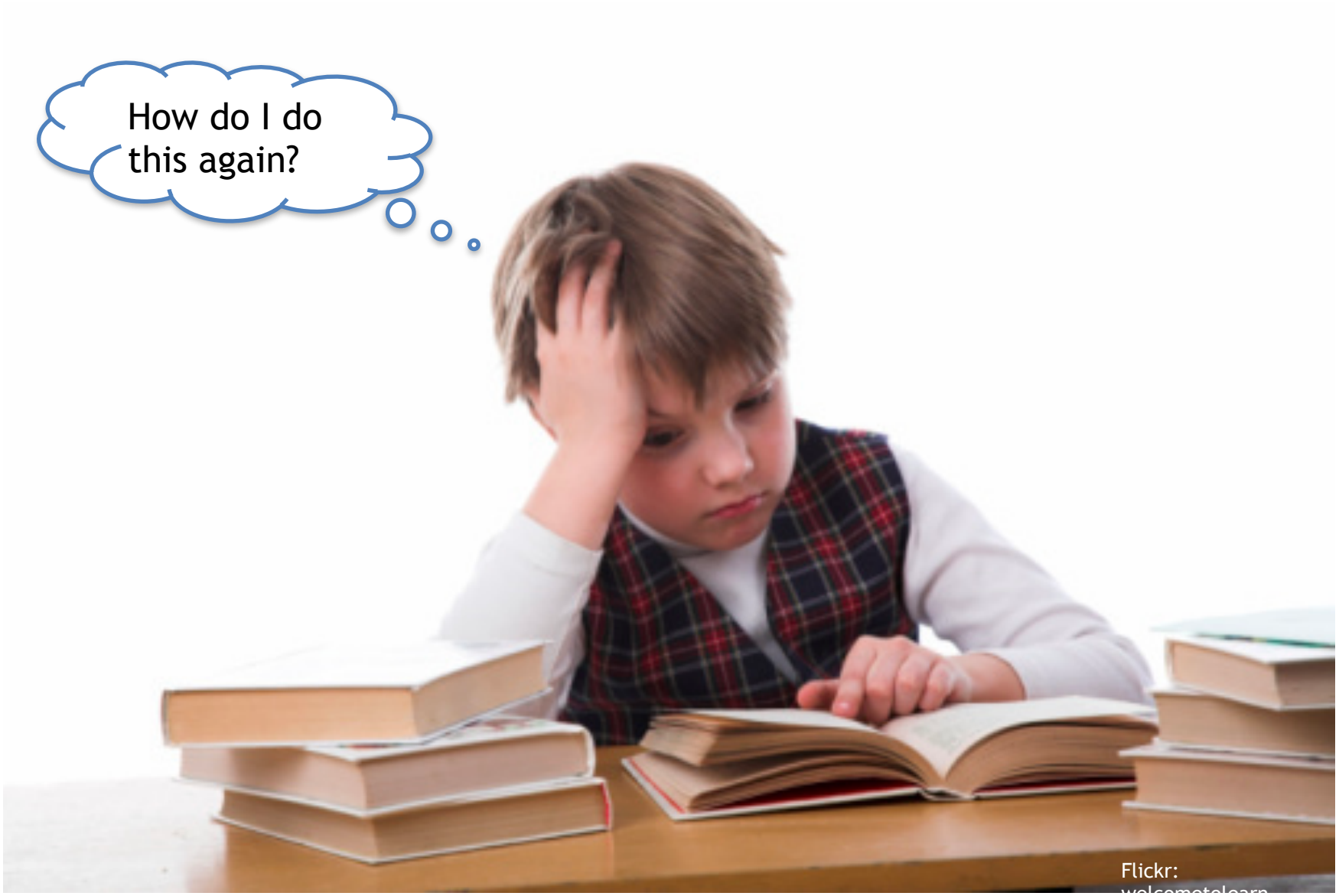
Git'n Pro with HTML/CSS

The Coding Bootcamp

It’s Okay!



Admin Items

Where to Get Help

- **Practice, Practice, Practice:** Work Individually or in Groups
- **Review In Class Material (Exercises and Slides):**
  <<<<PROVIDE LINK HERE>>>>>
- **Re-Watch Class Videos:**
  <<<<PROVIDE LINK HERE>>>>>
- **In Class Office Hours:** 45 minutes before class, 30 minutes after
- **One-on-One Sessions:** By Announcement through SSM
- **Contact Student Success:** Anytime!

Homework #1 - Assignment

- **Also, at this point everyone should have access to the class content and homework repository.**

  <<<<< LINK HERE>>>>

- **Homework Assignment #1 is due next week**

  - **MW Class: Next Wednesday (<<<<DATE HERE>>>>)**

  - **TTH Class: Next Thursday (<<<<DATE HERE>>>>)**

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:

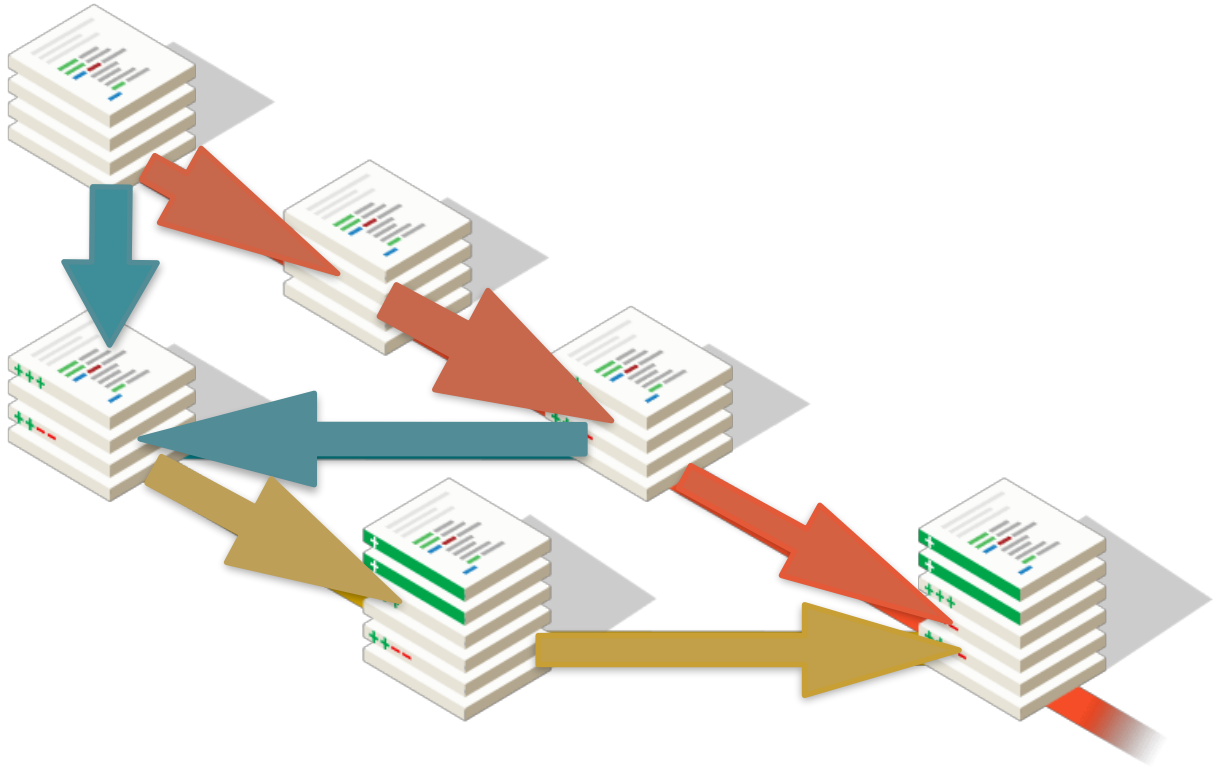
- Continue getting comfortable with HTML.
- Be able to completely write a basic HTML document (like in last class).
- Understand what CSS is, what it's for, and how it works with HTML.
- *Be able to use Git and GitHub to upload code.*

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.

What could go wrong?

The Group Project

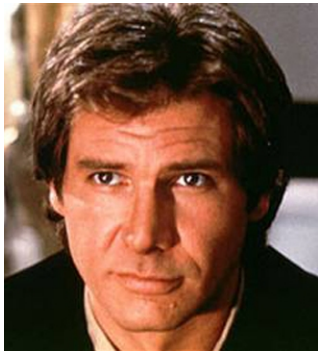
Lesson:

You should use Version Control.

….and watch your teammates' work

Through the Git, things you will see. Other branches. The future… the past…

…You will find only what you commit.



