

# Welcome to University of Pennsylvania Coding Bootcamp

**Wifi Network:** Coding Class  
**Password** - SASPennovation18!

**Sign up for GitLab (if you haven't already):**  
[upenn.bootcampcontent.com](http://upenn.bootcampcontent.com)

**Open:**  
Slack desktop app & **ADD PHOTO!**  
**Log in to BCS:**  
Mark yourself present & **ADD PHOTO!**



# **By the end of this session, you will:**

- Come to know your classmates as the community that you will rely on for collaborative learning.
- Know the staff who will be providing holistic support throughout the program.
- Understand the minimum requirements in order to successfully graduate from this boot camp.
- Be able to list ways to get help and support at your moments of need.

# **Activity: Present & Accounted For**

- Navigate to [www.bootcampspot.com](http://www.bootcampspot.com)
- Sign in
- Click Sessions
- Find Today's Date
- Mark Your Attendance

# **Josh Hochman**

**Student Success Manager (SSM)**

---

My goal is simple: help you successfully complete the program!

[jhochman@bootcampspot.com](mailto:jhochman@bootcampspot.com)



# **Stacy Clements**

**Associate Director, Professional and Organizational  
Development, College of Arts and Sciences**



**[stacymc@sas.upenn.edu](mailto:stacymc@sas.upenn.edu)**

# Rita McGlone

Executive Director, Professional and Organizational  
Development, College of Arts and Sciences



[mcgloner@sas.upenn.edu](mailto:mcgloner@sas.upenn.edu)

# Dr. Robert Hollebeek

Professor

Research Areas: Particle Physics, Proton Therapy



[bobh@physics.upenn.edu](mailto:bobh@physics.upenn.edu)

# Scott Wittrock

## Teaching Assistant

---

- Developer, designer, and entrepreneur
- Previously worked at a number of tech companies including Slyce, Blippar, Ipreo, and Lumi
- Supporter of cycling and public transit, love to travel, I'm a productivity and process enthusiast, nature lover and I'm most relaxed when cooking



# Vipinjeet Sandhu

Teaching Assistant

---

- Learned how to program at age 8 on my Apple IIgs and have been in love with coding ever since!
- Took formal classes in coding throughout my life and worked as a desktop programmer.
- Extensive history teaching college students and adult learners in Computer Science
- Through empathy and tenacity, I eternally chase an unending series of "aha!" moments and look forward to racking up more of those victories as your ally in your quest to become a full-stack developer.

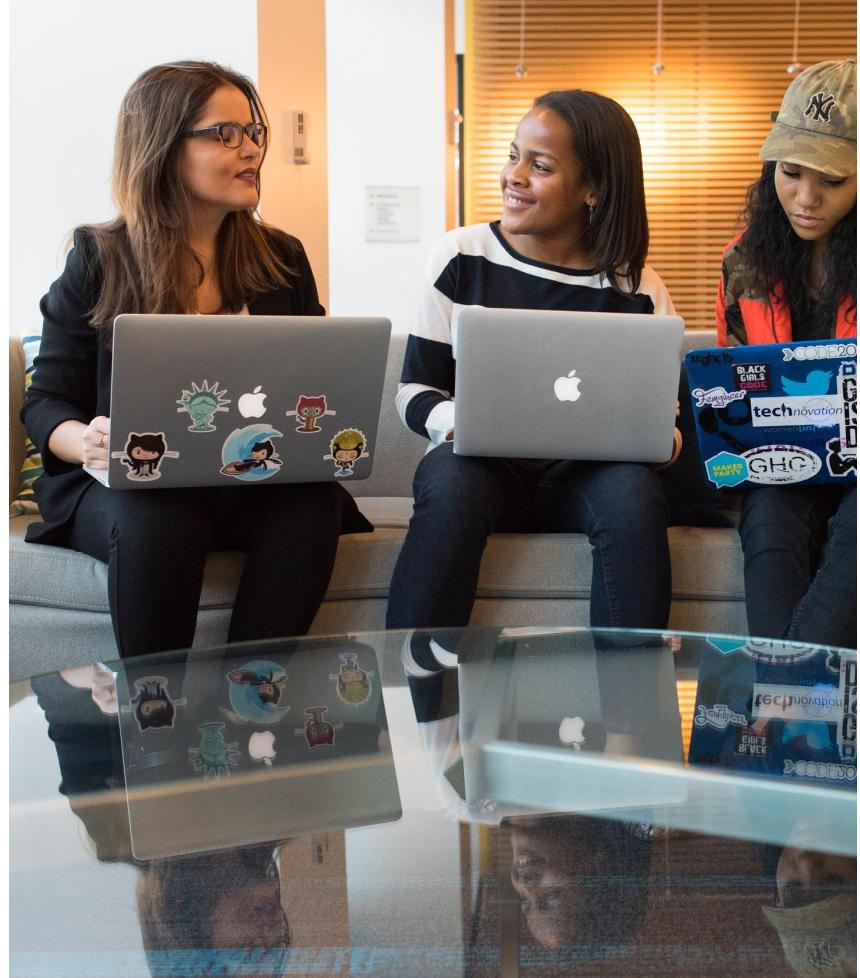


# Your turn!

---

In 30 seconds or less, share your:

- Name
- Location
- Background (Career, Education, Interests)
- What excites you about this class!



# Dave Leonhardt

Instructor

---

- I grew up in southern New Jersey and now live in the Fishtown area
- I have a B.S. in Computer Science from Rowan University and am a former Application Engineer/Scrum Master
- Outside of coding I enjoy playing guitar & bass, cooking, video games, and card games





## YOUR CAREER SERVICES TEAM

---

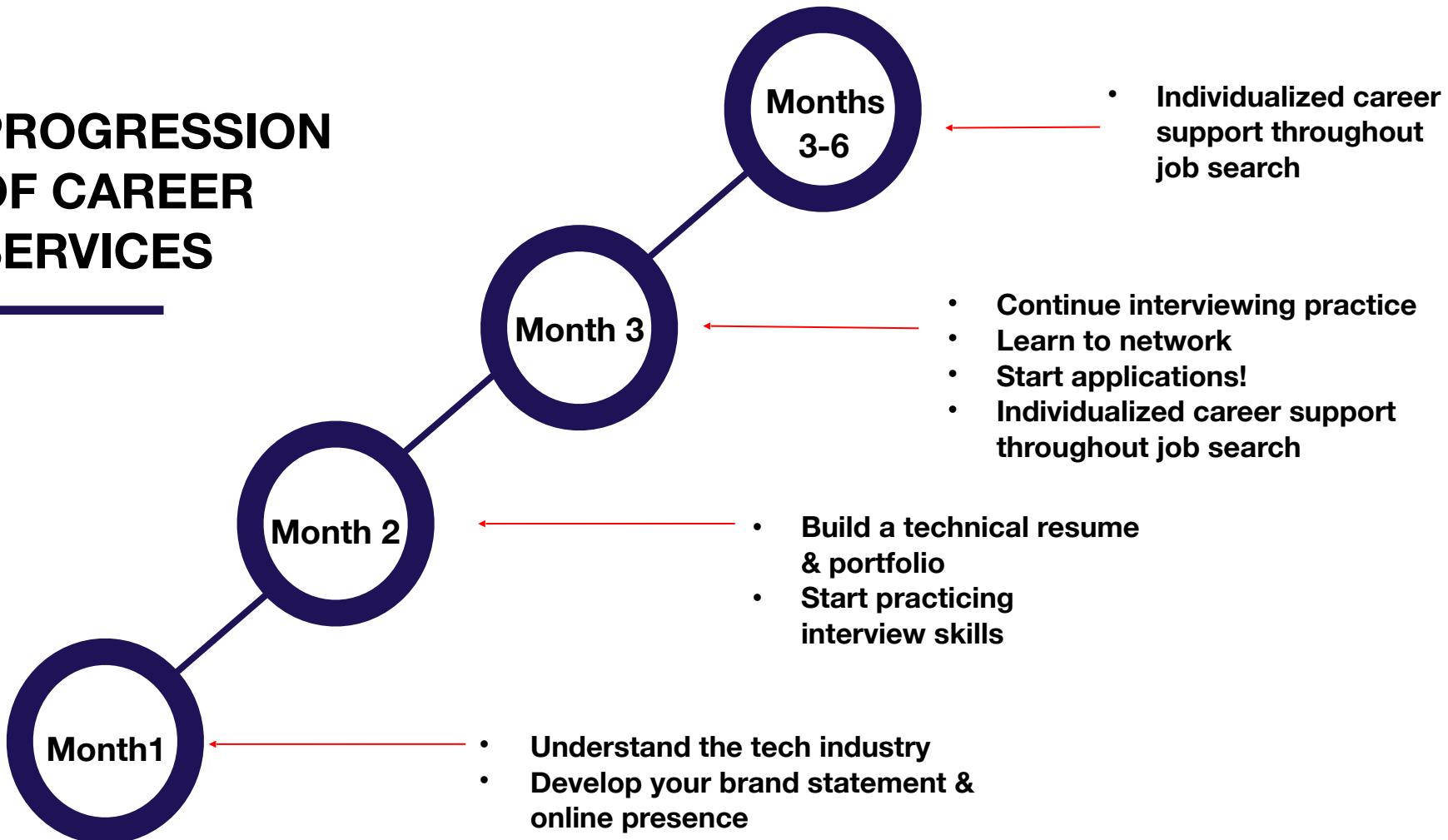
PROFILE  
COACHES

CAREER  
COACH

YOU

# PROGRESSION OF CAREER SERVICES

---



# SUPPORT STRATEGIES

You are responsible for your **succes**, but you're not alone!



