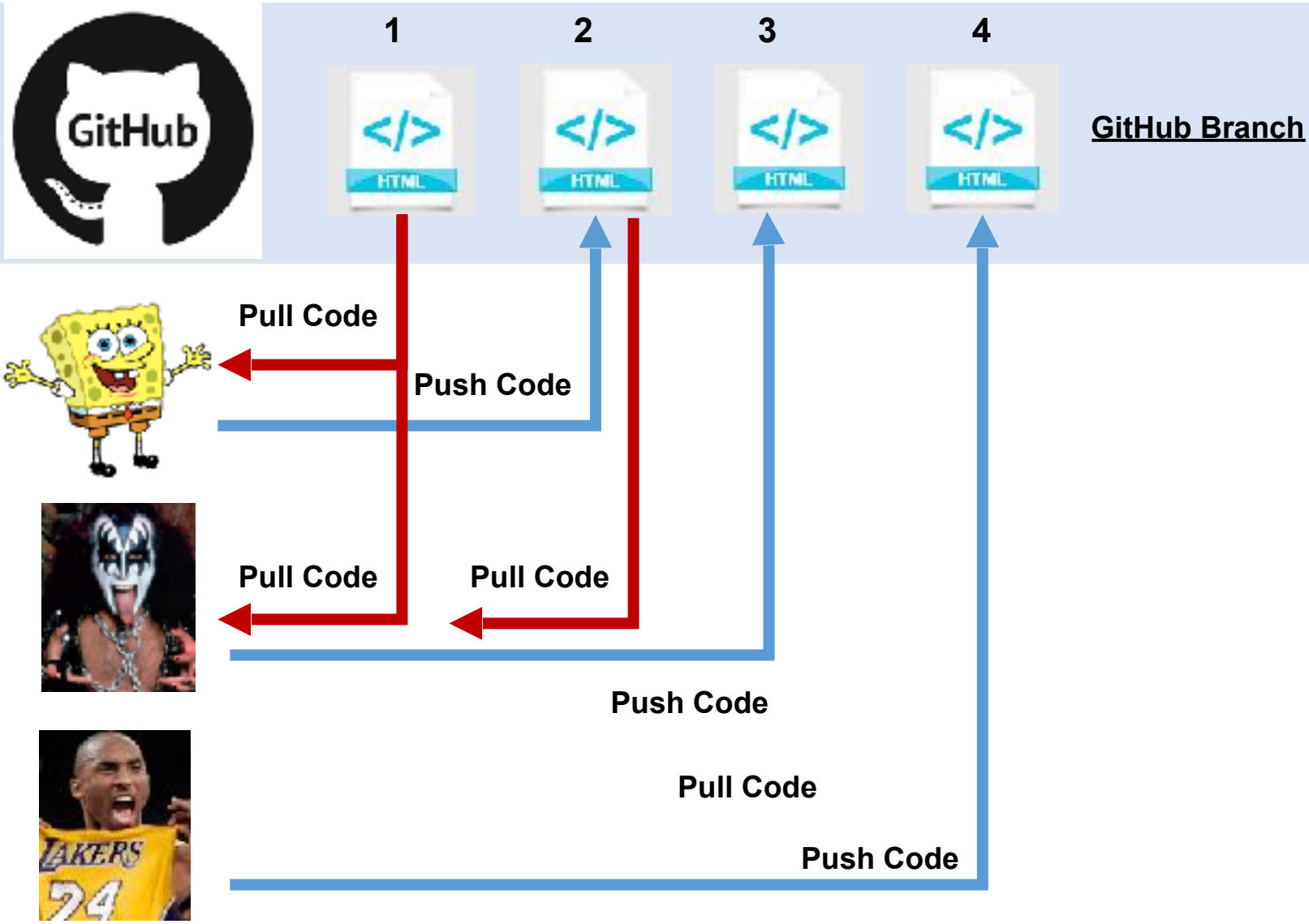
> Intro to Console

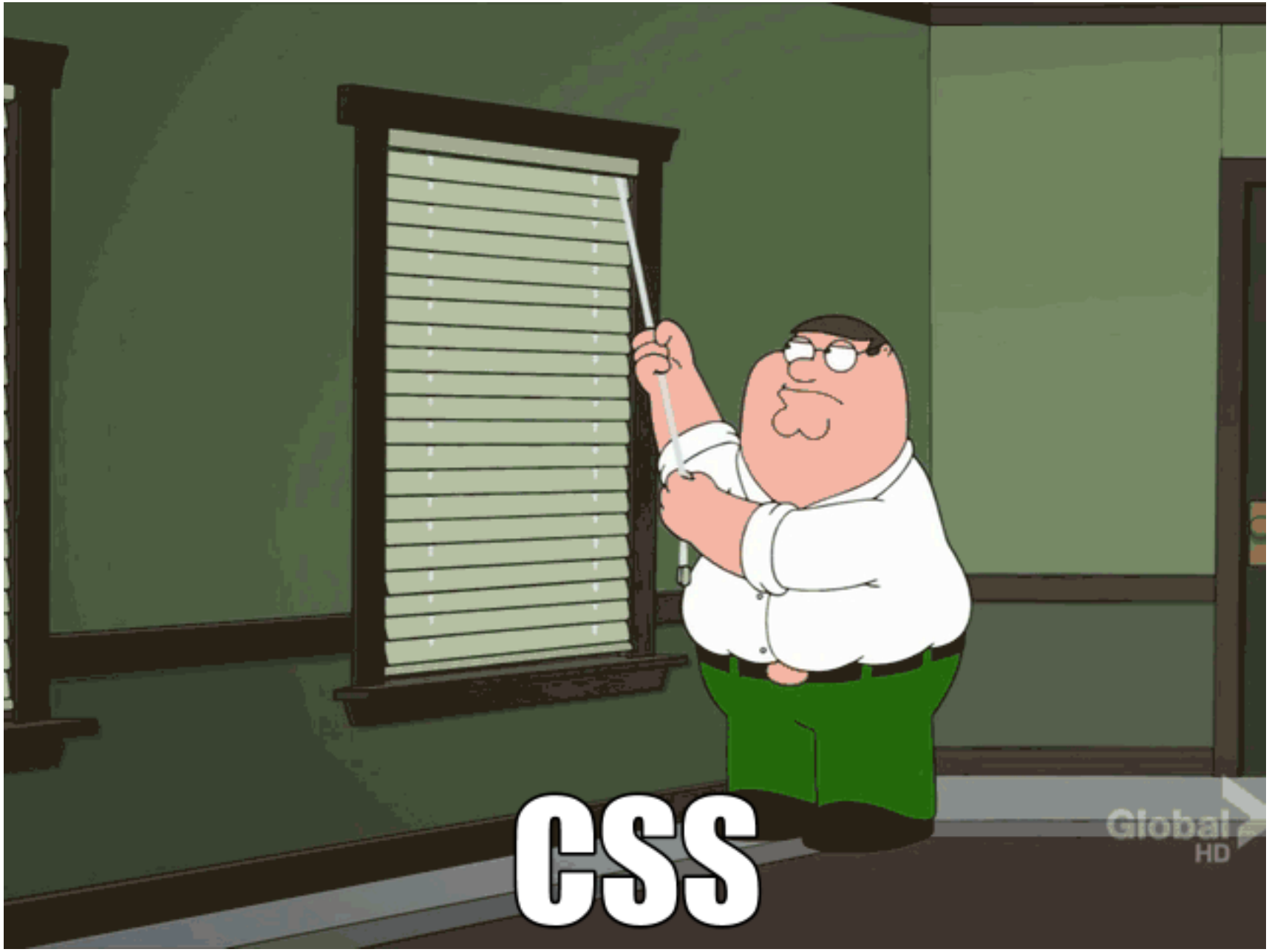


<title> Intro to HTML </title>

- **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

