

# ***Git'n Pro with HTML/CSS***

---

The Coding Bootcamp

September 20, 2018

# It's Okay!

How do I do  
this again?



Flickr: welcometolearn

# ***Admin Items***

---

# Build your Professional Skills

---

- **Practice, Practice, Practice:** Work Individually or in Groups
- **Review In Class Material**
- **Re-Watch Class Videos:**
- **In Class Office Hours:** 45 minutes before class, 30 minutes after
- **One-on-One Sessions:** By Announcement through SSM
- **Contact Student Success:** Anytime!

**Remember – Its all about time management. You can do this!**

# Homework #1 - Assignment

---

- Also, at this point everyone should have access to the class content and homework repository.
- Homework Assignment #1 is due **Saturday, September 30**

# ***Today's Class!***

---

# Today's Objectives

---

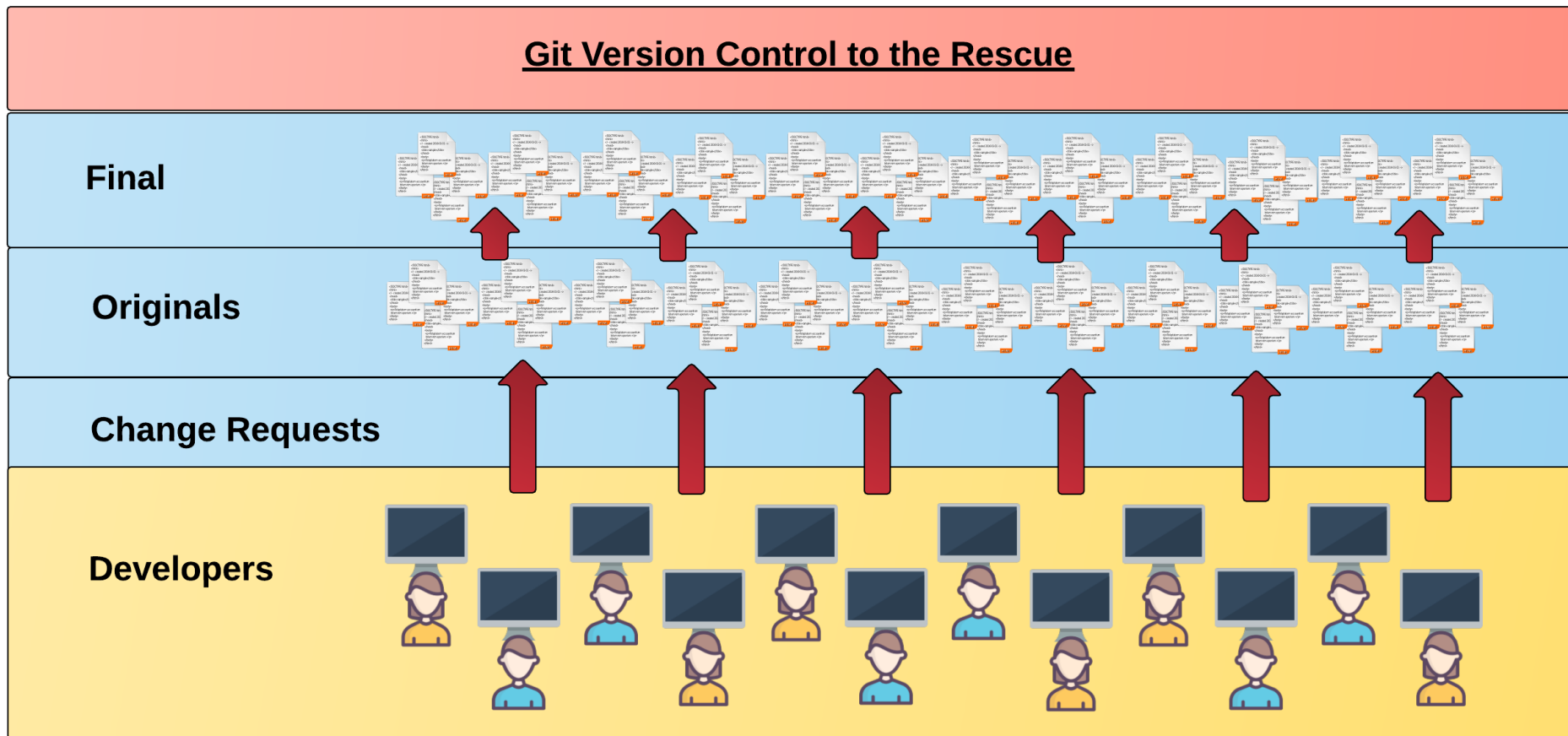
- You will learn about the importance of Git Version Control and of how to use it.
- You will create GitHub Repositories, push code into them, and share with class.
- You will make more HTML documents.
- You will learn to properly use basic HTML tags.
- You will implement basic CSS styling to HTML documents.

