



Git'n Pro with HTML/CSS

Web Development Boot Camp
Lesson 1.2



Admin Items

How do I do this again?



How to Get Help

01

Practice, practice, practice: work individually or in groups.

02

Review in-class material (activities and slides).

03

Watch the class videos again.

04

Attend office hours, which are held 45 minutes before and 30 minutes after class.

05

Attend one-on-one sessions with your Student Success Manager (SSM) (to be announced by your SSM).

06

Contact your Student Success Manager anytime!

Today's Class!

Today's Objectives

Today we will:

01

Understand the importance of Git version control and how to use it.

02

Create GitHub repositories, push up code, and share with the class.

03

Create more HTML documents.

04

Learn how to properly use basic HTML tags.

05

Apply basic CSS styling to HTML documents.

Know Thyself

If you are a beginner to HTML/CSS and coding, your objectives are to:

- Continue to get comfortable with HTML.
- Be able to write a complete, basic HTML document (like in the one we just did).
- Understand the function of CSS and how it works with HTML.
- Be able to use Git and GitHub to upload code.

If you've had past exposure to HTML/CSS and coding and felt comfortable with the last lesson, your objectives are to:

- Aim to build up your skills.
- Clear up any questions or confusion you have 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.

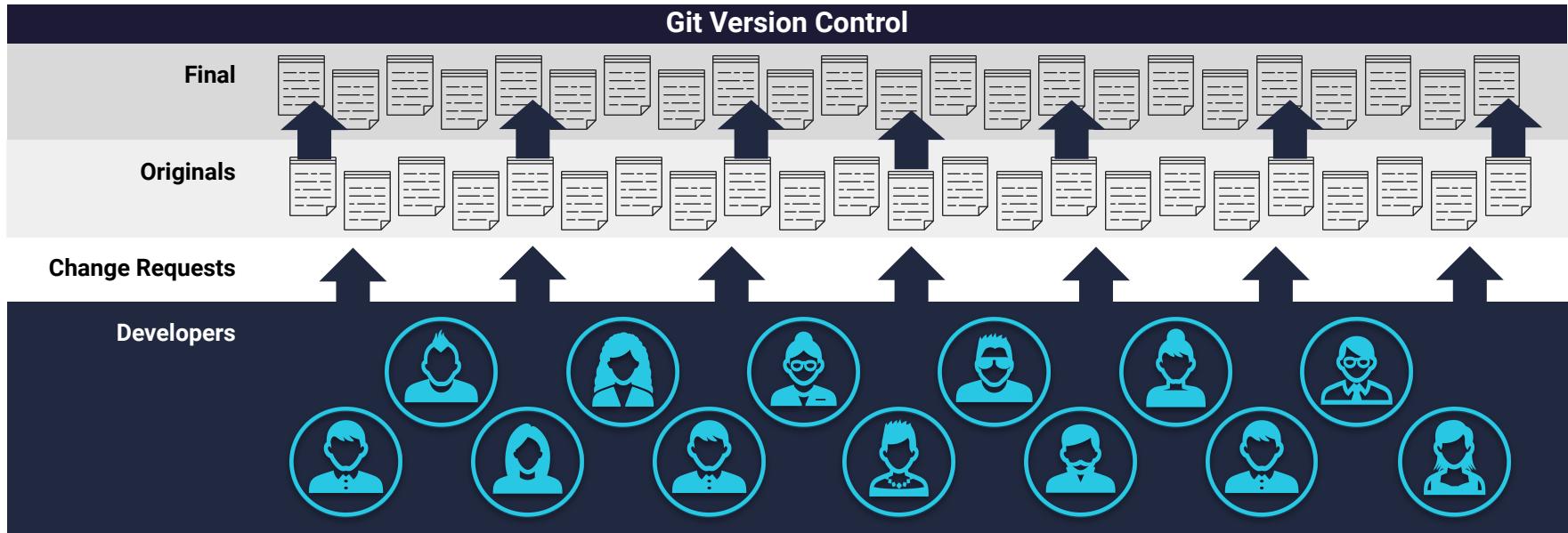
What Is Git?

Collaborative Coding

Modern web development is *highly* collaborative.

Teams are often extremely large and spread out across the country or world.

Apps are sometimes made up of hundreds or even thousands of files.



The Team's Task

Make a list of creative works you've written in the past.

Programming Team		
Maya Angelou	Anne Sexton	Gil Scott Heron
 A circular portrait of Maya Angelou. She is wearing a dark jacket over a white top and a blue sash with a gold medal around her neck. She is smiling and looking towards the camera.	 A circular black and white portrait of Anne Sexton. She has short, dark hair and is looking directly at the viewer with a neutral expression.	 A circular portrait of Gil Scott Heron. He is wearing a cap and a light-colored shirt, sitting at a piano and singing into a microphone. A band is visible in the background.

Maya Angelou & Gil Scott Heron Make Their Edits



Maya Angelou is programming away.



Maya Angelou's version

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



Gil Scott Heron is programming away.



Gil Scott Heron's version

```
<ul>
  <li>Free Will</li>
  <li>Pieces of a Man</li>
  <li>The Revolution Will Not Be
    Televised</li>
</ul>
```

Different Solutions

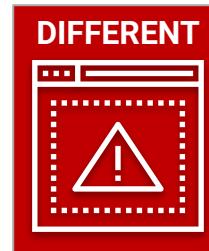


Maya Angelou's version

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



Gil Scott Heron's version



```
<ul>
  <li>Free Will</li>
  <li>Pieces of a Man</li>
  <li>The Revolution Will Not Be Televised</li>
</ul>
```

Resolution



Maya Angelou's version

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



Gil Scott Heron's version

```
<ul>
  <li>Free Will</li>
  <li>Pieces of a Man</li>
  <li>The Revolution Will Not Be Televised</li>
</ul>
```

Let's settle on this:

```
<ul>
  <li>Poems</li>
  <li>Albums</li>
  <li>Songs</li>
</ul>
```

Anne Sexton Writes Her Own Version



Anne Sexton's version

```
<ul>
    <li>The Double Image</li>
    <li>Heart's Needle</li>
    <li>Baby Picture</li>
</ul>
```

Anne Sexton Overwrites the Work of Her Teammates



Delete. Delete.
Delete. Delete.
Delete. Delete.

```
<ul>  
    <li>Poems</li>  
    <li>Albums</li>  
    <li>Songs</li>  
</ul>
```

```
<ul>  
    <li>The Double Image</li>  
    <li>45 Mercy Street</li>  
    <li>The Road Back</li>  
</ul>
```

The Group Project

Lesson: You should use version control because it helps you manage multiple developers working on a single codebase.



"Today we fret and pull on wheels, ignore our regular loss of time..." Or maybe we should just use Git.