You

Live Chat  
on  
Bootcampspot

Schedule a  
Check-in with  
your SSM

Attend  
Office  
Hours

Form  
Study  
Groups

Tutoring  
sessions

## **Minimum Graduation Requirements:**

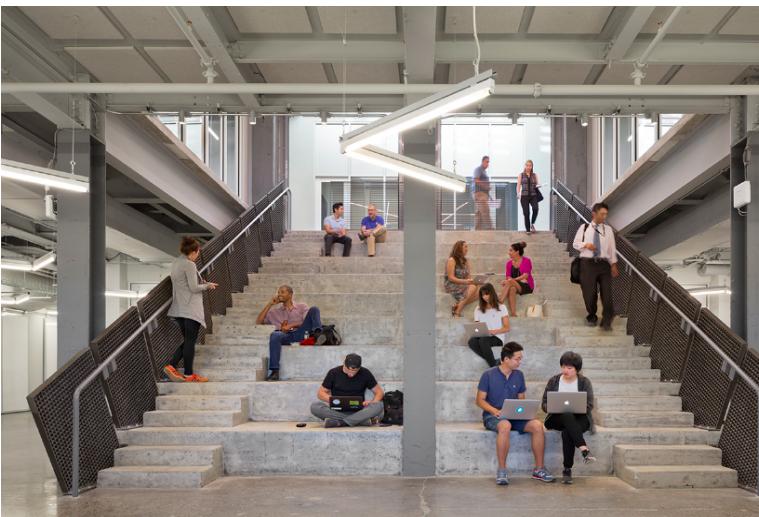
- No more than 4 regular absences and 4 remote/excused absences (8 total)
- Have no more than 2 incomplete homeworks
- Hold at least a 70% average on homeworks
- Hold at least a 70% average on group projects
- Participate in all projects

# Add/Drop Period

- Ends Friday, January 31st at 11:59pm
- Can withdraw with full refund
- Must let me know

# Pennovation Center

- Co-working space/incubator owned by the University of Pennsylvania
- Roughly 40 startups using the space
- 9 AM - 3 PM on weekdays (first floor only)
- First floor use only
- Cafeteria
- Food trucks on Wednesdays!



# **Session Objectives Completed:**

- ✓ Come to know your classmates as the community that you will rely on for collaborative learning.
- ✓ Know the staff who will be providing holistic support throughout the program.
- ✓ Understand the minimum requirements in order to successfully graduate from this bootcamp.
- ✓ Be able to list ways to get help and support at your moments of need.

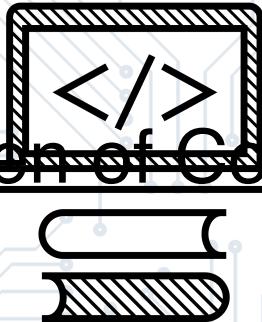
# **Questions?**



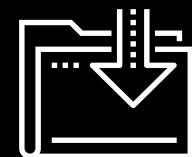
# Break

See you back in 15 minutes!

# The [Zen of Coding]



Web Development Boot Camp  
Lesson 1.1



# The Path of Learning

# Your Goals

---

Basically, 100% of you said **new career**.



# Your Goals

---

And why do you want a new career?



To pursue a **dream**



To escape a **dead-end job**



To be able to **create**



To follow a **fascination**



To attain **financial stability**



To obtain **financial freedom**



To **challenge yourself**



To be a **role model** to kids



## **Your Goal = Our Goal**

As instructors, we take your goals very, very seriously.

# Support Team Promise

---

If you're willing to put in the time and take our advice, we're here to help you 100% of the way.

This goes for everyone working on this program:

- Instructors
- TAs
- Student Success Team
- Career coaches
- Everyone else!