# ***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 Group Project

---

**The key:**

**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.

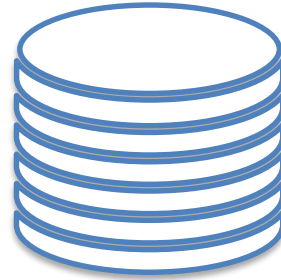
## The Benefits of Git:

1. Supports a process of dividing up work and assigning it to different individuals
2. Supports a process for resolving conflicts in code.
3. Maintains version History.

# The Group Project

Each grabs a copy of the  
same index.html file ...  
which only has a <body>  
tag

Master Branch



*'Branch' = personal copy*

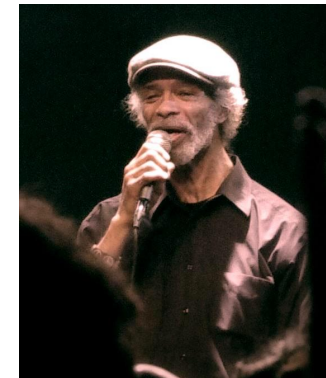
Personal branch



Maya's branch



Anne's branch



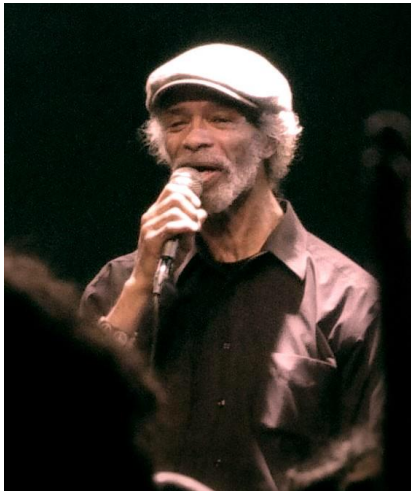
Gil's branch

# The team goes to work



```
<ul>  
  <li>On the Pulse of Morning</li>  
  <li>I Know Why the Caged Bird Sings</li>  
  <li>And Still I Rise</li>  
</ul>
```

**Notice Gill and Maya  
plan to add the same  
tags to their own copy  
of index.html**



```
<ul>  
  <li>Free Will</li>  
  <li>Pieces of a Man</li>  
  <li>The Revolution will not be Televised</li>  
</ul>
```

# Maya pushes first

## Master Copy



1

Maya **pushes (uploads)** her code changes into the main branch.