Git Version Control:

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

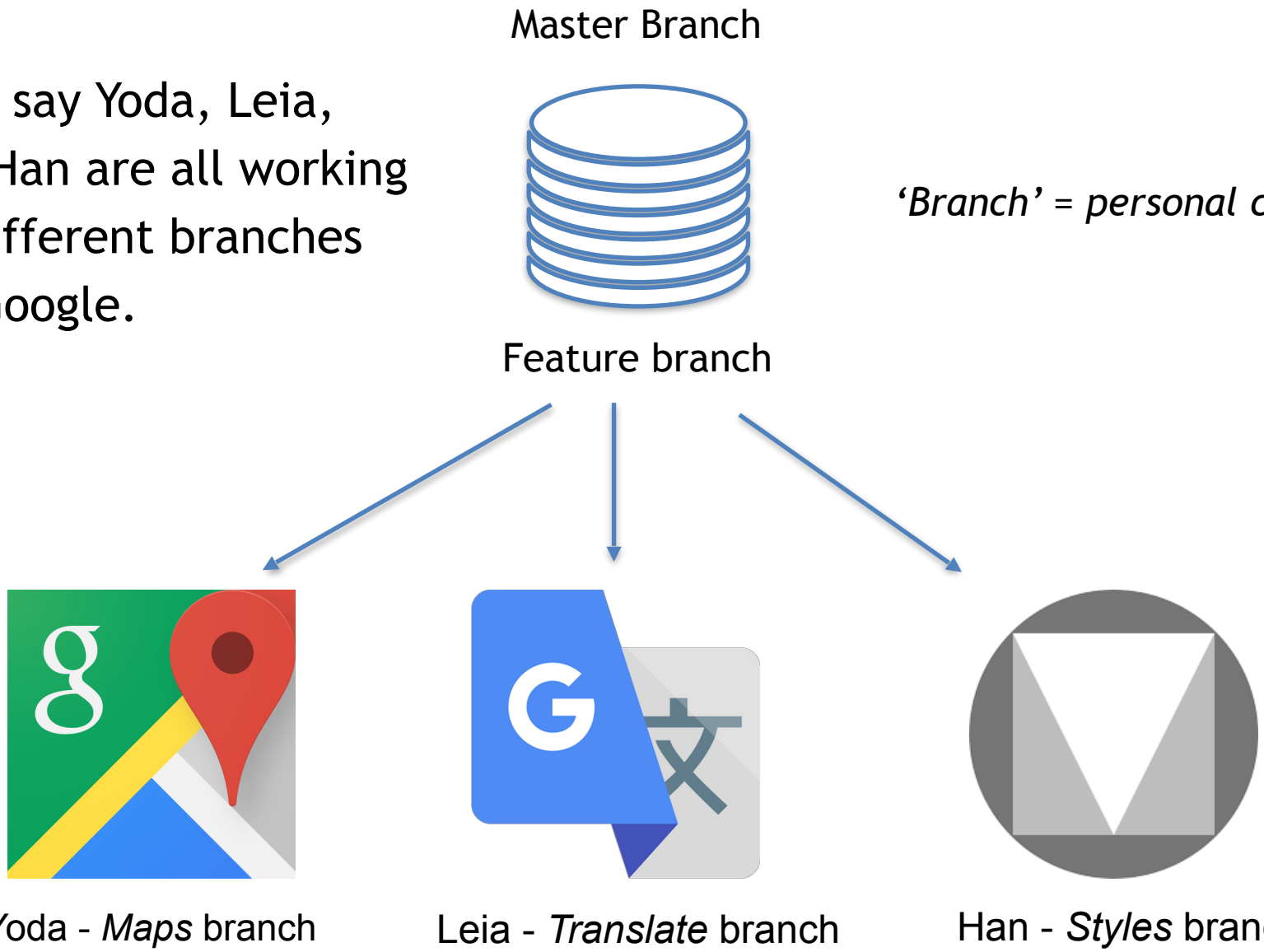
The Benefits of Git:

1. A process for resolving conflicts in code.

2. Version History.

The Group Project

Let's say Yoda, Leia, and Han are all working on different branches for Google.



The Group Project

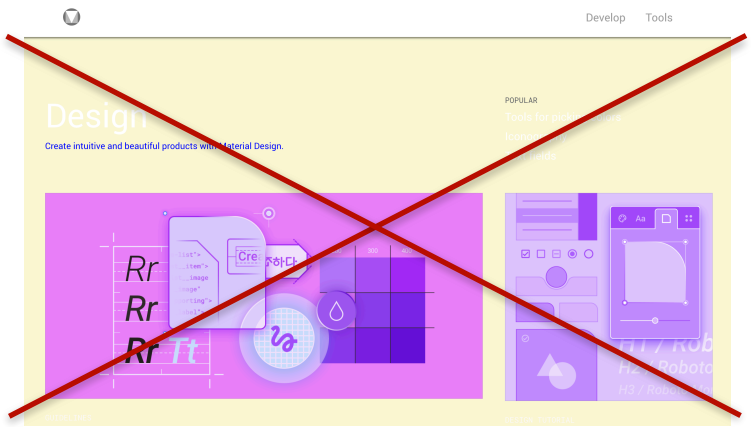
- If Leia, Yoda, and Han are all working on different branches, they can work separately without affecting each other's code.

- If Yoda added a Maps feature and Leia wants to use this new feature in his code, all Leia has to do is merge in the new version of Yoda's branch into her own branch and keep coding!
  (This is *much* better than the alternative of copy and pasting files)

The Group Project

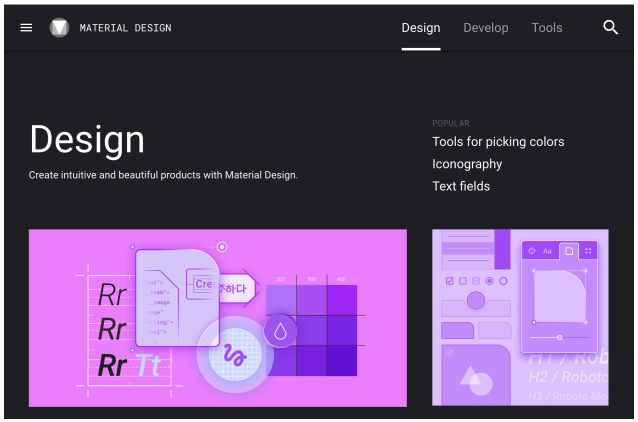
- Let’s say Han was working on a site-wide change in styling, but one of his changes had unintended consequences on a page Leia was working on.
- Instead of having to debug the problem while users stare at a broken page, Git makes it easy to instantly revert or “rollback” the mistake.

New Broken Version



Revert

Old Working Version



Lesson:

You should use Version Control!

Quick Activity!

Suggested Time: 3 min

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?

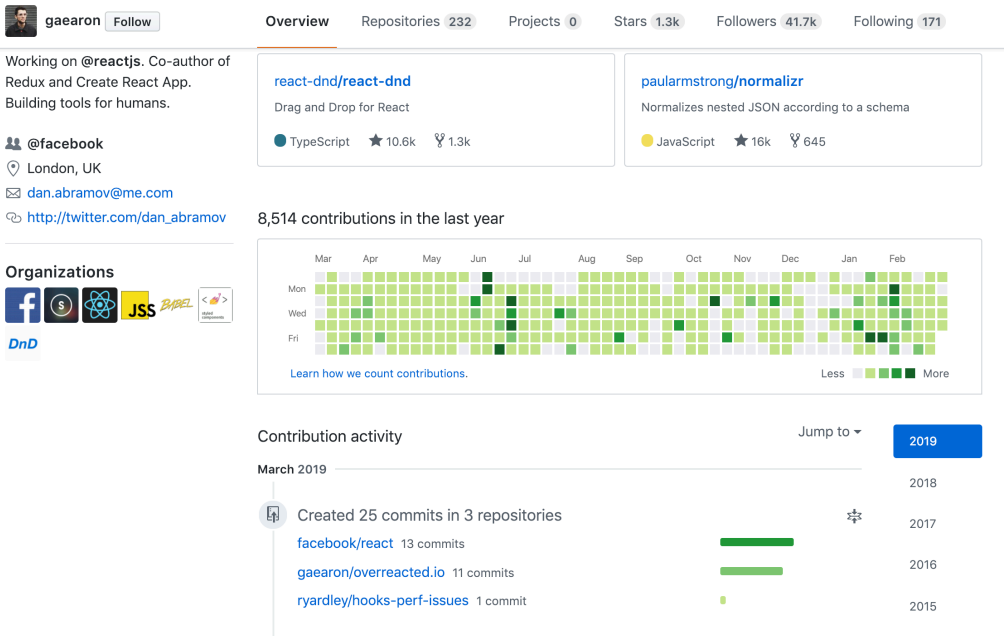
- GitHub is a Web-Based hosting service to store code online.
- It allows developers to **pull** (download) code or **push** (upload) code to the same **repository** (directory).
- It also allows developers to **view histories** of code changes and to **track issues**.

So… What's this GitHub?

- GitHub also serves as a social network for developers and their code.
- It allows developers to **freely** contribute to open source projects and discuss issues/features.

So… What's this GitHub?

- GitHub is a great way to showcase your work.