Git Version Control

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

The Benefits of Git



01

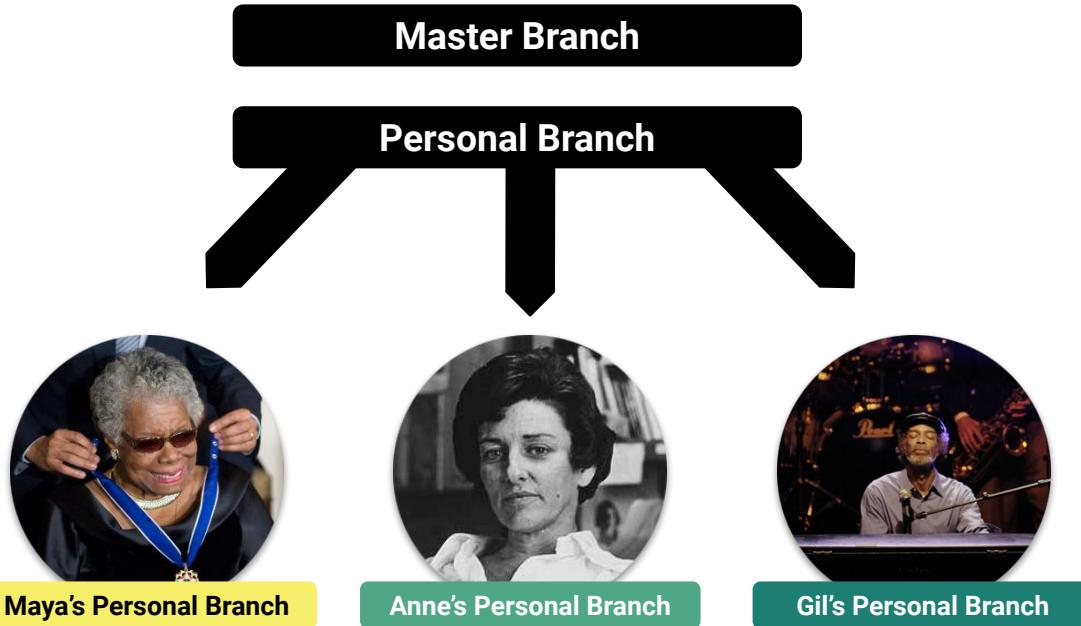
A process for resolving conflicts in code

02

Version history

The Group Project

Branch = personal copy



The Team Goes to Work



Maya Angelou's version

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



Gil Scott Heron's version

```
<ul>
    <li>Free Will</li>
    <li>Pieces of a Man</li>
    <li>The Revolution Will Not Be Televised</li>
</ul>
```

Maya Angelou Pushes First

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

No code conflicts

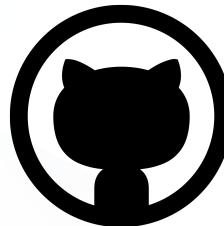


Master Branch



Maya's Personal Branch



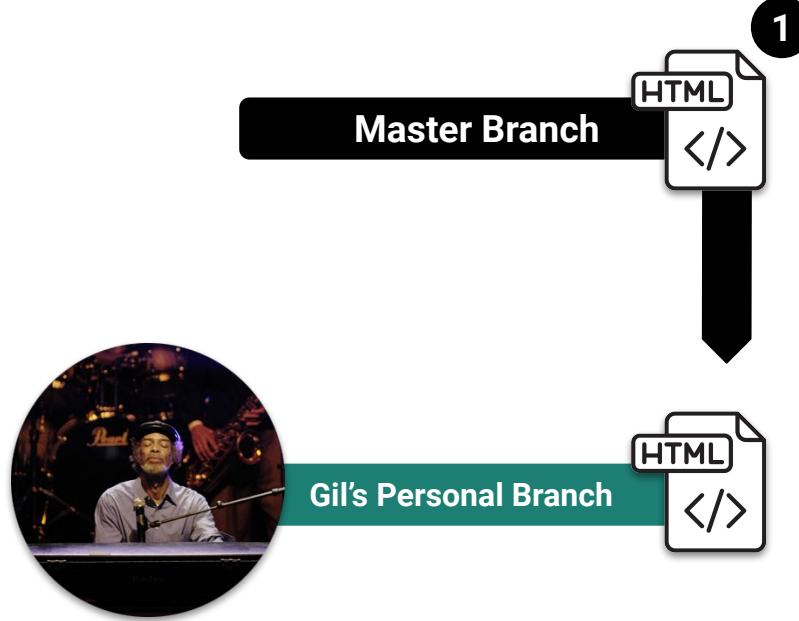


Rule: Pull first, and then push your changes.

Gil Scott Heron's Edits Are Ready



Rule: Pull first, and then push your changes.



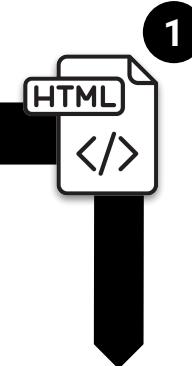
Gil Scott Heron Conflicts with the Master Branch



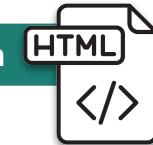
Git sees a conflict.

```
<ul>
<<<<<<< HEAD
    <li>Free Will</li>
    <li>Pieces of a Man</li>
    <li>The Revolution Will Not Be
Televised</li>
=====
    <li>On the Pulse of Morning</li>
    <li>I Know Why the Caged Bird Sings</li>
    <li>And Still I Rise</li>
>>>>>>> master
</ul>
```

Master Branch



Gil's Personal Branch



Gil Resolves

```
<ul>
<<<<<<< HEAD
    <li>Free Will</li>
    <li>Pieces of a Man</li>
    <li>The Revolution Will Not Be
Televised</li>
=====
    <li>On the Pulse of Morning</li>
    <li>I Know Why the Caged Bird Sings</li>
    <li>And Still I Rise</li>
>>>>>>> master
</ul>
```

```
<ul>
    <li>Poems</li>
    <li>Albums</li>
    <li>Songs</li>
</ul>
```



Gil's Personal Branch



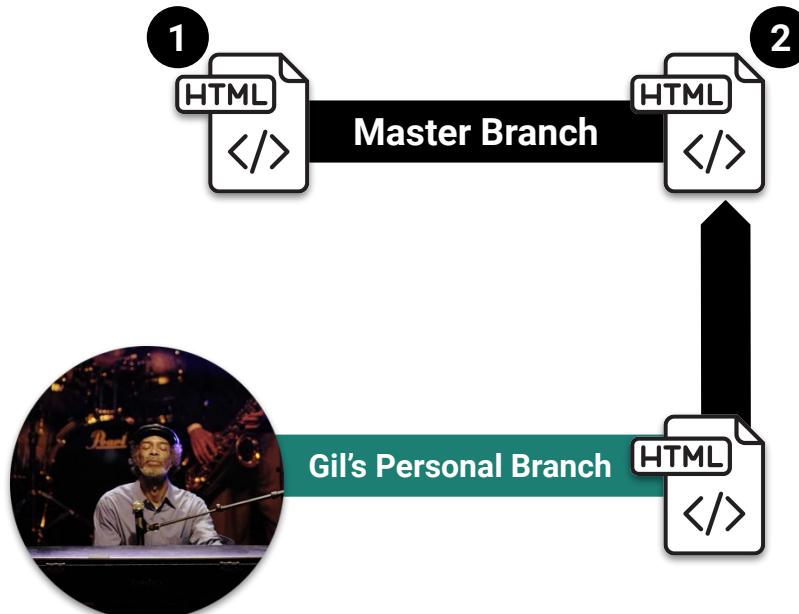
Gil Scott Heron Fixes and Pushes

Gil pushes (uploads) his revision to the main branch.