**No code conflicts.**

**Maya pushes her code first – so now the master branch with index.html has a body tag with the <ul> tagged data**



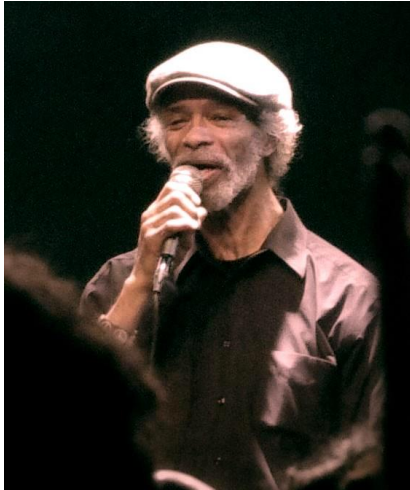
## Maya's Branch



**Gil's edits are ready**

---

**Rule: pull first, then push your changes**



Ok

# Gil pulls latest changes

## Master Copy



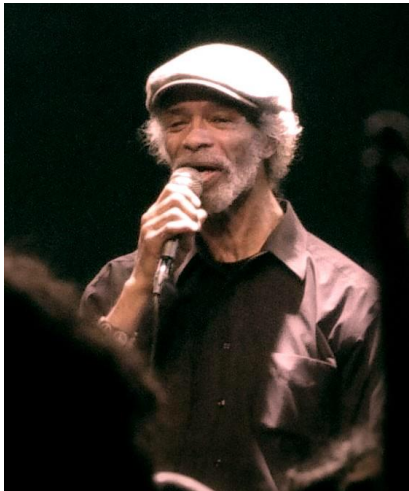
1



**Gil refreshes his branch with a new pull from Master .....**



## Gil's Branch





# Gil conflicts with master branch

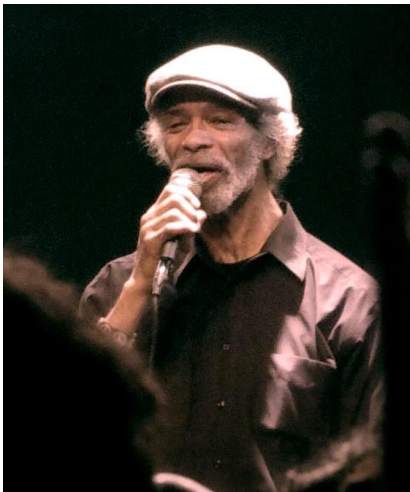
## Master Branch



1



..... And he now has a conflict. Git does not know if the `<li>` tags are duplicates and should be replaced or are additional data and should be appended ...



```
<li>On the Pulse of Morning</li>
<li>I Know Why the Caged Bird Sings</li>
<li>And Still I Rise</li>
<li>On the Pulse of Morning</li>
<li>I Know Why the Caged Bird Sings</li>
<li>And Still I Rise</li>
```

Git sees a conflict.

**Lesson:**

**Use Version Control!**

**Pull frequently**

# 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.

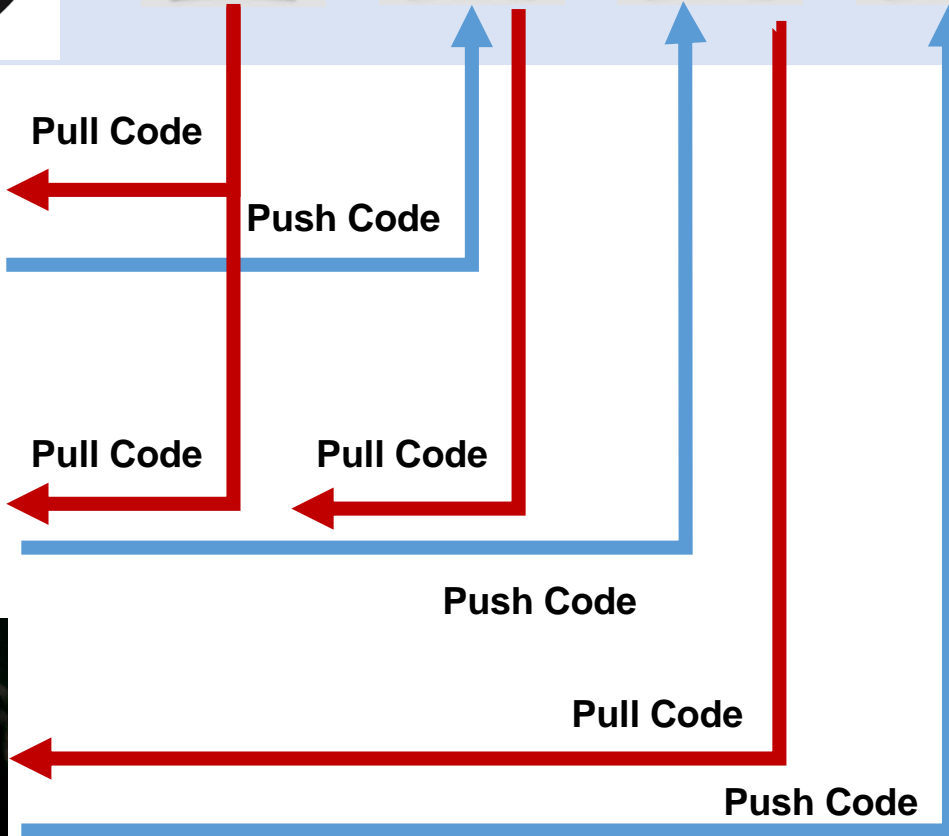
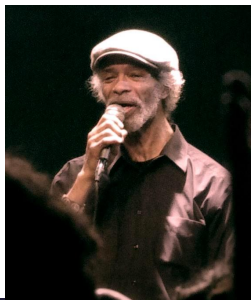
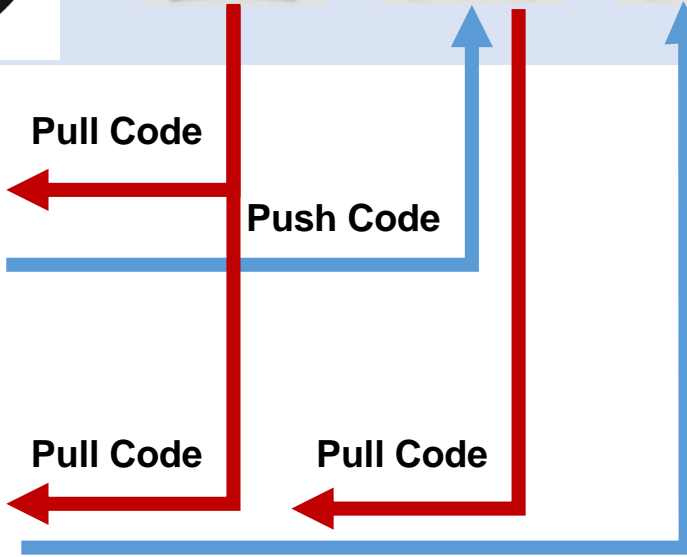
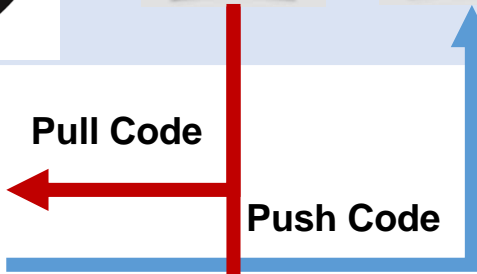
# So... What's this GitHub?