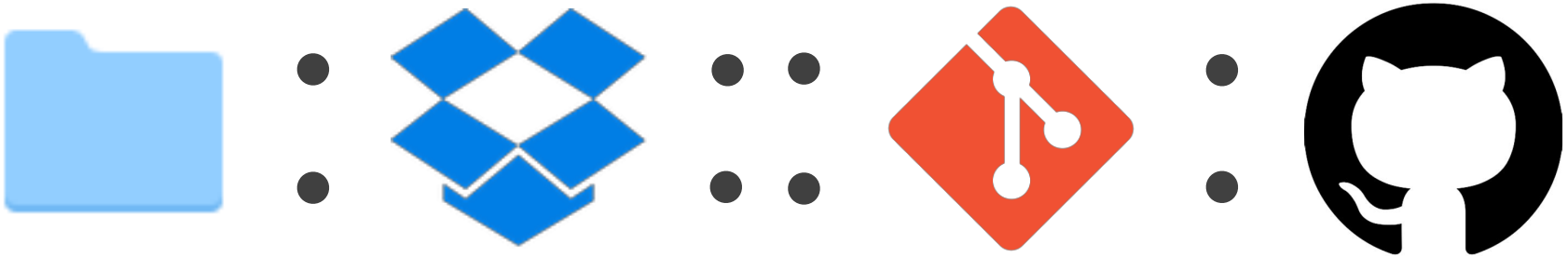
- Hiring managers check GitHub activity and samples to evaluate candidates.
- Seeing regular activity shows recruiters that you're continuously learning and developing.

To Sum It Up…

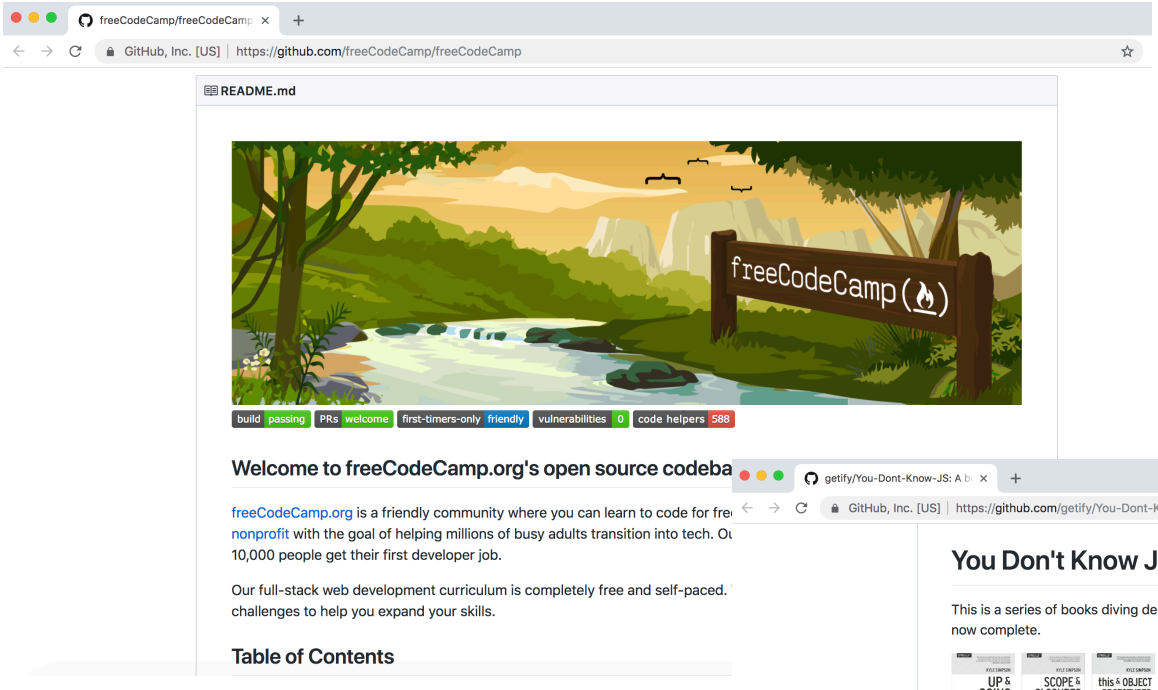
- Think of GitHub like Dropbox for Git. It serves as a backup for your repository and allows you to share your code with other people.

- One big advantage GitHub has over a tool like DropBox is that there are many builtin tools to help with developers with collaboration.

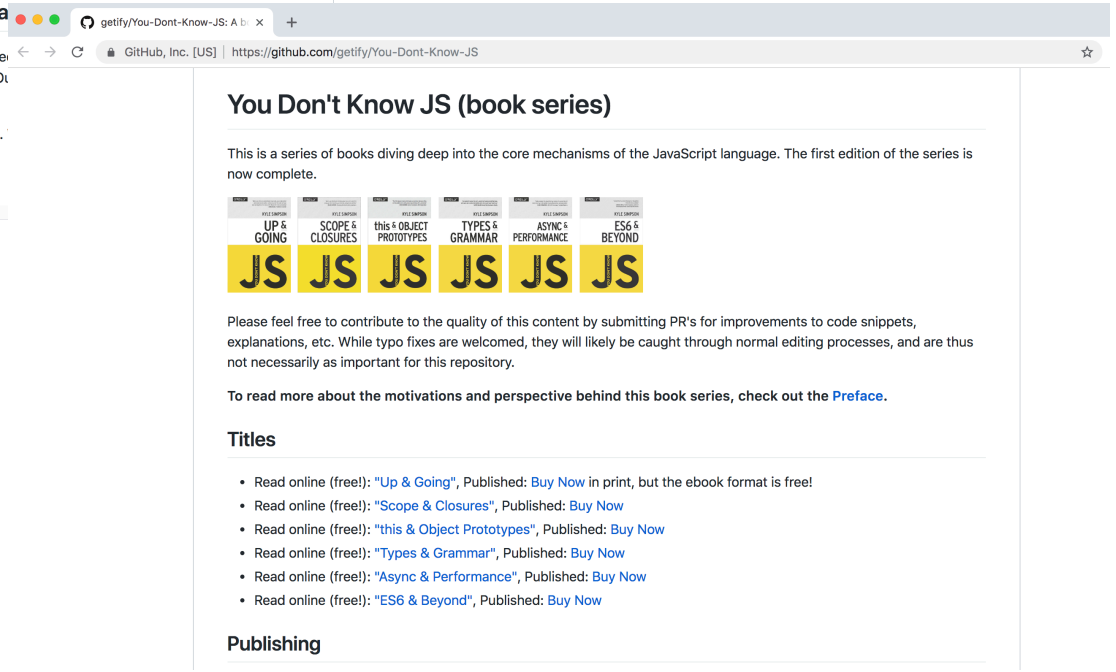
- Remember, both Git and GitHub aren't tied to specific programming languages. For our purposes, we've been using code examples, but they could just as easily be image files or PowerPoint presentations.



GitHub Examples



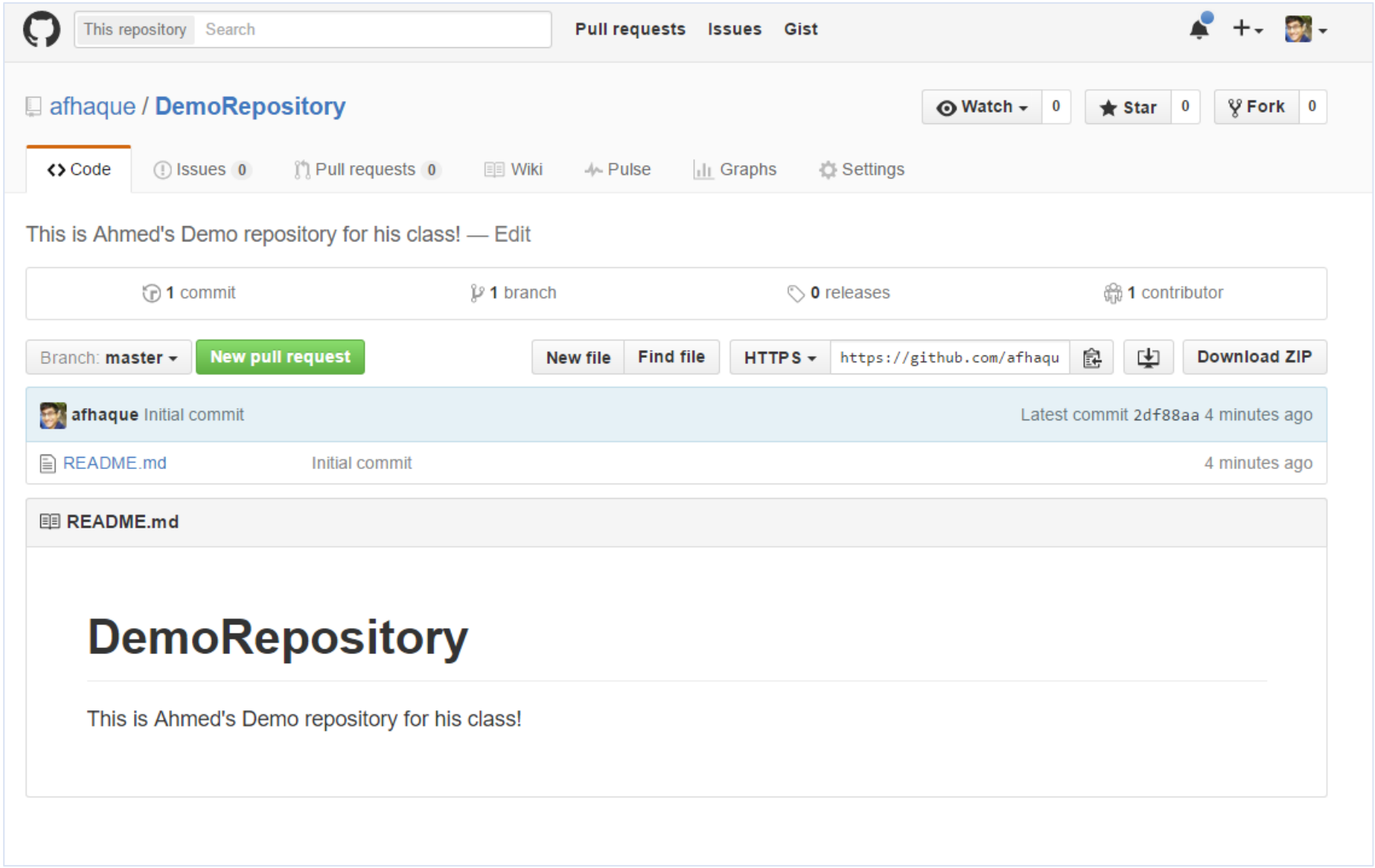
Repos...