**But Remember:**  
Nothing good comes easily.

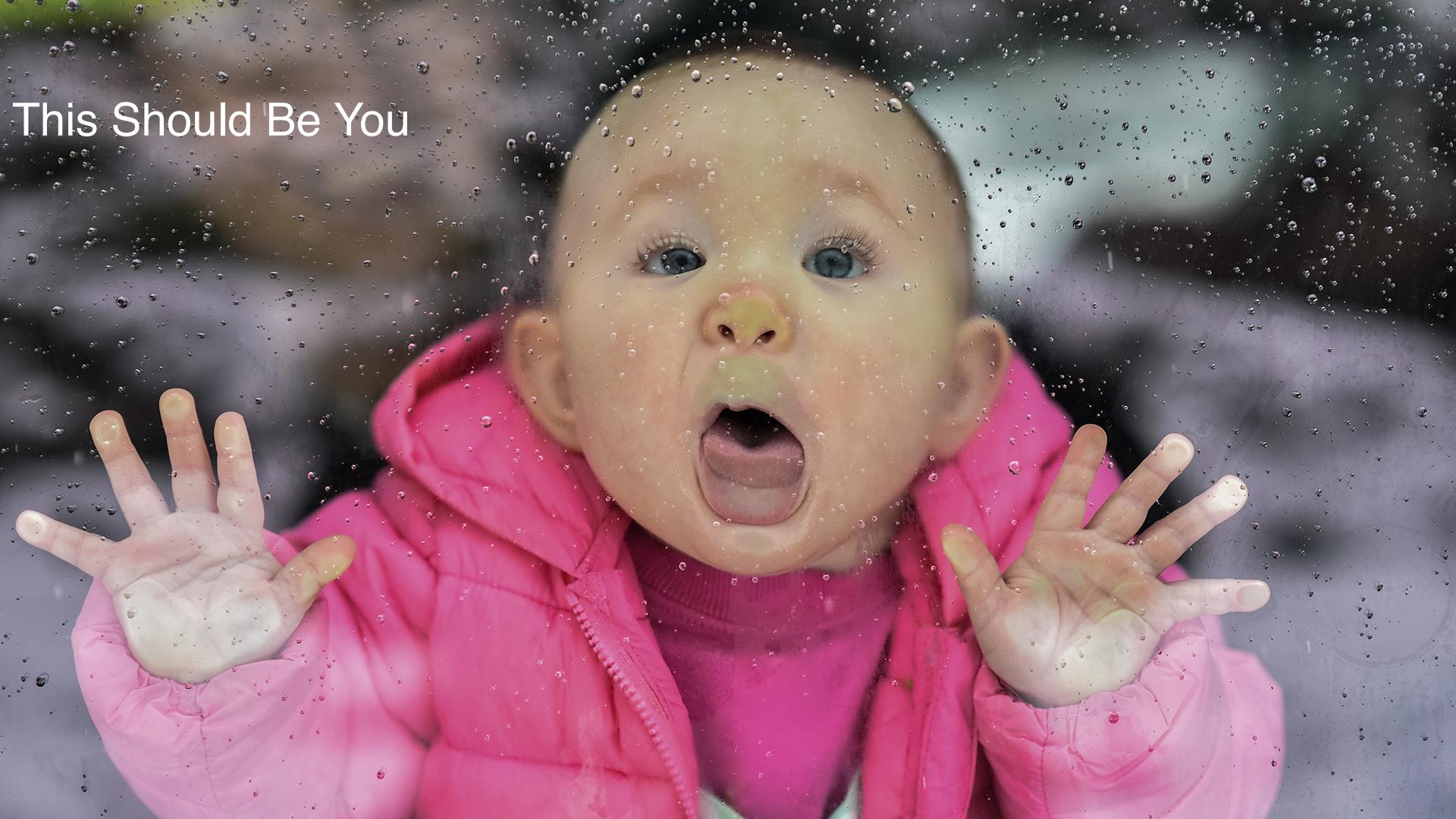
# Our Keys to Success

# Don't Be This Person:

---

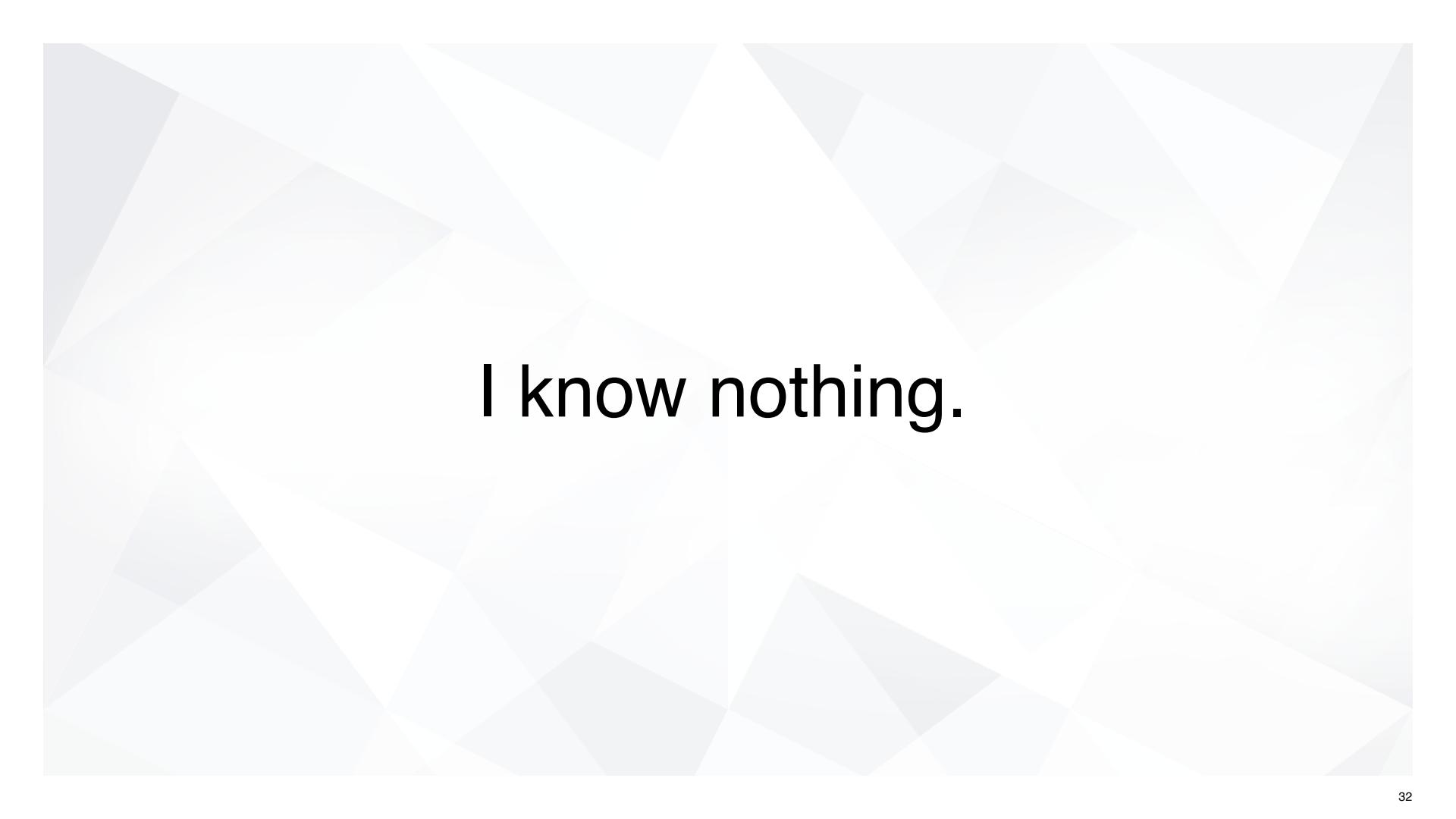


This Should Be You



## **Our Mantra for Today and Beyond**

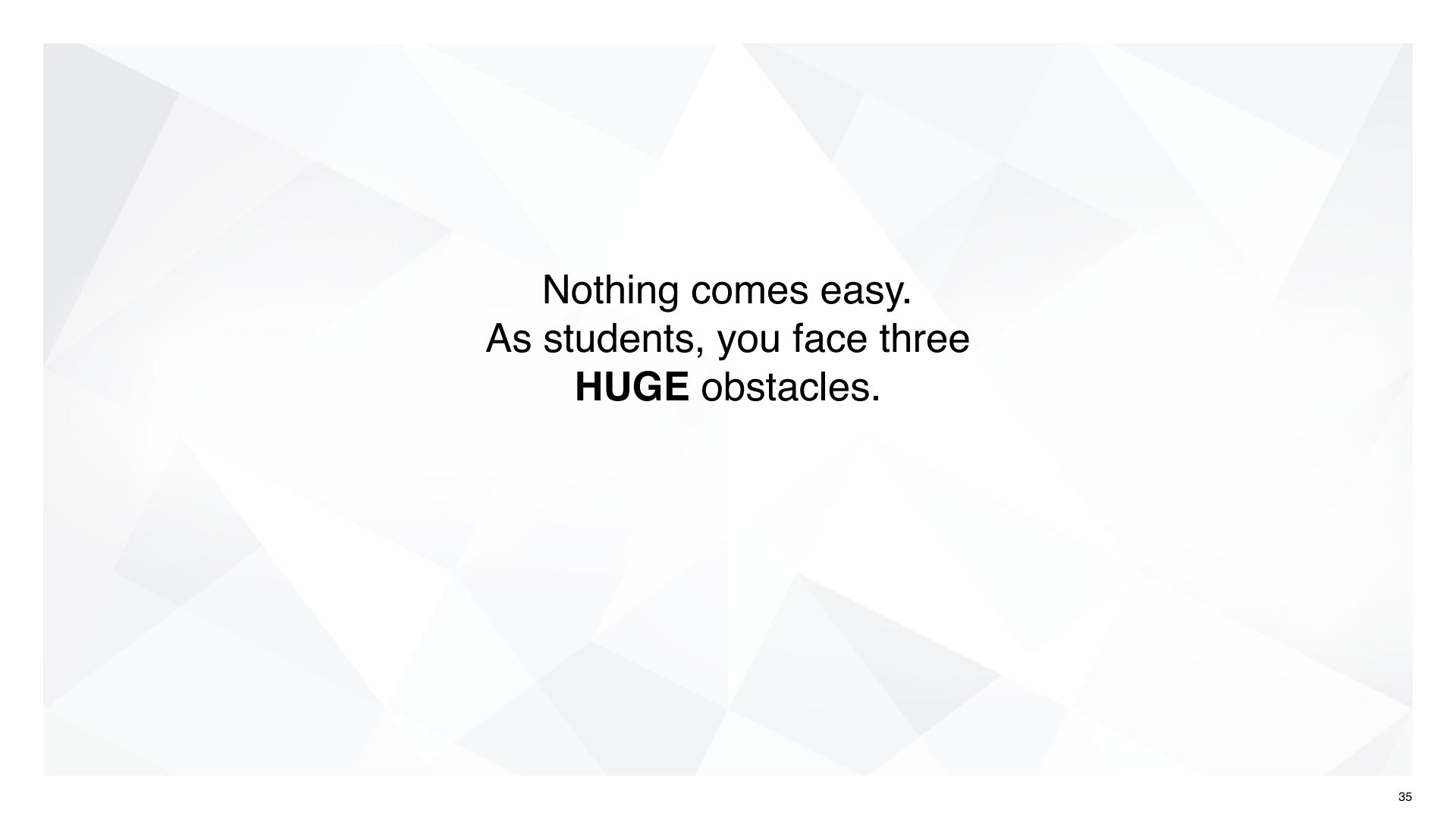
When it comes to web development...



I know nothing.

You.

# The Path of Learning



Nothing comes easy.  
As students, you face three  
**HUGE** obstacles.

# Obstacle #1: The Great Confusion

---

**My code doesn't work and I don't know why.**

---

**My code works and I don't know why.**



“Maybe I’m just dumb.”



# Obstacle #3: The Great Distance

---



# Nothing Comes Easy

---

Learning to code requires two things:

01

Persisting in the face of something that feels incredibly hard and confusing.

02

Maintaining the self-confidence necessary to believe that YOU CAN DO THIS.

# Learning Can Be Frustrating

---



You can't tell whether you're learning something when you're learning it—in fact, *learning feels a lot more like frustration.*

What I've learned is that during this period of frustration is actually when people improve the most, and their improvements are usually obvious to an outsider. If you feel frustrated while trying to understand new concepts, try to remember that it might not feel like it, but you're probably rapidly expanding your knowledge.

—Jeff Dickey, author of *Write Modern Web Apps with the MEAN Stack: Mongo, Express, AngularJS, and Node.js* (Peachpit Press, 2014)

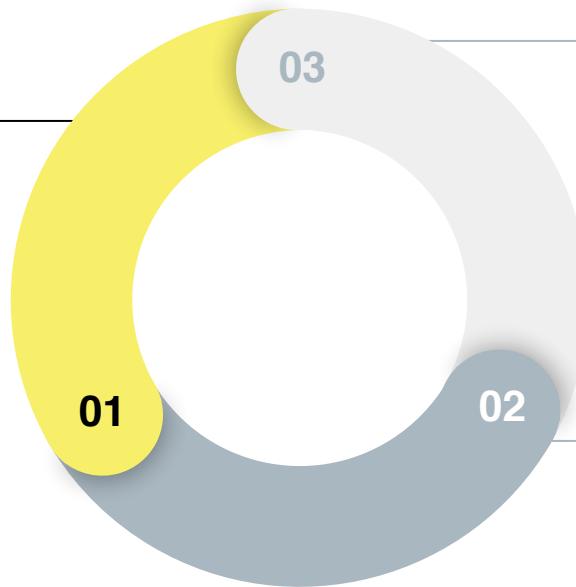


# Advice for the Journey

---

Throughout this course, always remember to:

Work hard!



Trust yourself

Celebrate your  
successes

---

Throughout this course, always remember to:



Work hard!

Trust yourself

Celebrate your  
successes

# Advice for the Journey

---

Throughout this course, always remember to:



# And Remember...

---

If you want to go fast, go alone. If you want to go far, go with a team.



# Google Fu: The Most Important Skill of All

What Is Google Fu?



# Course Structure

# Daily Schedule

---

In each class, we'll cover the following:

01

Set objectives

02

Brief background lecture

03

Watch me / coding demos

04

Code discussions

05

In-class exercises

06

Project work

# Daily Schedule

---

In each class, we'll cover the following:

01

Set objectives

02

Brief background lecture

03

Watch me / coding demos

04

Code discussions

05

In-class exercises

06

Project work



This is the super important stuff  
—ALWAYS BE CODING!

# Prework



How'd it go?

# Software Checklist

---

At this point, you should have all of these installed:

- Slack
- Visual Studio Code
- Git
- Git Bash (Windows) or Terminal (Mac)
- Node.js
- Heroku-CLI
- Google Chrome

# Accounts Checklist

---

You should also have accounts for:

- GitHub (with SSH Integration)
- Heroku
- LinkedIn
- Stack Overflow

# Self-Check

---

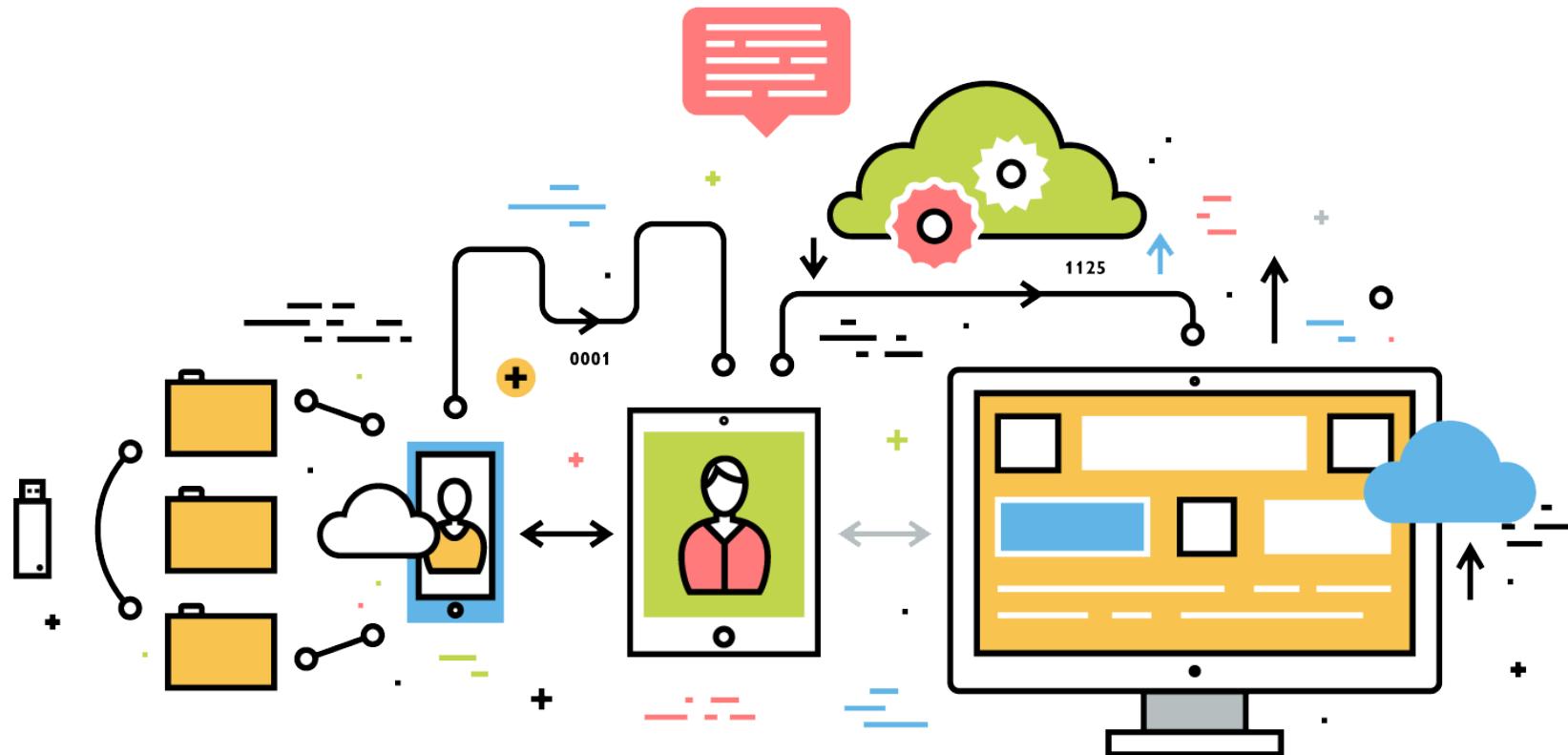
Let's do some quick checks of the following:

- Visual Studio Code Check
- Git Bash/Terminal Check
- Node Check
- Git Check
- Heroku Check

