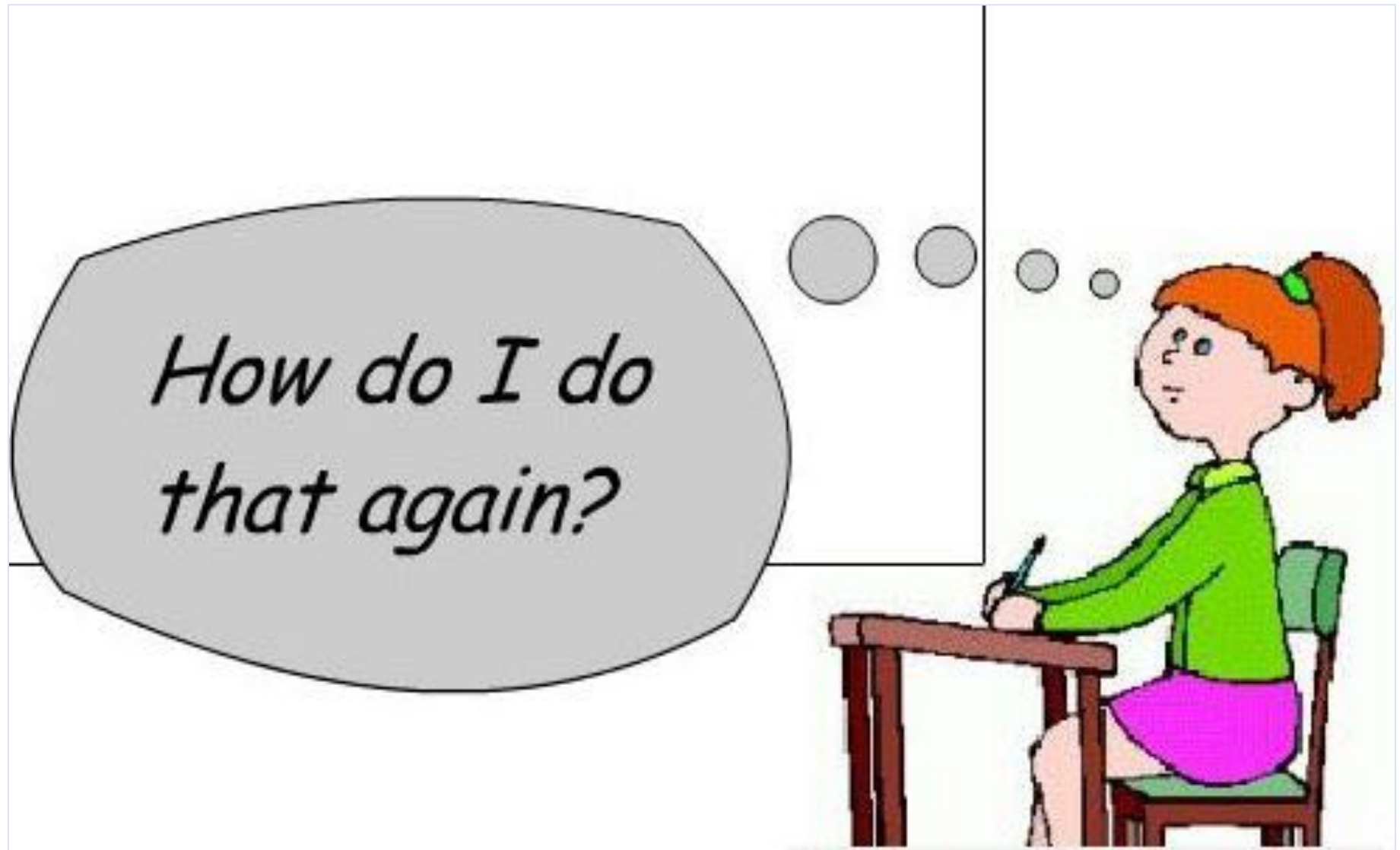


Day 2

Git'n Pro with HTML/CSS

The Coding Bootcamp |

It's Okay!



Admin Items

Today's Class!

Today's Objectives

- Students will understand the importance of Git Version Control and of how to use it.
- Students will create GitHub Repositories, push code into them, and share with class.
- Students will make more HTML documents.
- Students will learn to properly use basic HTML tags.
- Students will implement basic CSS styling to HTML documents.

Know Thyself

If you are a *complete* beginner to HTML/CSS and Coding:

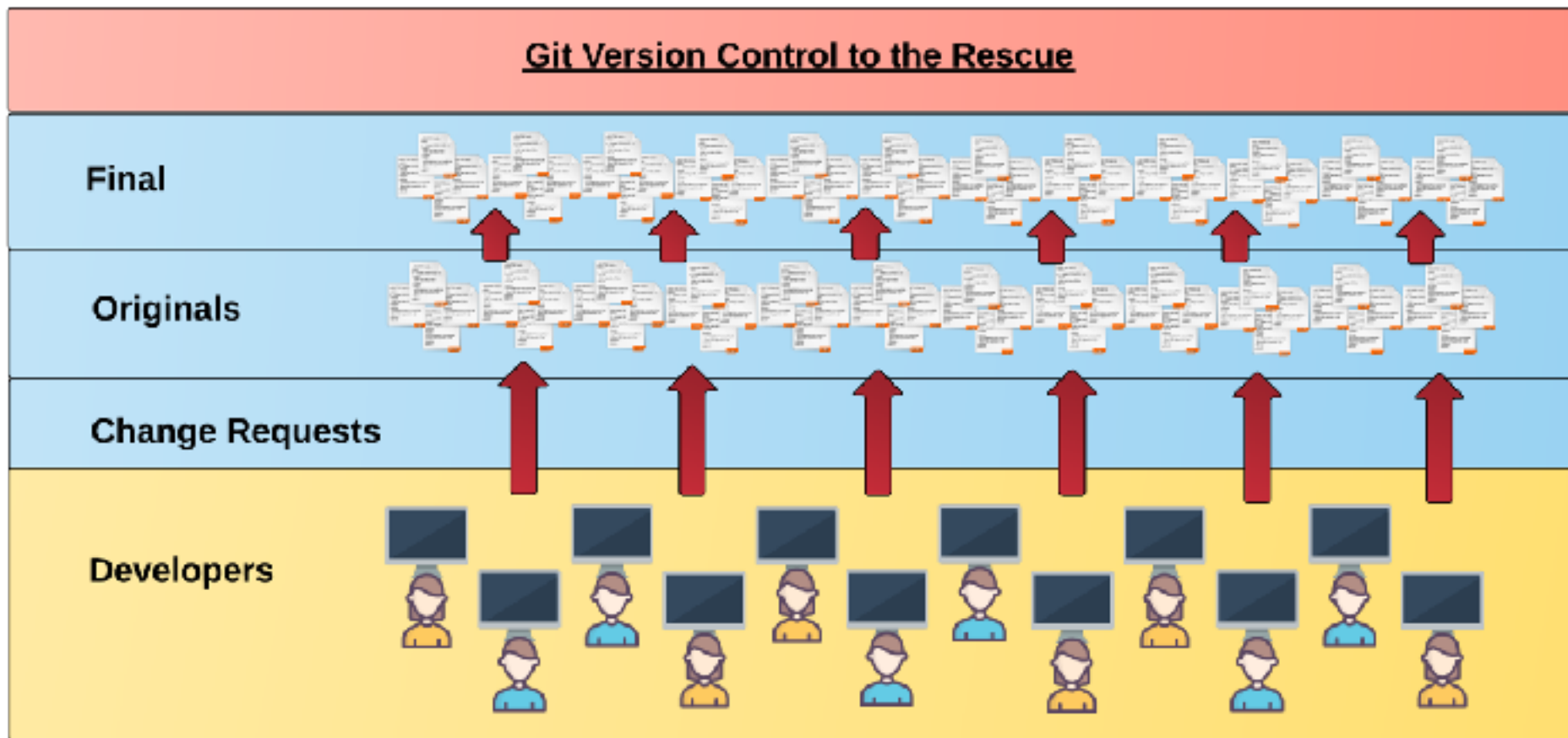
- Continue getting comfortable with HTML.
- Be able to completely write a basic HTML document (like in last class).
- Understand what CSS is, what it's for, and how it works with HTML.
- *Be able to use Git and GitHub to upload code.*