---

- 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**.



# Pushing and Pulling to GitHub



# ***Get Started with Git***

---

---

**Let's Create a Repo Together!**



# 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!**

**See guide and instructions in your slack channel**




# To improve your GIT Mastery... Here's a (Free) Course

 **Code School**  
a Pluralsight company

CoursesScreencastsPricing

Create Free AccountSign in



tryGit

GIT

Try Git

Play Course for Free

Paths > 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


Discover Git 25 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



**Olivier Lacan**  
Olivier was an innocent French web designer when Rails for Zombies infected his brain. The course motivated him to learn to build what he previously could only design. Now set free by the eloquence of Ruby, he spends his time learning, teaching, and shipping code for Code School.



**Gregg Pollack**  
Gregg is passionate about taking complex topics and teaching them efficiently. 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.

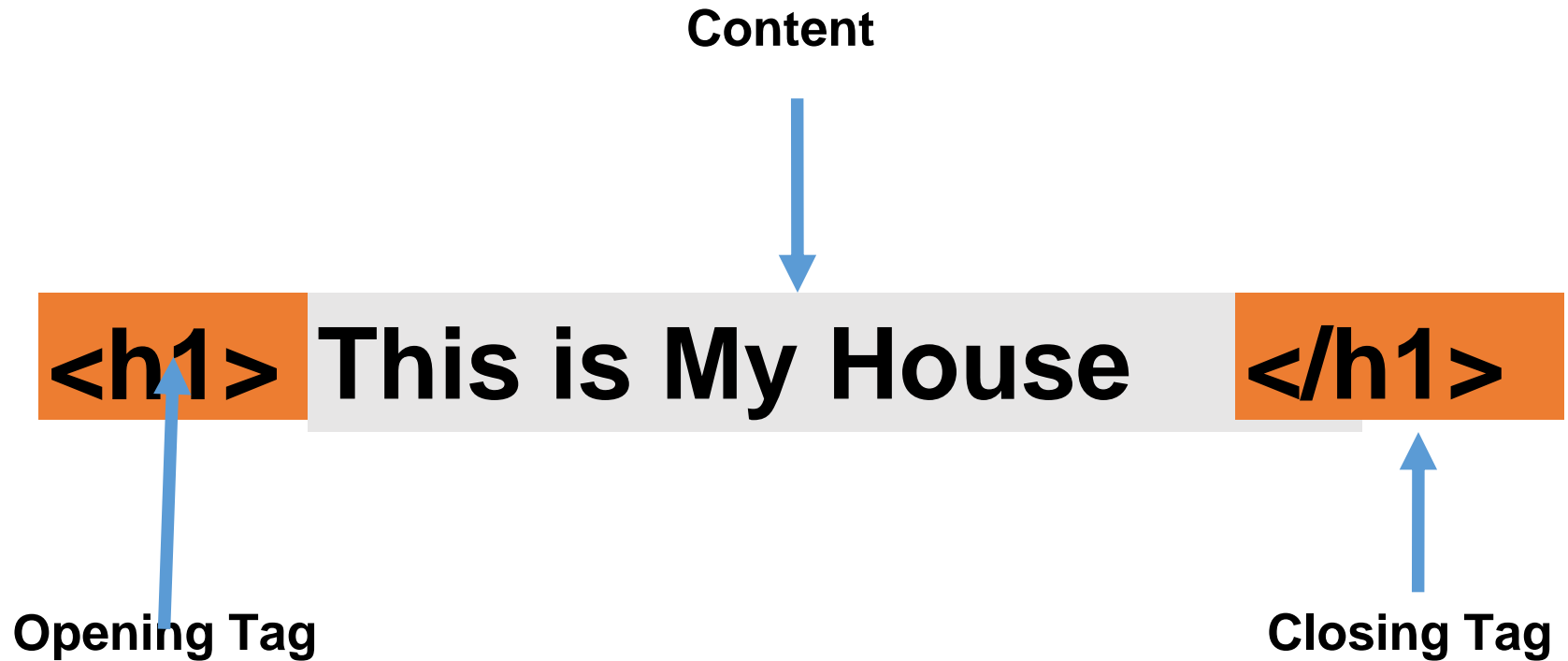
<https://www.codeschool.com/courses/try-git>