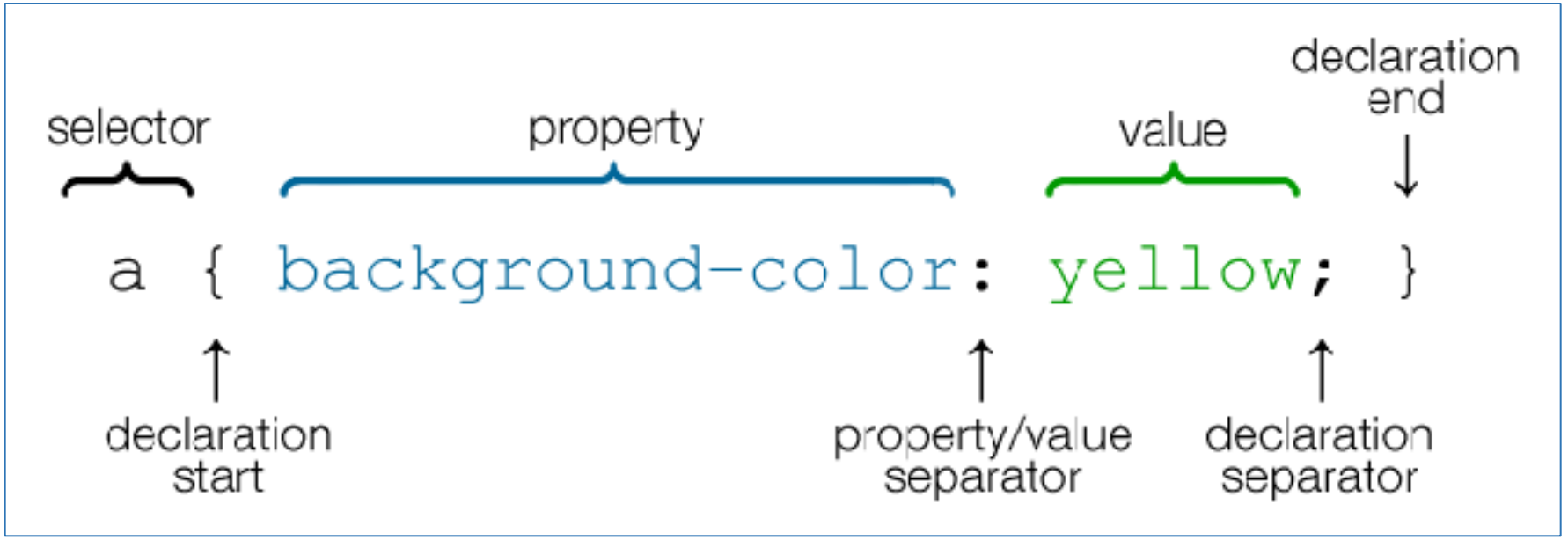



CSS
Global
HD

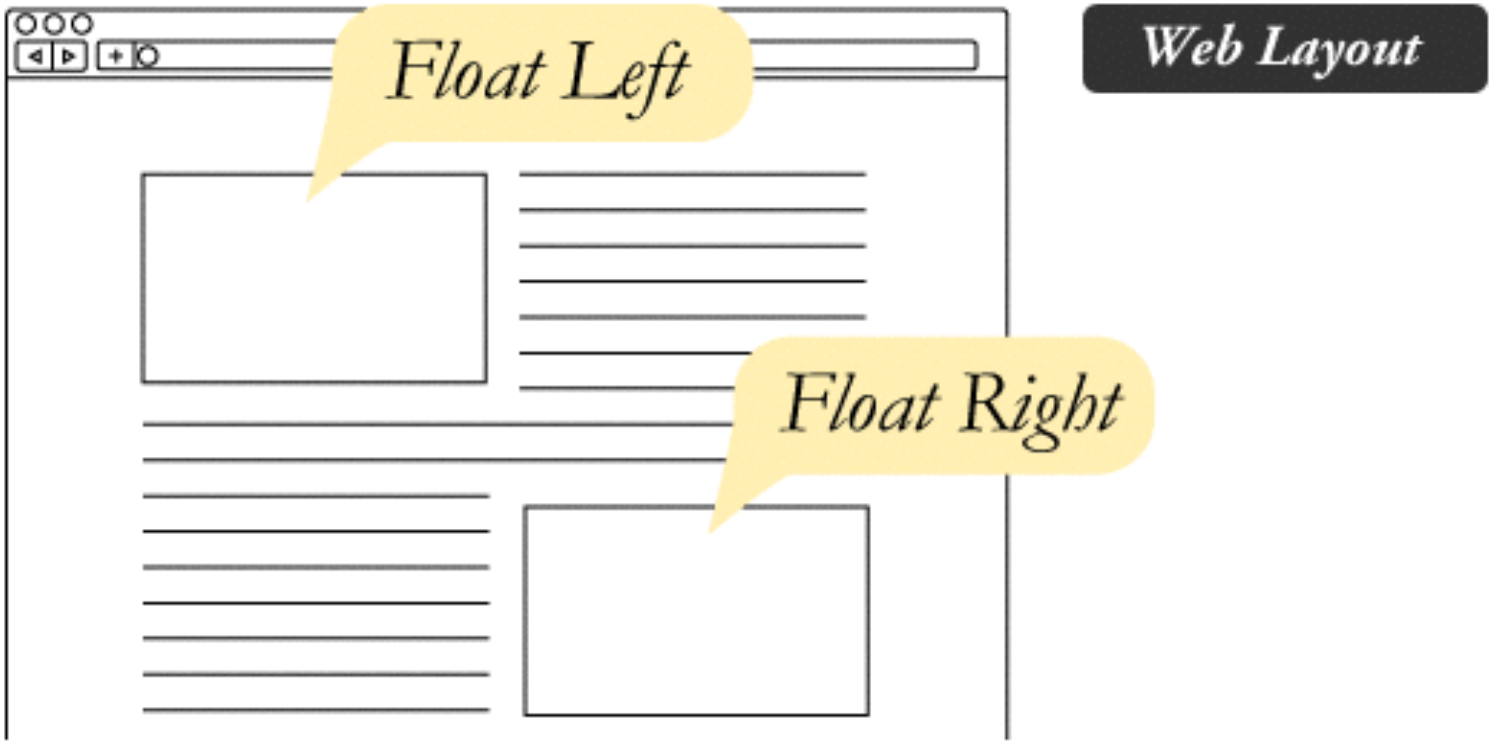
CSS Syntax

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