No code conflicts

```
<ul>
    <li>Poems</li>
    <li>Albums</li>
    <li>Songs</li>
</ul>
```



Anne Sexton Starts Her Work



Rule: Pull first, and then push your changes.

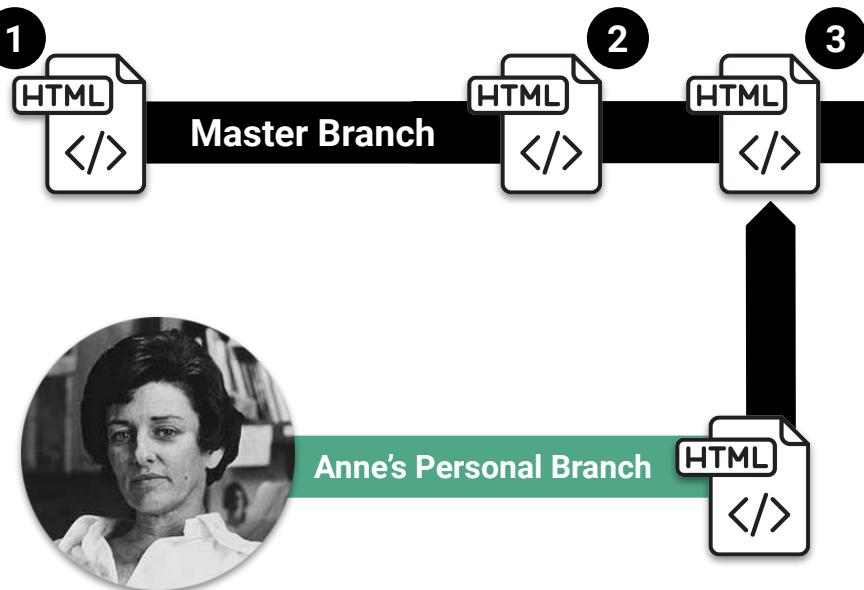


*look into my face
and you will know that crimes dropped upon me
as from a high building...
...by which I mean, I broke the rules.*

Anne Sexton Pushes

Anne Sexton pushes (uploads) her revision to the main branch, but she doesn't pull first. Because she did not pull first, she sees no conflicts in the code (and doesn't get Gil's work!). **This is not what we want.**

```
<ul>
  <li>The Double Image</li>
  <li>Heart's Needle</li>
  <li>Baby Picture</li>
</ul>
```



If Anne Had Made a Pull First...

Conflict!

```
<ul>
<<<<<<<< HEAD
    <li>The Double Image</li>
    <li>Heart's Needle</li>
    <li>Baby Picture</li>
=====
    <li>Poems</li>
    <li>Albums</li>
    <li>Songs</li>
>>>>>>> master
</ul>
```

CONFLICT!

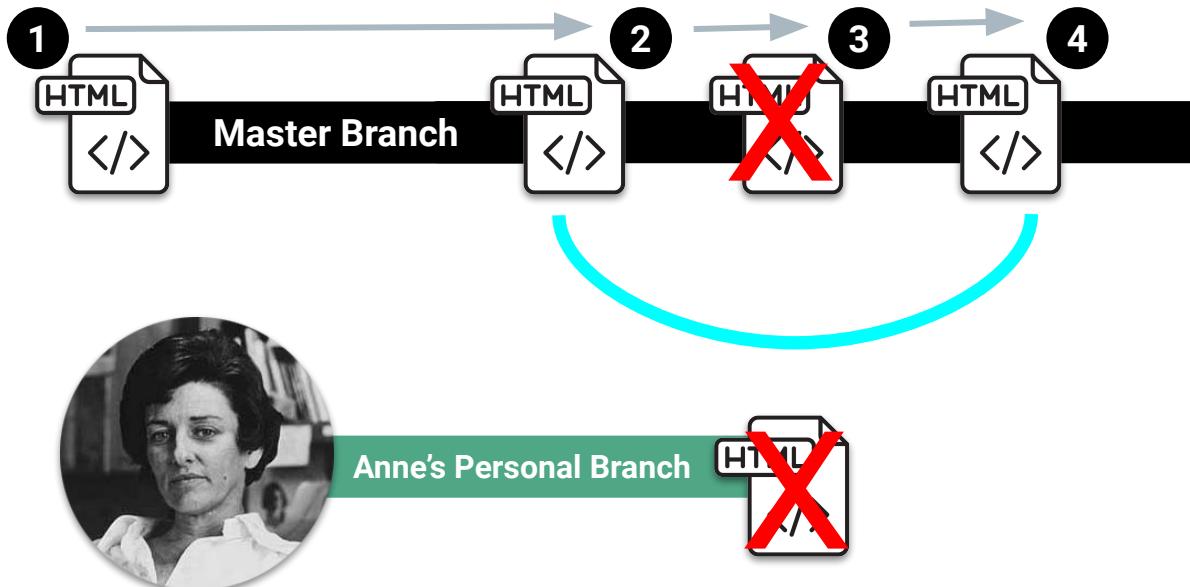




The **overwritten** work
is discovered!

Roll Back

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





Activity:

Explain Git (Version Control)

Suggested Time:
3 Minutes



Activity: Explain Git (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 key advantages of using a version control system

Suggested Time: 3 Minutes



What Is GitHub?

01

GitHub is a web-based hosting service to store code online.



