The Zen of Coding

The Coding Bootcamp

Quick Introductions! (30 seconds)

- Name
- Location
- Background (Career, Education, Interests)
- Why learn web development?
- Excited about class?

Instructor = … ?

Your Name Here

- <insert education or experience tidbit>
- <insert education or experience tidbit>
- Over <#> years coding, (with plenty of experience being a n00b at it.)

- *<personal fact>*
- Secretly, a <funny fact about yourself>

Some Cool Stuff I Made…

SNAPSHOT OF SOMETHING YOU MADE GOES HERE!

The Path of Learning

Your Goals...

Basically 1000% of you said… **New Career**

Your Goals...

And why do you want a new career?

To escape a "dead-end job"

To pursue a "dream"

To be able to "create"

To follow a "fascination"

To attain "financial stability"

To attain "financial freedom"

To "challenge" yourself

To be a "role model" for kids

Your Goal = Our Goal

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

Our Promise:

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

This goes for everyone working behind the program:

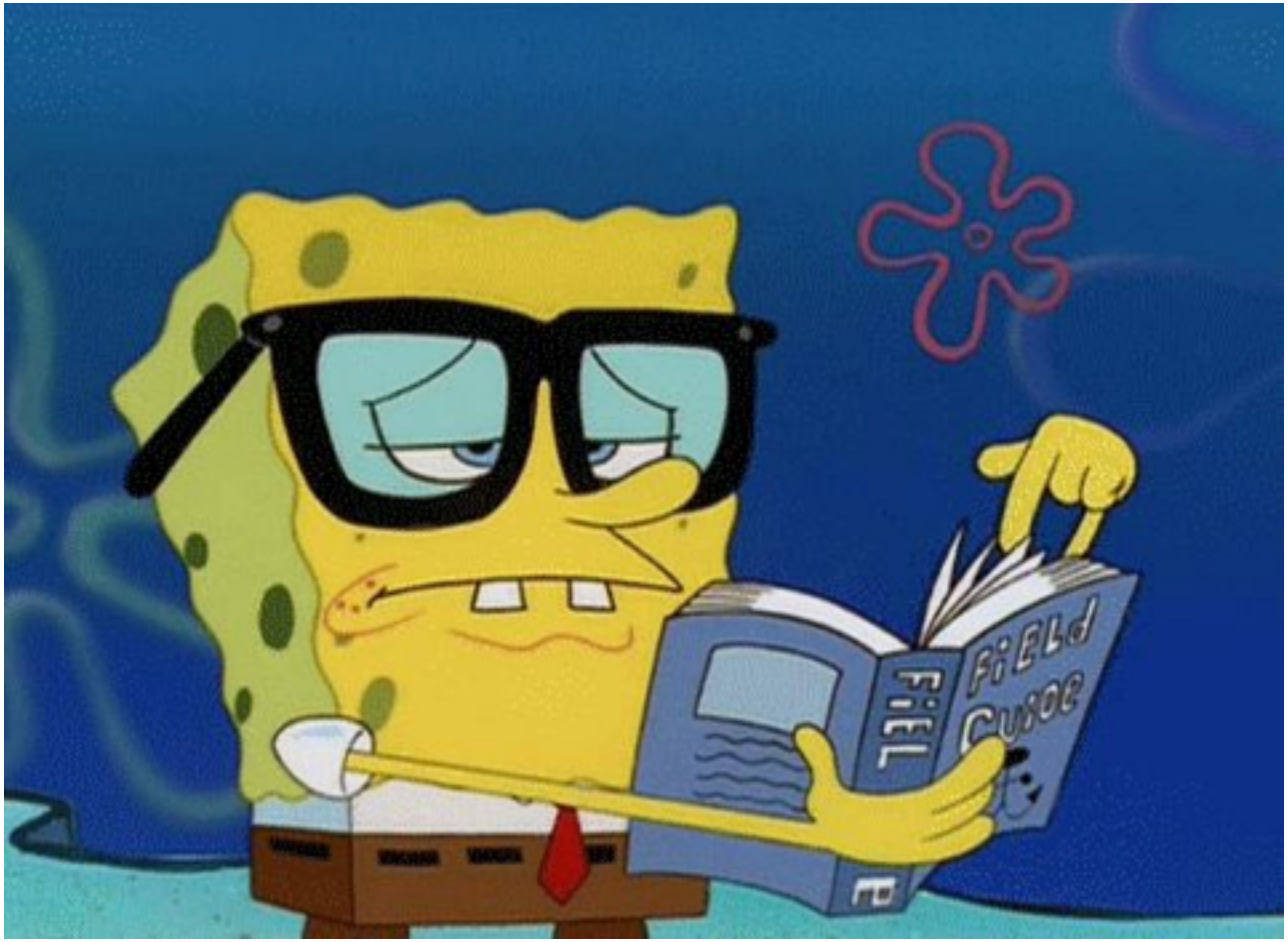
- Instructors
- TAs
- Student Success Team
- Career Coaches
- Everyone Else!

But Remember…

Nothing good comes easy.

On Keys To Success…

Don’t Be This Guy…



This Should Be You.

Our Mantra for Today and Beyond…

When it comes to web development…

Our Mantra for Today and Beyond…

*I know **nothing.***

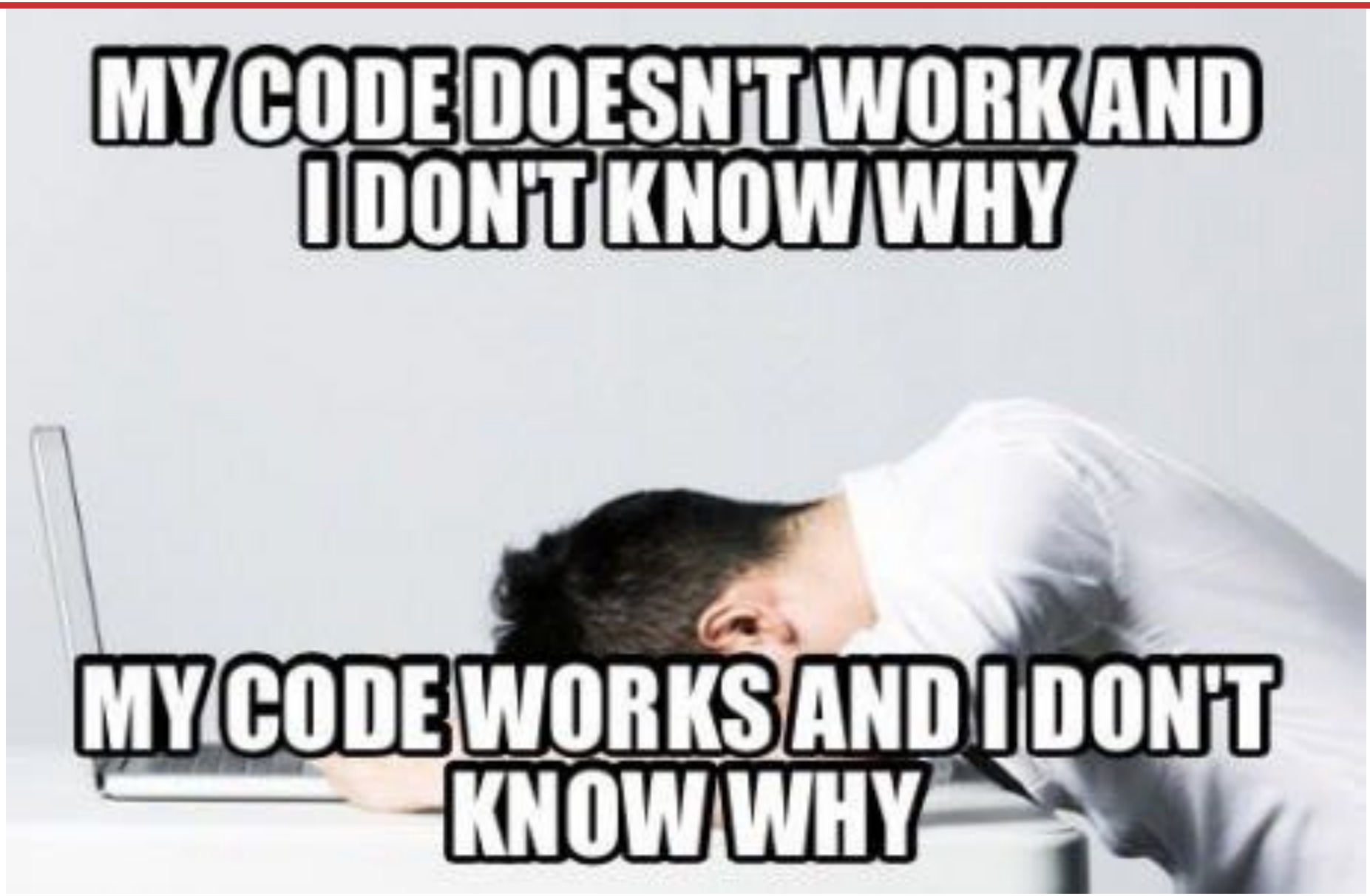
You.

