

# Let's get Started!

Welcome to week 1

# Status Check

1. Let's talk Office Hours!
2. Quick status check: Do we have all the tools needed to be productive in class? (Sublime, Github, and Chrome)
3. GitHub Setup
4. Walking through an HTML request
5. File structure, traversals, and you
6. Homework: Yep, you've got to do it!..BUT HOW?!

# Getting Started

Do we have all the tools needed to be productive in class?

Specifically, have you gone through the pre-work that GA sent over, installed Chrome and Sublime Text on your computer and watched this GitHub video to help get you setup.

If not no worries we can go through some of that now.

# Github...what now?

- GitHub is a platform makes it easy to manage git repositories.
- Stores files like Dropbox or Google Drive, but stores code.
- Stores a history of files and the changes that happens within each changed document.
- Hosts files on the cloud so you can share the finished product with other people.
- Git - the technology that Github is based on top of - was designed to allow for multiple engineers to work on the same project.

# Why is Github Valuable?

- It allows developers to go back in time if something break.
- GitHub allows multiple developers to work on the same project.
- GitHub tracks changes so you can see who worked on what.
- GitHub allows for feedback to be given on the code, which hopefully, increases code quality.

# Github Vocabulary

- git - a version control program that saves the state of your project's files and folders
- repository - a central location in which data - typically about a project - is stored and managed
- clone - download data from the cloud to your local machine
- branch - a version of a project that developers can work on
- commit - save a version of your project to git
- pull request - a request to combine branches of the same project
- push code - a request to push your latest code to your project repo
- sync - a push & pull request

# Working with others

## sharing is caring...

- Remember: "Clone" means that we are copying our GitHub repository from the cloud and saving it as a local folder on our computer.
- The main, stable version of the codebase is on the default "branch" in GitHub which is called "master".
- Engineers typically create new branches for certain features or portions of the code they will work on.
- As they are working they "commit" changes to their branch. This allows them to create a history of what they are working on so they can go back in time if an issue pops up.
- When the engineer feels confident that the code is stable and the feature they are working on is complete, they open a "pull request" to "merge" their branch to master.
- If there are multiple engineers working on a project, other engineers will review the "pull request" and provide feedback.

# Lab Time

- Let's Setup a Repo on Github together. You'll do the following:
- Log on to GitHub.com and press the big green button for "New Repository".
- Finish setting up the repo by adding a name to the repository. The default settings (public, no organization, and no readme) are fine. Name your repository "FEWD#\_yourname".
- To get started, click on the "Clone in Desktop" button. This should bring everyone back to the GitHub App.
- When the app opens, a message should pop-up that says "Clone As" with the name of our new repository pre-entered.



# Walking Through A HTML Request

WHAT HAPPENS WHEN YOU TYPE IN, “HTTP://WWW.GOOGLE.COM” INTO THE OMNIBAR?

- The browser has no idea what “http://www.google.com” means, so it looks up IP address for google.
- DNS stands for Domain Name System, it is essentially a phone book.
- The browser sends HTTP(HyperText Transfer Protocol) request — which basically asks google.com for HTML — to that IP address.
- Once the response is received by the browser, figures out what to do.  
(Display it? Download it? etc)
- (Elves)
- Yay!

# File Structure, Traversals, and You

As we all know the text, is stored in files. And files live in folders. And herein lies the next big idea of web development:

**A WEBSITE IS JUST A FOLDER STRUCTURE THAT CONTAINS A VERY SPECIFIC SET OF FILES.**

No seriously, that's all it is. Every web page you've ever visited on the internets is just a bunch of text organized in a certain way on a computer.

# Significance of index.html

You've probably seen both of these structures:

1. `http://www.somedomain.com/`
2. `http://www.somedomain.com/about.html`

You will usually see the latter after clicking on a menu item on that website. But if you go to the former, you end up on the homepage. In most setups, `http://www.somedomain.com/` is the same as `http://www.somedomain.com/index.html`

In other words: `index.html` is the default file that gets returned when you go to any website, it should always be the name of the homepage file.

# Website Folder Structure

myAwesomeFirstSite (main-folder)

- >– assets (folder)

- >–>– backgroundImage.jpg

- >–>– ...

- >–>– myHandsomeSelf.png

- >– styles (folder)

- >–>– reset.css

- >–>– ...

- >–>– main.css

- >– javascript (folder)

- >–>– main.js

- >– index.html

- >– about.html

- >– contact.html

# Notes about Structure and Traversal

Suppose we are inside our Documents folder, like so:

Documents

>- someFolder1

>- someFolder2

>- ...

>- myAwesomeFirstSite

>- ...

>- someFolderN

And I wanted to access the myHandsomeSelf.png (to update my facebook profile picture, let's say).

To access that image from the Documents, I would do something like:

myAwesomeFirstSite/assets/myHandsomeSelf.png

The forward slashes are the same as clicking into the folder with your mouse.

# Notes about Structure and Traversal

That was easy. But what if I was inside the styles/ folder, like so:

```
>- styles  
>->- reset.css  
>->- main.css
```

And I wanted to access myHandsomeSelf.png from here — what would I do?

In this case, I must do something like:

```
../assets/myDashingMug.png
```

In this case, the ../ signifies going up one level and then going inside of the assets/ folder to find the image asset. I won't belabor the point any further right now, but keep this note in mind, it will be super useful down the line.