If you've had past exposure and felt comfortable with the last lesson:

- Aim to build up your skills. Clear up any questions or confusions about HTML.
- Become knowledgeable about a wider range of HTML and CSS tags.
- Be able to selectively apply CSS to specific HTML elements.
- *Be able to use Git and GitHub to upload code.*

What / Why Git?

Collaborative Coding



- Modern web development is highly collaborative.
- Teams are often extremely large and separated across the country — or planet.
- Apps sometimes comprise hundreds or even thousands of files.

The Team's Task

Task: Make a list in HTML showing the three branches of US government.

Programming Team:



SpongeBob & Kobe make their edits



Programming Away...

SpongeBob's Version



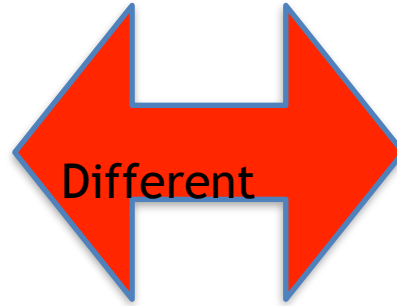
Programming Away...

Kobe's Version



Different Solutions

```
<ul>  
  <li>Legislative</li>  
  <li>Supreme Court</li>  
  <li>Executive</li>  
</ul>
```



```
<ul>  
  <li>Executive</li>  
  <li>Congress & Senate</li>  
  <li>Judicial</li>  
</ul>
```



Resolution



```
<ul>  
  <li>Legislative</li>  
  <li>Supreme Court</li>  
  <li>Executive</li>  
</ul>
```

“Let’s settle on this...”

```
<ul>  
  <li>Legislative</li>  
  <li>Judicial</li>  
  <li>Executive</li>  
</ul>
```



```
<ul>  
  <li>Executive</li>  
  <li>Congress & Senate</li>  
  <li>Judicial</li>  
</ul>
```

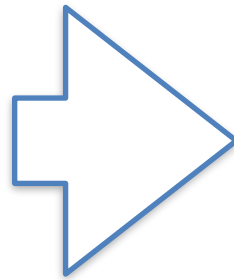


**Hai guyz!!!
How R Kan help??**



**Delete. Delete.
Delete. Delete.
Delete. Delete**

~~~~
~~Legislative~~
~~Judicial~~
~~Executive~~
~~~~



<list>
 Washington
 Dudes in Robes
 Mr. Hot Shot
</list>

The Group Project



Lesson:
You should use Version Control.

....and watch your teammates' work

Version Control

Git Version Control:

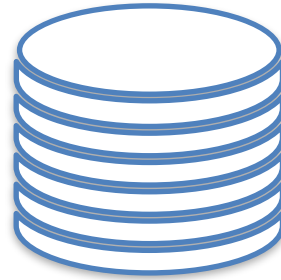
Provides a organized system for managing code for when multiple developers work on a project at the same time.

The Benefits of Git:

1. A process for resolving conflicts in code.
2. Version History.

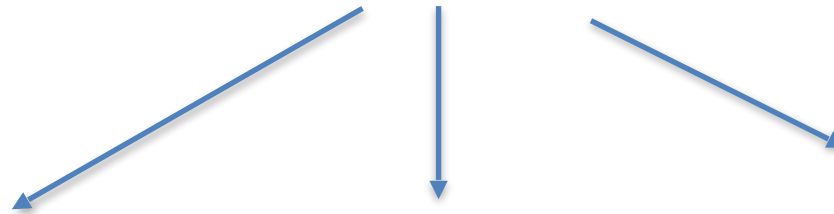
The Group Project

Master Branch



'Branch' = personal copy

Personal branch



Kobe's branch



Sponge Bob's branch



Kiss dude's branch

The team goes to work



```
<ul>  
  <li>Legislative</li>  
  <li>Supreme Court</li>  
  <li>Executive</li>  
</ul>
```



```
<ul>  
  <li>Executive</li>  
  <li>Congress & Senate</li>  
  <li>Judicial</li>  
</ul>
```

Sponge Bob pushes first

Master Copy



1



Sponge Bob **pushes (uploads)** his code changes into the main branch.

No code conflicts.



Sponge Bob's Branch



Kobe's edits are ready

Rule: pull first, then push your changes



Ok

Kobe pulls latest changes

Master Copy



1



Kobe's Branch

Kobe conflicts with master branch

Master Branch



1



Executive
Congress & Senate
Judicial
Legislative
Supreme Court
Executive

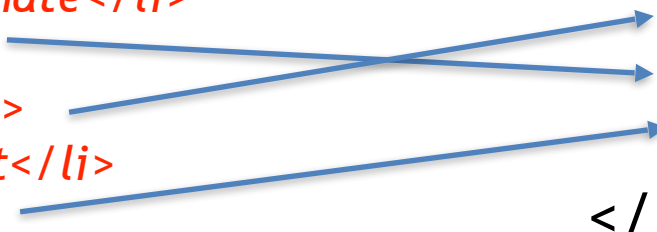


Git sees a conflict.

Kobe resolves

Executive
Congress & Senate
Judicial
Legislative
Supreme Court
Executive

``
`Legislative`
`Judicial`
`Executive`
``



Kobe's Branch

Kobe fixes and pushes

Master Branch



1



2

Kobe **pushes (uploads)** his revision the main branch.

No code conflicts.



Kobe's Branch

```
<ul>
  <li>Legislative</li>
  <li>Judicial</li>
  <li>Executive</li>
</ul>
```


Kiss Dude starts his work

Rule: pull first, then push your changes



Rules R 4 suckers!!!

Kiss dude pushes

Master Branch



1



2



3

Kiss dude **pushes (uploads)** his revision the main branch. No code conflicts.

Not what we want.



Kiss dude's Branch

<list>

Washington

Dudes in Robes/li>

Mr. Hot Shot

</list>

If Kiss dude had made a pull first...

Conflict!

```
<list>  
  <li>Washington  
  <li>Dudes in Robes/li>  
  <li>Mr. Hot Shot<li>  
</list>  
<ul>  
  <li>Legislative</li>  
  <li>Judicial</li>  
  <li>Executive</li>  
</ul>
```

The overwritten work is discovered



Git GUI interface showing a commit history and a diff view.