The Path of Learning

Nothing Comes Easy…

As students, you face three
HUGE obstacles!

Obstacle #1 – The Great Confusion



Obstacle #2 – The Great Doubt

Obstacle #3 – The Great Distance

Learning to Code Requires Two Things:

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

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

Learning is “Frustrating”

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

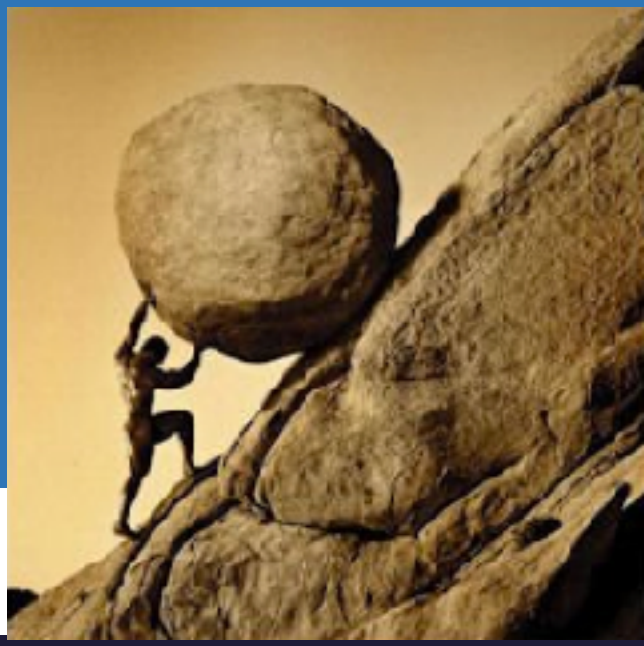
“What I’ve learned is that during this period of frustration is actually when people improve the most, and their improvements are usually obvious to an outsider. If you feel frustrated while trying to understand new concepts, try to remember that it might not feel like it, but you’re probably rapidly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

Jeff Dickey, Author of Write Modern Web Apps with the MEAN Stack: Mongo, Express, AngularJS, and Node.JS

Advice for the Journey

Throughout this course, always remember to:

1. ***Work Hard!!***



Advice for the Journey

Throughout this course, always remember to:

1. *Work Hard!!*

2. ***Appreciate your successes***

Advice for the Journey

Throughout this course, always remember to:

1. *Work Hard!!*
2. *Appreciate your successes*
3. ***Trust yourself***

But remember…

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

Google Fu – The Most Important Skill of All

What is Google Fu?

Course Structure

Daily Schedule

For each class we'll run through the following:

- Set Objectives
- Brief Background Lecture
- Watch Me / Coding Demos
- Code Discussions
- In-Class Exercises
- Project Work

Daily Schedule

For each class we'll run through the following:

- Set Objectives
- Brief Background Lecture
- Watch Me / Coding Demos
- Code Discussions
- In-Class Exercises
- Project Work

The Super Important Stuff!!!

i.e. Always be coding!

Pre-Work

Software Checklist

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

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

Accounts Checklist

You should also now have accounts for:

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

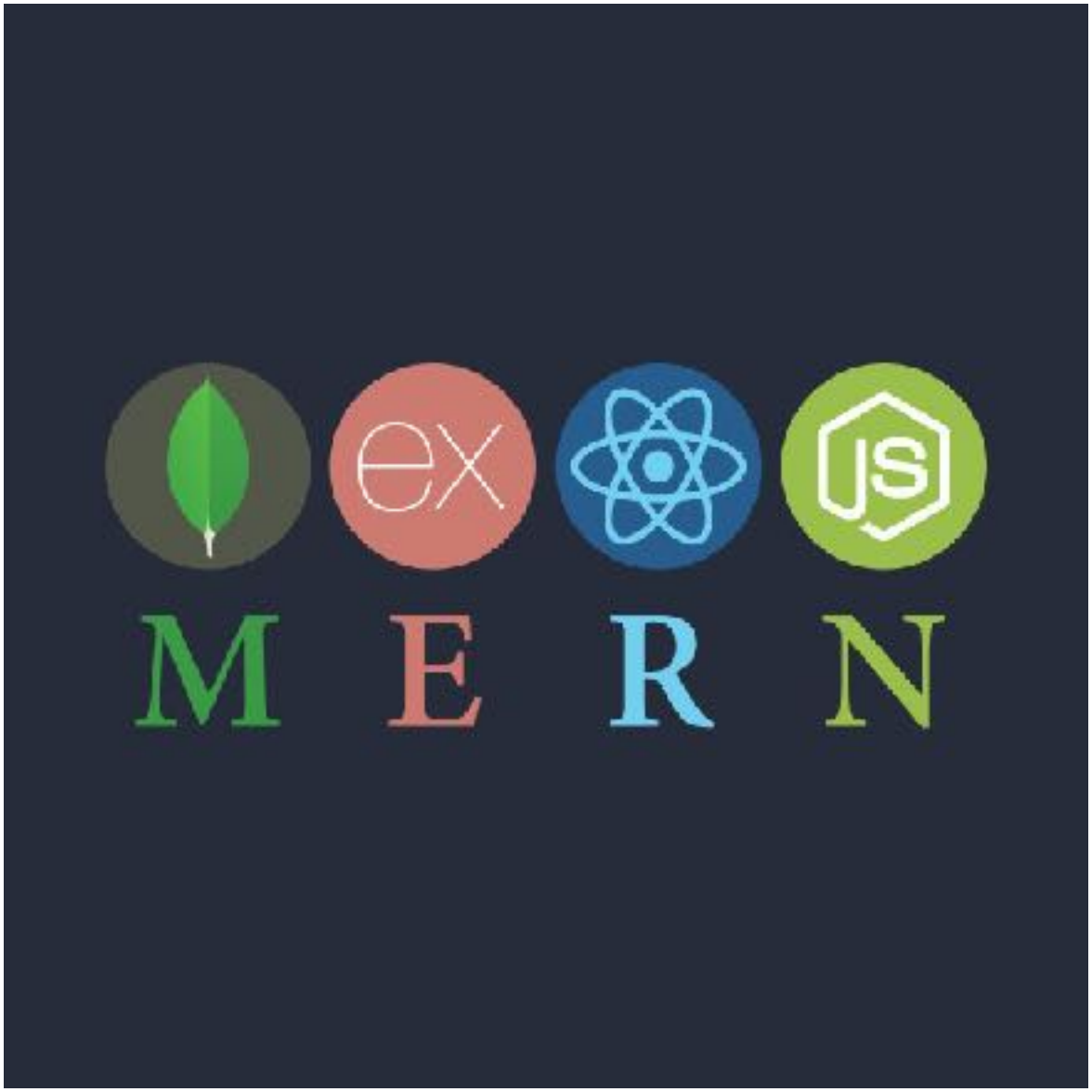
Self-Check

Let’s do some quick checks of the following

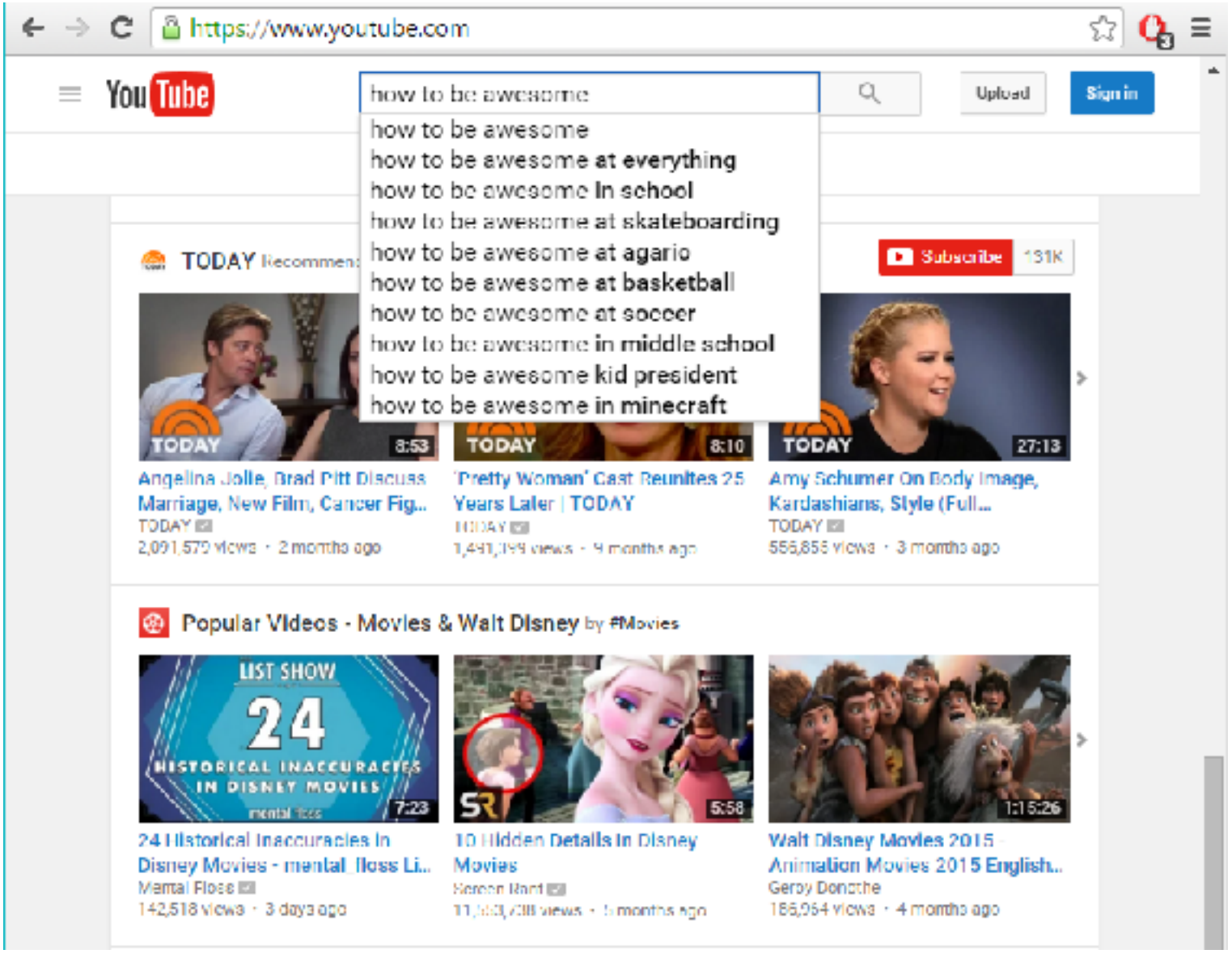
- ✓ Visual Studio Code Check
- ✓ Git Bash / Terminal Check
- ✓ Node Check
- ✓ Git Check
- ✓ Heroku Check

On the Modern Web

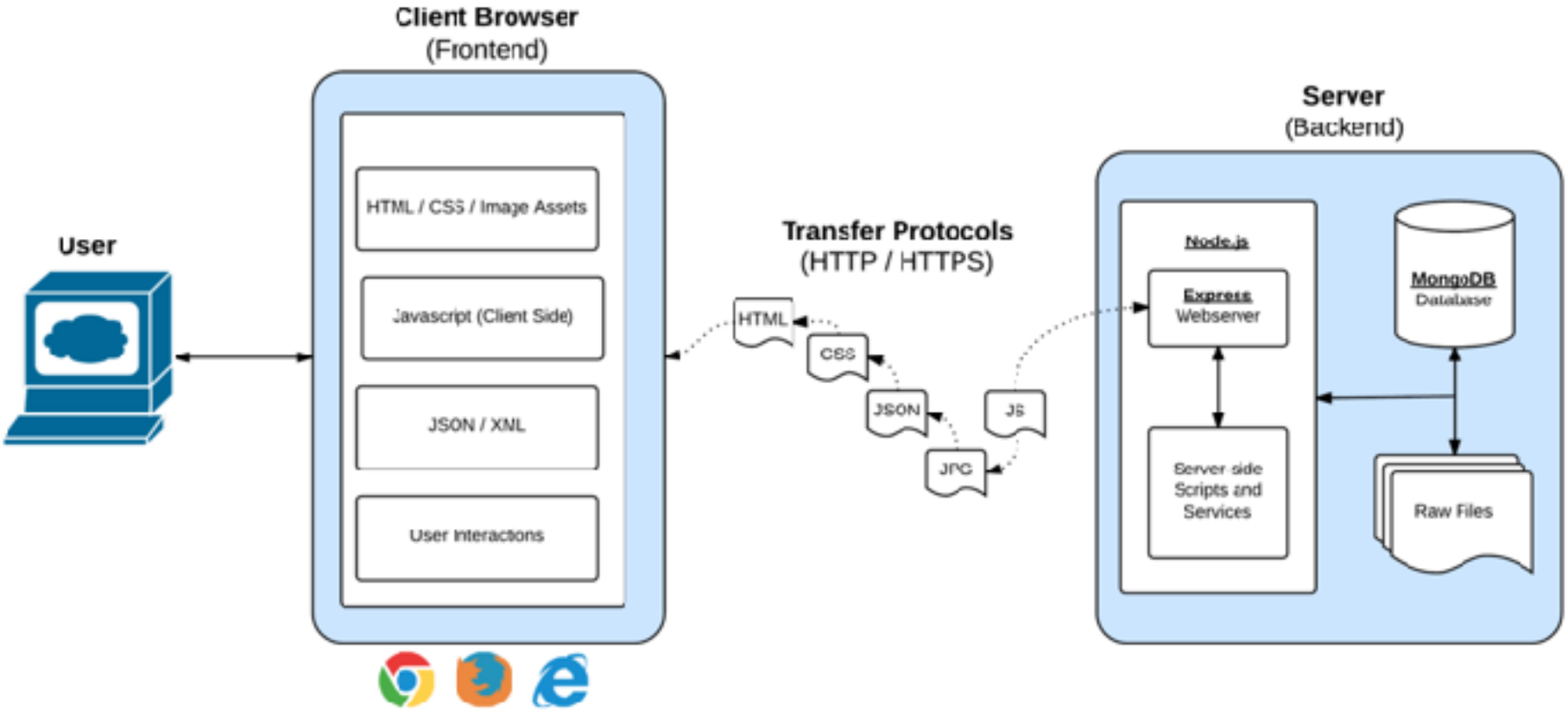
Full-Stack Development?



