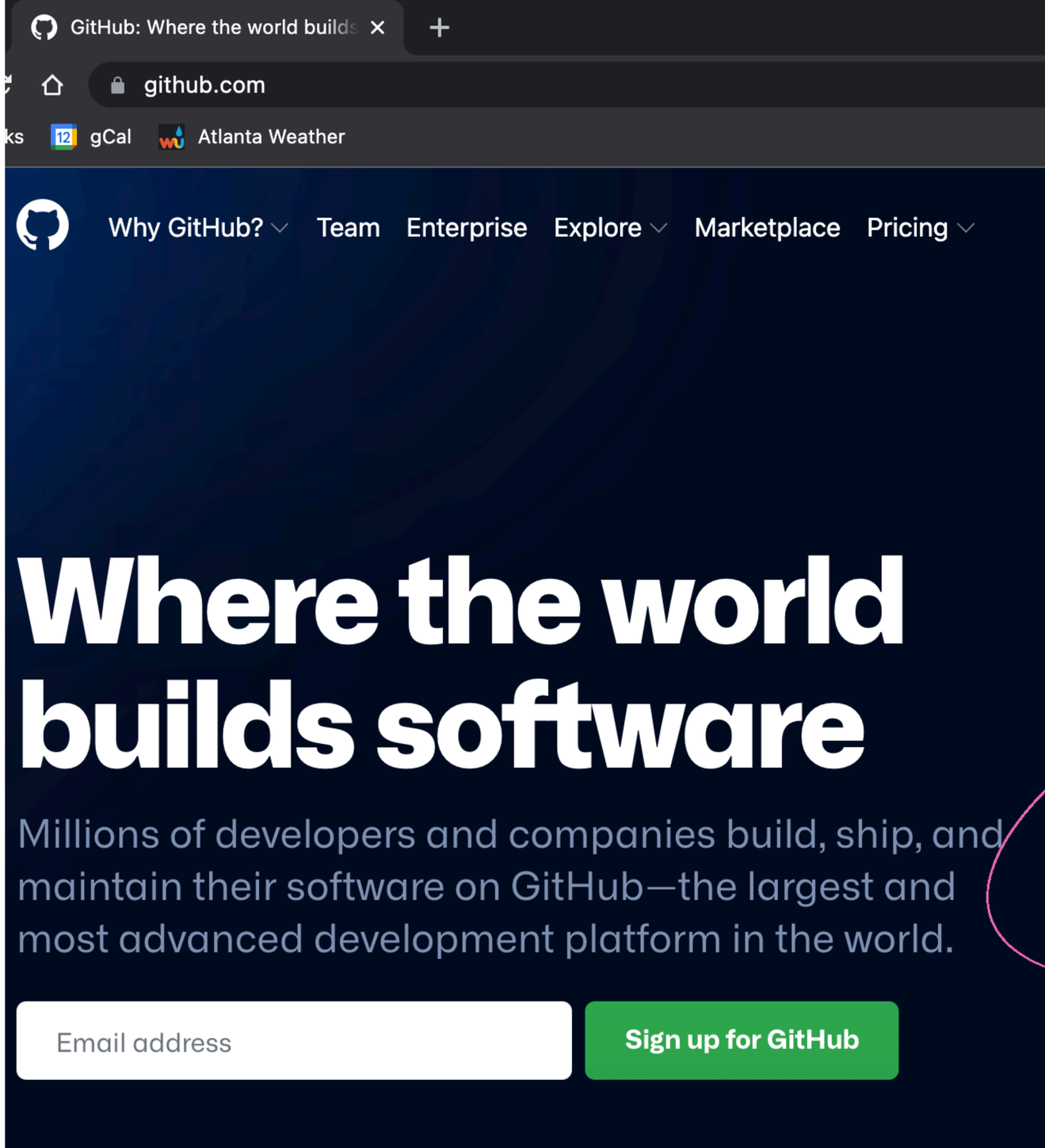
Intro to GitHub

Earth System Modeling