Commit History:

Graph	Description	Commit	Author	Date
	Uncommitted changes	*		Today, 3:04 PM
	Conflict experiment: slight e...	0b1e0a	Greg Sandell <g...>	Today, 2:56 PM
	Conflict experiment: restore rogue...	1199e0	Greg Sandell <g...>	Today, 2:56 PM
	Conflict experiment: undo rogue changes	2feb29d	Greg Sandell <g...>	Today, 2:53 PM

Diff View (test.html):

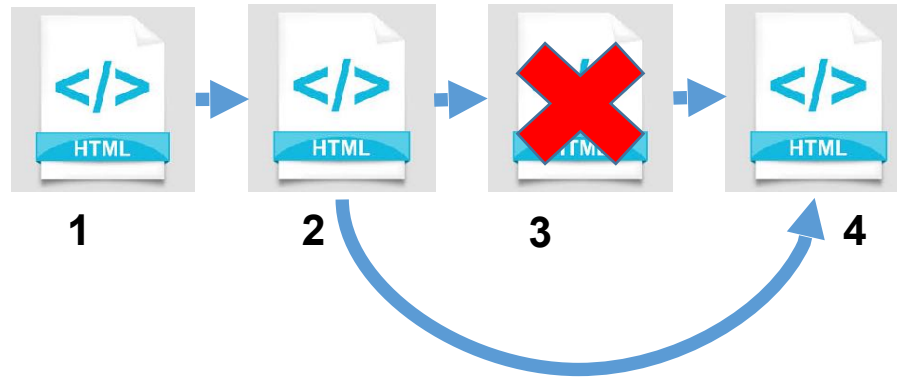
Hunk 1 : Lines 1-10 | Reverse hunk

```
1 1 <idestype.html>
2 2 <html>
3 3 <body>
4 4 <ul>
5 5 <li>Legislative</li>
6 6 <li>Judicial</li>
7 7 <li>Executive</li>
8 8 </ul>
9 9 <list>
10 10 <li>Washington
11 11 <li>Gades in Robes</li>
12 12 <li>Mr. Not Shot</li>
13 13 </list>
14 14 </body>
15 15 </html>
```

Roll Back



Main Branch

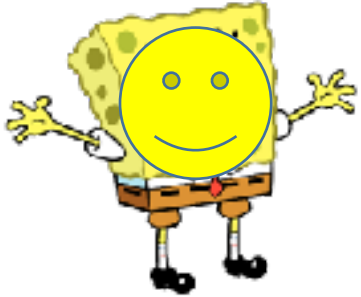


SpongeBob **rolls back** the code to an earlier version.



Kiss Dude's Branch

The Group Project



Lesson:
You should use Version Control!

Turn to your neighbor, and have one of you explain to the other:

- **The concept of version control.**

Then the other should explain:

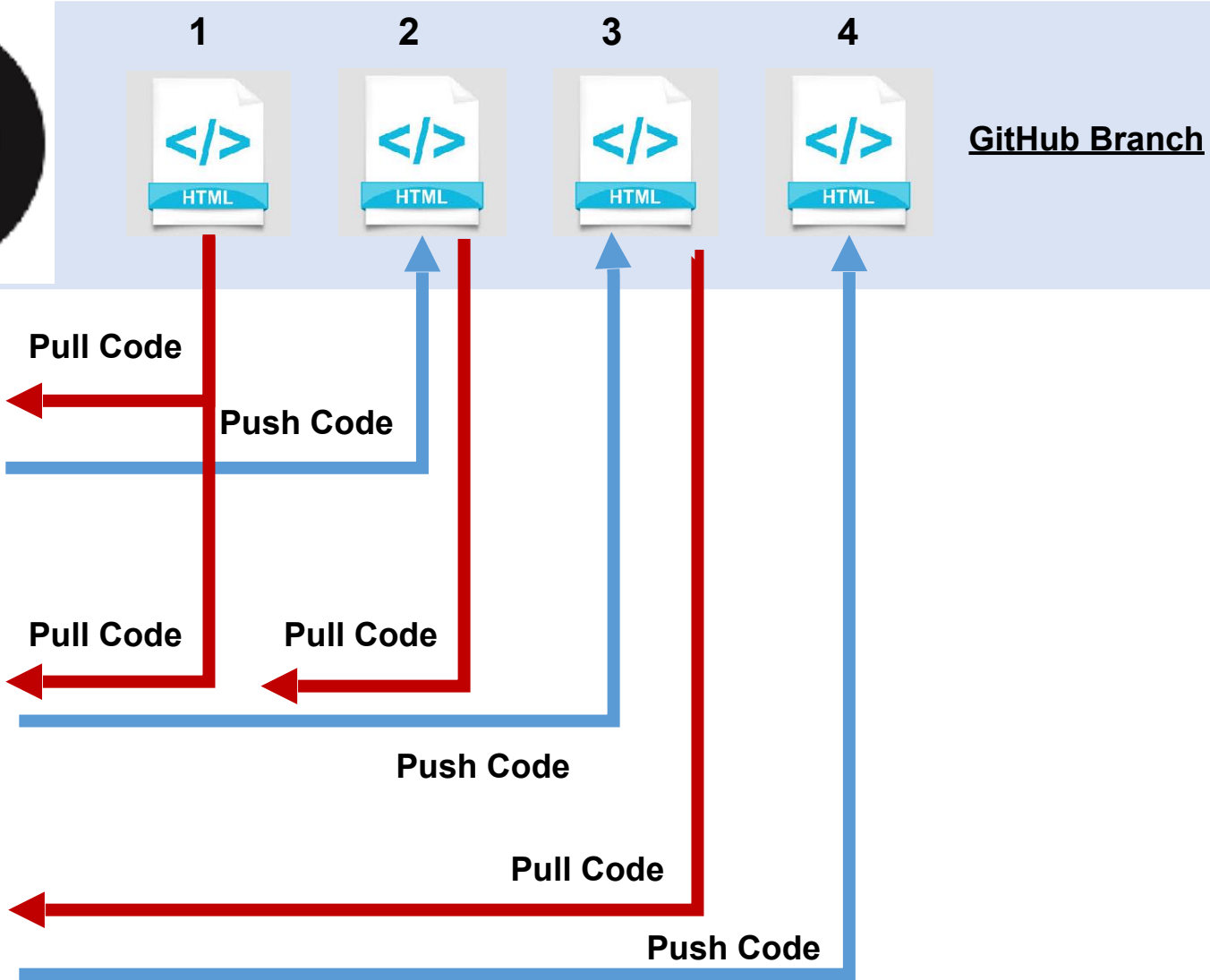
- **Two of the key advantages to using a version control system.**

So... What's this GitHub?

- GitHub is a Web-Based hosting service to store code online.
- It allows developers to **pull** (download) code or **push** (upload) code to the same **repository** (directory).
- It also allows developers to **view histories** of code changes and to **track issues**.



Pushing and Pulling to GitHub



Get Started with Git

Instructor Git Demo!

The screenshot shows the GitHub interface for a repository named 'DemoRepository' by user 'afhaque'. At the top, there's a navigation bar with 'Pull requests', 'Issues', and 'Gist' links. Below this, the repository name is displayed along with 'Watch', 'Star', and 'Fork' buttons. A secondary navigation bar includes 'Code', 'Issues', 'Pull requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The main content area features a description: 'This is Ahmed's Demo repository for his class!'. Below the description, statistics show '1 commit', '1 branch', '0 releases', and '1 contributor'. Action buttons include 'New pull request', 'New file', 'Find file', 'HTTPS', a clone button, and 'Download ZIP'. A commit history section shows an 'Initial commit' by 'afhaque' with a link to the commit details. Below this, a file list shows 'README.md' as the only file. At the bottom, a large heading 'DemoRepository' is followed by the same description text.