The “Magic” of YouTube

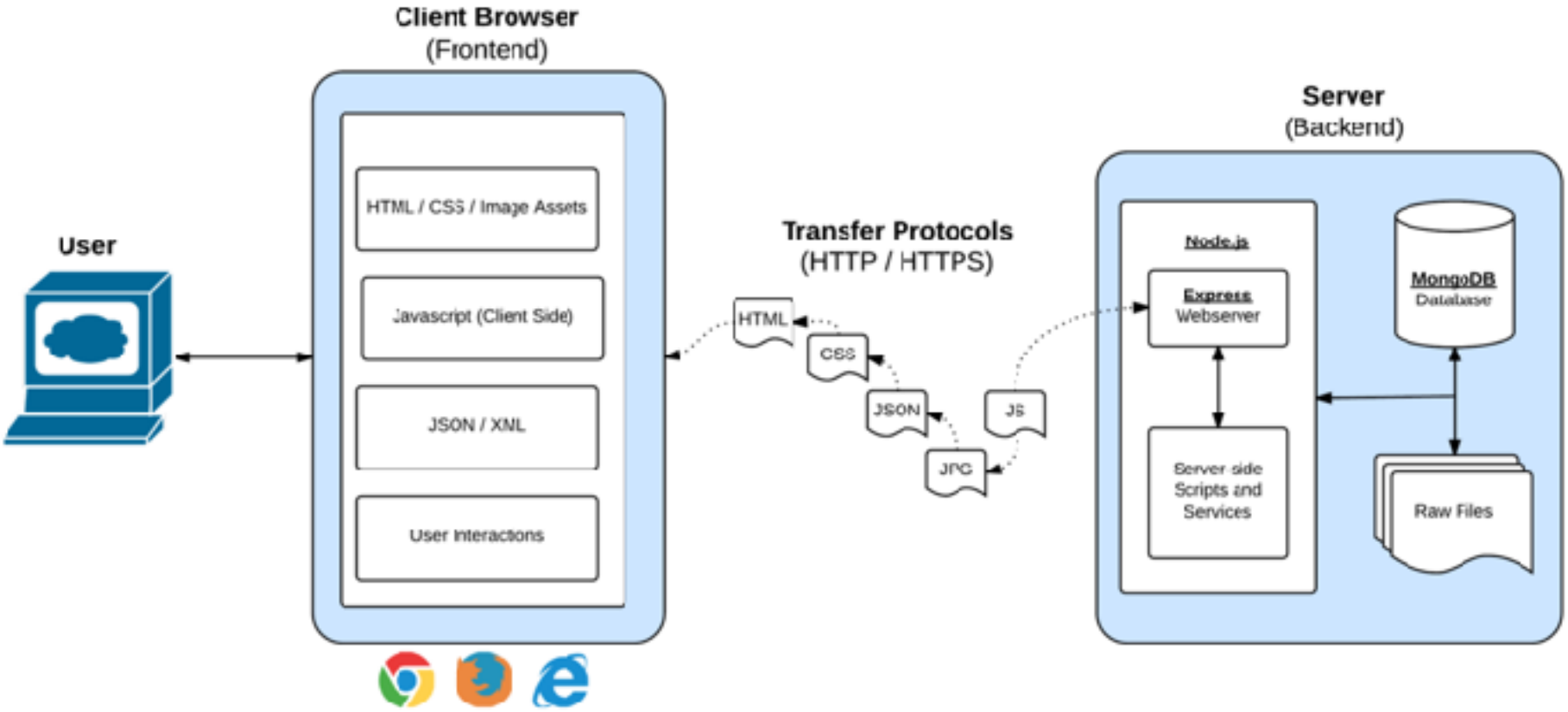


Full-Stack Development



- In modern **web apps,** there's a constant back-and-forth communication between two key components: the visuals displayed on the user's browser (**frontend)** and the data and logic stored on the server (**backend).**

Full-Stack Development



- **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.

Full-Stack Development

The Browser

- HTML
- CSS
- JavaScript
- jQuery
- Bootstrap
- SEO

Dev Tools

- Heroku
- Git
- GitHub

Databases

- MySQL
- MongoDB

Server Side

- Templating Engines
- Sessions
- Writing tests
- Node.js
- Express.js
- Creating APIs
- MVC
- User Authentication
- ORM (Sequelize)

API Interaction

- APIs (Consuming)
- JSON
- AJAX
- Real Time Cloud Database via Firebase

CS Fundamentals

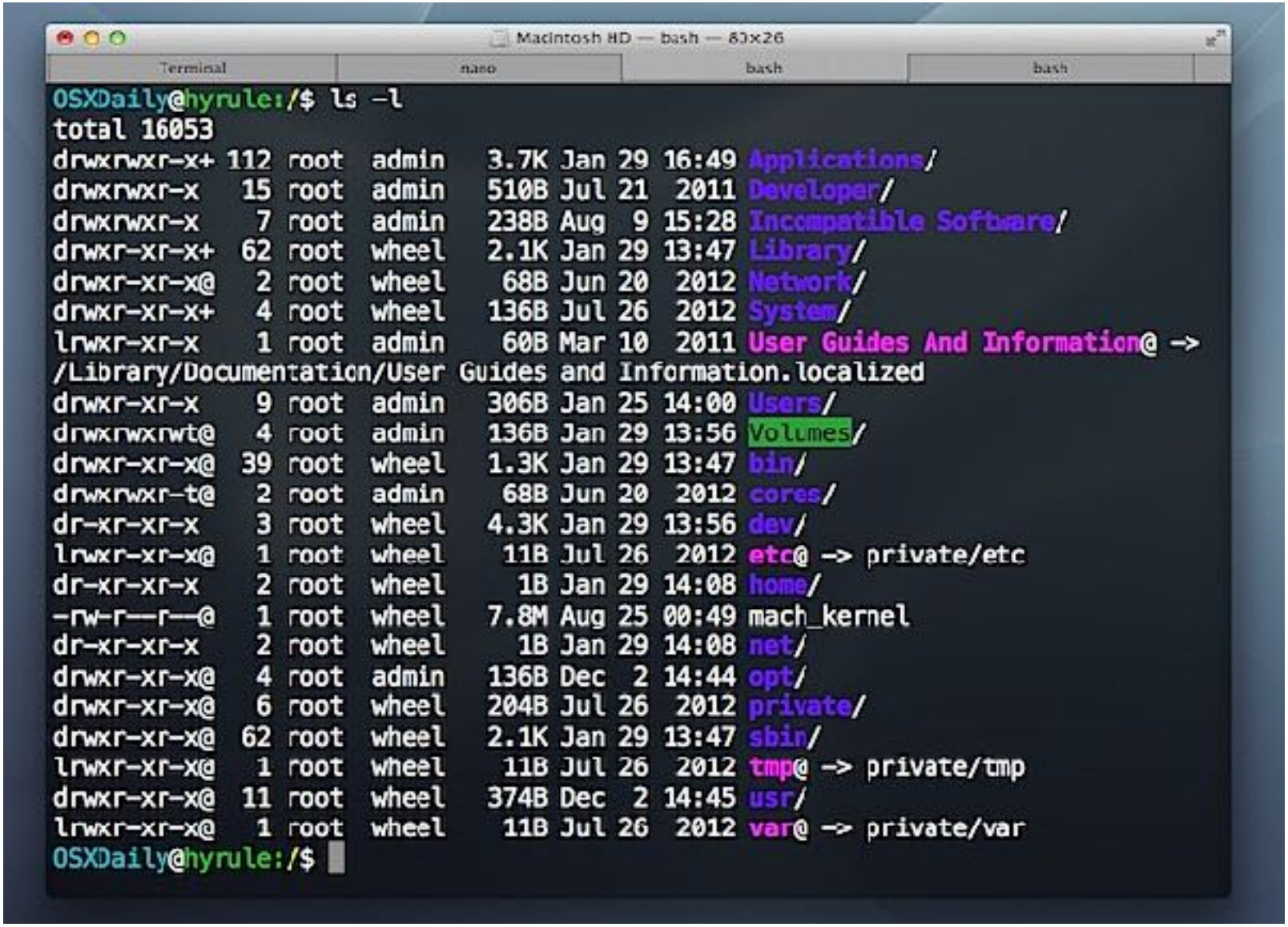
- Algorithms
- Design Patterns

Cutting Edge Development

- React.js

Let's Get Crackin!