# ***HTML Round 2***

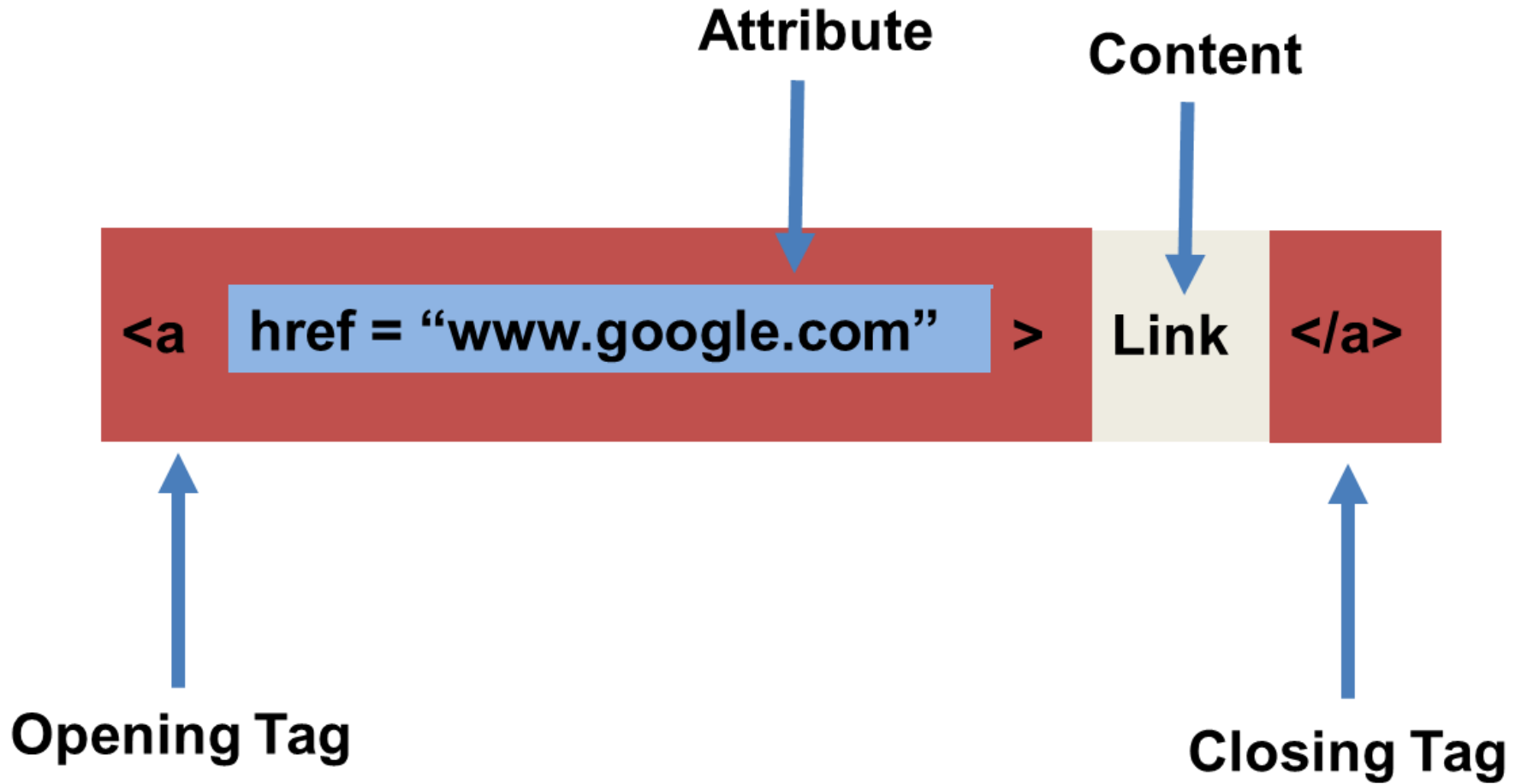
---

# HTML Syntax (Basic)

---

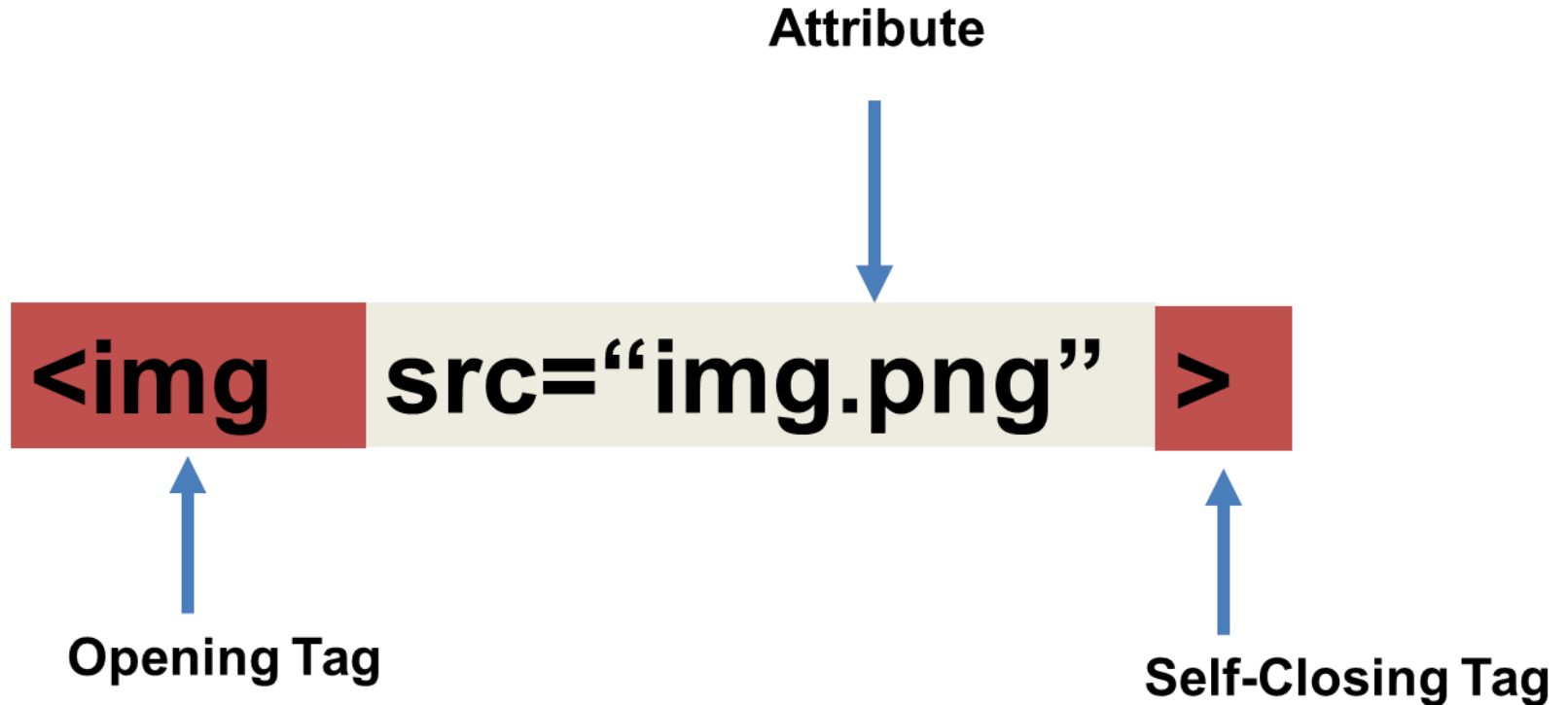


# HTML Syntax (with Attribute)



# 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:

- `<strong>` (bold), `<em>` (emphasis)
- `<img>` (images), `<a href>` (links), `<li>` (list items) , `<title>` (title), `<br>` (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

```
firstfile.html Quick.html
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.

**BUT – BEFORE PANIC SETS IN .....**

# > YOUR TURN!

## Student Bio

Your Name