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



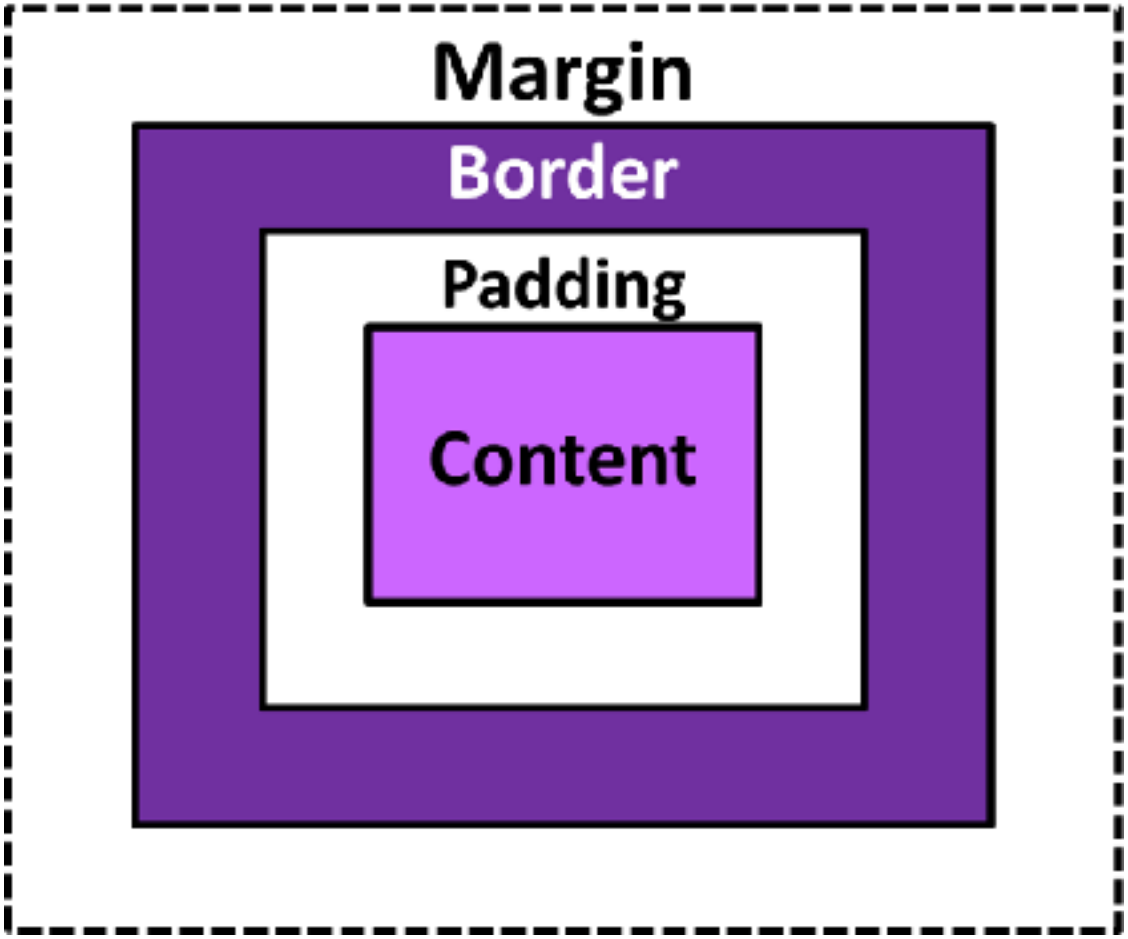
The Concept of “Flow”