Intro to Console / Terminal



INSTRUCTOR DEMO

Instructor: Demo

(01-ConsoleCommands)

> YOUR TURN!

Activity: Get Situated | **Suggested Time:** 1 min

Assignment:

- Make a folder on your desktop named code.
- Put all of your code that you do inside of that folder.

Best Practices:

- Always use lowercase for folder and file names.
- Never put in spaces in your folder and file names.
- Use dashes to separate.

> YOUR TURN!

Activity*:* Console Commands **| Suggested Time:** 12 min

Assignment:
From the Terminal / Console and using only the command line, create:

- A new folder with the name of first_day_stuff.
- A new HTML file with the name of first_day.html.
- Open the current folder containing the new HTML file.

Bonus:

- Create multiple directories/folders with the names one_folder and second_folder in one command.
- Create multiple files with the names one.html and two.html in one command in the first_day_stuff directory.

Discuss with Neighbors

Hello, HTML

<title> Intro to HTML </title>

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

> YOUR TURN

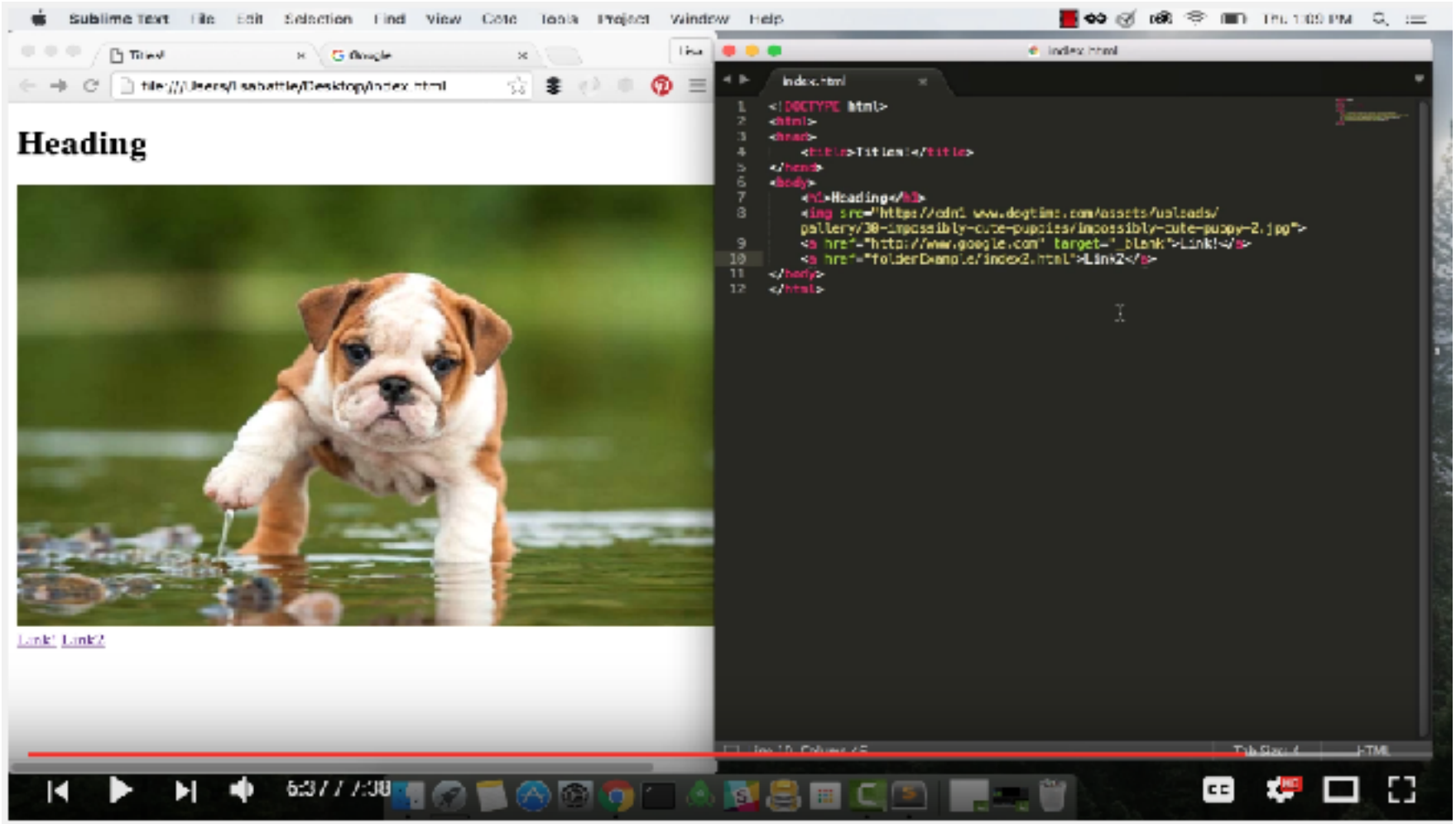
Assignment:
In a new HTML file, create the basic structure of an HTML document and include the following in it:

- DOCTYPE declaration.
- Head tag with a title tag.
- H1 tag with a title of your choice.
- Embed an image.
- Create the following three links on your page:
  - One link that is target="_blank" so that it opens a new tab when clicked on.
  - Make the second link bold.
  - Make the third link a placeholder so it goes nowhere.

Bonus:

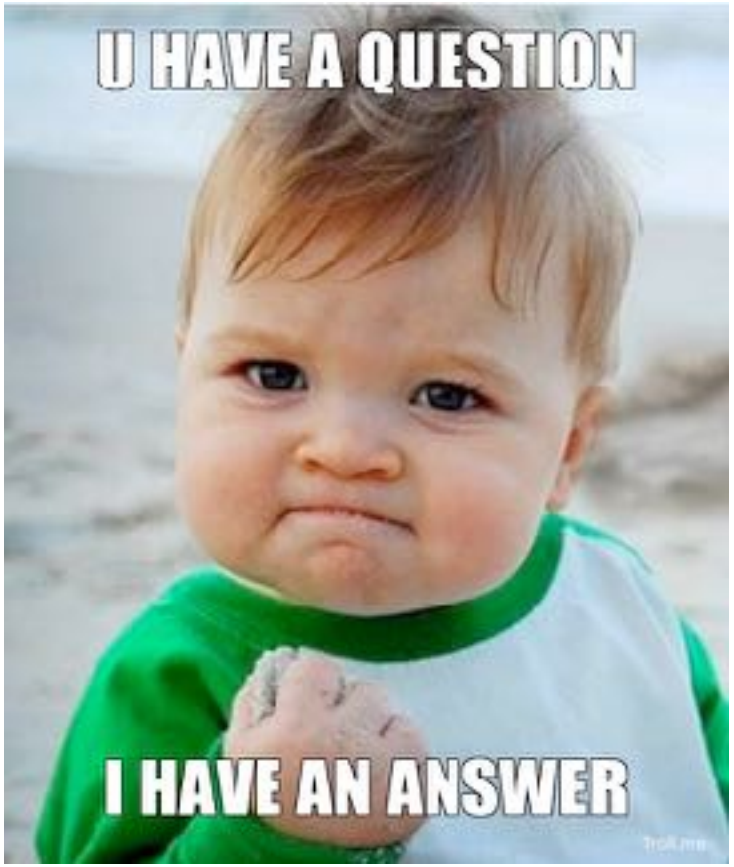
- Create an ordered list of steps to make a sandwich.
- Create an unordered list of 5 bands/musicians you like.
- Create a table with 2 columns (animal class and animal name) and 4 rows of animals.
- Use an alternate way of separating links without line breaks.
- Embed a YouTube video of your favorite band/musician.