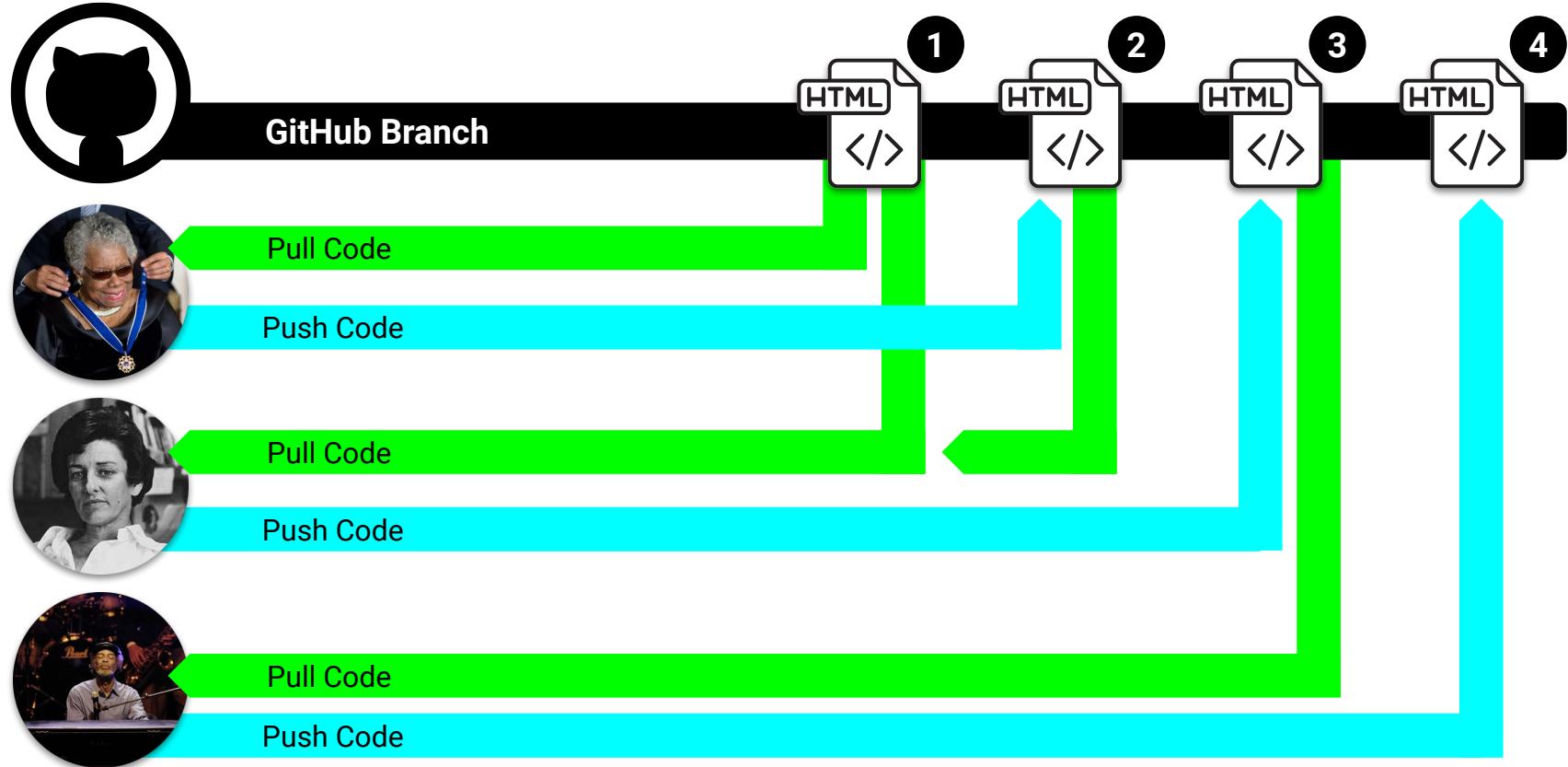
02

It allows developers to pull (download) code or push (upload) code to the same repository (directory).

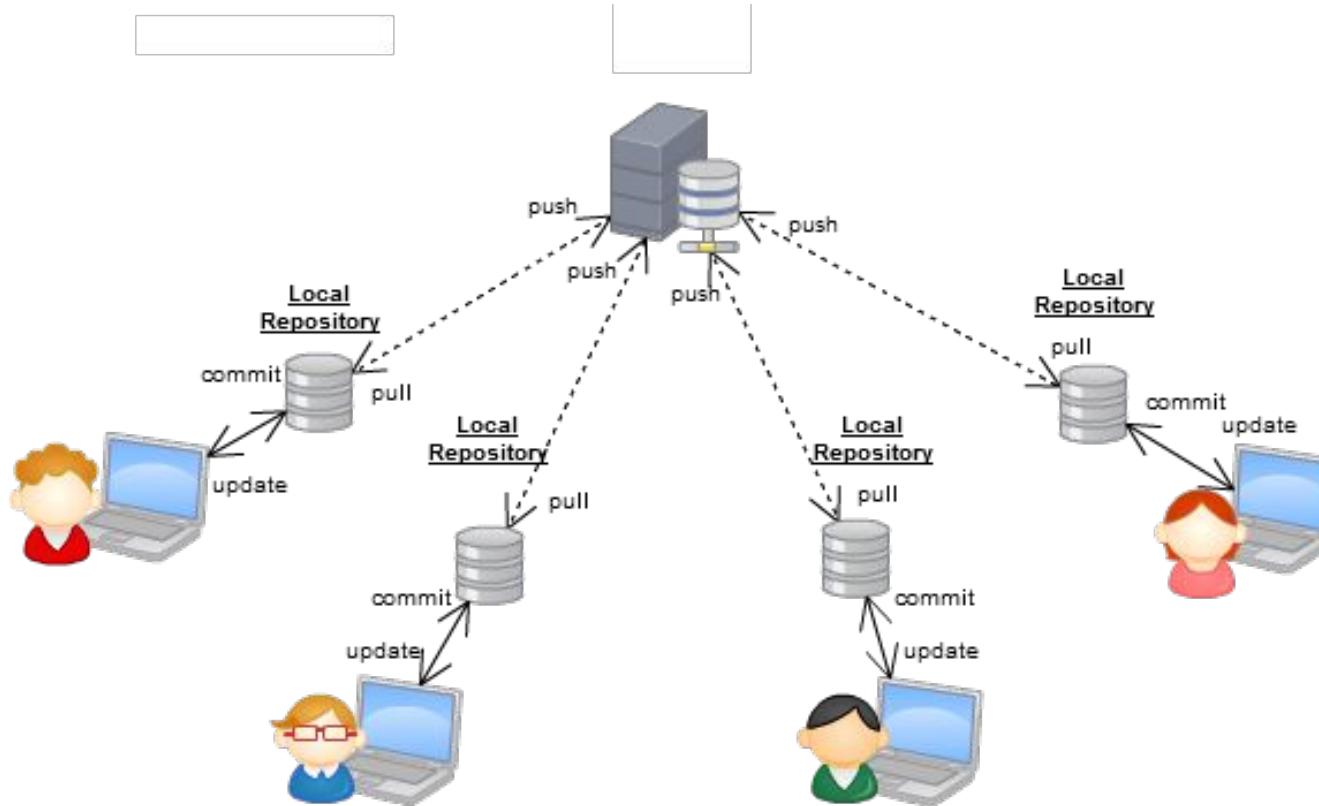
03

It also allows developers to view histories of code changes and track issues.

Pushing and Pulling to GitHub



How Git Manages Change



Central Repository

a.k.a. Remote Repo

a.k.a. 'origin'