Georgia Tech EAS 4610

Prof. Alex Robel

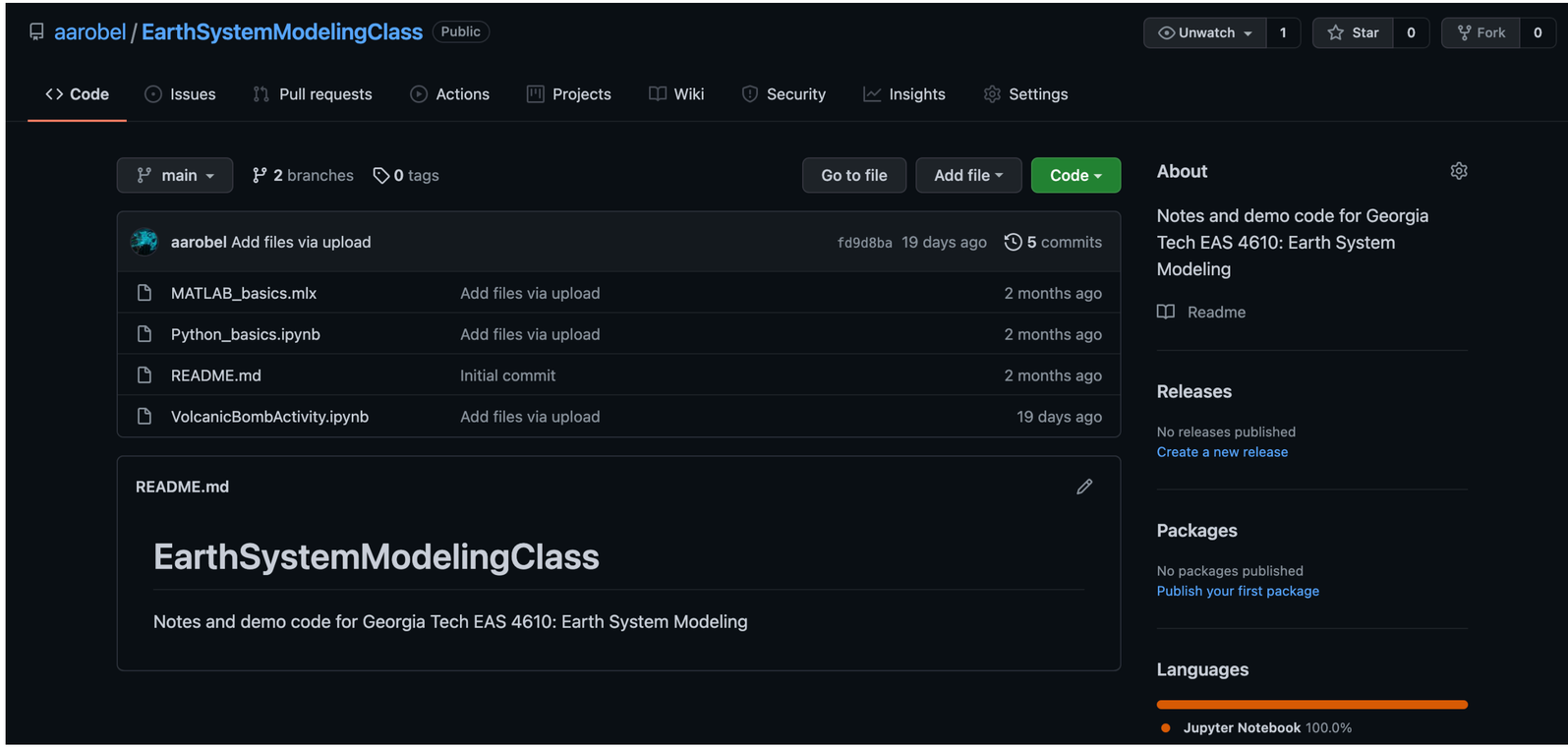
What is GitHub?