YouTube Video Walkthrough!



https://www.youtube.com/watch?v=ieb6Svbc10E&index=1&list=PLgJ8UgkiorCnMLsUevoQRxH8t9bt7ne14

<title> Intro to HTML </title>

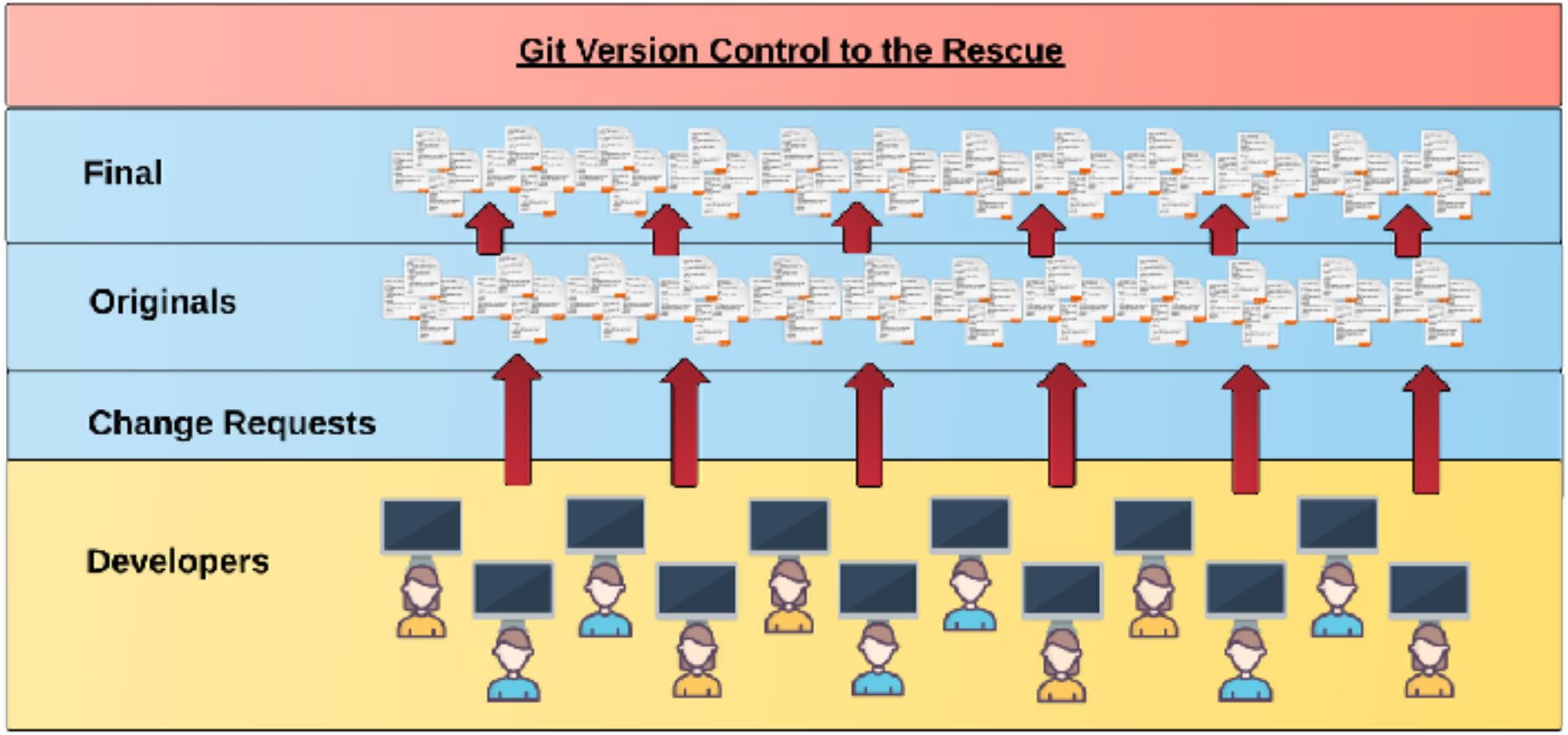


How'd it go?

LUNCH (30 Mins)

What / Why Git?

Collaborative Coding



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

The Team's Task

Task: Make a list of creative works you've written in the pa

Programming Team:

Maya Angelou

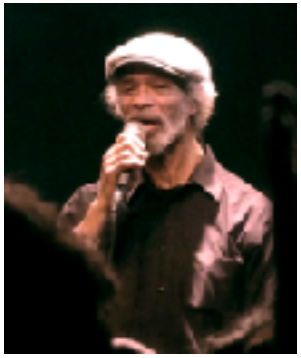
Anne Sexton

Gil Scott Heron

Maya & Gil make their edits

Programming Away…

Maya's Version



Programming Away…

Gil's Version

Different Solutions

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

Different

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

Resolution

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

“Let’s settle on this…”

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

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

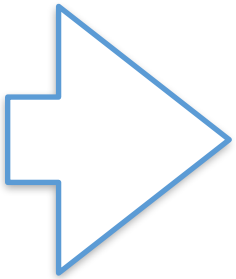
Anne writes her own stuff…

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

Anne overwrites 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</l
<li>45 Mercy Street</li>
<li>The Road Back</li>
</ul>

The Group Project

Lesson:

You should use Version Control.

….and watch your teammates' work

Today we fret and pull
on wheels, ignore our regular loss
of time…

…or maybe we should just use git

Git Version Control:

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

The Benefits of Git:

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

The Group Project

Master Branch

'Branch' = personal copy

Personal branch

Maya's branch

Anne's branch

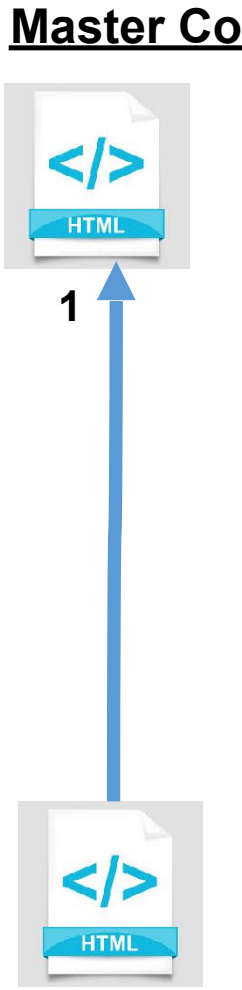
Gil's branch

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>
```

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

Maya pushes first



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

No code conflicts.

Gil's edits are ready

Rule: pull first, then push your changes

Ok

Gil pulls latest changes

Master Copy

1

Gil's Branch

Gil conflicts with master branch

Master Branch

1

```
<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.

Kobe resolves

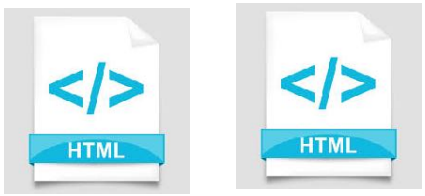
```
 <li>On the Pulse of Morning</li>
 <li>I Know Why the Caged Bird Sings</
li>
 <li>And Still I Rise</li>
<li>On the Pulse of Morning</li>
 <li>I Know Why the Caged Bird Sings</
li>
 <li>And Still I Rise</li>
```

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

Gil's Branch

Gil fixes and pushes

Master Branch



1 2

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

No code conflicts.

Gil's Branch

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

Anne starts her work

Rule: pull first, 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 pushes

Master Branch

Anne dude **pushes (uploads)** her revision the main branch. No code conflicts.

Not what we want.