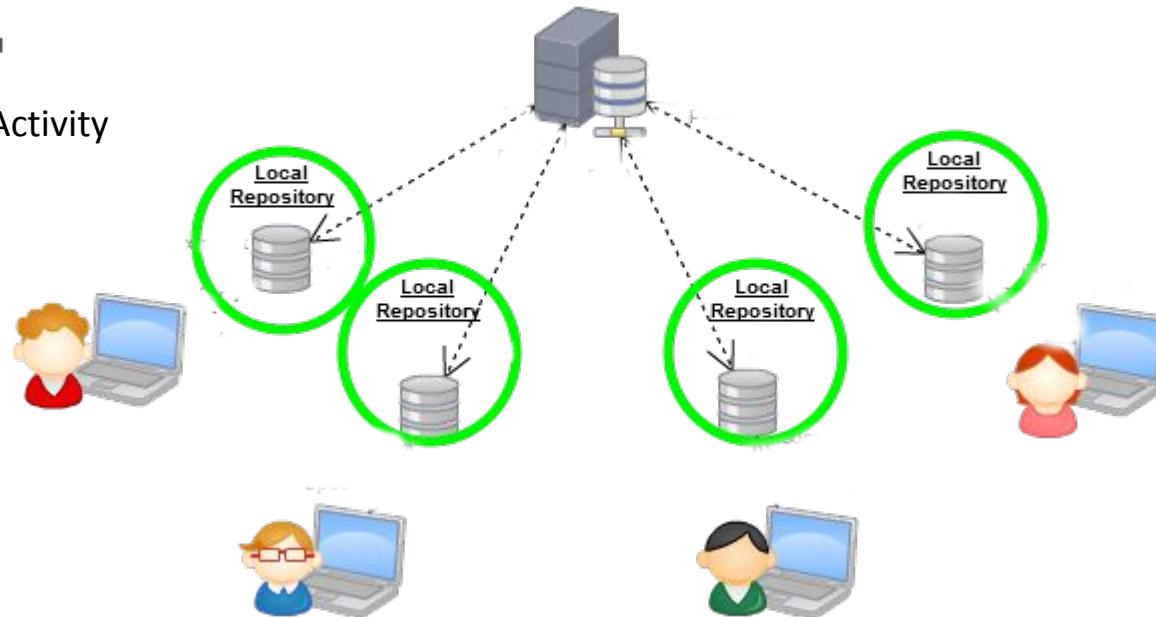


Easy User Gets a Copy of the Repo



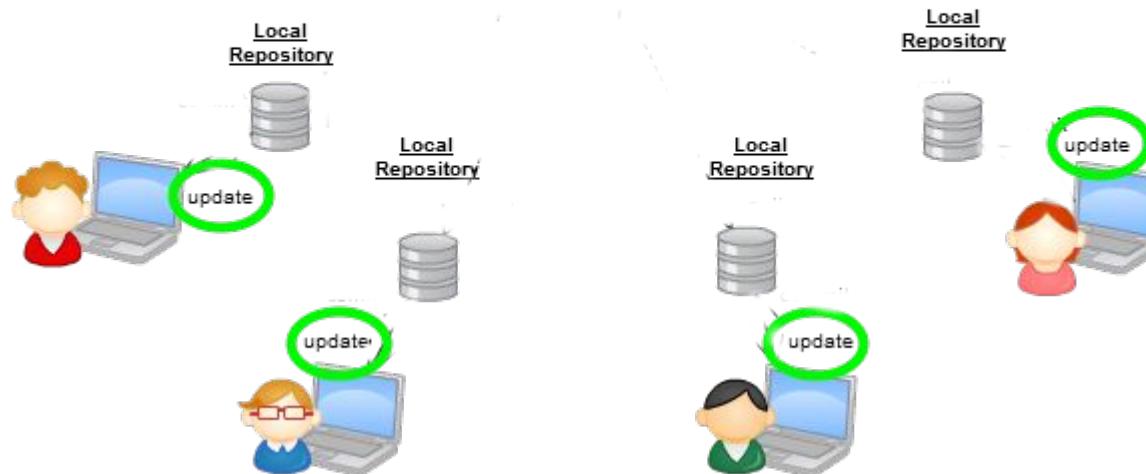
a.k.a. Cloning the Repo

Network Activity



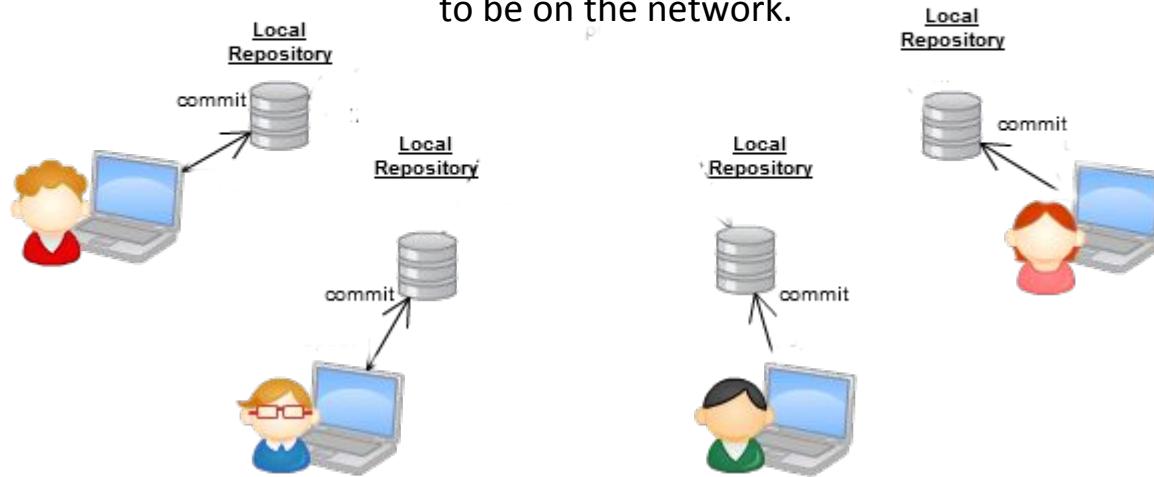
Each User Alters a File

- Edit the file (in VS Code, whatever)
- Does not involve the Local or Remote Repos



Users Check In File to Local Repo

- We call this action a “commit”
- This is an action on your Local Repo
- Does not involve the Remote Repo. You don’t need to be on the network.

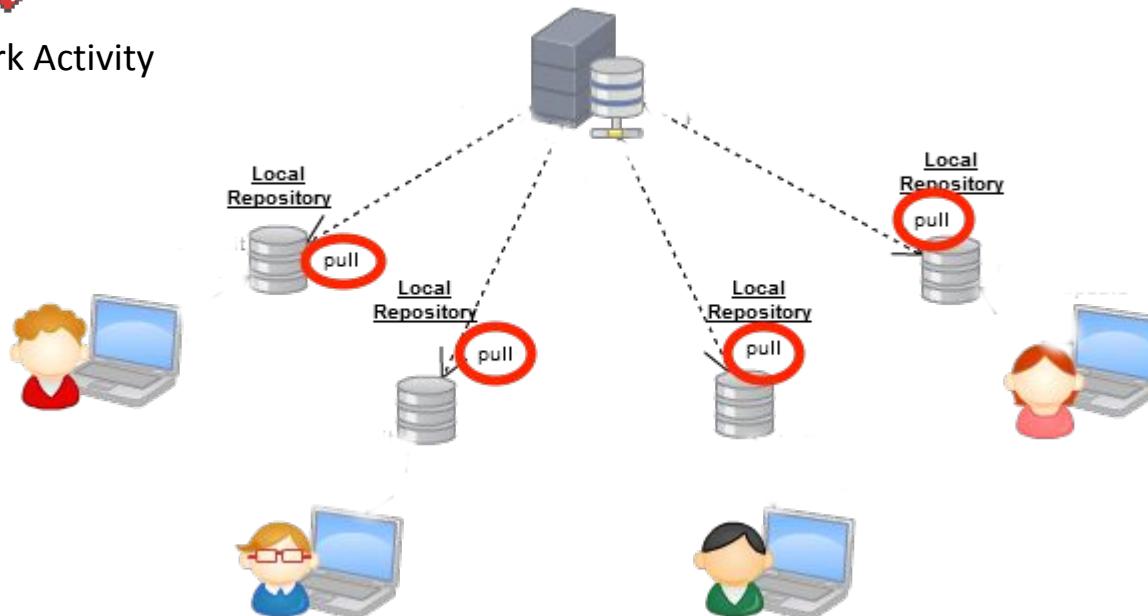


Users Pull Latest From Origin



- Do a 'pull' from the Remote Server
- This syncs your Local with any new changes on the Remote
- ...for example, changes your teammates may have made to the file