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

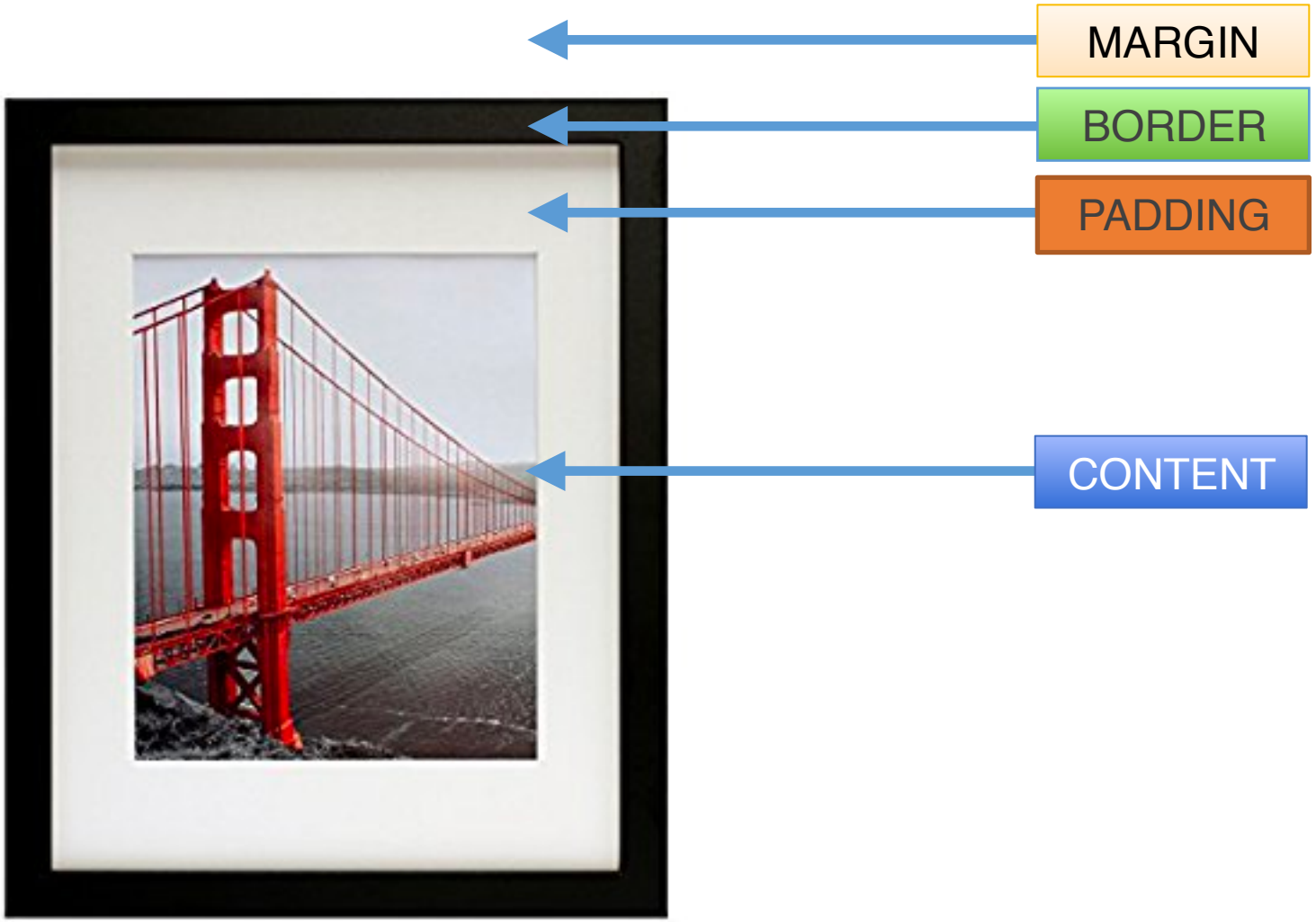
• This means that HTML elements force their adjacent elements to **flow around** them.