Write a short paragraph or two about yourself, or use placeholder text from [www.lipsum.com](http://www.lipsum.com)

## Contact Info

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

Take a minute to design

How do I do these?

Tip: <https://placeholder.it/200x200>

# ***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 adipiscing 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>
  

  <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](#)

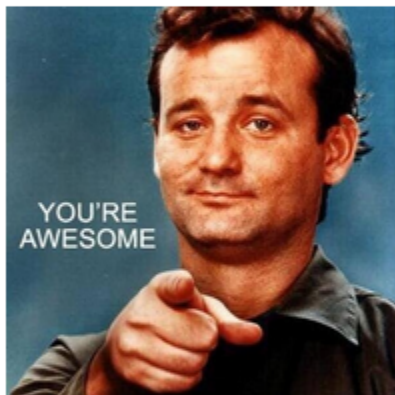
# Basic HTML Page - Result

## 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](#)

**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?

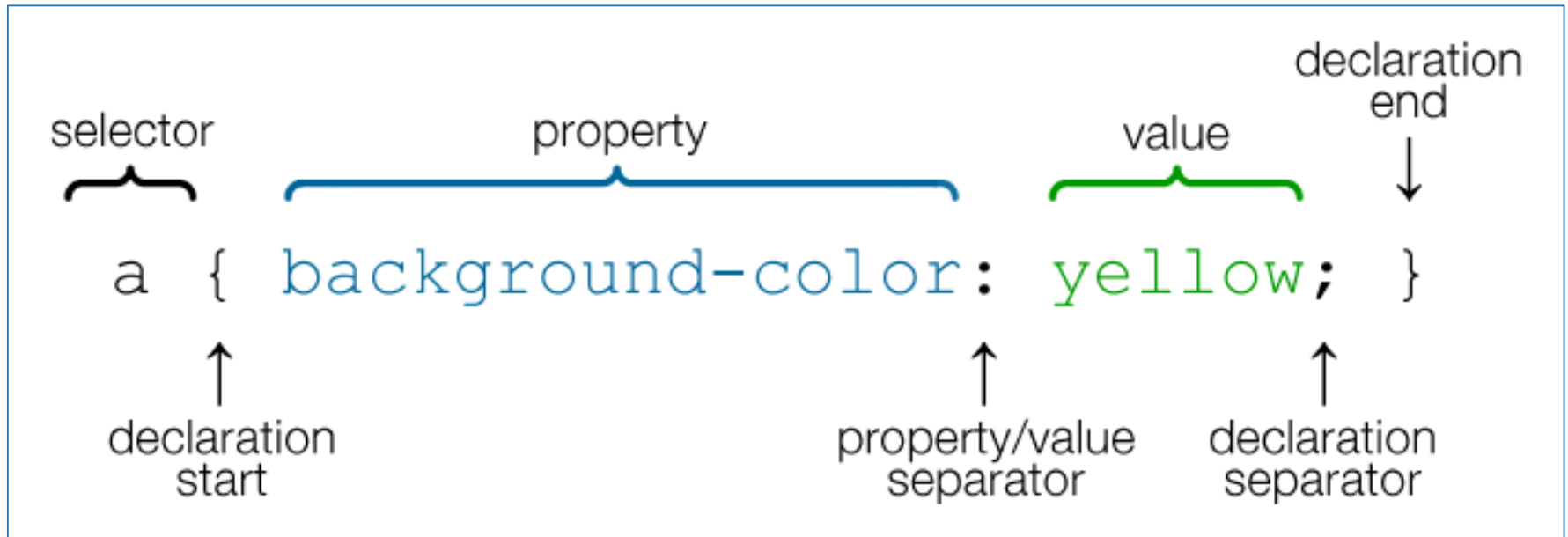


#### Menu Links

- 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;  
}
```



# Another CSS Example

---

- Example of using id .... Note the selector uses a #
- Note that with the use of float:left property and value, the page will line up on the horizontal

## **Example (HTML):**

```
<section id="main-bio">
```

```
  <h2>Your Name</h2>
```

```
  
```

```
  <p>Write a short paragraph or two about yourself</p>
```

```
</section>
```

## **Example (CSS):**

```
#main-bio,  
#contact-info,  
#bio-image {  
float: left;  
}
```

```
#main-bio {  
width: 100%;  
}
```