Network Activity



Fix any conflicts

- If someone recently made edits your files may be in conflict
- Git will tell you if there is a conflict
- Edit the file (a.k.a. “resolve the conflict”)

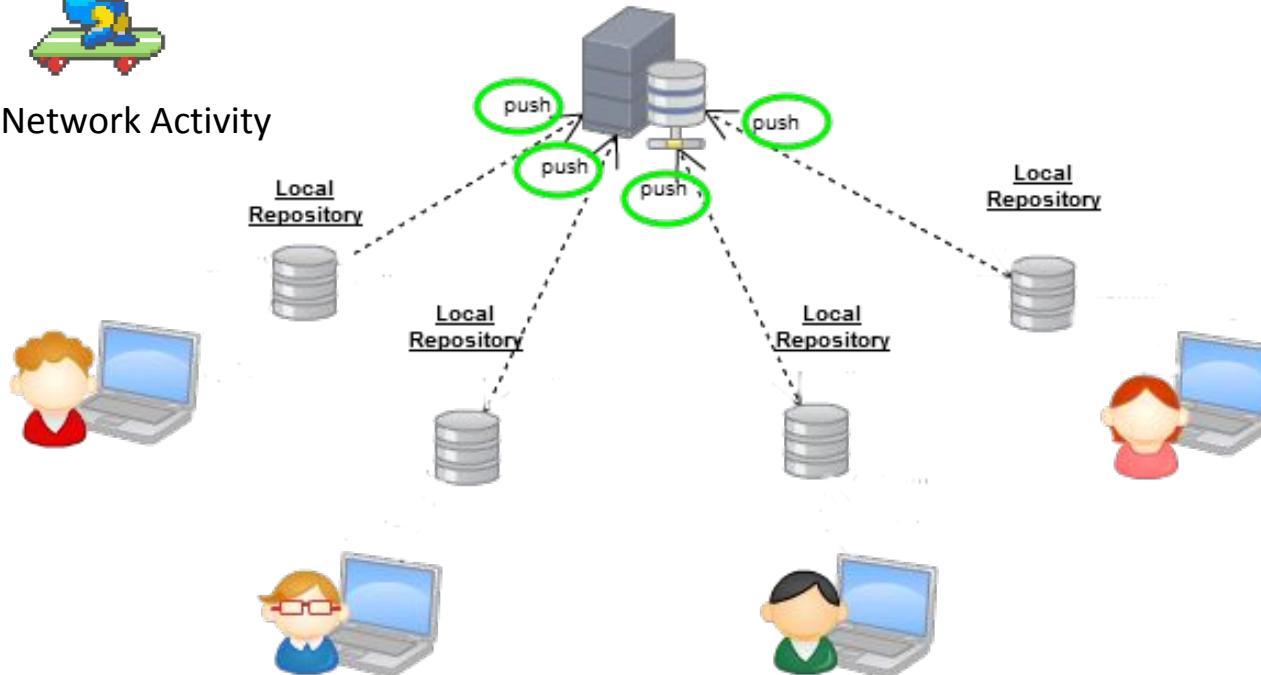


Push Your Changes



- Push your changes to the Remote Repo. You're done!
- What if...there are some new changes on the Remote?
- Git will tell you to do another pull first

Network Activity



Get Started with Git



Instructor Demonstration

Git

Basic Git Commands

These are the five basic Git commands to get started:

01

`git clone`

02

`git add`

03

`git commit`

04

`git push`

05

`git pull`

Basic Git Commands

These are the five basic Git commands to get started:

git clone

Copies an entire repo (to begin)

git add

Adds file changes for inclusion in Git, AKA “Staging”

git commit

Makes the change to the local repo

git push

Sends changes to hosting service

git pull

Downloads latest version of repo from hosting service

The most important Git command of all

\$ git status

When you need a gut check run git status

Let Git tell you the current status of your repository. For example, Git will indicate which files have been changed or added as a result of the changes you've made on your branch. Then Git offers suggestions on what to do, offering commands on how to stage or commit those files.

When in doubt, run git status and read what Git tells you!



Activity:

Git Add, Commit, Push

Suggested Time:
20 minutes



Activity: Git Add, Commit, Push

Using GitHub and the command line:

1. Create a new **public GitHub repository** and name it whatever you like. Be sure to check the box to initialize this repository with a README.
2. **Clone** the repo to your local directory.
3. Create an **HTML file** inside the local directory.
4. **Add, commit, and push** the code to GitHub.

Bonus:

1. Find a partner in class, and **fork their** repository to your own GitHub account.
Clone this forked repository to your local directory.
2. **Add, commit, and push** the code back to your forked copy.
3. Submit a **pull request** to send your changes to your partner's repo.

Suggested Time: 20 Minutes



Still a Bit Lost? Don't Worry!

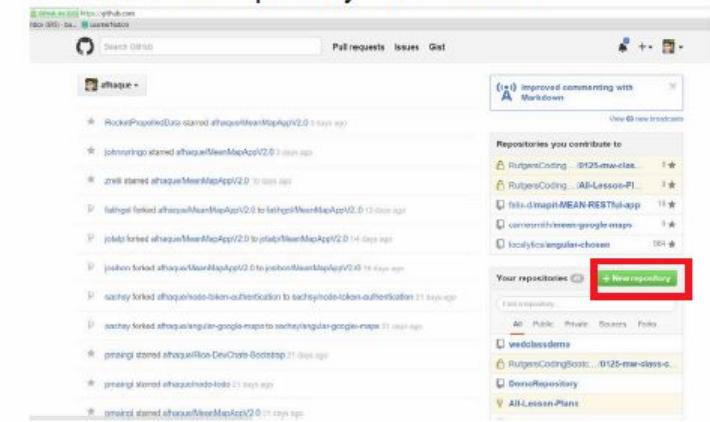
Follow this handy guide!

Practice a few times on your own before the next class.

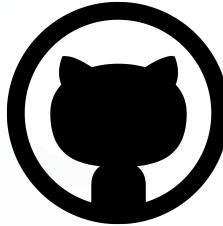
Steps to Uploading Your Code to GitHub

Step 1

Create a New Repository in GitHub.com



The screenshot shows a user's GitHub profile page. At the top, there's a search bar and navigation links for Pull requests, Issues, and Gist. Below the search bar, there's a list of repositories contributed to by the user 'ahfauque'. On the right side, under 'Your repositories', there's a green button labeled 'Create New Repository'. This button is highlighted with a red rectangular box. Below the button, there are filters for Public, Private, Sectors, and Parks, and a dropdown menu showing 'webclassdem'.