The Box Model

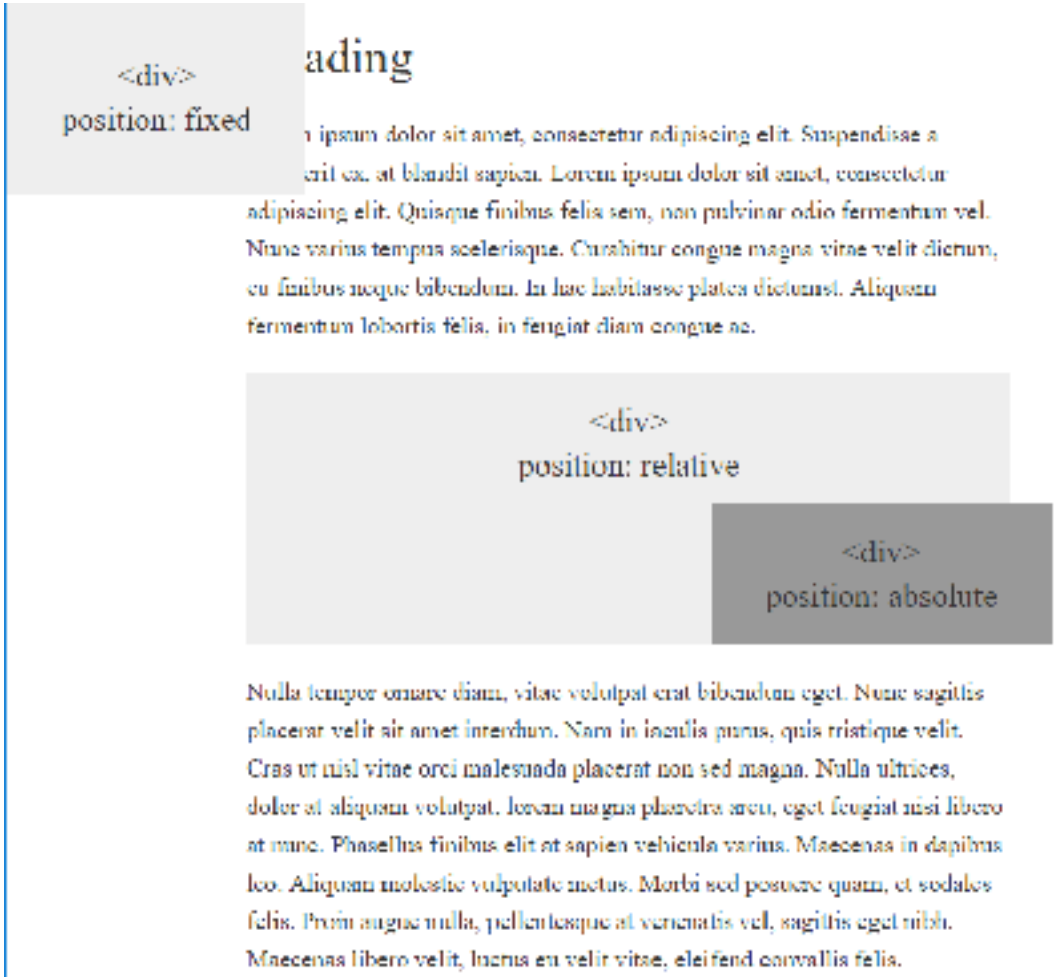


• The Box Model wraps every CSS element in **padding, border and margin** – allowing developers to modify spacing styles.