Repos everywhere...

Get Started with Git

Instructor Git Demo!



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, while establishing a connection between a local and remote repo.

2. **git add** – tells git to track one or more files.

3. **git commit** – notes a change to the local repo.

4. **git push** – sends changes to a remote repo.

5. **git pull** – downloads freshest version of the repo (from a remote repository) into the local repo.

> YOUR TURN!

Activity*:* Git Add, Commit, Push **| Suggested Time:** 15 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:

- Create a new public GitHub repository, and name it `zen-garden`.
- Go to CSS Zen Garden (http://www.mezzoblue.com/zengarden/alldesigns/). Navigate to a few of the examples and choose a page that you like.
- Download the HTML/CSS, add it to your new repo, and change the CSS in any way you'd like.
- **Add**, **Commit**, and **Push** the code to GitHub.

Git Workflow

The .git Folder

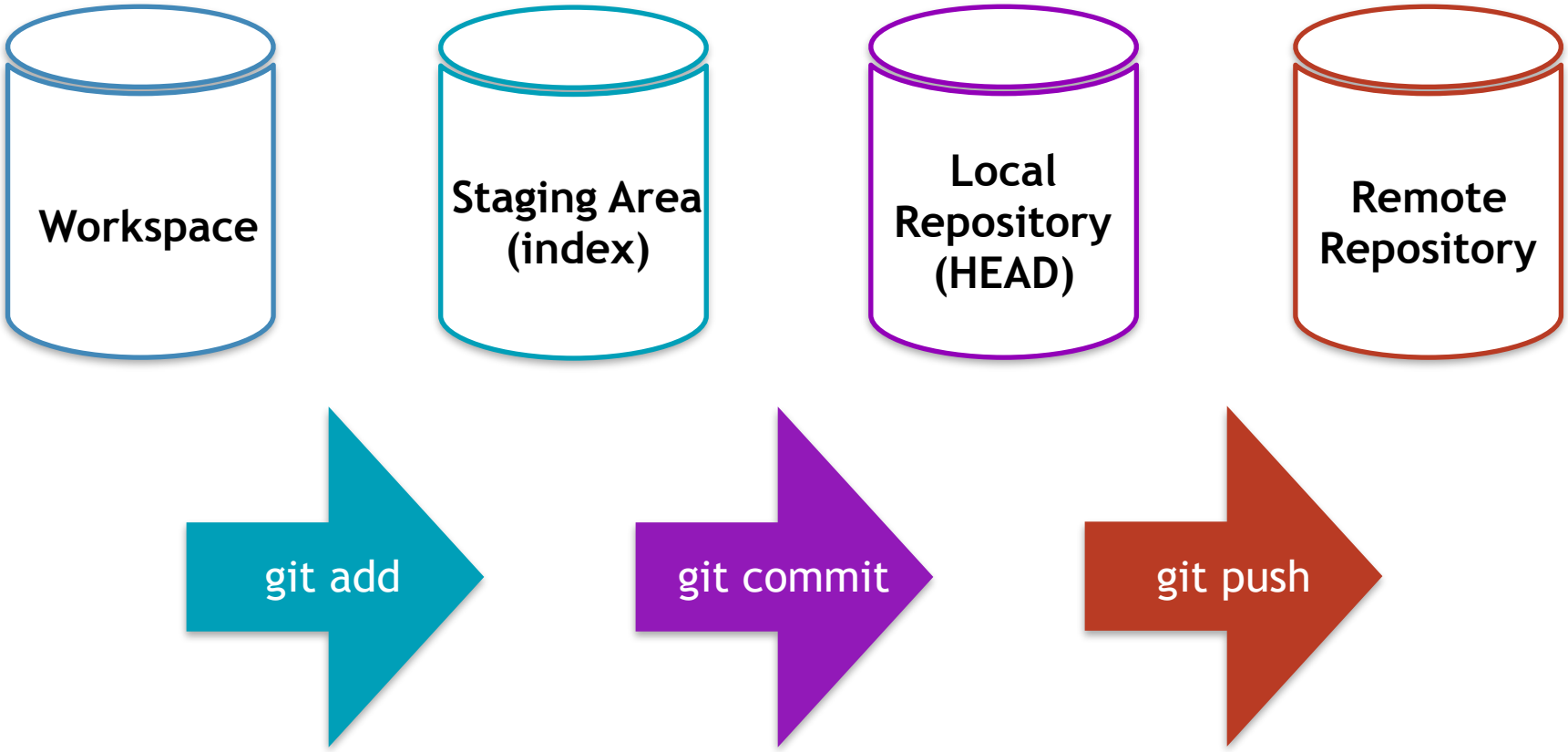
- When you cloned your repo, you might have noticed a hidden *.git* folder was created.

- This folder stores information about your repository, including a history of all changes, along with who made them.

This is what makes your local directory a *repo*, as opposed to a regular folder.

Git Workflow

In Git, there are *four* conceptual areas your code can be in.



Workspace

The **workspace** is where you're used to dealing with your files.

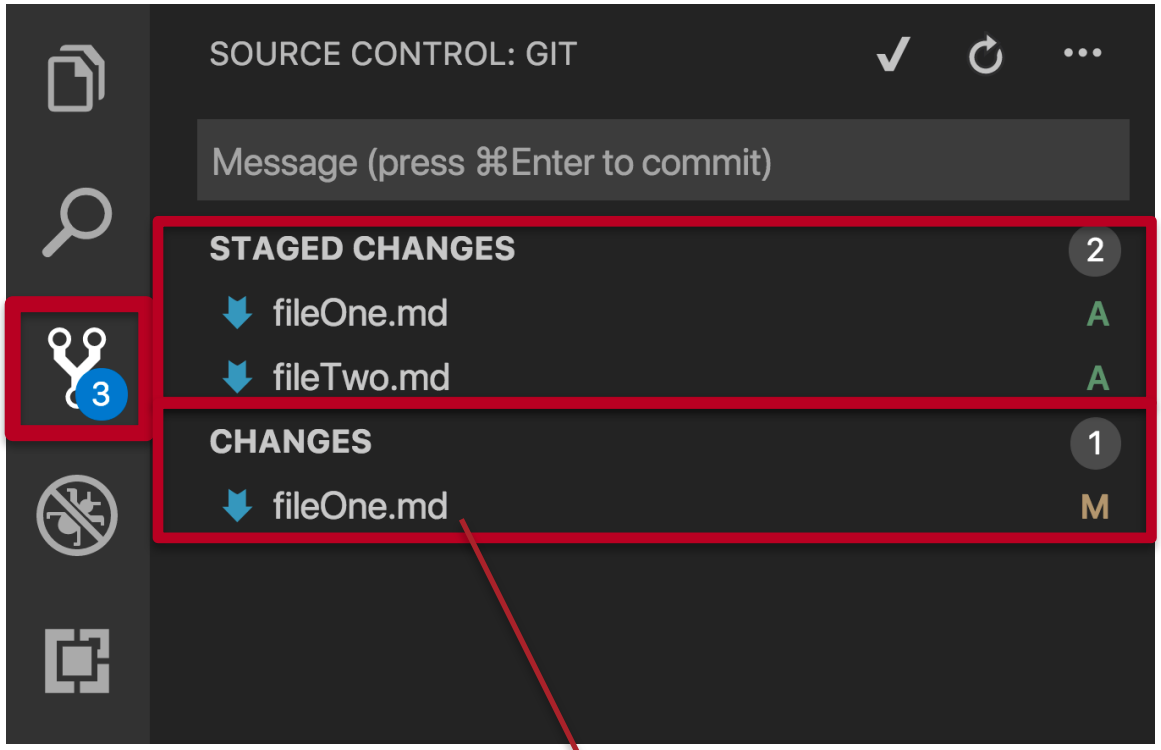
Any time you **save** a file, you're replacing the old version of a file with a new one.