If you're still lost, here's a (free) course on how to use GitHub:

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

HTML Round 2

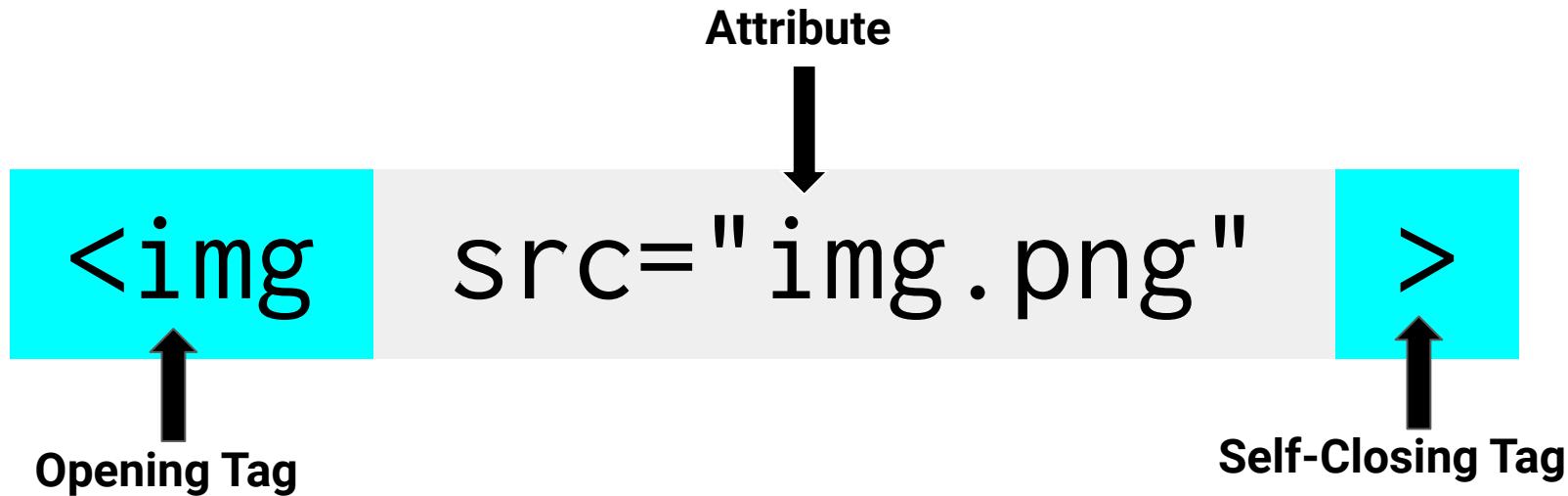
HTML Syntax (Basic)



HTML Syntax (with Attribute)



Tricky Tags (Self-Closing)



Important Common Tags

Headings:	Containers:	Others:	
<h1> </h1>	Heading 1 (Largest heading)	<html> </html>	Wraps the entire page
<h2> </h2>	Heading 2 (Next largest heading)	<head> </head>	Wraps the header of the page
<h3> </h3>	Heading 3	<body> </body>	Wraps the main content
		<div> </div>	Logical container
		<p> </p>	Wraps individual paragraphs
		<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.

<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 text box

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:

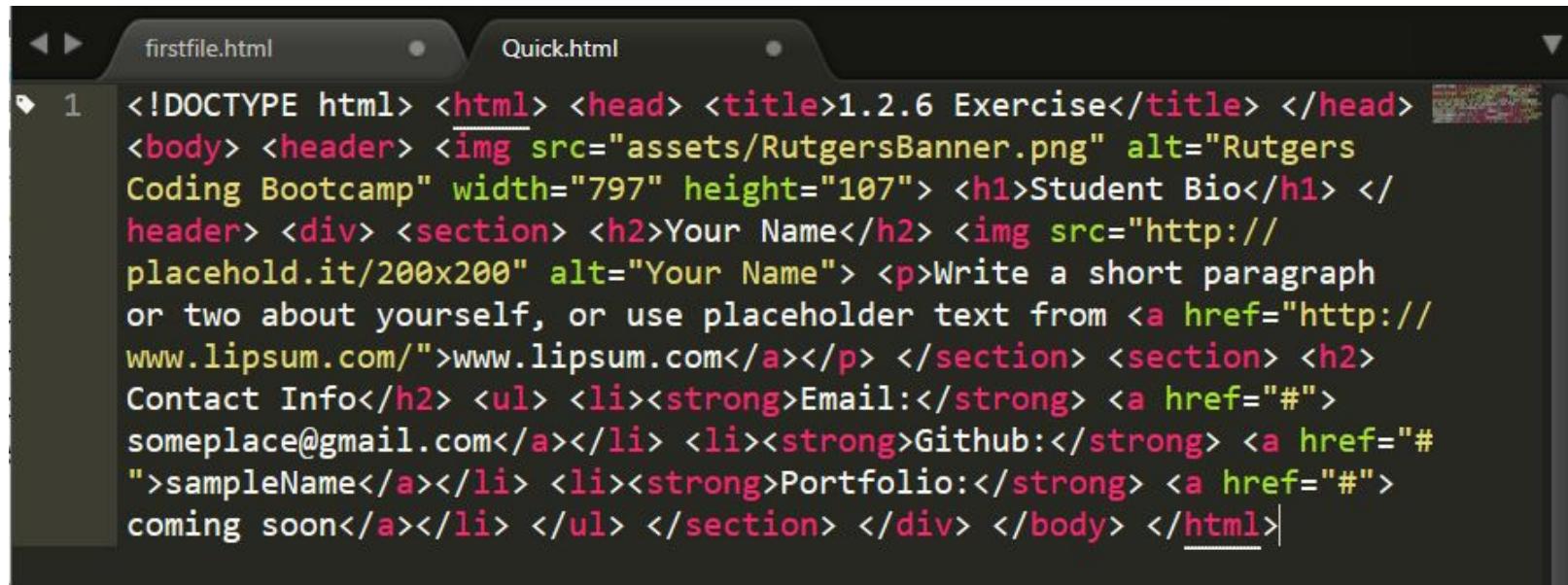
Last name:

Note that the form itself is not visible.

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

Ugly HTML

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



```
<!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>
```



Activity: Basic Student Bio

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 will be sent via Slack)

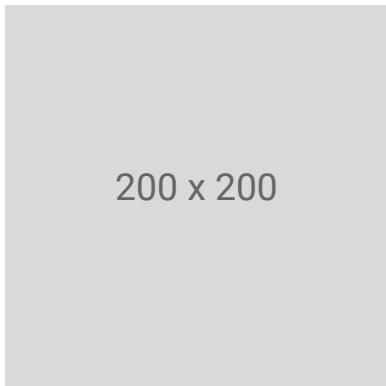
Suggested Time:
20 minutes



Activity: Basic Student Bio

Student Bio

Your Name



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

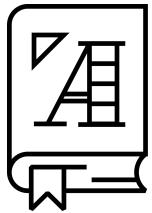
Contact Info

- Email: someplace@gmail.com
- GitHub: [Sample Name](#)
- Portfolio: [Coming Soon](#)



CSS Stylin'

HTML and CSS Definitions



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 aesthetics of a webpage, including layouts, colors, fonts, and content placement. (JavaScript is the language that deals with logic, animation, etc.)

HTML/CSS Analogy

HTML Alone	HTML and CSS
Like writing papers in Notepad	Like writing papers in Microsoft Word
Can only write unformatted text	Can format text, page layout, alignment, and more 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>
        

        <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 Awesome Header

 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.



Menu Links

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

Basic HTML Page: Result

Awesome Header

Smaller Awesome Header

Even Smaller Awesome Header

 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.