This repository Search

Pull requests Issues Gist

afhaque / DemoRepository

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

This is Ahmed's Demo repository for his class! — Edit

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request

New file Find file HTTPS <https://github.com/afhaque> Download ZIP

afhaque Initial commit Latest commit 2d460aa 4 minutes ago

README.md Initial commit 4 minutes ago

README.md

DemoRepository

This is Ahmed's Demo repository for his class!

Basic Git Commands

At its most basic, these are the five git commands to get started:

1. **git clone**
2. **git add**
3. **git commit**
4. **git push**
5. **git pull**

Basic Git Commands

At its most basic, these are the five git commands to get started:

1. **git clone** – copies an entire repo (to begin).
2. **git add** – adds a file for inclusion in Git.
3. **git commit** – notes a change to the local repo.
4. **git push** – sends changes to hosting service.
5. **git pull** – downloads freshest version of repo.

> YOUR TURN! **Activity:** Git Add, Commit, Push | **Suggested Time:** 20 min

Assignment:

Using GitHub and the Command Line:

- Create a new **public GitHub repository** and name it whatever you like. Be sure to check the box for “initialize this repository with a README.”
- Next, **clone** the repo to your local directory.
- Then create an HTML file inside the local directory.
- **Add, Commit, and Push** the code to GitHub.

Bonus:

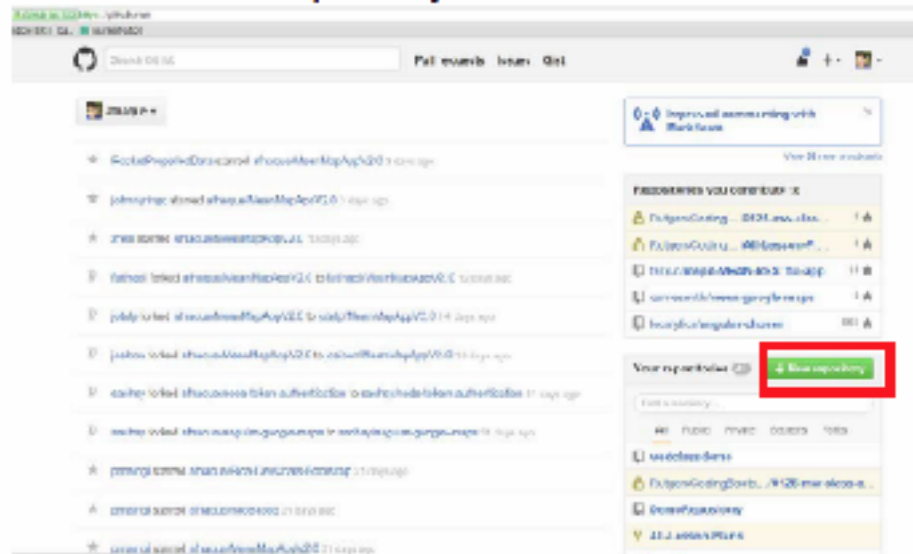
- Find a partner in class, and **fork their** repository to your own GitHub account. Clone this forked repository to your local directory.
- Add, Commit, and Push the code back to your forked copy.
- Finally, submit a **pull request** to send your changes to your partner’s repo.

Still a Bit Lost? Never Worry!

Steps to Uploading Your Code to GitHub

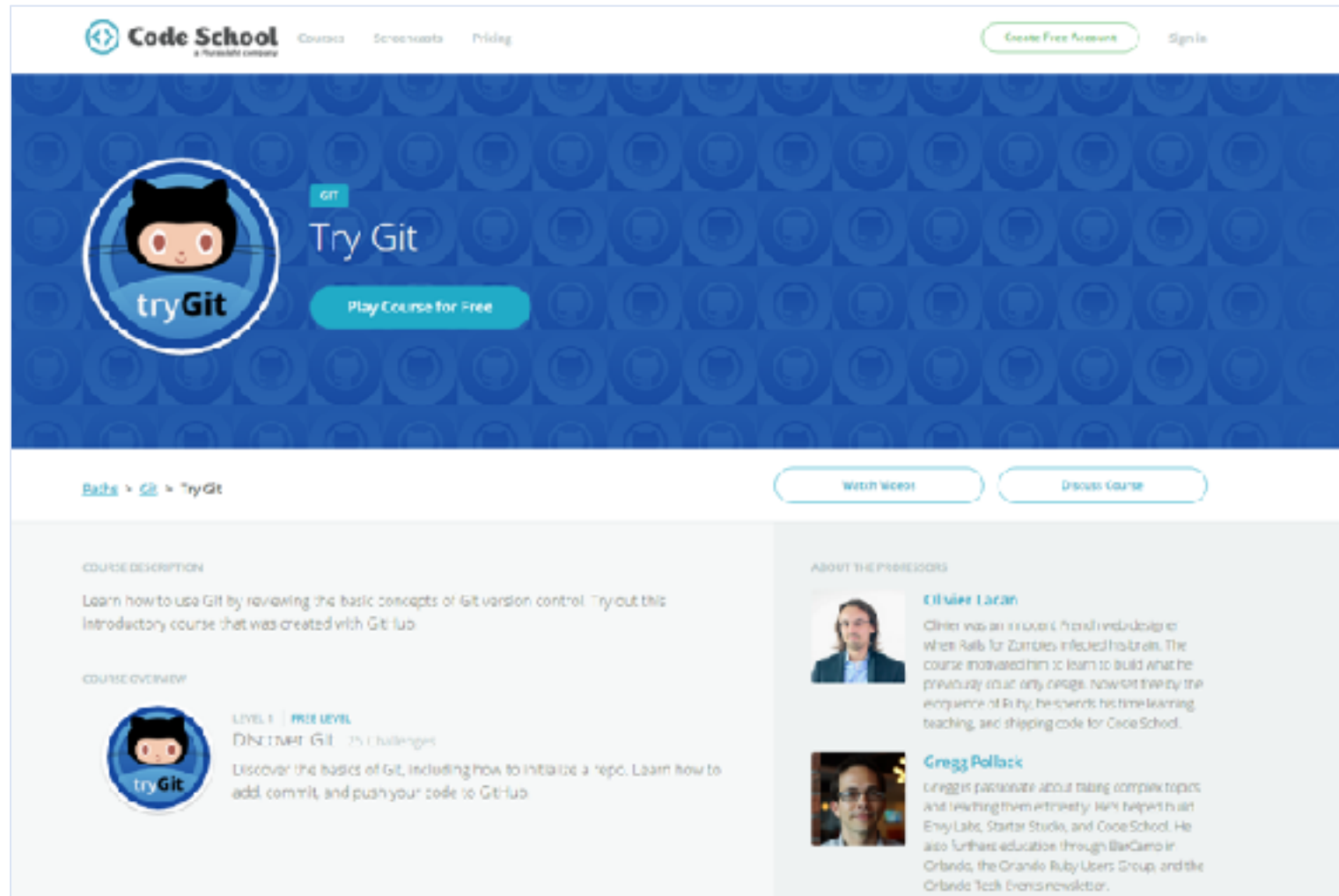
Step 1

Create a New Repository in GitHub.com



- Follow this handy Guide!
- Practice a few times on your own before our next class.