This step is referred to as **Untracked** since Git has no knowledge of your workspace.

Setting the Stage

- **git add -** Adds a file to the **Staging Area.**
- This command takes an **argument**, which means we must provide the name of the file we want to add. Ex: `git add myFile.md`
- Since we want to add *all* the files in our directory, we'll give it the option **-A**. Ex: `git add -A`

Setting the Stage Continued



These files have been *staged* with `git add`, but not committed.

This file has been *changed* (and saved) since it was staged.
We should probably run `git add -A` again.

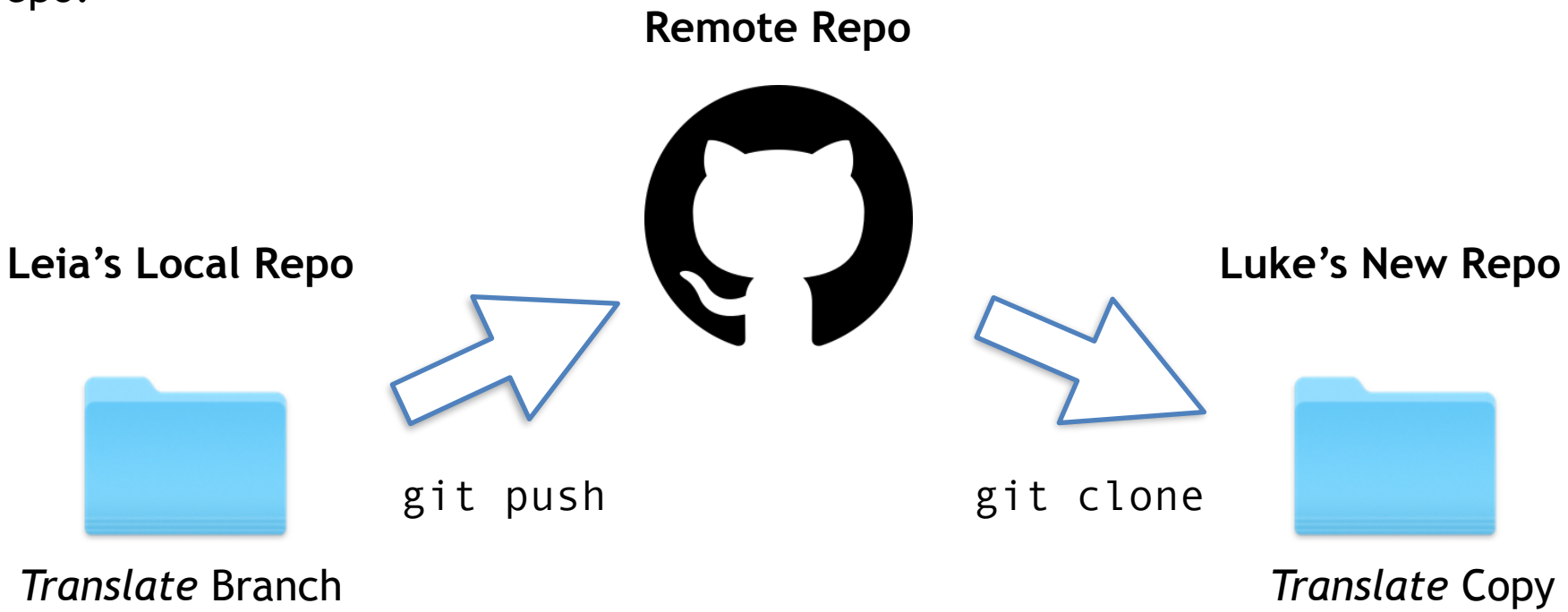
Keeping It Local

- To add files to the **local repository,** run `git commit`
- This command takes the files from your **staging area** and adds them to the history kept in the *.git* folder.

The Remote Repository

- Lastly, we add our **local repo** to our **remote repo** with `git push`.

- Once things are pushed up to GitHub, GitHub will have a history of every change you've committed, along with any changes your teammates have pushed to the repo.



Still a Bit Lost? Never Worry!

- Remember that you will use these concepts every day.
- It's most important that you familiarize yourself with the **5 basic git commands**.
- If you're ever in the middle of making changes and want to see what step you're at, run `git status`.

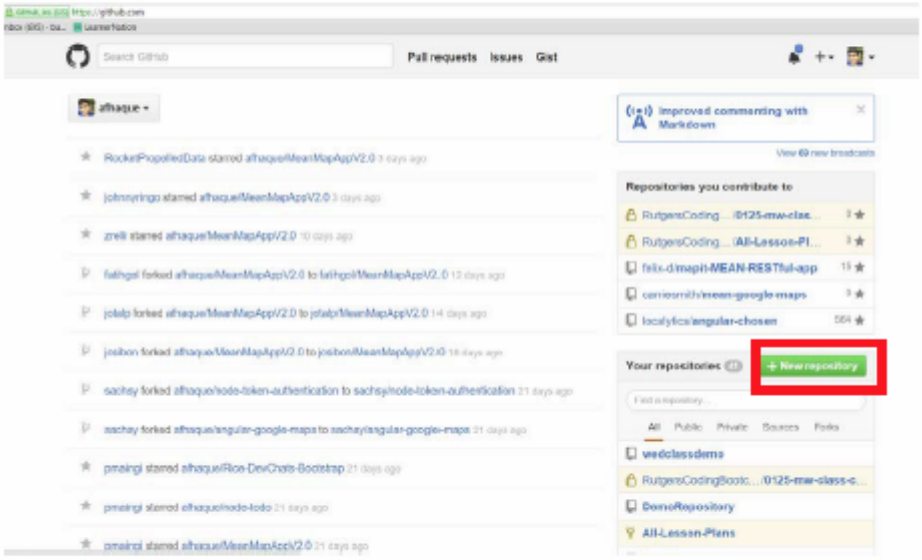
```
Alecs-MacBook-Pro:git-example alecbranaa$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   fileOne.md
        new file:   fileTwo.md

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   fileOne.md
```

Still a Bit Lost? Never Worry!

Steps to Uploading Your Code to GitHub

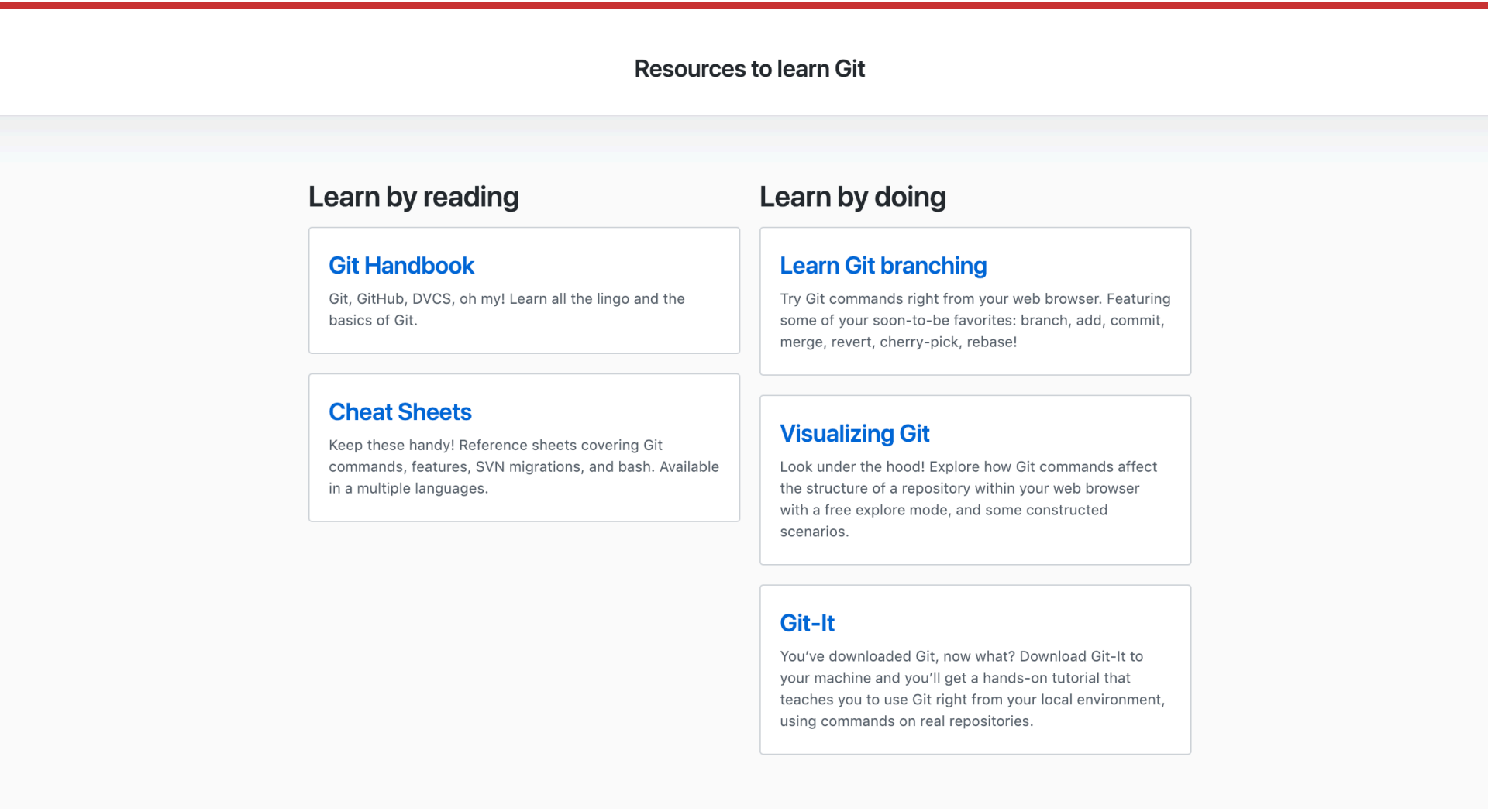
Step 1

Create a New Repository in GitHub.com



- **Follow the handy pdf guide provided to you!**
- Practice a few times on your own before our next class.