Anne's Branch

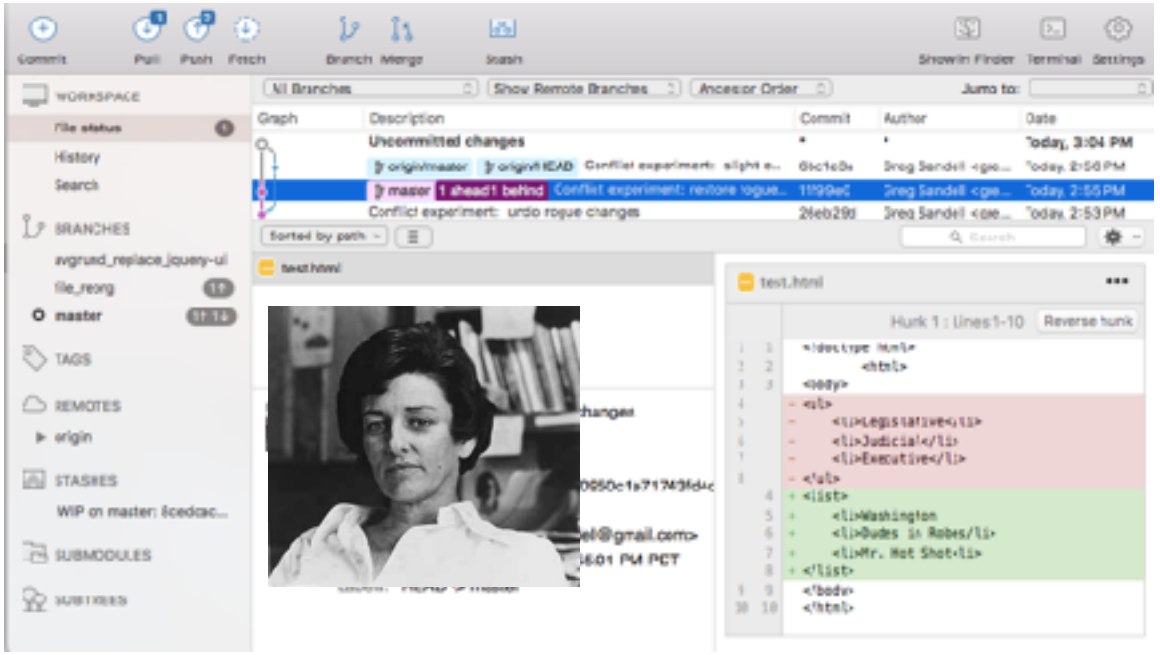
```
<ul>
    <li>The Double Image</li>
    <li>Heart's Needle</li>
    <li>Baby Picture</li>
</ul>
```

If Anne had made a pull first…

Conflict!

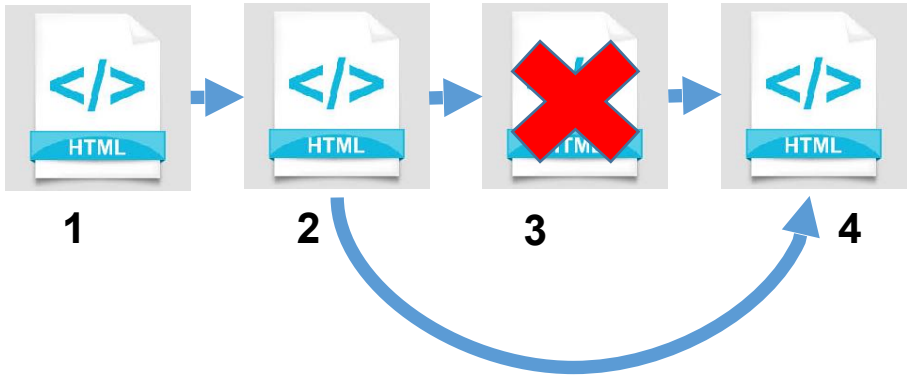
```
<ul>
    <li>The Double Image</li>
    <li>Heart's Needle</li>
    <li>Baby Picture</li>
</ul>
<ul>
    <li>Poems</li>
    <li>Albums</li>
    <li>Songs</li>
</ul>
```

The overwritten work is discovered



Roll Back

Main Branch



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

Anne's Branch

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**.

Pushing and Pulling to GitHub

1 2 3 4

GitHub Branch

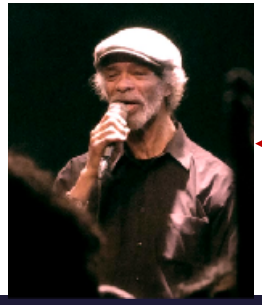
Pull Code

Push Code

Pull Code

Pull Code

Push Code

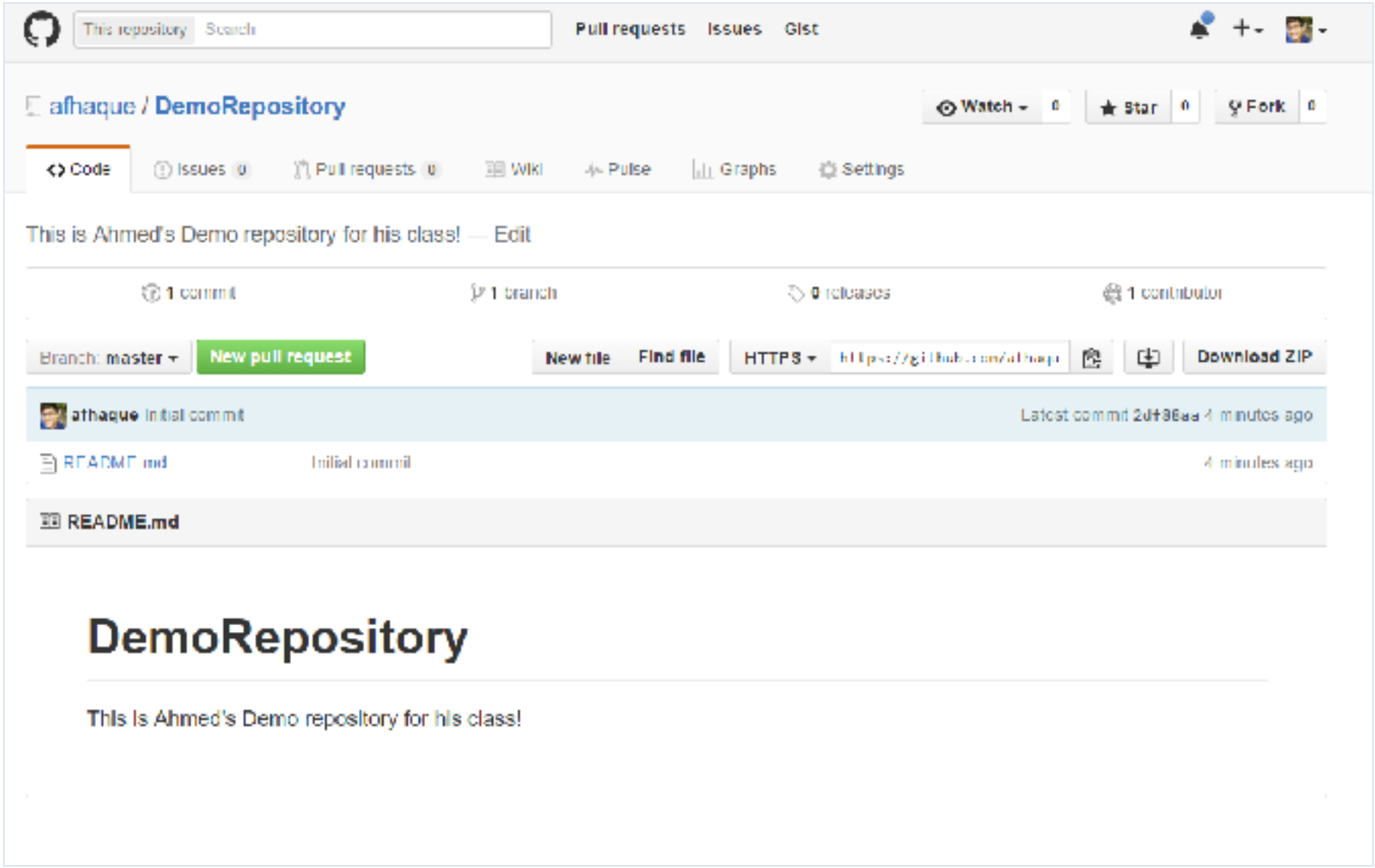


Pull Code

Push Code

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

> YOUR TURN!