If You're Still Lost... Here's a (Free) Course



Code School
a Frontend Collective

Courses Screencasts Pricing

Create Free Account Sign In

Try Git

Play Course for Free


[Baths](#) > [Git](#) > [Try Git](#)

[Watch Videos](#) [Discuss Course](#)

COURSE DESCRIPTION


Learn how to use Git by reviewing the basic concepts of Git version control. Try out this introductory course that was created with GitHub.


COURSE OVERVIEW

 **LEVEL 1 | FREE LEVEL**
DURATION: Git 75 Challenges

Discover the basics of Git, including how to initialize a repo. Learn how to add, commit, and push your code to GitHub.

ABOUT THE PROFESSORS

 **Eli Sizer Larian**
Eli was an ex-Ruby Friend and designer when Rails for Zombies infected his brain. The course motivated him to learn to build what he previously could only design. Now, at twenty-three years old, he spends his time learning, teaching, and shipping code for Code School.

 **Gregg Pollack**
Gregg is passionate about taking complex topics and teaching them effectively. He's helped build Envy Labs, Starter Studio, and Code School. He also furthers education through BarCamp in Orlando, the Orlando Ruby Users Group, and the Orlando Tech Events newsletter.

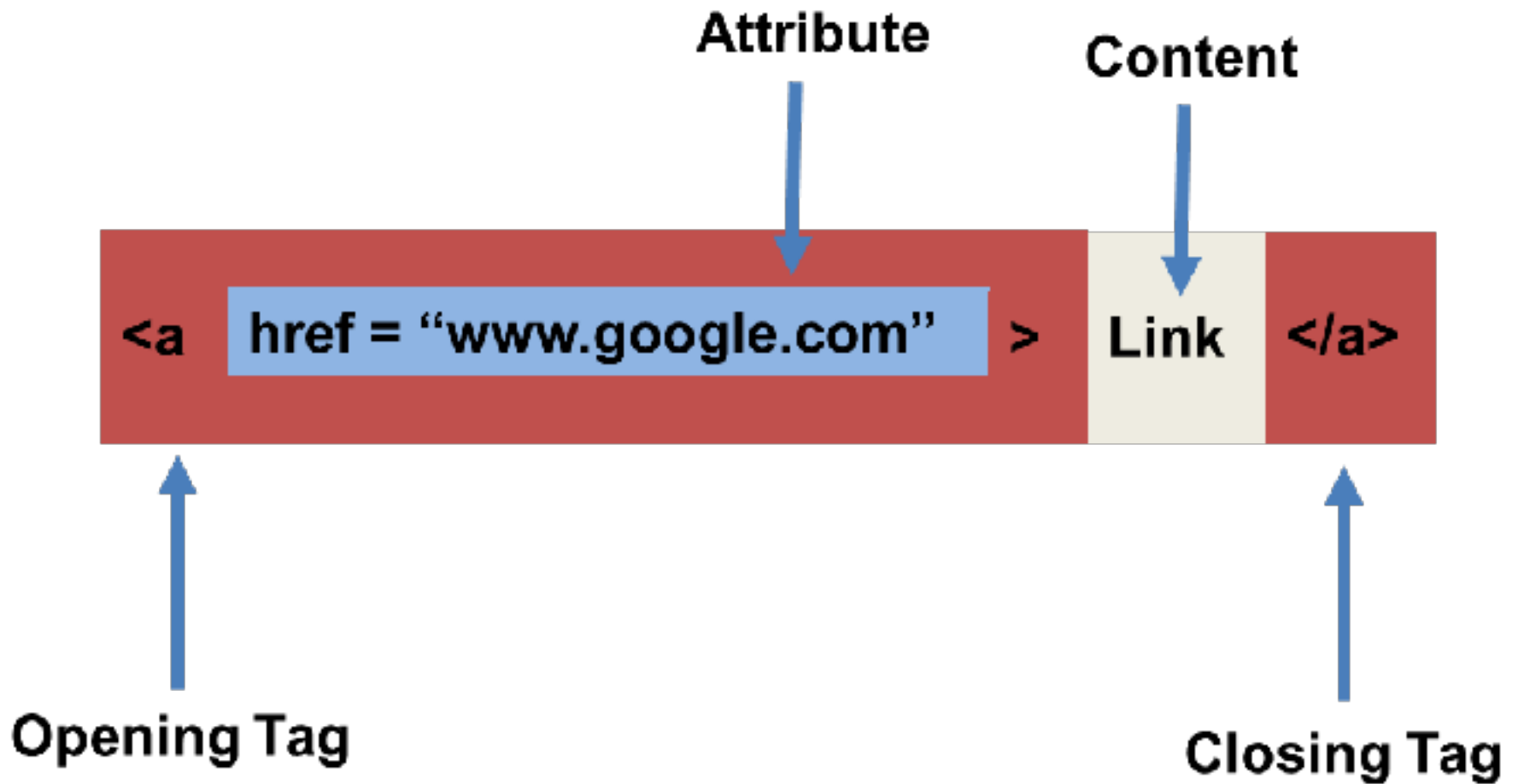
<https://www.codeschool.com/courses/try-git>

HTML Round 2

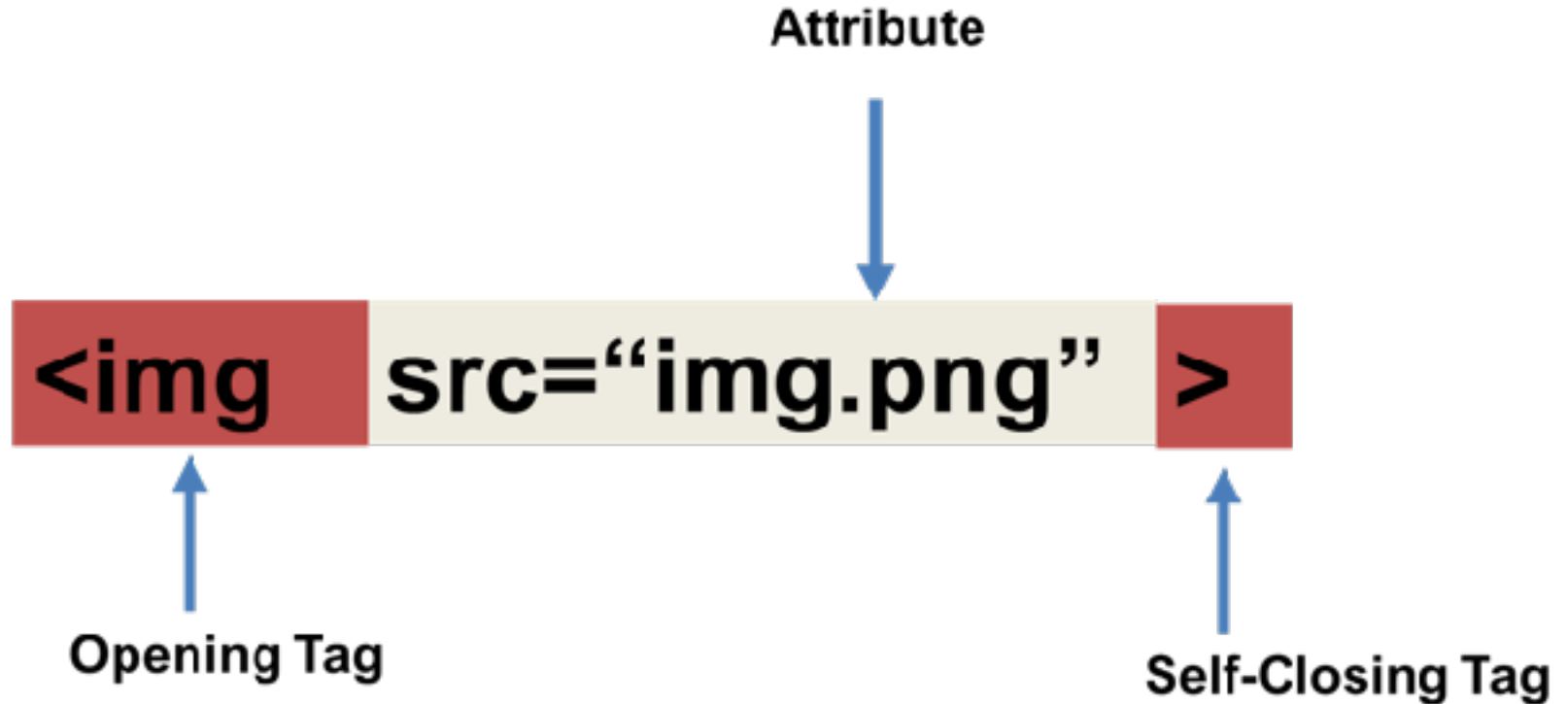
HTML Syntax (Basic)