Chris' Painting Example



CSS Positioning



- 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 ?

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?

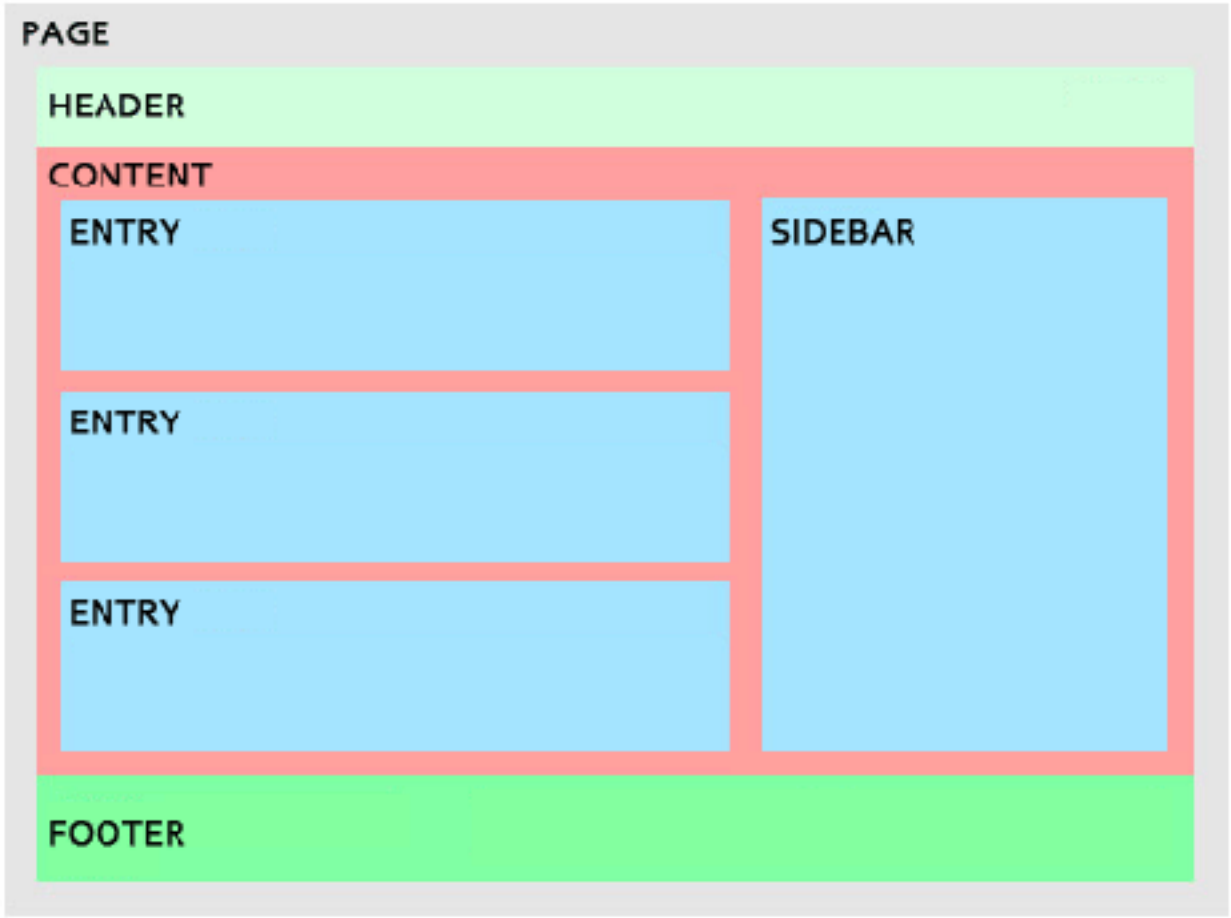
Need Help

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….

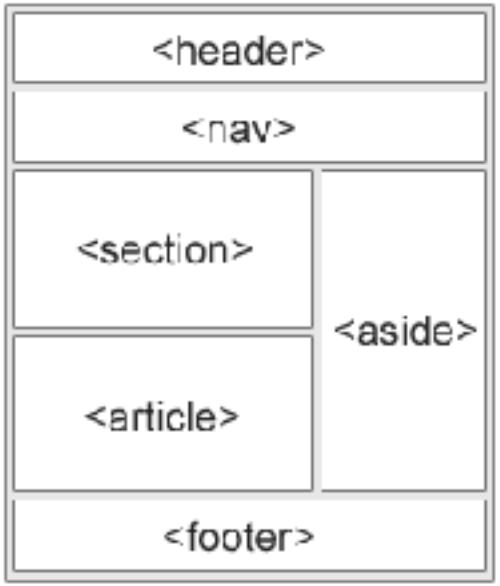


- All web layouts are inherently composed of containers, traditionally called “**divs.**”

Divs, Sections, Navs, Etc….