# The Modern Web

# What Exactly Is Full-Stack Development?

---

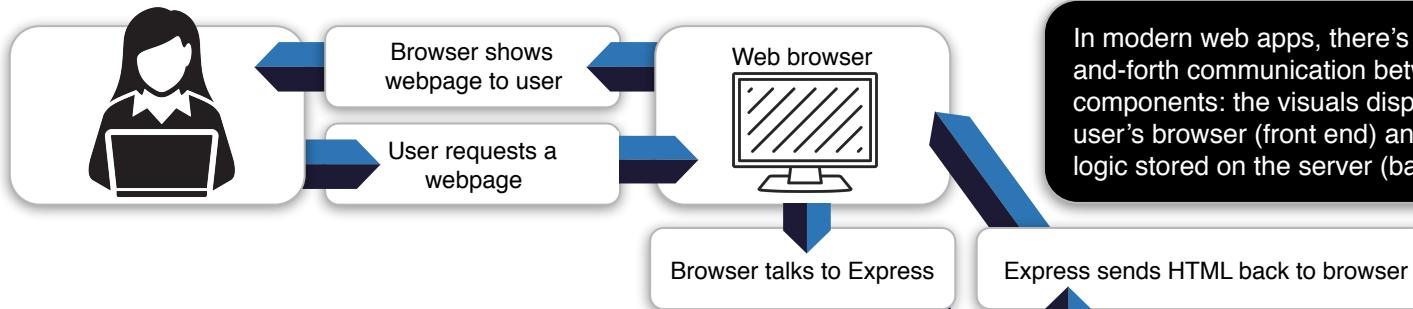


# The “Magic” of YouTube

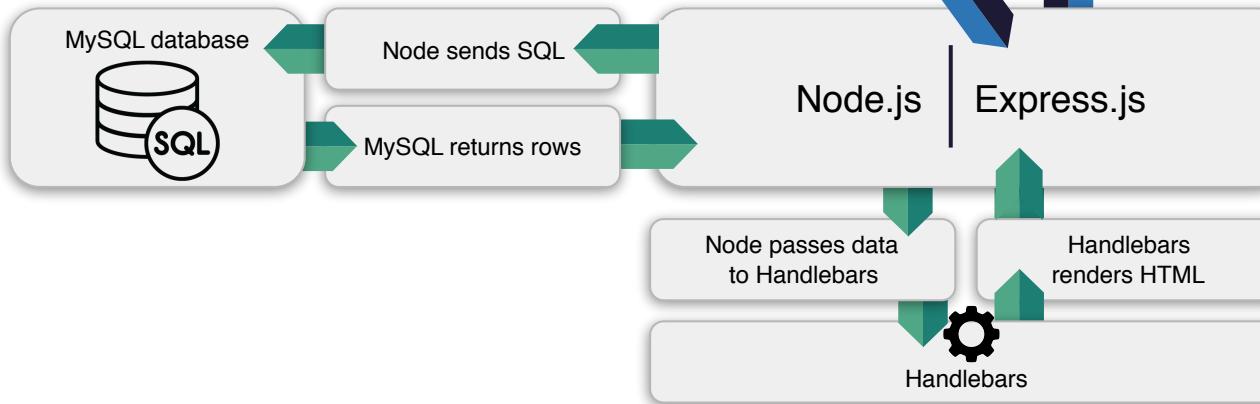


## FSF FLOW

Front end

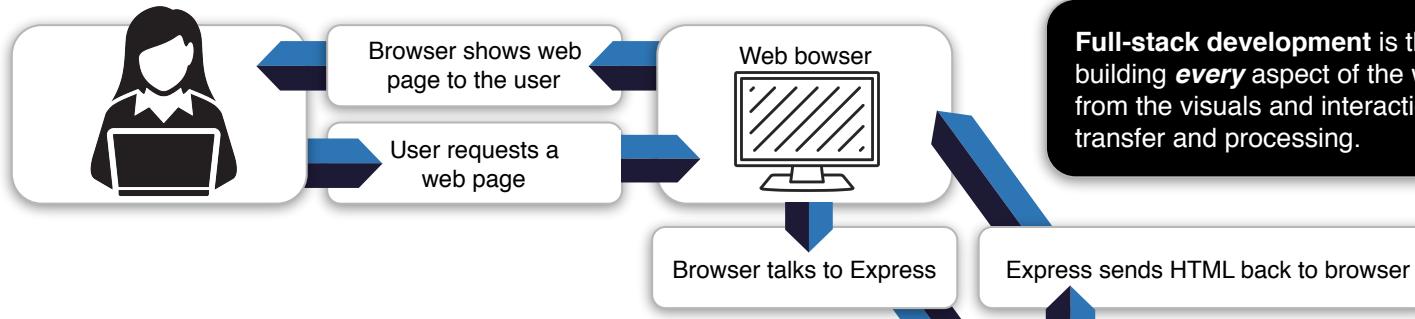


Back end

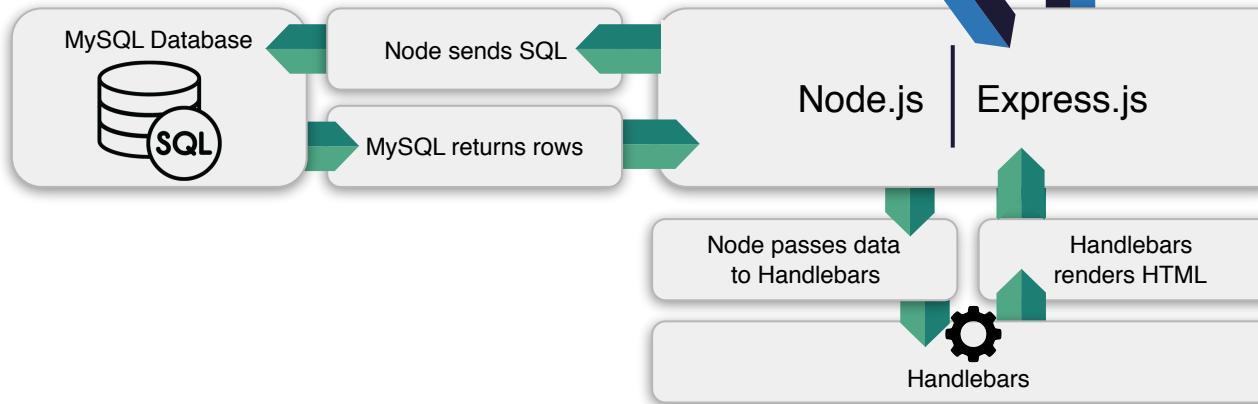


## FSF FLOW

Frontend

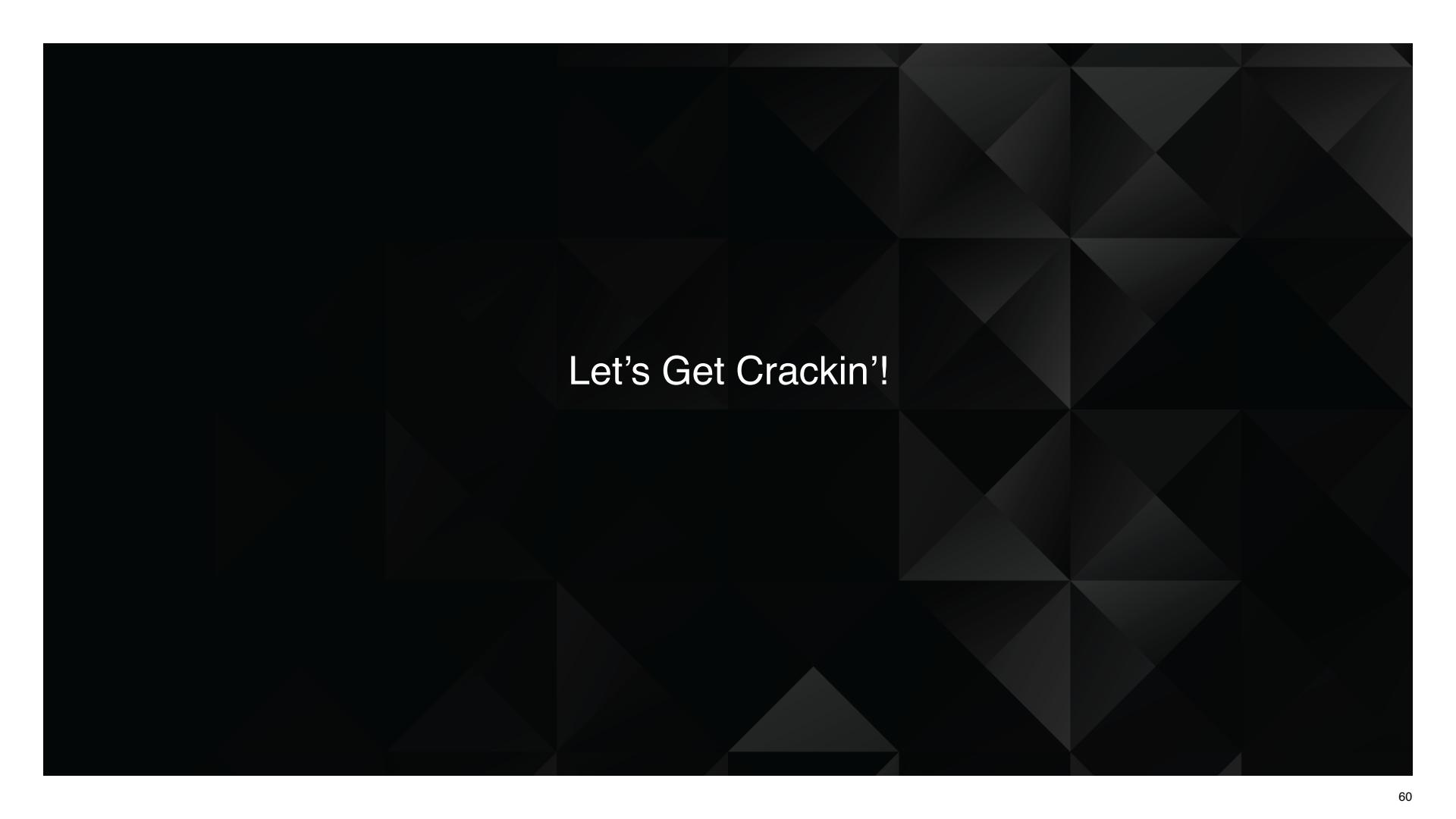


Backend



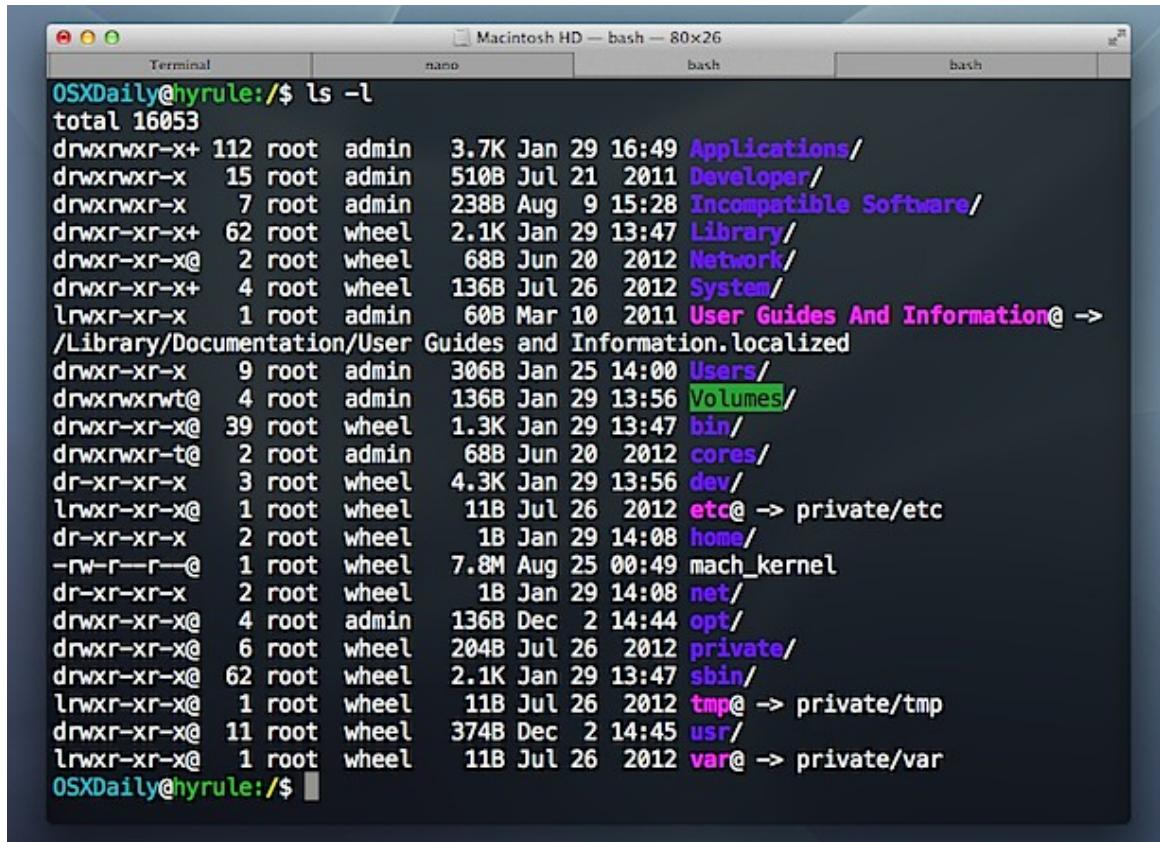
**Full-stack development** is the concept of building *every* aspect of the web application—from the visuals and interactions to the data transfer and processing.

The Browser	Dev Tools	Server Side	
HTML	Heroku	Templating engines	
CSS	Git	Sessions	
JavaScript	GitHub	Writing tests	
jQuery	Databases	Node.js	
Bootstrap	MySQL	Express.js	
SEO	MongoDB	Creating APIs	
<b>API Interaction</b>		MVC	
APIs (Consuming)		User authentication	
JSON		ORM (Sequelize)	
AJAX		<b>CS Fundamentals</b>	
Real-time cloud database via Firebase		Algorithms	
<b>Cutting-Edge Development</b>		Design patterns	
React.js			



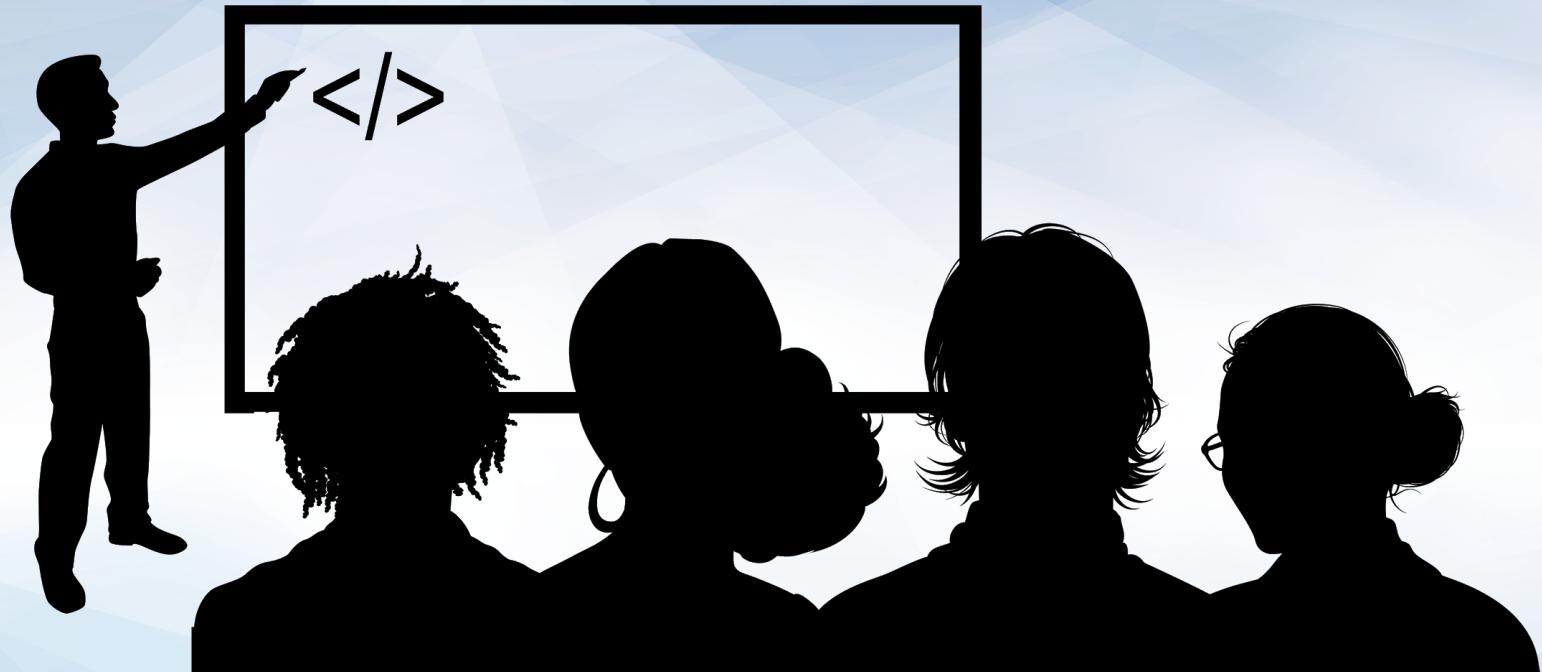
Let's Get Crackin'!