HTML Syntax (with Attribute)



Tricky Tags (Self-Closing)



Important Common Tags

Headings:

- `<h1> </h1>` - Heading 1 (Largest Heading)
- `<h2> </h2>` - Heading 2 (Next Largest Heading)
- `<h3> </h3>` - Heading 3
- ...

Containers:

- `<html> </html>` - Wraps the entire page
- `<head> </head>` - Wraps the header of the page
- `<body> </body>` - Wraps the main content
- `<div> </div>` - Logical Container ***
- `<p> </p>` - Wraps individual Paragraphs

Others:

- `` (bold), `` (emphasis)
- `` (images), `<a href>` (links), `` (list items) , `<title>` (title), `
` (line break), `<table>` (tables), `<!-- -->` (comments)

Less Common Tags

- All HTML Tags are listed here: <http://www.w3schools.com/tags/>
- Don't try to memorize them! Simply refer back to documentation as needed.
- Other tags:
 - <video> for Videos
 - <audio> for Audio files
 - <embed> for Embedded files
 - <code> for including computer code
 - <header> for headers
 - <nav> for navigation bars
 - <footer> for footers

HTML for Forms

Common UI (User Interface) Form Elements:

- **<form>** - Creates a form section in HTML
- **<input>** - Input boxes
- **<label>** - Labels for boxes
- **<button>** - Button
- **<textarea>** - Large textbox

HTML for Forms

```
<!DOCTYPE html>
<html>
<body>

<form>
  First name:<br>
  <input type="text" name="firstname">
  <br>
  Last name:<br>
  <input type="text" name="lastname">
</form>

<p>Note that the form itself is not visible.</p>

<p>Also note that the default width of a text input field is 20 characters.</p>

</body>
</html>
```



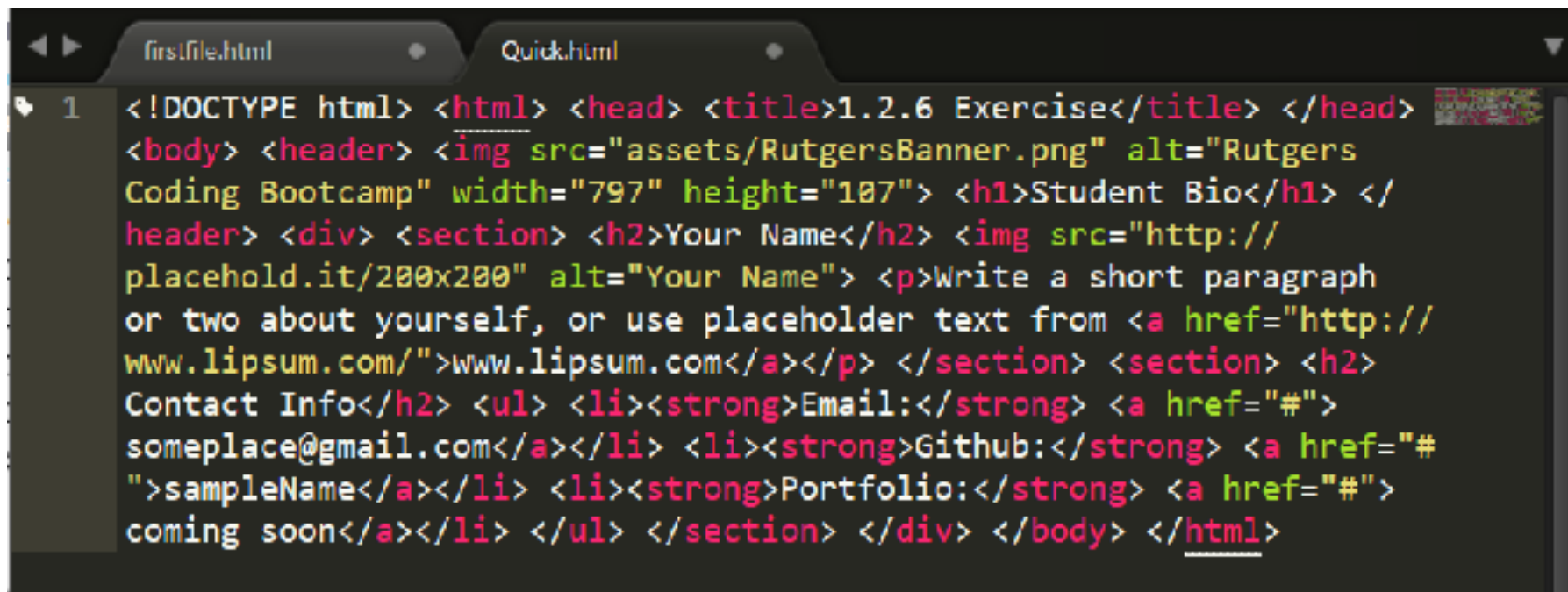
First name:

Last name:

Note that the form itself is not visible.

Also note that the default width of a text input field is 20 characters

On Ugly HTML

A screenshot of a code editor with two tabs: 'firstfile.html' and 'Quick.html'. The 'Quick.html' tab is active, showing HTML code with poor formatting. The code is written in a monospaced font with syntax highlighting. It starts with a DOCTYPE declaration, followed by a single-line <html> tag. The <head> section contains a <title> tag. The <body> section contains a <header> with an tag and an <h1> tag. This is followed by a <div> containing a <section> with an <h2> tag, an tag, and a <p> tag with an <a> tag. This is followed by another <section> with an <h2> tag, a tag with three tags, each containing a tag and an <a> tag. The code ends with </div>, </body>, and </html> tags. The code is not indented, making it difficult to read and maintain.

```
1 <!DOCTYPE html> <html> <head> <title>1.2.6 Exercise</title> </head>
  <body> <header>  <h1>Student Bio</h1> </
header> <div> <section> <h2>Your Name</h2>  <p>Write a short paragraph
or two about yourself, or use placeholder text from <a href="http://
www.lipsum.com/">www.lipsum.com</a></p> </section> <section> <h2>
Contact Info</h2> <ul> <li><strong>Email:</strong> <a href="#">
someplace@gmail.com</a></li> <li><strong>Github:</strong> <a href="#"
">sampleName</a></li> <li><strong>Portfolio:</strong> <a href="#">
coming soon</a></li> </ul> </section> </div> </body> </html>
```

- Don't do this... Use proper indentation and sectioning.
- Readable code is easier to maintain.
- Invest time to get better about this now. It will pay dividends!