- A website widely used to share, find and collaborate on software code (and other things as well)
- Just about any modern software development (scientific, commercial and otherwise) is done through GitHub or very similar “versioning” systems



What is a “repository”?

- A directory filed with files (code, README text, and other)
- Local repository lives on your computer
- Remote/public repository is on the GitHub website and may be available for public use or only for private use by collaborators
- Called “repo” for short



What is Git?

- The software that lives on your computer, that allows you to easily, seamlessly:
  - Upload repositories from your computer to Github site
  - “Clone” or copy repositories for GitHub to your computer
  - “Merge” or sync changes from a local repository to an existing online repository

Installing Git

- Mac
  - If you have installed Xcode already, you are finished
  - Can install Xcode (warning: large file): https://developer.apple.com/xcode/
  - Or just Git itself: https://sourceforge.net/projects/git-osx-installer/
- Windows
  - https://git-scm.com/download/win
- Linux
  - Fedora: sudo dnf install git-all
  - Ubuntu: sudo apt install git-all

Command line vs. GUI

- On Mac/Windows, you can use Git through a GUI (graphical user interface) that makes GitHub as easy as drag-and-drop. Download here: https://desktop.github.com/. You can also drag and drop easily through the browser.
- In most cases, and in most of the internet support of Git/GitHub, command line is used. That is how we will proceed here.
- On Mac, open Terminal
- On Windows, open command prompt
  - Windows+R
  - Type ‘cmd’ and ‘Enter’

Some simple UNIX shell commands/terminology

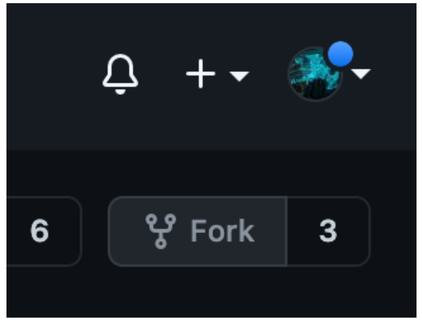
- path - location of a file in the file structure of a computer
- pwd - list current path
- cd PATH - “change directory” - go to a different path
- cp PATH1 PATH2 - copy a file located at PATH1 to PATH2
- mv PATH1 PATH2 - move/rename a file located at PATH1 to PATH2
- mkdir PATH - make a new directory at PATH
- ls PATH - list files in current directory or at PATH