Activity*:* Git Add, Commit, Push **| Suggested Time:** 20 min

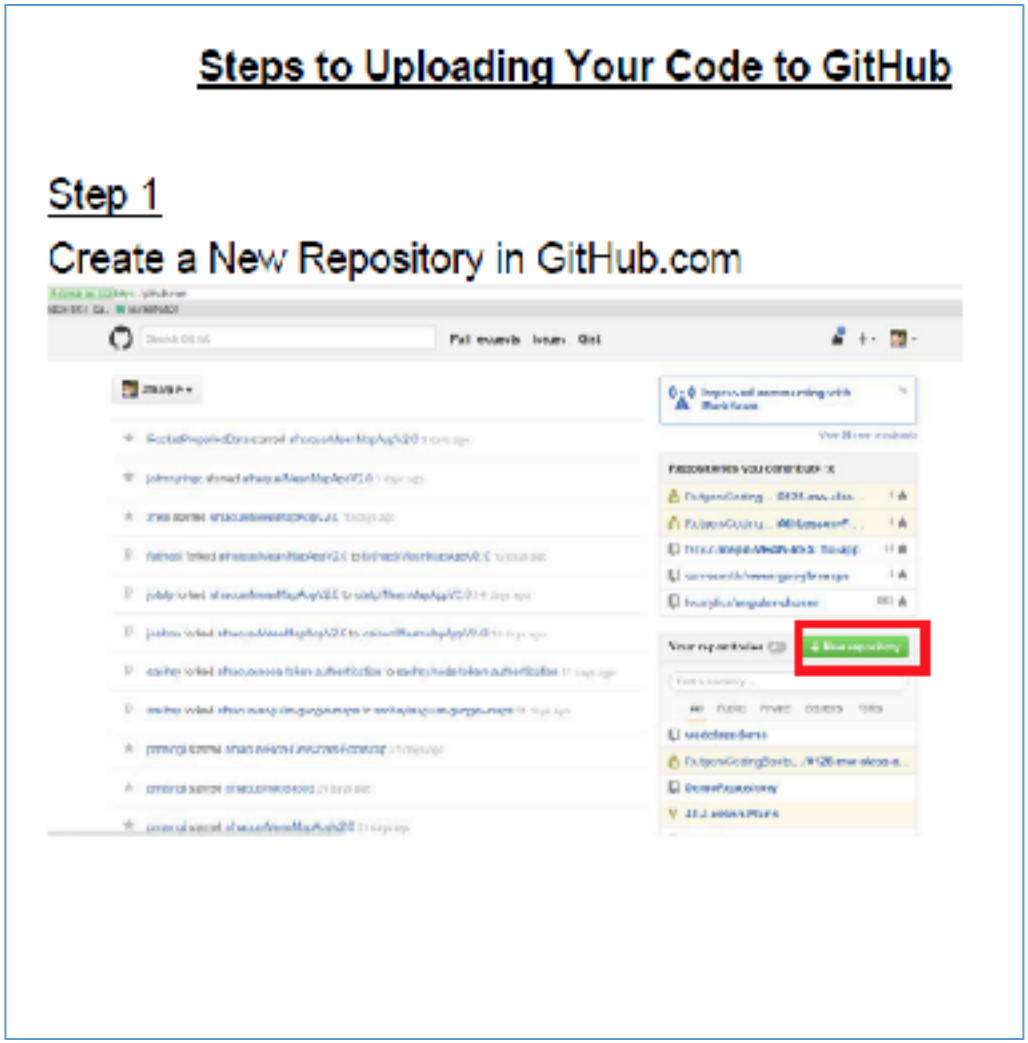
Assignment:
Using GitHub and the Command Line:

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

Bonus:

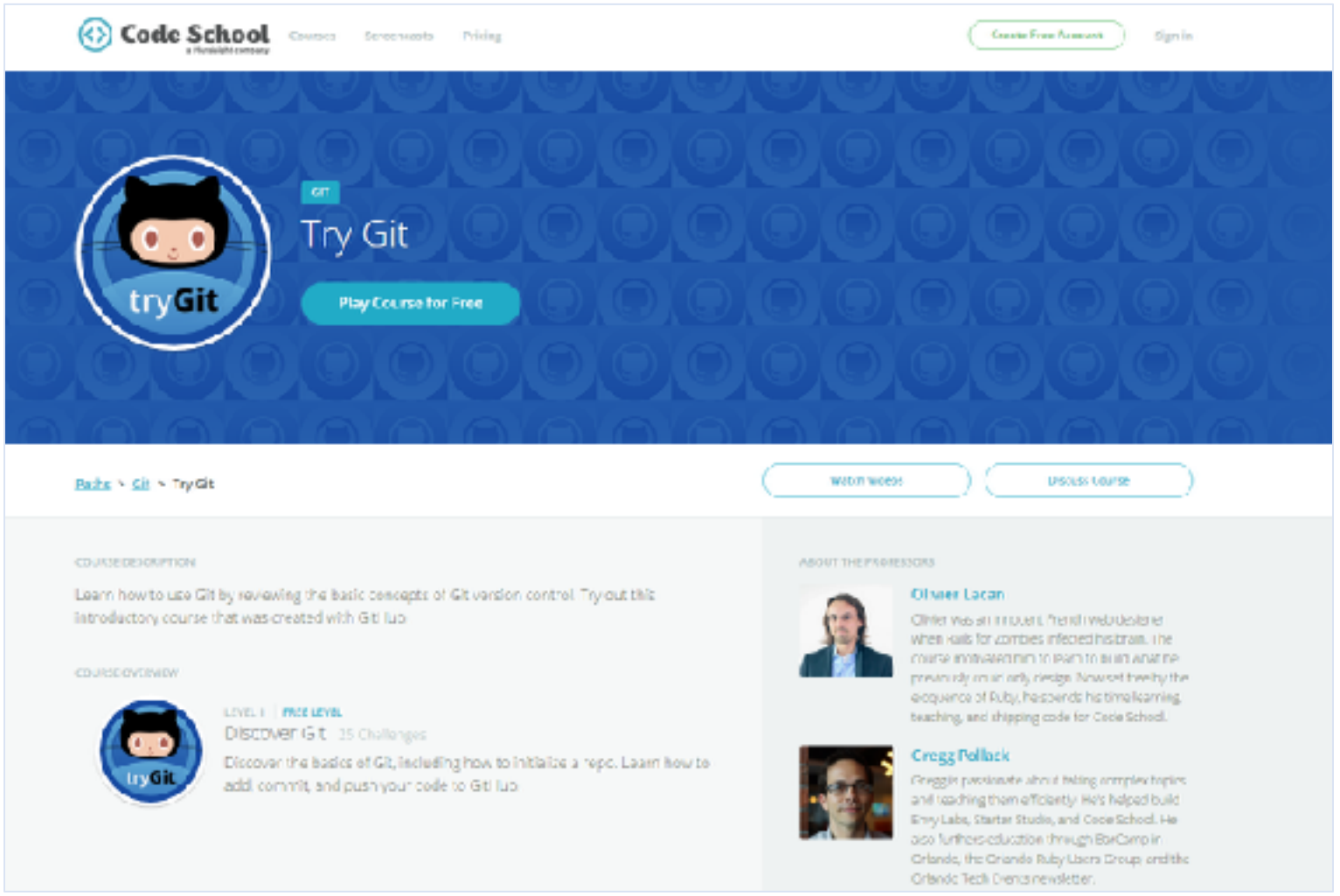
- Find a partner in class, and **fork** *their* repository to your own GitHub account. Clone this forked repository to your local directory.
- Add, Commit, and Push the code back to your forked copy.
- Finally, submit a **pull request** to send your changes to your partner’s repo.

Still a Bit Lost? Never Worry!



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

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



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

Homework!

Homework “Due”: This Week

By Next Class You Should:

- Figure out where the GitHub Repo is for our class.
- Re-do the Terminal example from class today.
- Re-do the HTML example from class today.
    - Watch the Walkthrough Video if you felt a bit lost.
- Review the Git commands we learned and how to use them