Boring

Menu Links

- Google
- Facebook
- Twitter

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>
```

Enter CSS: Result

Awesome Header

Smaller Awesome Header

Even Smaller Awesome Header

Lore 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.



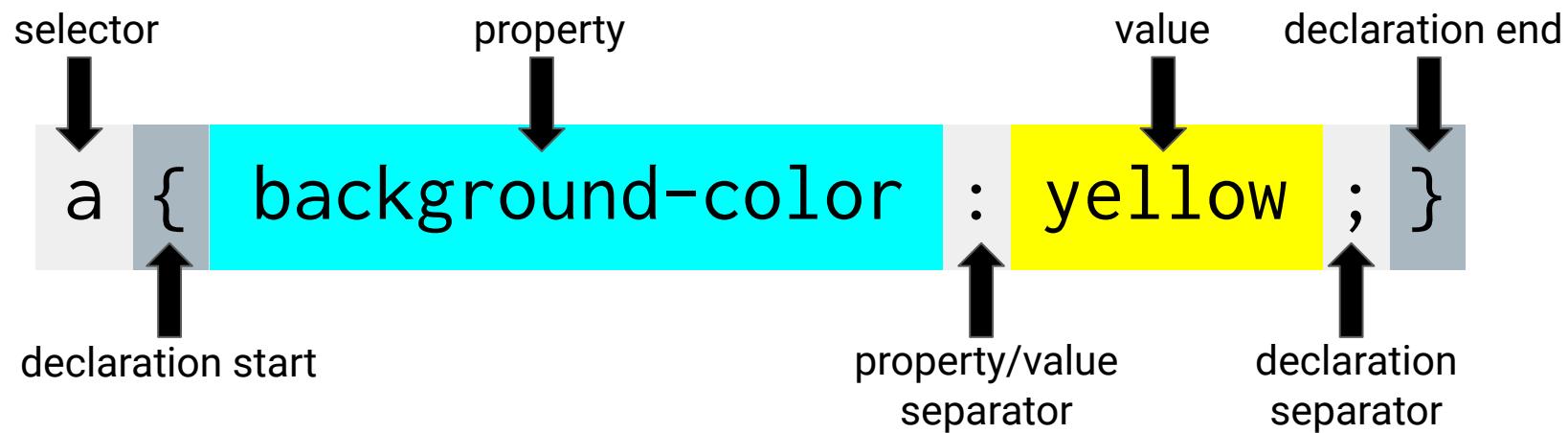
Menu Links

- 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 following example, the header would become 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)	Example (CSS)
<p class="bigBlue">Header</p>	<pre>.bigBlue { font-size: 100px; color: blue; }</pre>

Key CSS Attributes

Font and Color:

color: sets color of text

font-size: sets size of the font

font-style: sets italics

font-weight: sets bold

Alignment and 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:

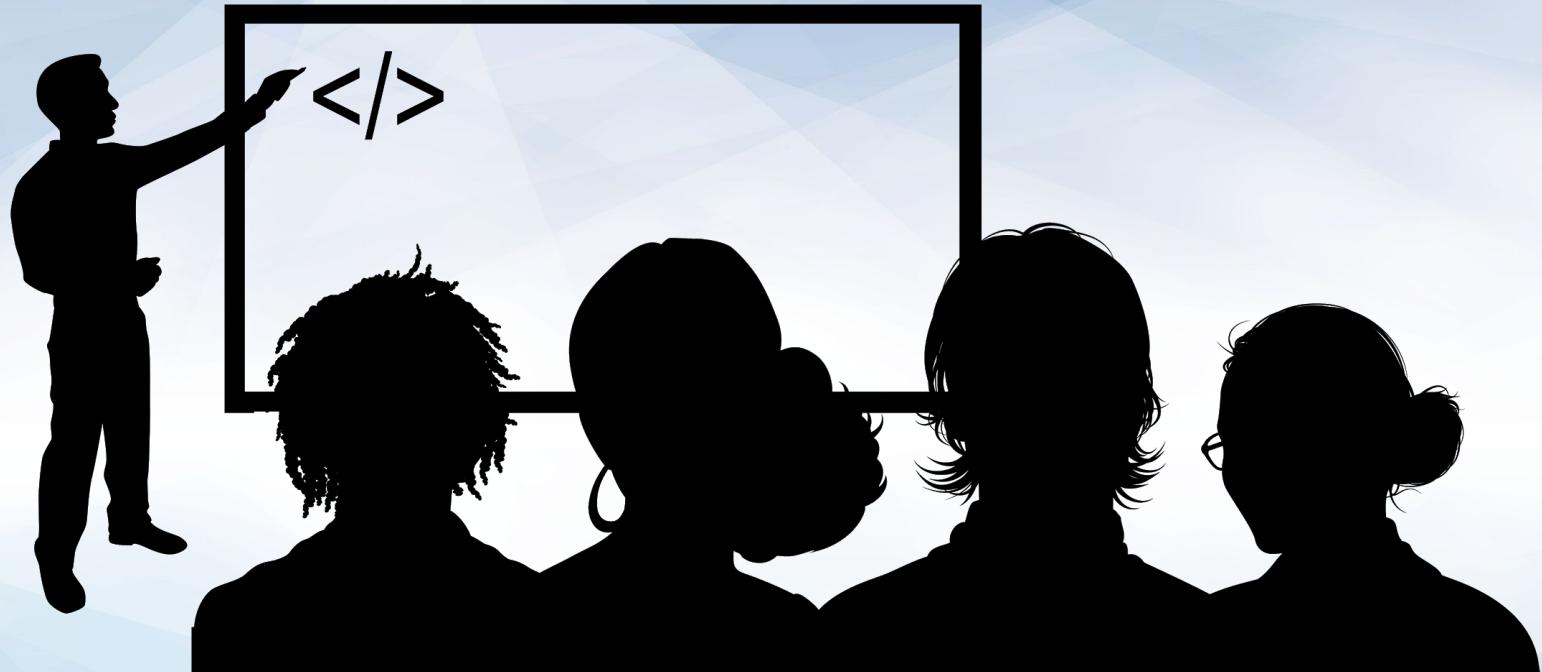
background-color: sets background color

background-image: sets background image

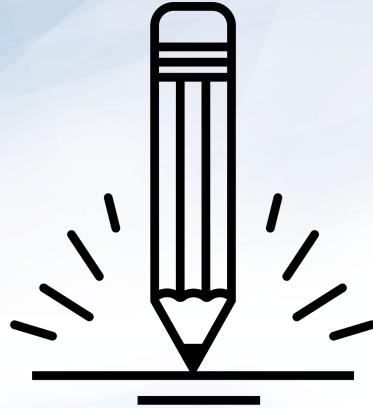


Powerful Duo

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



Instructor Demonstration CSS Basics



Activity: CSS-Styled Bio Page

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

(Additional instructions will be sent via Slack)

Suggested Time:
20 minutes



Activity: CSS-Styled Bio Page

Student Bio

Your Name



200x200

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](#)





Time For a Quick Video

[Student Bio Layout](#)

Still a Bit Confused?

- Remember the video guides for key activities (such as the last one).
- If you EVER feel like you are falling behind, use the video walk-throughs to catch 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 however we can.



Questions?