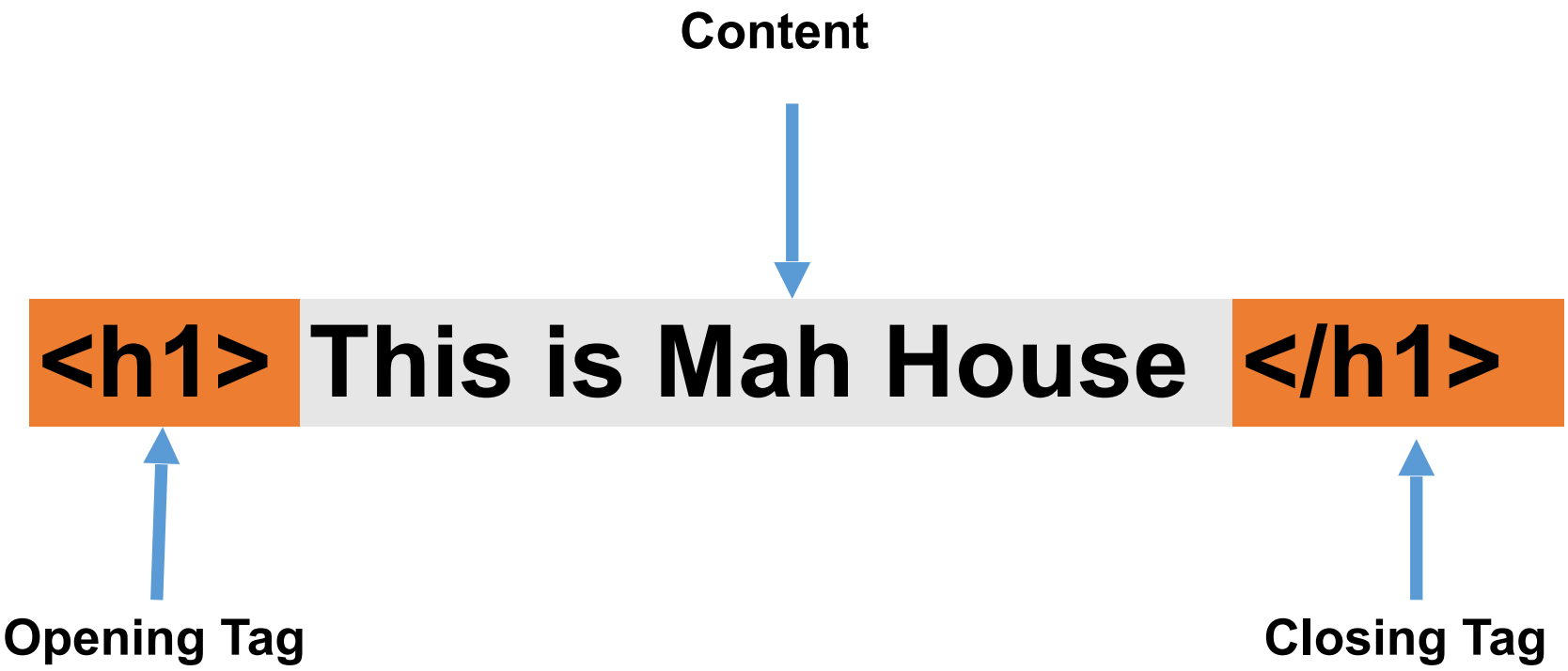
If You’re Still Lost… Here’s Some (Free) Resources