Assignment

In this activity, you'll create a student bio using HTML. You will then add, commit, and push your completed HTML to GitHub for the world to see.

Additional instructions, sent via Slack.

> YOUR TURN!

Student Bio

Your Name



Write a short paragraph or two about yourself, or use placeholder text from www.lipsum.com

Contact Info

- Email: someplace@gmail.com
- Github: [sampleName](#)
- Portfolio: [coming soon](#)

CSS Stylin'

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

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>My First Website!</title>
6 </head>
7 <body>
8
9   <h1>Awesome Header</h1>
10   <h2>Smaller Awesome Header</h2>
11   <h3>Even Smaller Header</h3>
12
13   <p>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?</p>
14   
15
16   <h3>Menu Links</h3>
17   <ul>
18     <li><a href="http://www.google.com"></a>Google</li>
19     <li><a href="http://www.facebook.com"></a>Facebook</li>
20     <li><a href="http://www.twitter.com"></a>Twitter</li>
21   </ul>
22
23 </body>
24 </html>
```

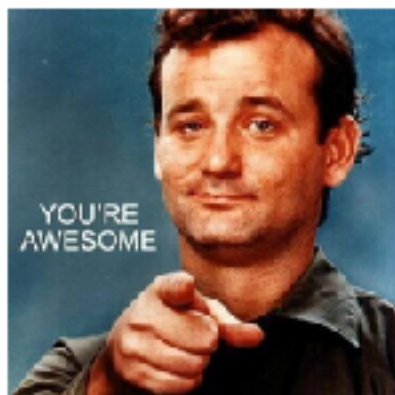
Basic HTML Page - Result

Awesome Header

Smaller Awesome Header

Even Smaller Header

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quidem consequat unde aut dolores odio hac, accusamus recusandae ipsam illum enim voluptatibus obcaecati totam tempora eum quod sapiente. Corporis, quidem, culpa?



Menu Links

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

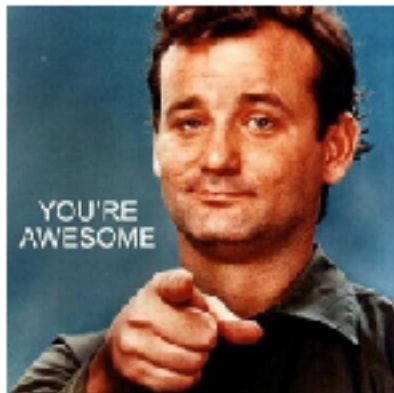
Basic HTML Page - Result

Awesome Header

Smaller Awesome Header

Even Smaller Header

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quidem consequat unde aut dolores odio hac, accusamus recusandae ipsam illum enim voluptatibus obcaecati totam tempora eum quod sapiente. Corporis, quidem, culpa?



Menu Links

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

Hella Boring...

Enter CSS

```
26 <style>
27   h1 {
28     font-size: 60px;
29     text-align: center;
30     margin-bottom: 15px;
31     text-decoration: underline;
32     background-color: black;
33     color: white;
34   }
35
36   h2 {
37     font-size: 40px;
38     text-align: center;
39     margin-top: 15px;
40     margin-bottom: 15px;
41   }
42
43   h3 {
44     font-size: 20px;
45     text-align: center;
46     margin-top: 15px;
47   }
48
```

```
49   img {
50     display: block;
51     margin-left: auto;
52     margin-right: auto;
53   }
54
55   p {
56     text-align: center;
57     font-size: 20px;
58     font-weight: bold;
59   }
60
61   ul {
62     text-align: center;
63     font-size: 35px;
64     list-style-position: inside;
65     border-style: solid;
66     border-width: 5px;
67   }
68 </style>
```

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?

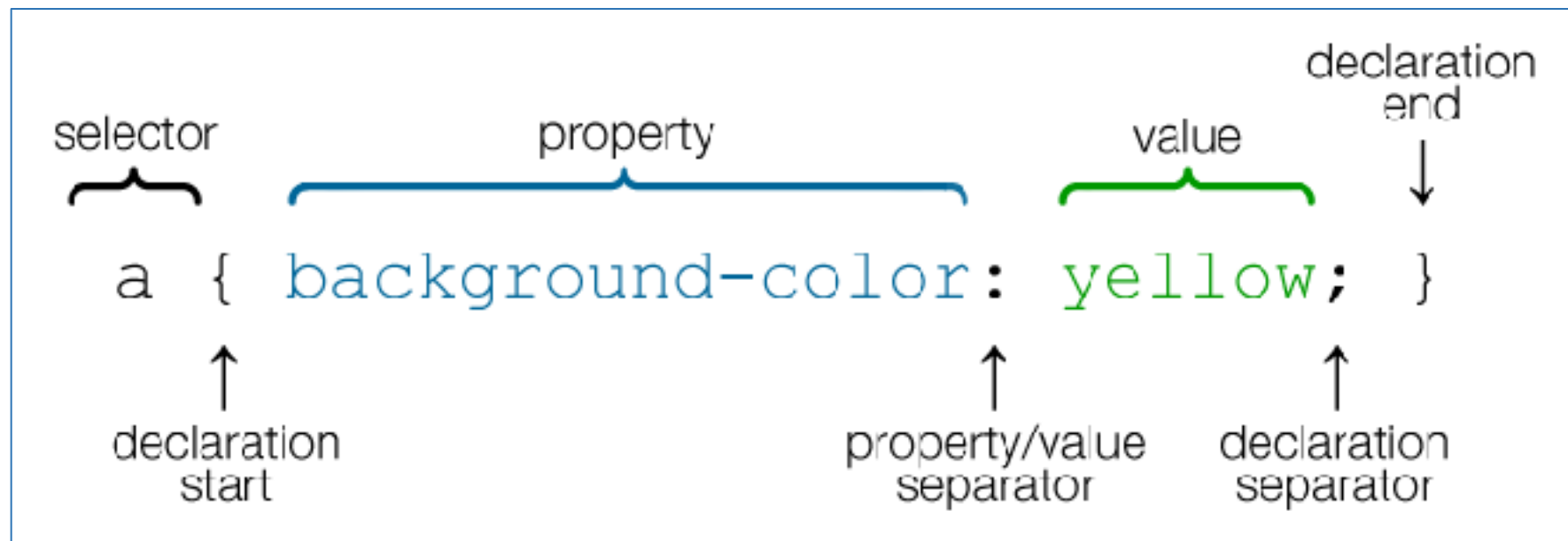


Menu Links

- Google
- Facebook
- Twitter

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.



CSS Example

- In the below example the “Header” would be turned blue and MUCH larger because of the CSS.
- We can incorporate an element’s class or ID to apply a CSS style to a particular part of the document.
 - Just remember to include the necessary symbol before the CSS: “.” for class, “#” for ID.

Example (HTML):

```
<p class="bigBlue">Header</p>
```

Example (CSS):