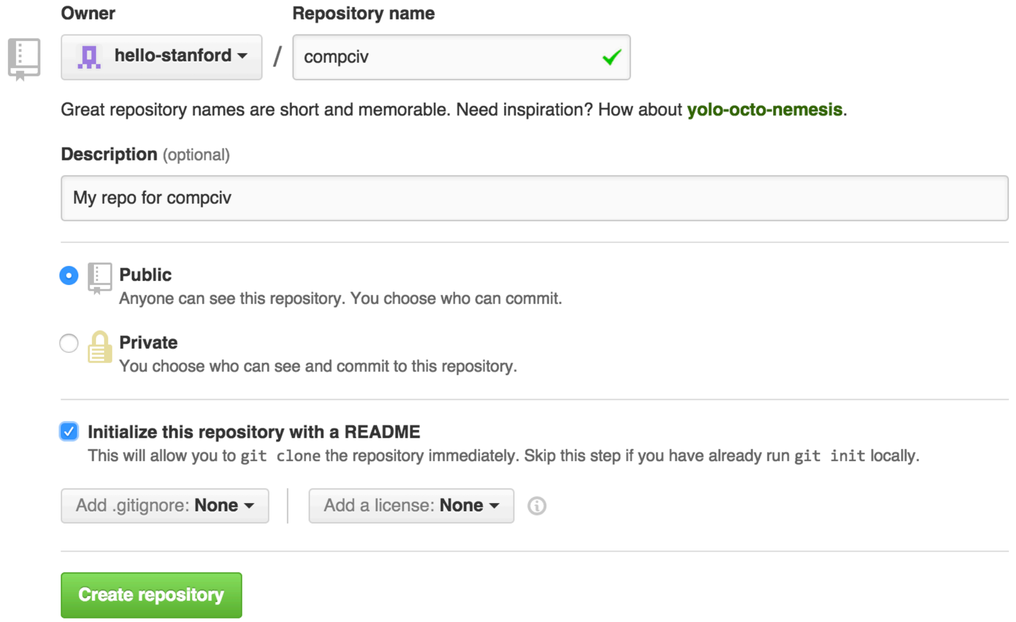
Getting started with GitHub

- Make an account: https://github.com/signup
- Let’s make a practice repository on your computer:
  - Go to terminal/command prompt
  - cd /Users/user/my_project (Mac) or cd C:/Users/user/my_project (Win)
    - Can make a new directory with mkdir
  - Can copy an existing file into the directory with cp
  - Now you have a project that you are ready to initiate as a GitHub repository

Create a new repository on GitHub

- ‘+’ sign next to profile picture in top right of GitHub website
- Name your repo
- Set it to public if you want people not in your project to see it, private if you only want invited people to see it
- The README file is the first thing someone will see if they visit your repository
- A license is a legal document that sets how other people are allowed to use your project code



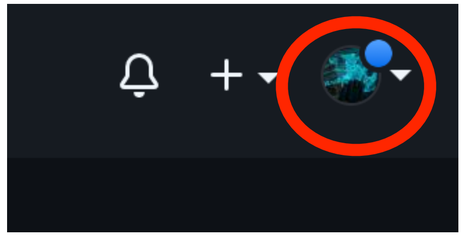


Initiating a new repository

- To start a repository from a project (made up of a directory with some files), you need to “initiate” the local repository inside the directory with all the project files:
  - git init
- Then you need to initiate a “commit” which is a new version of the local repository that you would like to then add to the public repository
  - git add * OR git add -A
    - This commands adds all the files in a directory to the commit
    - Replace * with a specific file path to only add some files to the commit
- Then add a note about the commit (i.e. a brief description of the version or what was changed in it):
  - git commit -m 'Project version 1.0’

Create a personal access token (PAT) for GitHub

- In the top right corner of any page, your profile picture and **Settings**
- On the left, click **Developer settings**
- Select Personal access tokens and click **Generate new token**
- Give the token a description/name and select **repo only** to facilitate pull, push, clone, and commit actions
- Click **Generate token**
- Copy the token somewhere – this is your new password for push/pull/cloning
- If you lose this Token, you can always generate a new one



Pushing a local repository to the remote/public

- To “push” the new commit to the remote repository (i.e. on GitHub website), we need to know three things:
  - Username
  - Personal Access Token
  - Name of .git repository
- Then we execute a push command
  - git push https://PAT@github.com/USERNAME/REPOSITORYNAME.git
- If you are successful git should let you know

Making a clone/copy of an existing repository

- Navigate to the path where you want to clone the repository
- This command will create a new directory with the same name as the repository
  - git clone https://github.com/aarobel/EarthSystemModelingClass
- You just use the URL of the GitHub repository
- This will also initiate a local repository that is already linked to the remote repository, such that you can easily make changes, initiate a new commit, and then push back to the repository.

Version control

- If you don’t own the repository (i.e. the PAT you use to push is not the same as the owner of the repository that is online) then your push command will initiate a “merge request” which the owner must approve for the changes to take effect.
    - Another GitHub user may be added as a “collaborator” on a repository in order to push without requiring the owner to approve the merge.
- All versions of the repository can be found online at the repository, and if something goes wrong, the owner can revert the repository back to an earlier version.
- This is the essence of “version control”, it allows for easy collaboration, but stores many versions in case mistakes occur

Invite a collaborator on a repository

- Under your repository name, click Settings
- In the left sidebar, click Manage access
- Click invite a collaborator