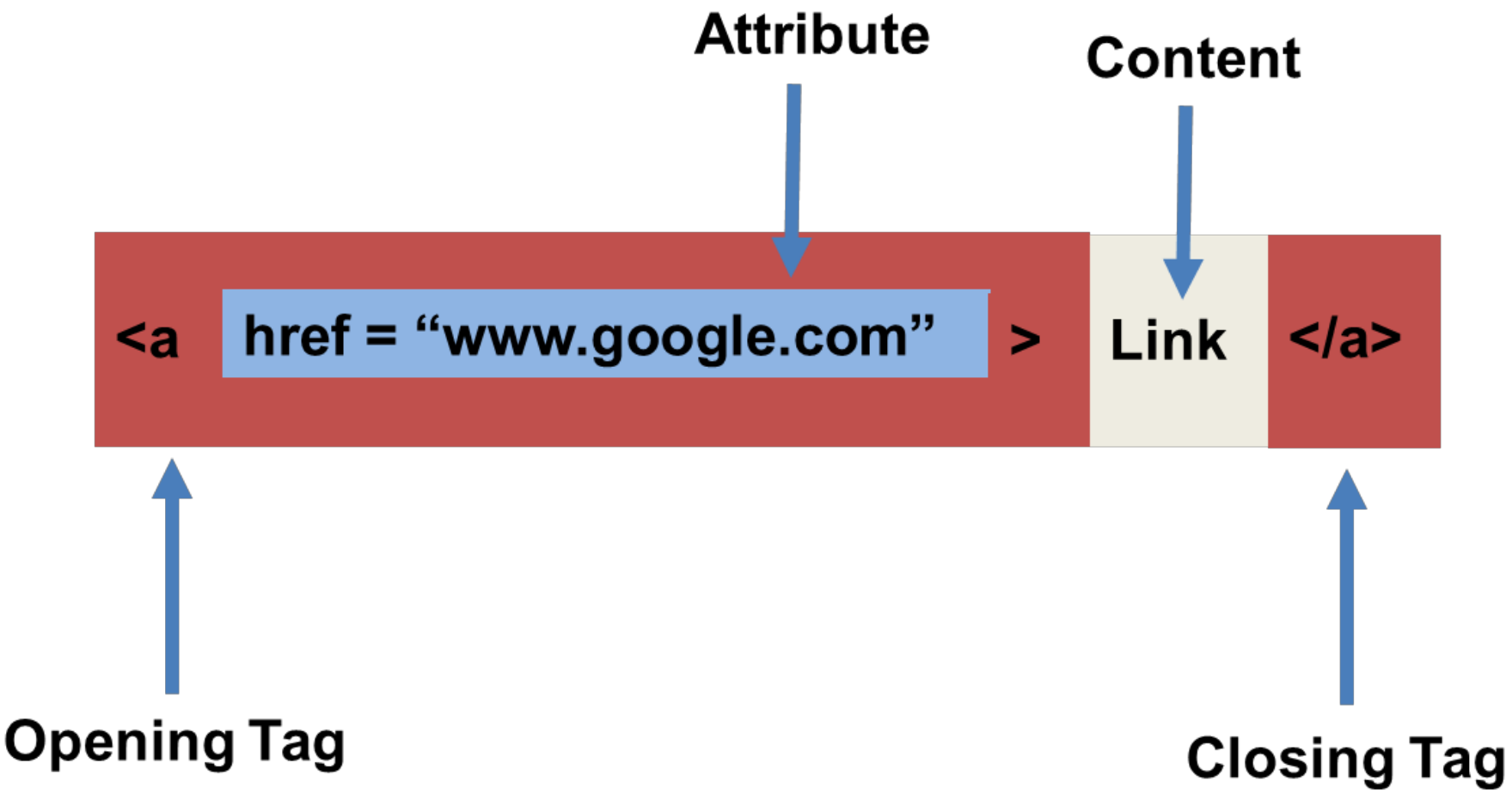
https://try.github.io/

HTML Round 2

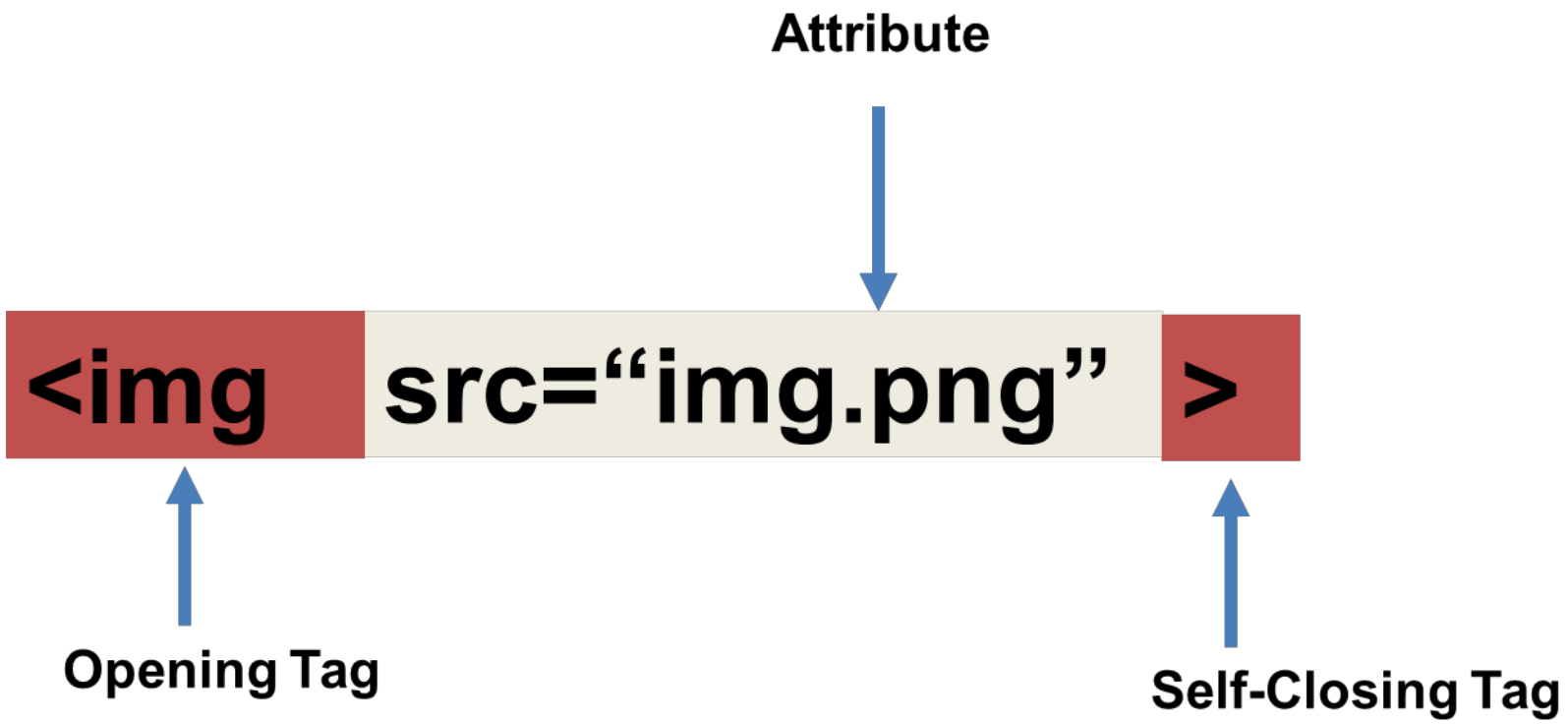
HTML Syntax (Basic)



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
- ** ** - Logical Container ***
- ** ** - Wraps individual Paragraphs

Others:

- **** (bold), **** (emphasis)
- **<img>** (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>
```

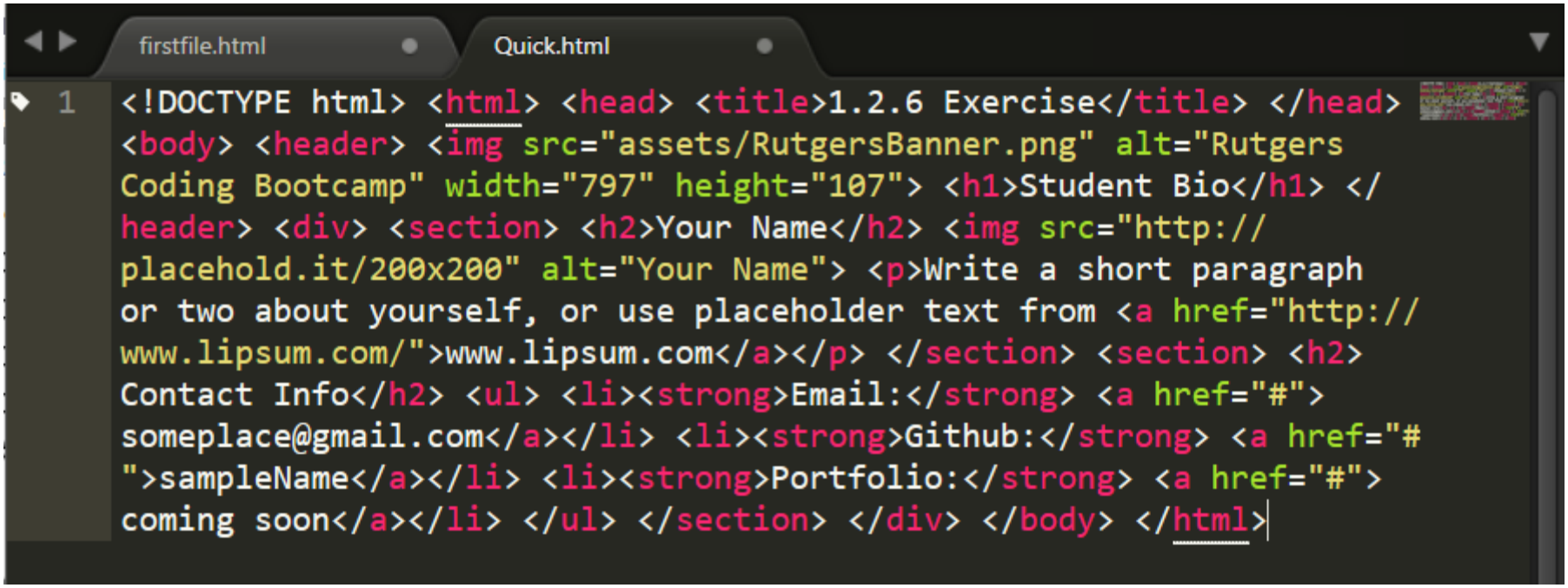
First name:

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



- 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!

> YOUR TURN!

Activity*:* 1-HTML_Git **| Suggested Time:** 20 min

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

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>My First Website!</title>
</head>

<body>

    <h1>Awesome Header</h1>
    <h2>Smaller Awesome Header</h2>
    <h3>Even Smaller Header</h3>

    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
        incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
        exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
    <img src="https://vignette.wikia.nocookie.net/Bill_Murray.jpeg/" alt="bill murray" width="25%" >

    <h3>Menu Links</h3>
    <ul>
        <li><a href="http//www.google.com">Google</a></li>
        <li><a href="http//www.facebook.com">Facebook</a></li>
        <li><a href="http//www.twitter.com">Twitter</a></li>
    </ul>

</body>
</html>
```

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

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

Hella Boring...

Enter CSS

```
<style>
    h1 {
        font-size: 60px;
        text-align: center;
        margin-bottom: 15px;
        text-decoration: underline;
        background-color: black;
        color: white;
    }

    h2 {
        font-size: 40px;
        text-align: center;
        margin-top: 15px;
        margin-bottom: 15px;
    }

    h3 {
        font-size: 20px;
        text-align: center;
        margin-top: 15px;
    }

    img {
        display: block;
        margin-left: auto;
        margin-right: auto;
    }

    p {
        text-align: center;
        font-size: 20px;
        font-weight: bold;
    }

    ul {
        text-align: center;
        font-size: 35px;
        list-style-position: inside;
        border-style: solid;
        border-width: 5px;
    }
</style>
```

Enter CSS - 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?