# Intro to Console/Terminal



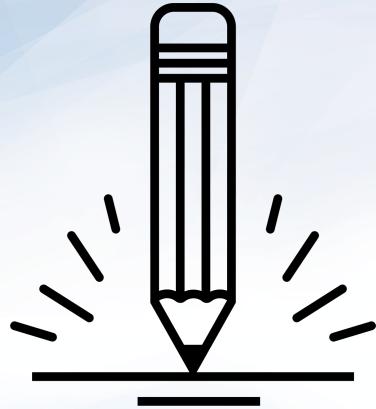
A screenshot of a Mac OS X Terminal window titled "Macintosh HD — bash — 80x26". The window shows the output of the command "ls -l". The output lists numerous system directories and files, including Applications, Developer, Incompatible Software, Library, Network, System, User Guides And Information, Users, Volumes, bin, cores, dev, etc, home, mach\_kernel, net, opt, private, sbin, tmp, usr, and var. The "User Guides And Information" directory is shown as a symbolic link (@ ->) to "/Library/Documentation/User Guides and Information.localized". The "etc" directory is also shown as a symbolic link (@ ->) to "private/etc". The "tmp" directory is shown as a symbolic link (@ ->) to "private/tmp". The "usr" directory is shown as a symbolic link (@ ->) to "private/usr". The "var" directory is shown as a symbolic link (@ ->) to "private/var". The terminal prompt at the bottom is "OSXDaily@hyrule:/\$".

```
OSXDaily@hyrule:/$ ls -l
total 16053
drwxrwxr-x+ 112 root admin 3.7K Jan 29 16:49 Applications/
drwxrwxr-x 15 root admin 510B Jul 21 2011 Developer/
drwxrwxr-x 7 root admin 238B Aug 9 15:28 Incompatible Software/
drwxr-xr-x+ 62 root wheel 2.1K Jan 29 13:47 Library/
drwxr-xr-x@ 2 root wheel 68B Jun 20 2012 Network/
drwxr-xr-x+ 4 root wheel 136B Jul 26 2012 System/
lrwxr-xr-x 1 root admin 60B Mar 10 2011 User Guides And Information@ ->
/Library/Documentation/User Guides and Information.localized
drwxr-xr-x 9 root admin 306B Jan 25 14:00 Users/
drwxrwxnwt@ 4 root admin 136B Jan 29 13:56 Volumes/
drwxr-xr-x@ 39 root wheel 1.3K Jan 29 13:47 bin/
drwxrwxr-t@ 2 root admin 68B Jun 20 2012 cores/
dr-xr-xr-x 3 root wheel 4.3K Jan 29 13:56 dev/
lrwxr-xr-x@ 1 root wheel 11B Jul 26 2012 etc@ -> private/etc
dr-xr-xr-x 2 root wheel 1B Jan 29 14:08 home/
-rw-r--r--@ 1 root wheel 7.8M Aug 25 00:49 mach_kernel
dr-xr-xr-x 2 root wheel 1B Jan 29 14:08 net/
drwxr-xr-x@ 4 root admin 136B Dec 2 14:44 opt/
drwxr-xr-x@ 6 root wheel 204B Jul 26 2012 private/
drwxr-xr-x@ 62 root wheel 2.1K Jan 29 13:47 sbin/
lrwxr-xr-x@ 1 root wheel 11B Jul 26 2012 tmp@ -> private/tmp
drwxr-xr-x@ 11 root wheel 374B Dec 2 14:45 usr/
lrwxr-xr-x@ 1 root wheel 11B Jul 26 2012 var@ -> private/var
OSXDaily@hyrule:/$
```



# Instructor Demonstration

## Let's Get Crackin'—Intro to Conso



# Activity: Console Commands

Instructions sent via Slack.

Suggested Time:  
12 minutes



# Activity: Console Commands

---

Instructions:

01

Create a new folder with the name of **self-destructing-folder**.

02

Create a new txt file with the name **secret-message.txt**.

03

Copy **secret-message.txt** to the inside of **self-destructing-folder**.

04

Delete the original **secret-message.txt** file.

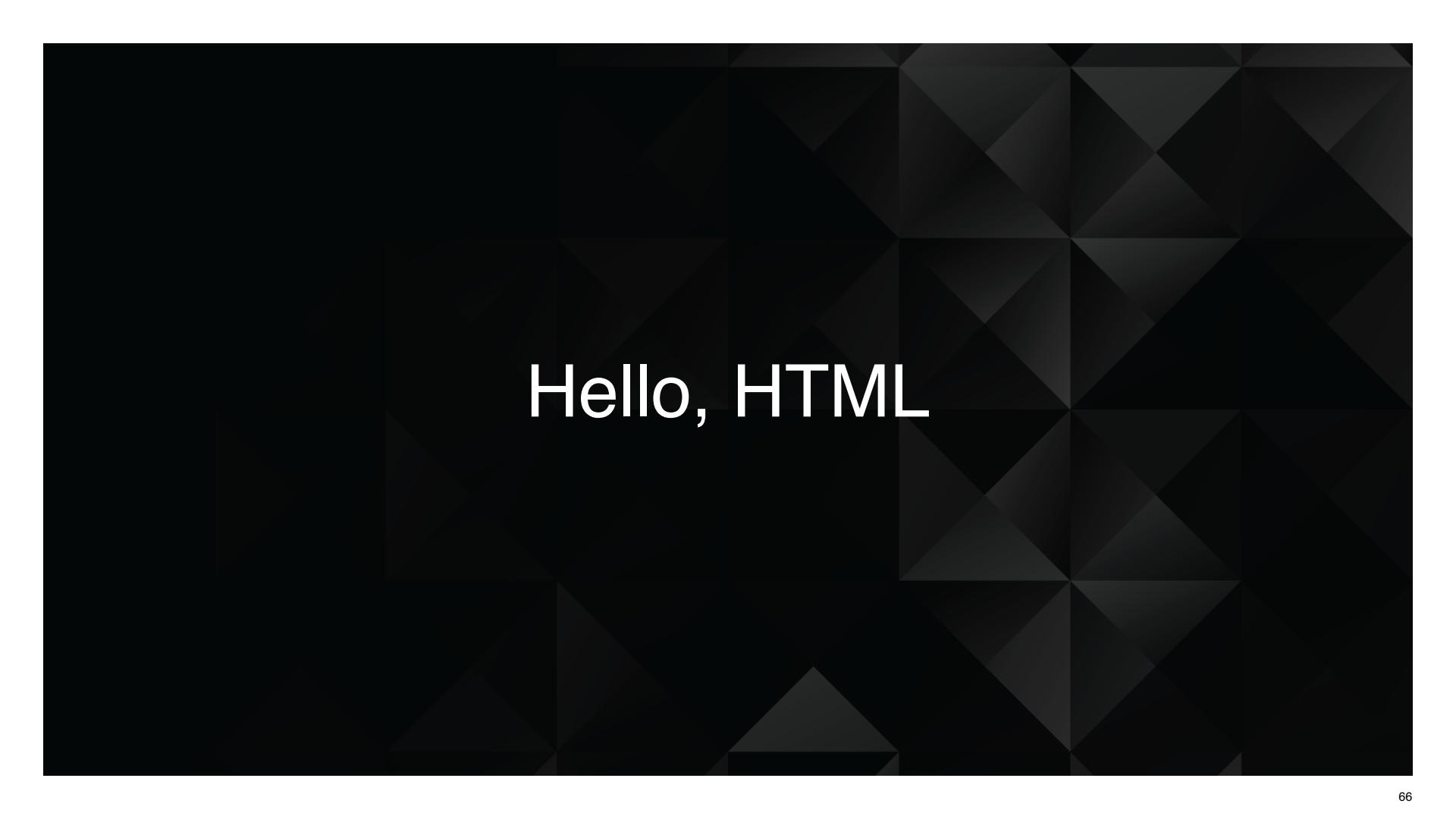
05

Delete the **self-destructing-folder**.

Suggested Time: 12 min







Hello, HTML

## HTML5

- HTML is one of the three base languages behind every website.
- It defines all the basic content and a bit of formatting.



# Activity

## Intro to HTML

Instructions sent via Slack.



Suggested Time:  
5 minutes



# Activity: Intro to HTML

---

In a new HTML file, create the basic structure of an HTML document and include the following in it:

- DOCTYPE declaration
- Head tag with a title tag
- H1 tag with a title of your choice
- An embedded image

Now add the following three links to your page:

- One link that has target="\_blank" so that it opens a new tab when clicked
- One link that is bold
- One link that is only a placeholder (it goes nowhere)

**BONUS!**

- Create an ordered list of steps to make a sandwich
- Create an unordered list of five bands/musicians you like
- Create a table with two columns (animal class and animal name) and four rows of animals
- Use an alternate way of separating links without line breaks
- Embed a YouTube video of your favorite band/musician

Suggested Time: 5 minutes





Time For a Quick Video

---

[My First HTML](#)