# 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.

---

**Let's take a closer look together**

**Open 05-BasicCSS in your code editor**

## 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

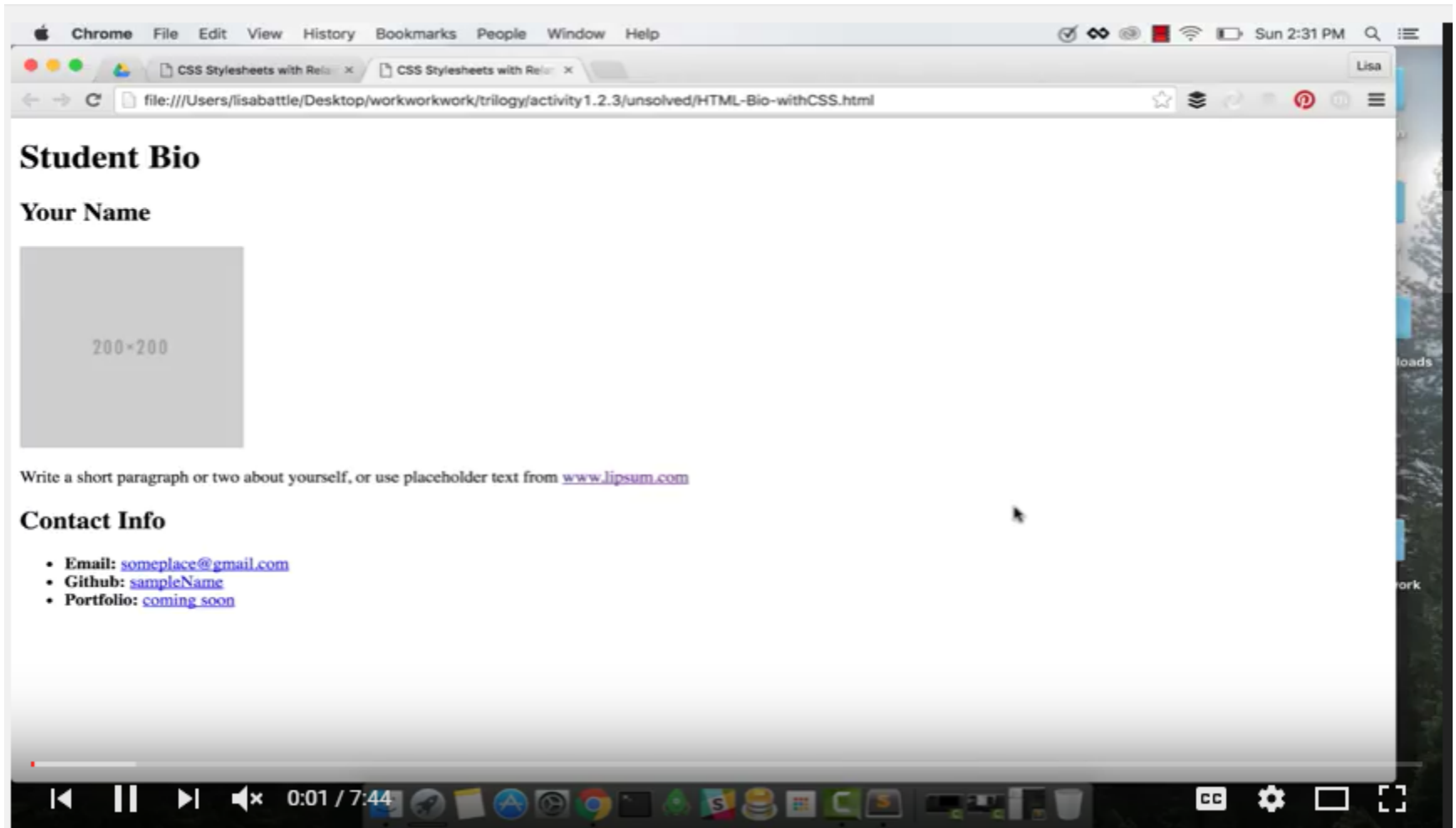


Write a short paragraph or two about yourself, or use placeholder text from [www.lipsum.com](http://www.lipsum.com)

### Contact Info

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

# Video Walkthrough!!

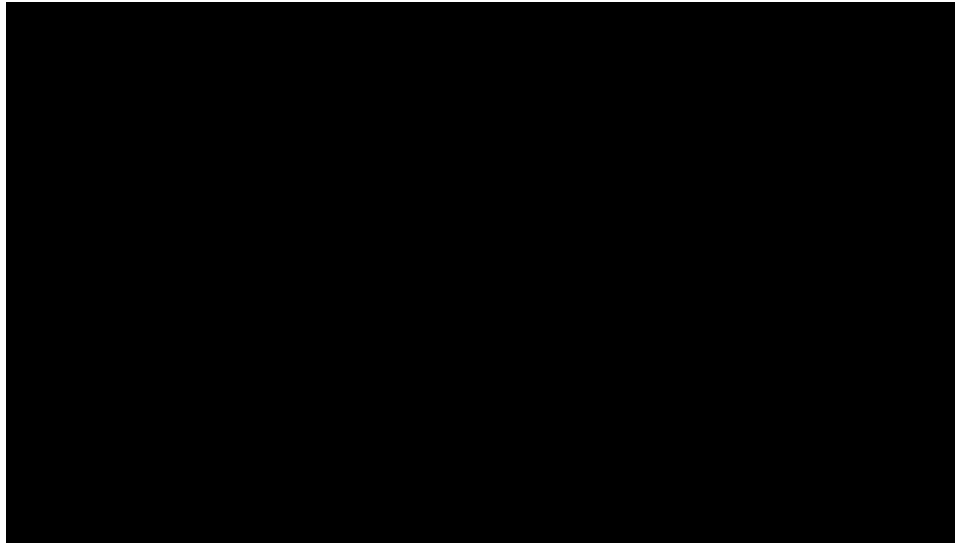


<https://www.youtube.com/watch?v=kMBinXTCrXI&list=PLgJ8UgkiorCnMLsUevoQRxH8t9bt7ne14&index=2>

# 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***

---