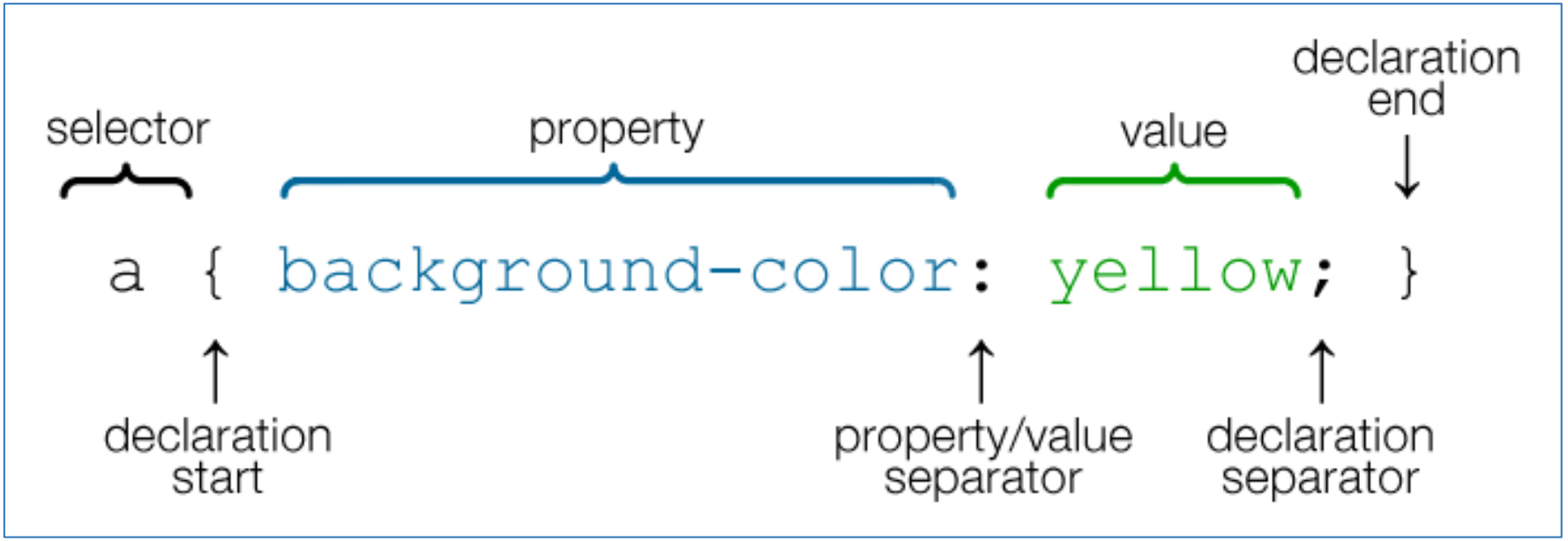
Menu Links

- Google
- Facebook
- Twitter

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.

Powerful Duo

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

INSTRUCTOR DEMO

Instructor: Demo

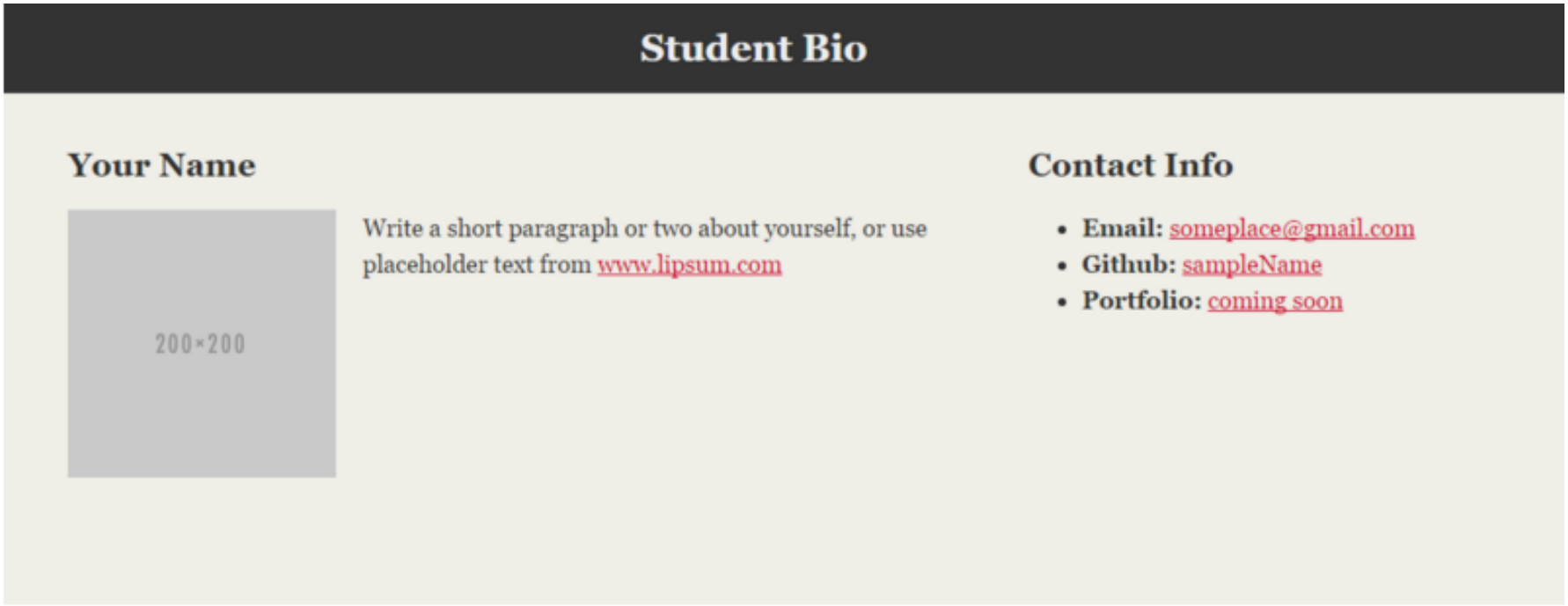
(quickexample_internalcss.html | 2-BasicCSS)

Assignment

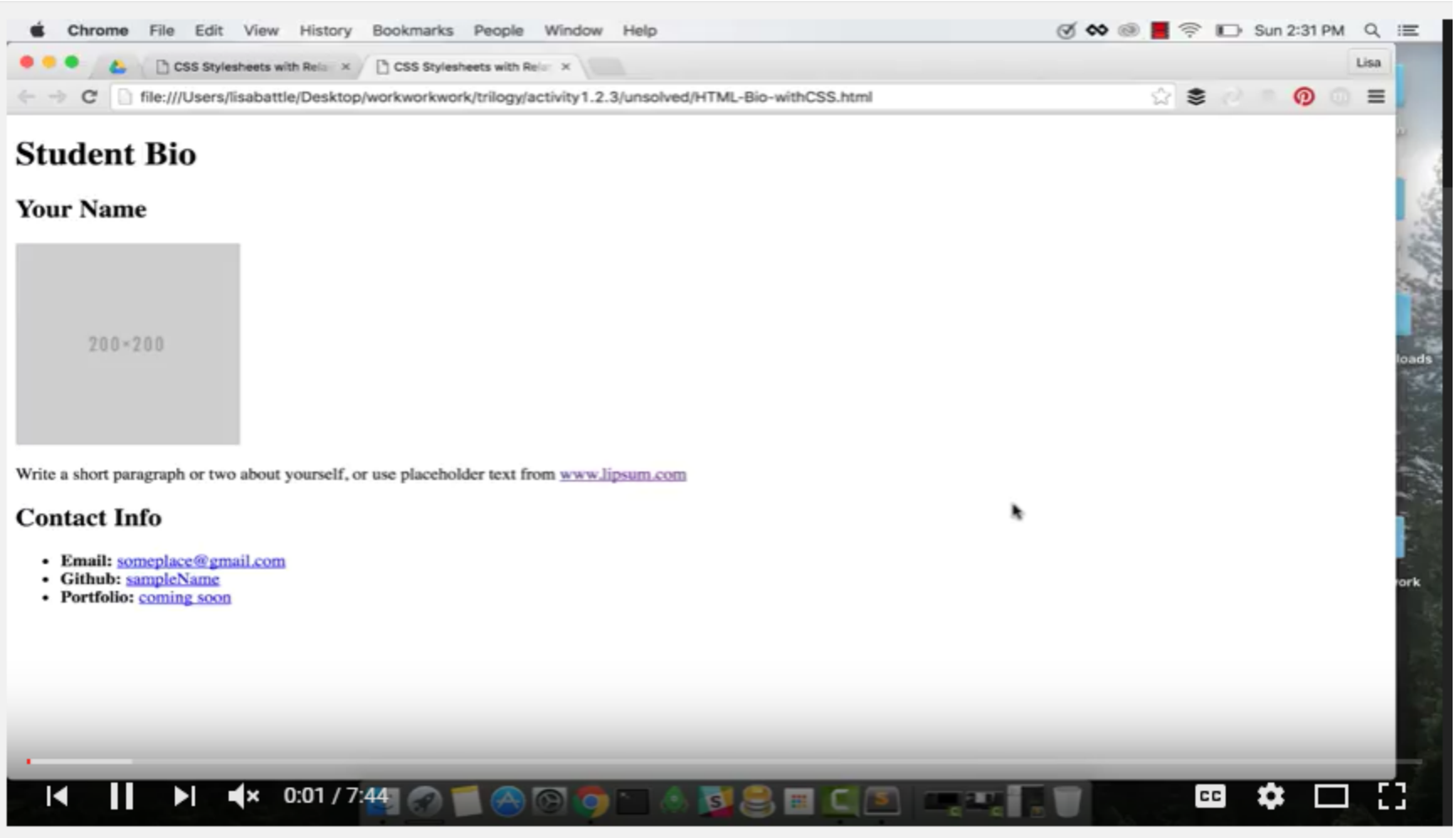
In this activity, you'll upgrade your previous HTML bio-page using CSS style rules. Once you're done, commit and push up your changes to GitHub.

We'll send you additional instructions via Slack.

> YOUR TURN!


Student Bio
Your Name
200×200
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