# Take a Break!

---



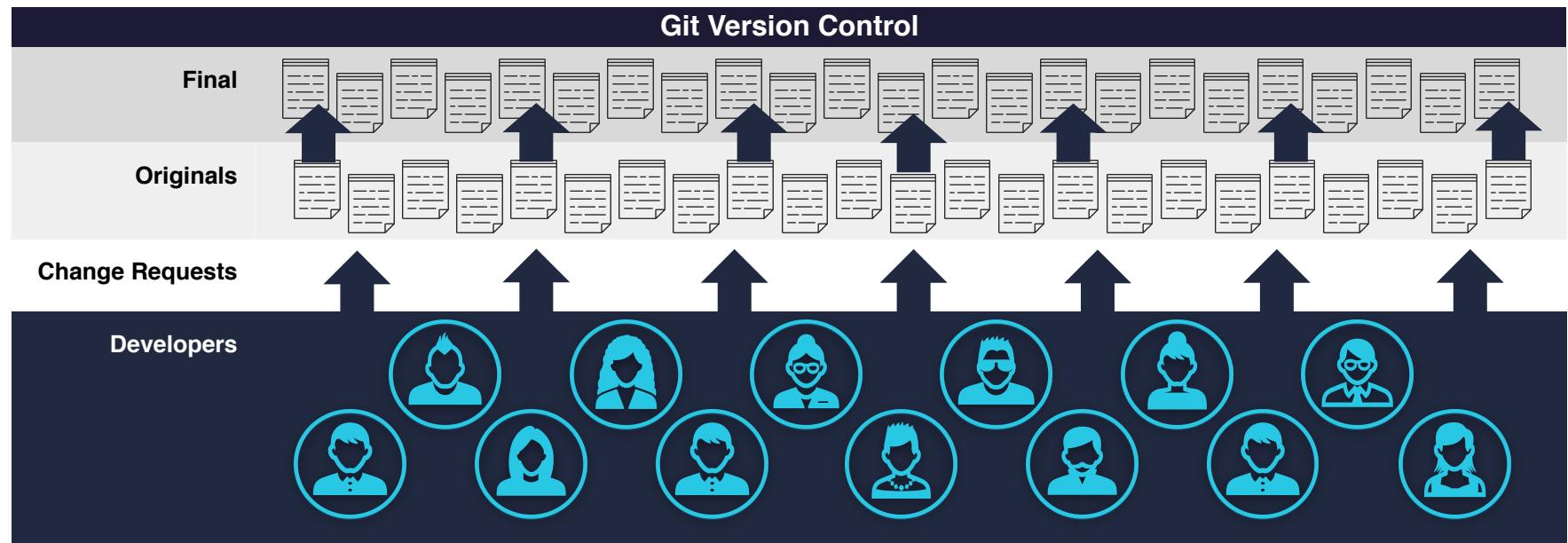
# What Is Git?

# 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



# Maya Angelou & Gil Scott Heron Make Their Edits

---



elou 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>
```



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>
```

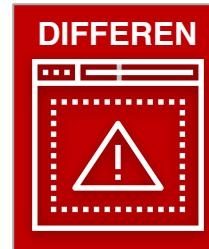
# Different Solutions

---



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's version

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



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



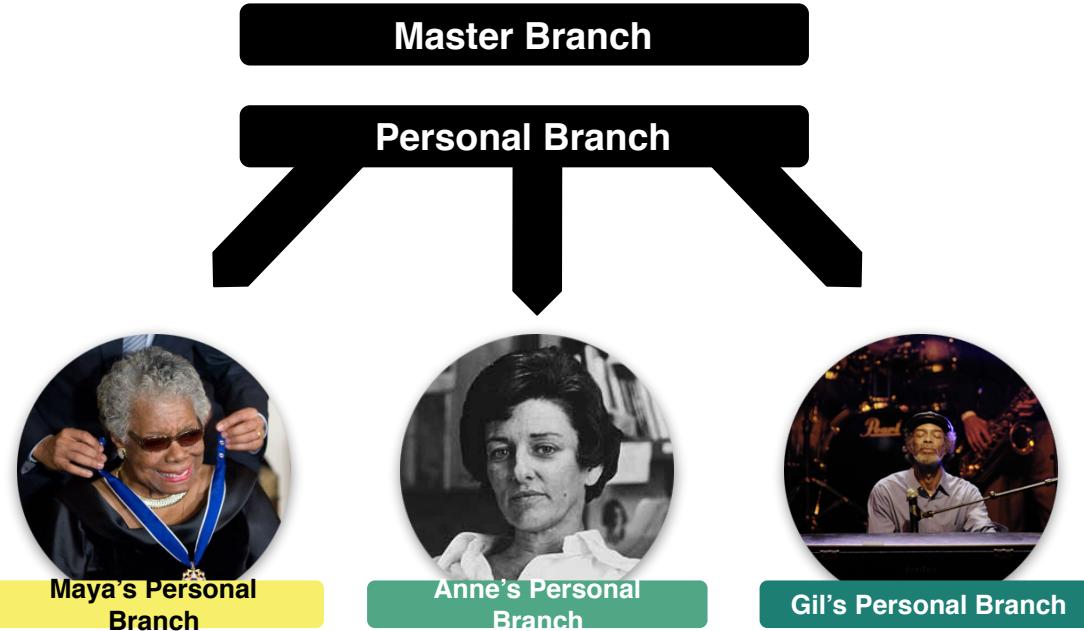
01

A process for resolving conflicts in code

02

Version history

Branch = personal copy



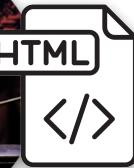
# The Team Goes to Work

---



**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

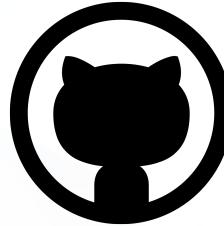


Master Branch

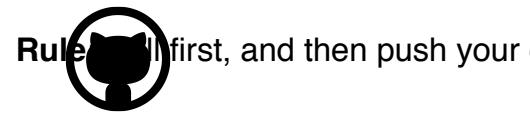


Maya's Personal Branch

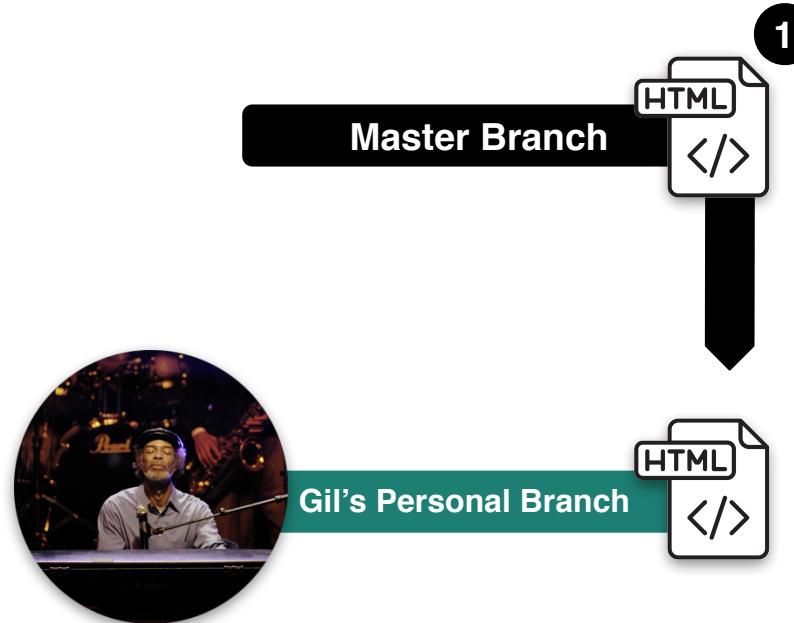




**Rule:** Pull first, and then push your changes.



Rule #1: Pull first, and then push your changes.



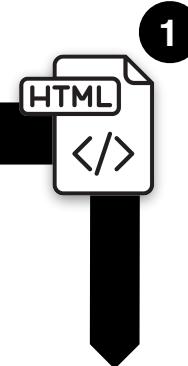
# Gil Scott Heron Conflicts with the Master Branch



Git sees a conflict.

```
<ul>
  <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>
</ul>
```

Master Branch



Gil's Personal Branch



# Kobe Resolves

---

```
<ul>
  <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>
</ul>
```

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



Gil's Personal Branch



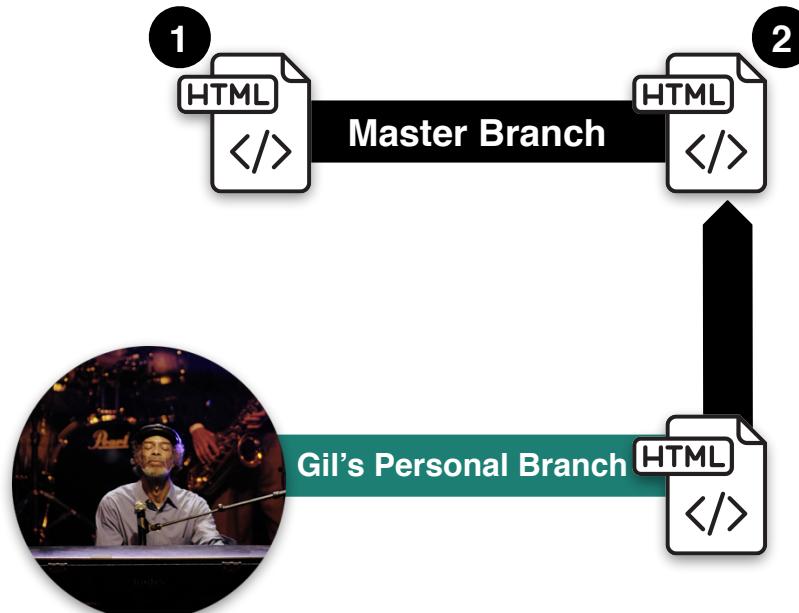
# Gil Scott Heron Fixes and Pushes

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 ~~all~~ first, and then push your changes.

