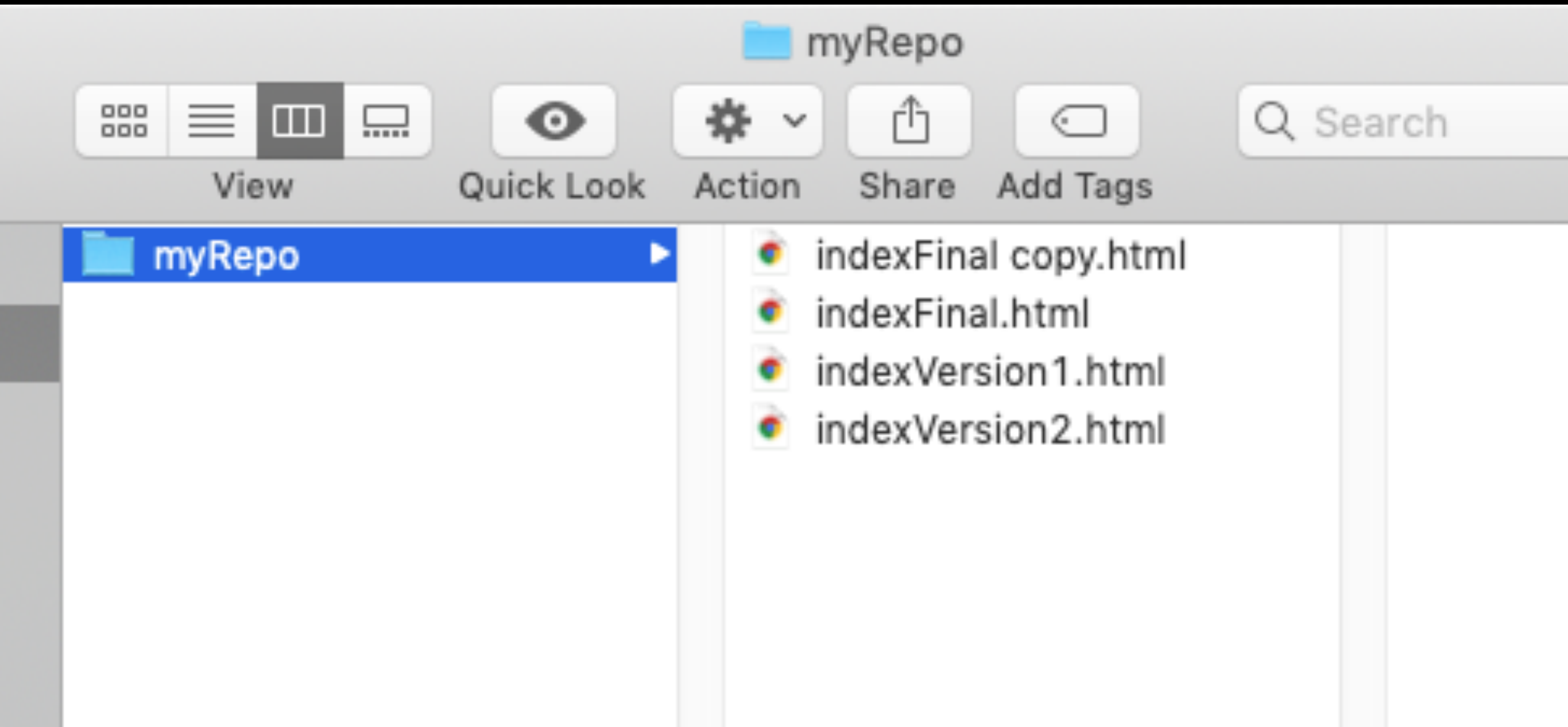


Git

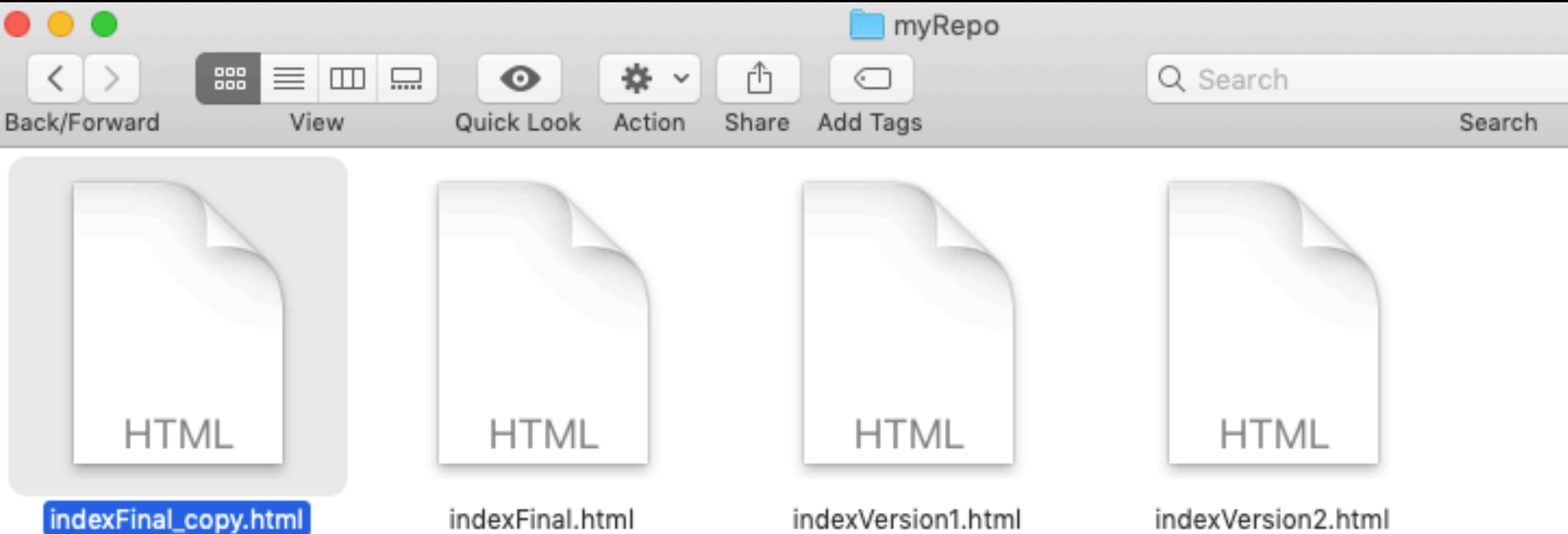
What is Git?

A version control system meant to make it easier to have multiple versions of code, sometimes across multiple developers or teams

At its most simple, git helps with the following problem:



At its most simple, git helps with the following problem: **AKA**



At its most complex, git allows professional developers to work together worldwide on software projects without over writing each other's work, causing erased code, bugs and generally breaking collaborative projects.

It is also a great resource (**web site**) to find code examples and inspiration. If you haven't already, you will likely be introduced to it and be asked to implement existing code in your Creative Code class - among many other contexts. It's like a library - only instead of taking out books, you can take out software.

****Note:** Like a library in English class you should **NEVER** take credit for someone else's intellectual property. There is a grey area between open source and plagiarism.

GitHub

Github is a service to host your code projects on the **WWW** in order to collaborate with other developers.

Code is **pushed** (uploaded) from a local **directory** (folder) on your computer thus becoming what we call in GitHub speak: a **repository** or **repo**

example - our class site:

http://www.github.com/rebleo/webDev_B_Fall2021

Git vs GitHub.com

git is a version control system that takes snapshots of your code at certain points in development

These snapshots are stored on your **local machine's storage** in a **repo**, or **repository** (in git hub speak) this translates to **directory** in **Unix** speak or **folder** in general operating system speak

GitHub.com is a website that hosts git repositories on a remote server + is available for all the web to see, copy + implement.

Git Terminology

repository - where data is managed. the directory containing your files.

local - the copy that exists on your machine, no one else can access this

remote - the copy in your github account, anyone with access to your github repo can access the remote instance (we won't be doing this!)

push - once you make changes to the local copy you *upload the changes to the remote copy

pull - if someone else makes changes to the remote copy (we won't be doing this this semester)

clone a repository - download the entire codebase of the repo you can pull in changes + and push your own changes if you are given access

Github pages

github.io

easily allows you to host web pages using github servers + workflow

url (uniform resource locator)

http://

yrUsername.github.io



yrUsername.github.io

prototype locally

prototype locally

w/ a local http server