Website Layout Using HTML5

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

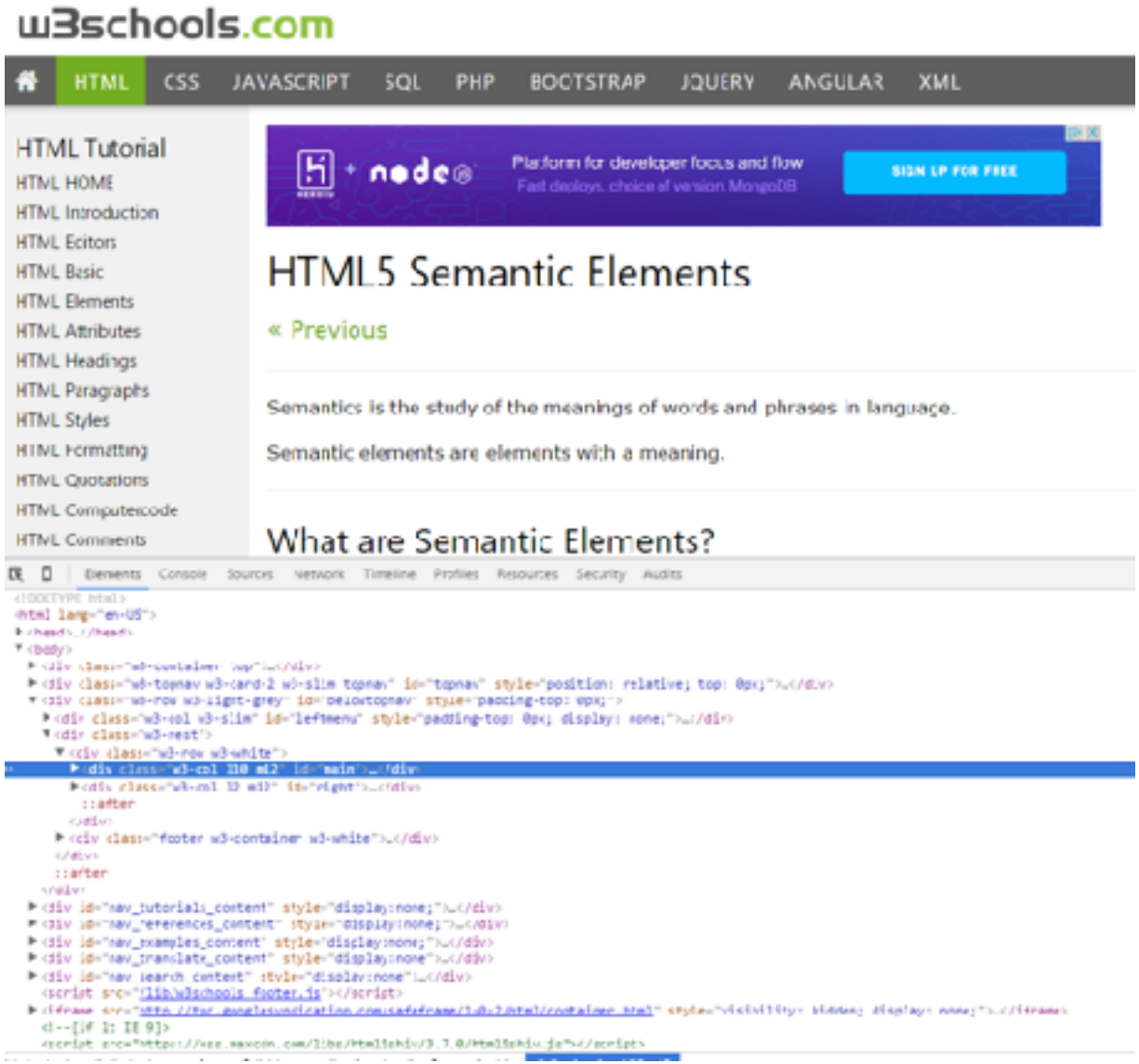


- <header> Defines a header for a document or a section
- <nav> - Defines a container for navigation links
- <section> Defines a section in a document
- <article> - Defines an independent self-contained article
- <aside> Defines content aside from the content (like a sidebar)
- <footer> - Defines a footer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



- 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:

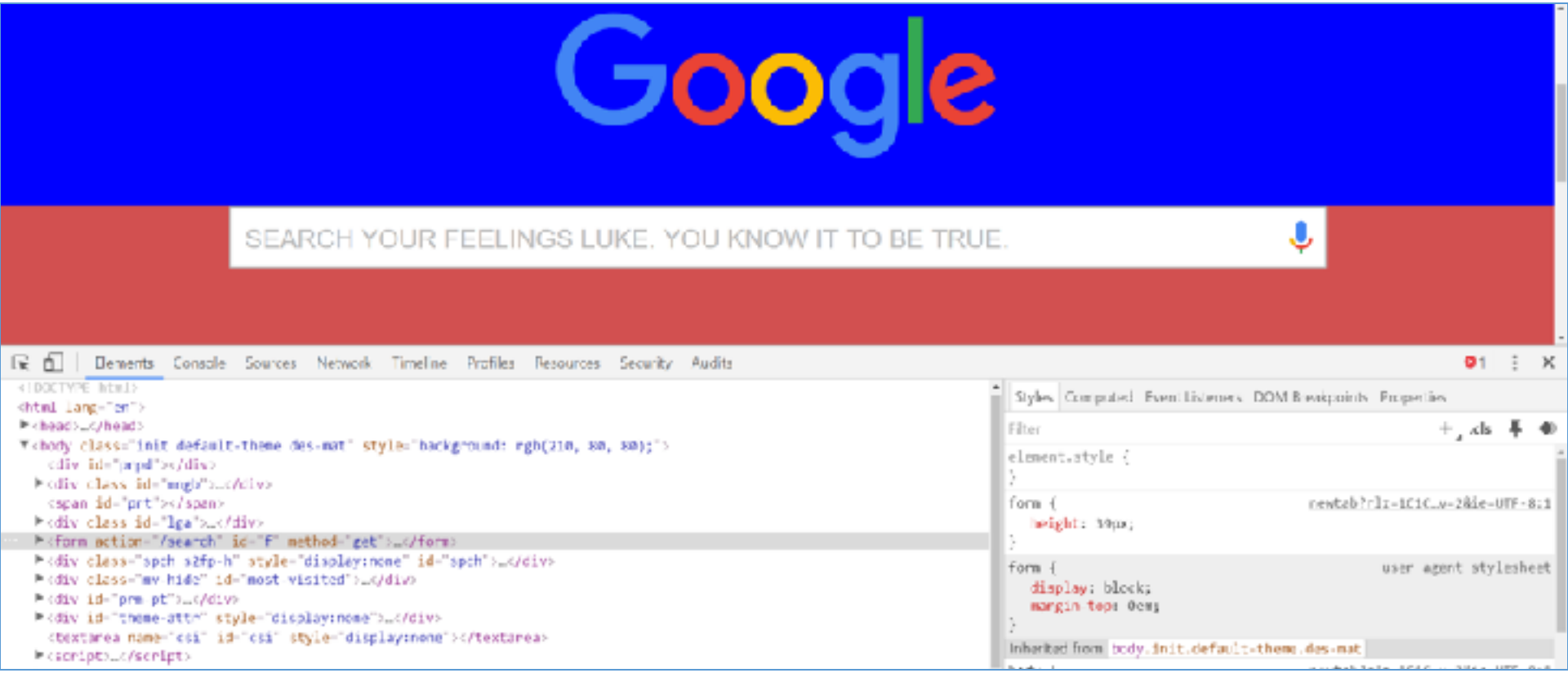
- **Classes (.classname)** are to be used if the same style will be used on multiple HTML elements.

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

Chrome Developer Tools (Inspector)

- This is one of the most frequent tools you will use in web development.
- It allows you to truly debug your web designs.
- **Start using it!**

Modifying Sites



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

INSTRUCTOR DEMO!

Instructor: Demo

(Chrome 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 Chrome 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 Chrome Developer Tools.

Be sure to at least modify:

- Content (Change words)
- Colors
- Spacing

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>
```

- 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 | 3-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)

The goal of a “reset” or “normalize” stylesheet is to reduce browser inconsistencies in things like default line heights, margins and font sizes of headings, etc!

- 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 | 4-ResetCSS)

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!!

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 GitHub!

The Internet

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

The World Will See Our Greatness!

GitHub provides hosting for static websites – which means we can deploy our websites and applications onto their servers for the world to see.

Together Now…

Let's all login to GitHub

INSTRUCTOR DEMO!

Instructor: Demo

(GitHub Pages Deployment - Personal)

Deploying Static Personal Site – GitHub Pages

Basic Steps:

1. Create a new repo that is named `_username_.github.io`
2. Navigate into a folder and clone the repo into it
3. Build your files
4. Add, commit, and push your changes into the repository

> YOUR TURN!!

Suggested Time: 15 min

Assignment

Time to take your newfangled website and deploy it to the cloud (in this case, GitHub Pages).

Additional instructions to be sent via Slack.

```
git clone https://github.com/username/username.github.io
cd username.github.io

git add –all
git commit -m "Initial commit"
git push -u origin master
```

INSTRUCTOR DEMO!

Instructor: Demo

(GitHub Pages Deployment - Project)

Deploying a Static Project Site – GitHub Pages

Basic Steps:

1. Create a new repository on your GitHub account. You can name this repository whatever you would like.

2. Once inside of the repository, create a new file and name it `index.html`

3. Add some very basic HTML into this file, save it, and then navigate into your repository's Settings tab.

4. Scroll down to the GitHub Pages section and then, in the section labeled "Source", select that you would like to use the master branch as your source.

5. Navigate to `<username>.github.io/<repositoryname>` and you will find that your new web page has gone live!

> YOUR TURN!!

Suggested Time: 15 min

Assignment

Build a newfangled website, and deploy it to GitHub Pages as a project instead of a personal site.

Additional instructions to be sent via Slack.

Keep Practicing!

It gets better.

Questions?

Homework 1 - Help?

EXTRA MATERIAL

And Back to Git…