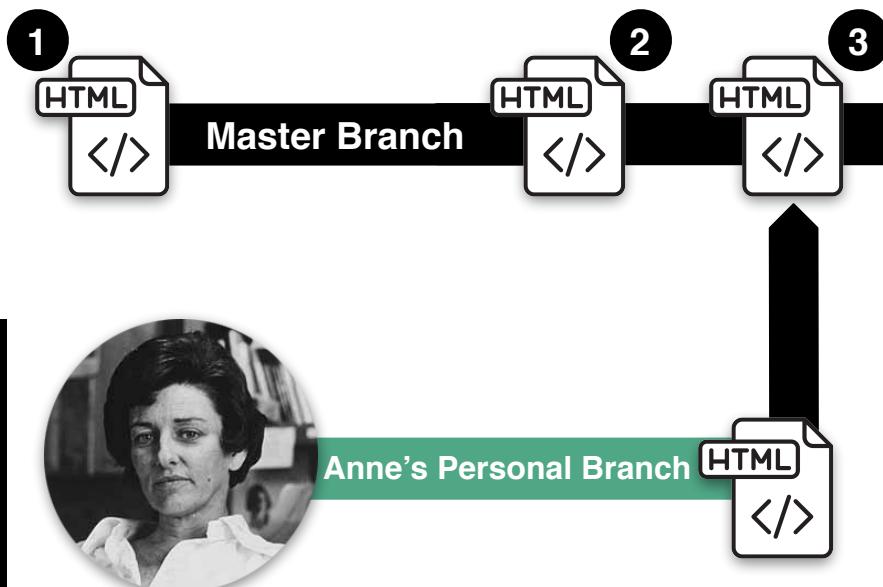


*look into my face  
and you will know that crimes dropped upon me  
as from a high building...  
...by which I mean, I broke the rules.*

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

CONFFLICT!

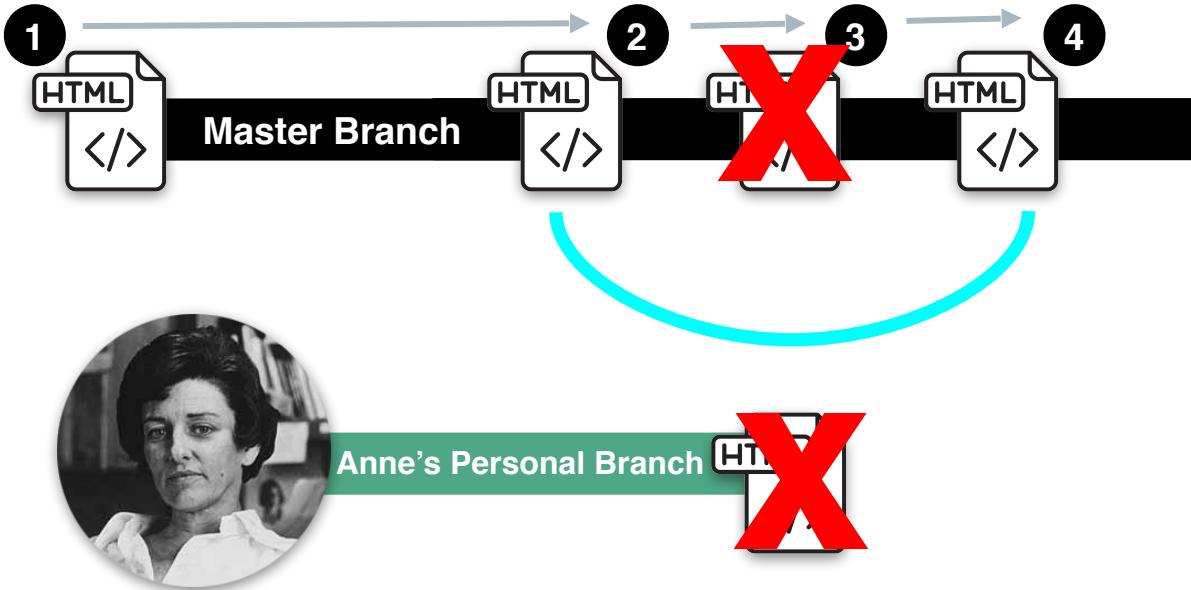


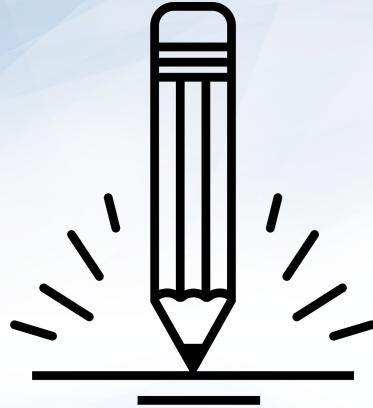


The **overwritten**  
work is discovered!

# 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.



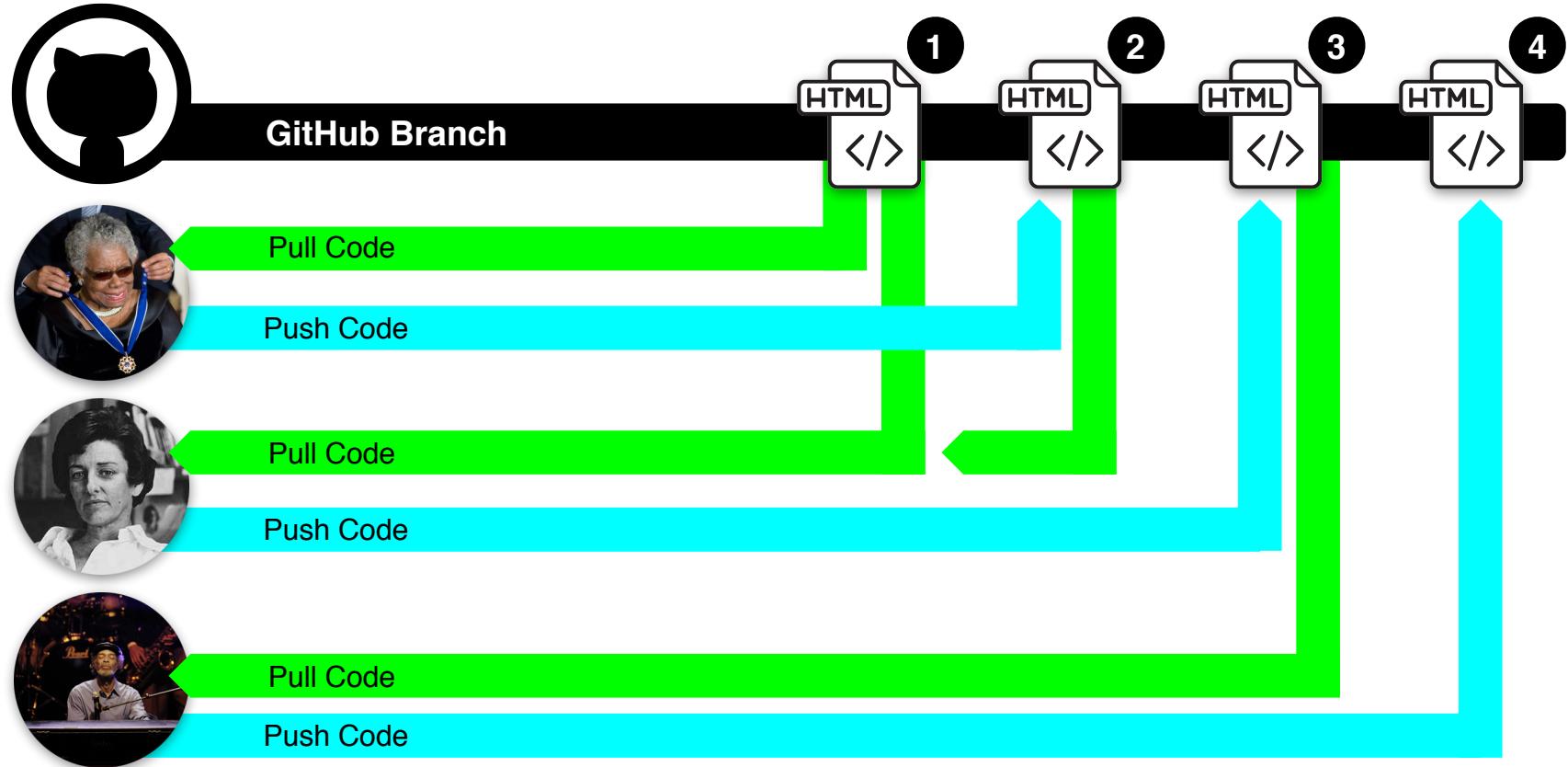
02

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



# Get Started with Git



## Instructor Demonstration

# 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**

**git clone** Copies an entire repo (to begin)

**git add** Adds a file for inclusion in Git

**git commit** Notes a change to the local repo

**git push** Sends changes to hosting service

**git pull** Downloads freshest version of repo

# Challenge

## Git Add, Commit, Push



Suggested Time:  
15 minutes



## 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: 15 Minutes



# Still a Bit Lost? Don't Worry!

Follow this handy guide!

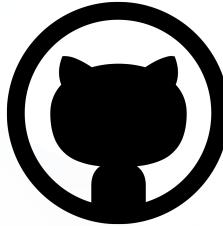
Practice a few times on your own before the next class.

### Steps to Uploading Your Code to GitHub

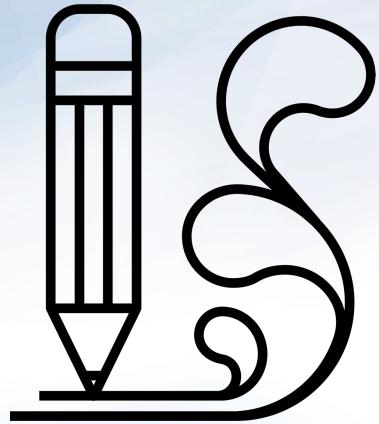
#### Step 1

##### Create a New Repository in GitHub.com

The screenshot shows the GitHub.com homepage with a user profile for 'ahogue'. The main feed displays various activity items. On the right side, there's a sidebar titled 'Repositories you contribute to' listing several repositories like 'RutgersCoding...' and 'All-Lesson-Plans'. Below that is a section for 'Your repositories' with a prominent green button labeled '+ New repository'. A red rectangular box highlights this button, indicating it's the next step in the process.



If you're still lost, here's a (free) course on how to use GitHub:  
[codeschool.com/courses/try-git](https://codeschool.com/courses/try-git)



# Homework Assignment

Due Date:  
**Next Class**



# Homework Assignment

---

01

Figure out where the GitHub repo is for our class.

02

Redo the Terminal example from today's class.

03

Redo the HTML example from today's class.  
(Watch the [walkthrough video](#) if you feel a bit lost.)

Due Date: Next Class