```
.bigBlue  
{  
  font-size: 100px;  
  color: blue;  
}
```

Key CSS Attributes

Font / Color:

- **color:** Sets color of text.
- **font-size:** Sets size of the font.
- **font-style:** Sets italics.
- **font-weight:** Sets bold.

Alignment / Spacing:

- **padding (top/right/bottom/left):** Adds space between element and its own border.
- **margin (top/right/bottom/left):** Adds space between element and surrounding elements.
- **float:** Forces elements to the sides, centers, or tops.

Background:

- **background-color:** sets background color.
- **background-image:** sets background image.

Believe it or not, HTML / CSS is all you need to develop a vivid, full-blown website.

Instructor: Demo
(quickexample_internalcss.html | 2-BasicCSS)

> YOUR TURN!

Activity: 3-HTML_CSS_Layout | Suggested Time: 20 min

Assignment

In this activity, you'll upgrade your previous HTML bio-page using CSS style rules. Once you're done, commit and push up your changes to GitHub.

We'll send you additional instructions via Slack.

> YOUR TURN!

Student Bio

Your Name

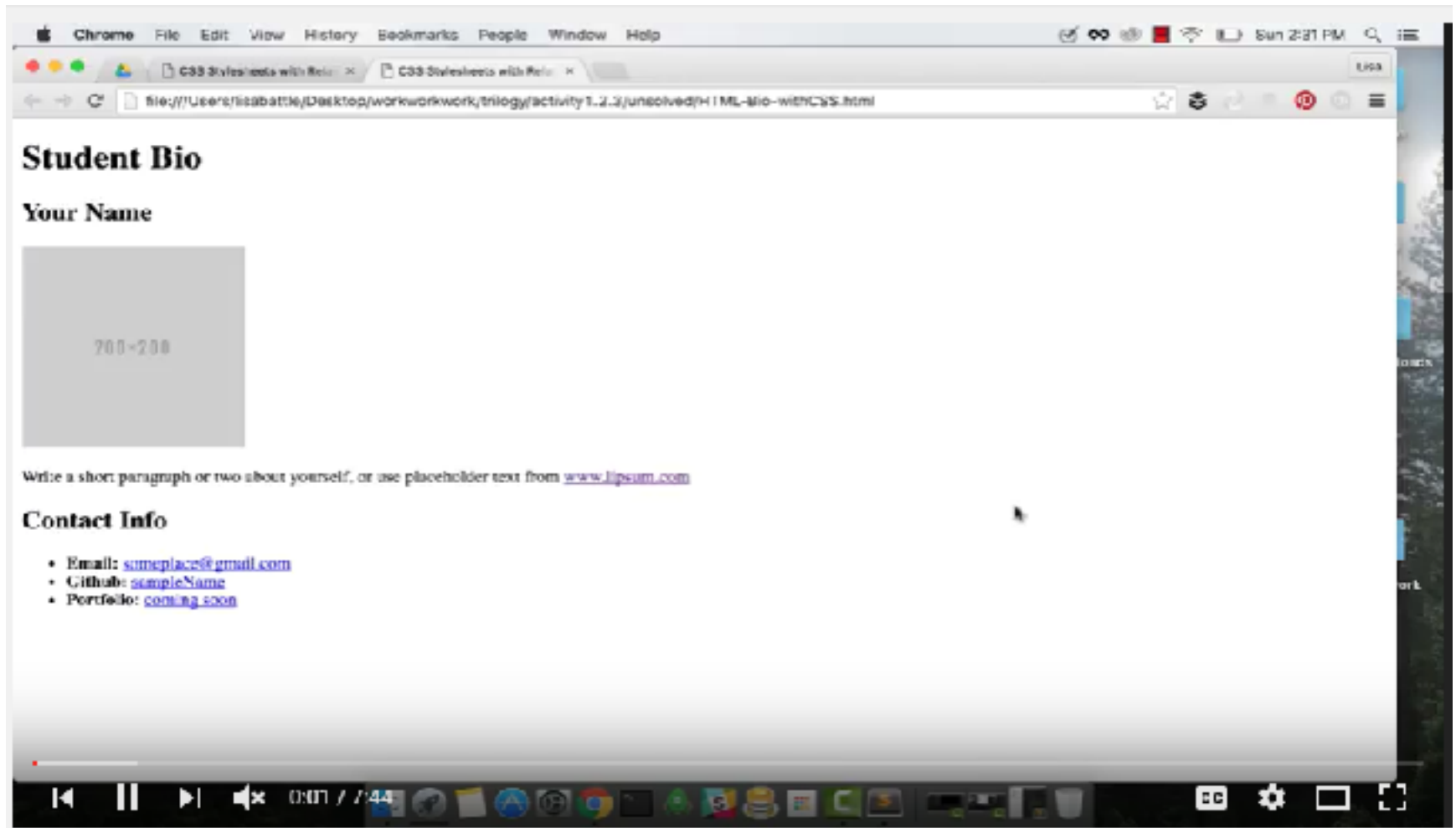


Write a short paragraph or two about yourself, or use placeholder text from www.lipsum.com

Contact Info

- Email: someplace@gmail.com
- Github: [sampleName](#)
- Portfolio: [coming soon](#)

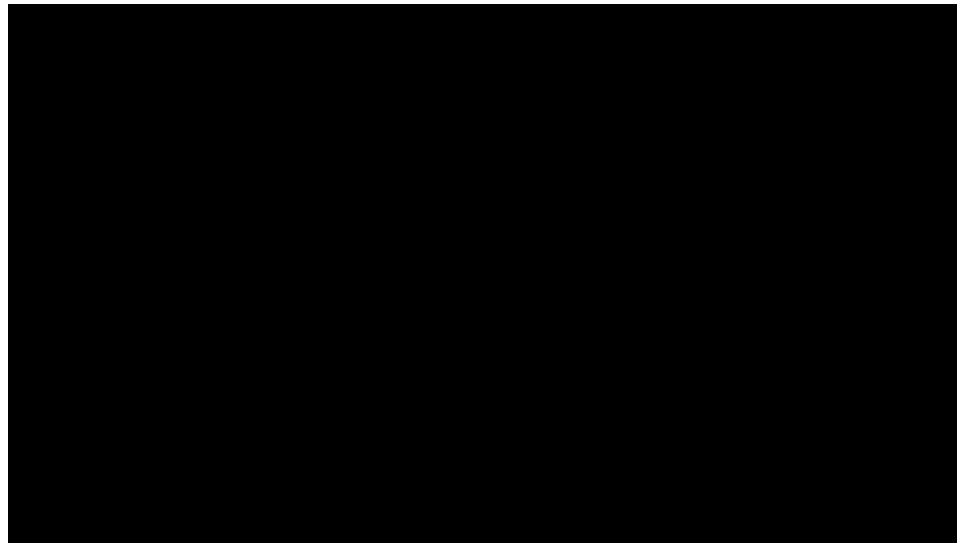
Video Walkthrough!!



<https://www.youtube.com/watch?v=kMBinXTCrXI&list=PLgJ8UgkiorCnMLsUevoQRxH8t9bt7ne14&index=2>

Still a Bit Confused?

Remember! We've got video guides for key activities like that last one.



If you feel like you are EVER falling behind, use those online walkthroughs to help catch back up. They are made to be easy to understand.

Still having trouble? Shoot your instructor or one of your TAs a message! We are here to help you out in whatever way we can!

Recap + Questions
