

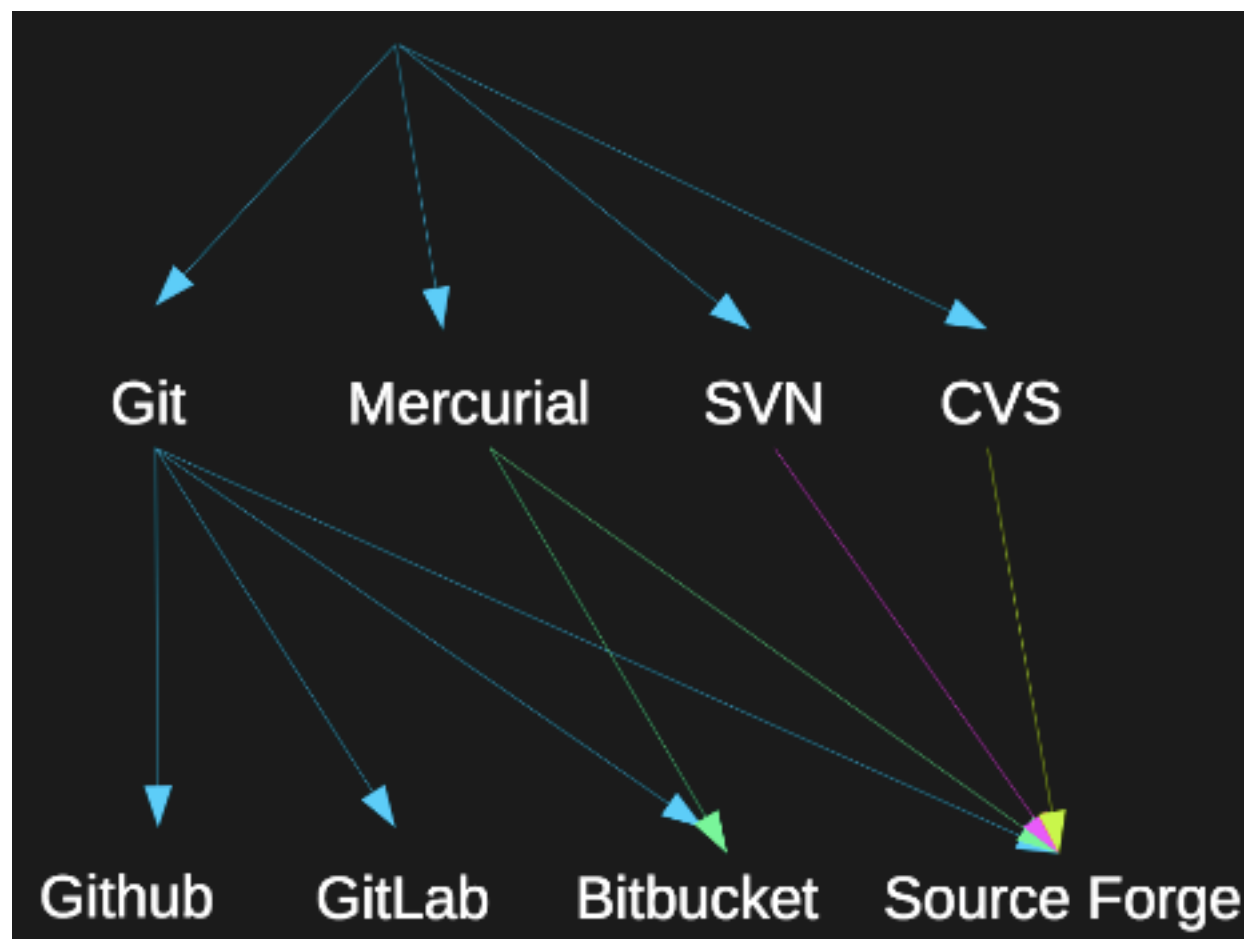
Github basics & good coding practices

ASTR400B
Ekta Patel



What is “version control”?

“Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.” —git-scm.com



various software to help track changes

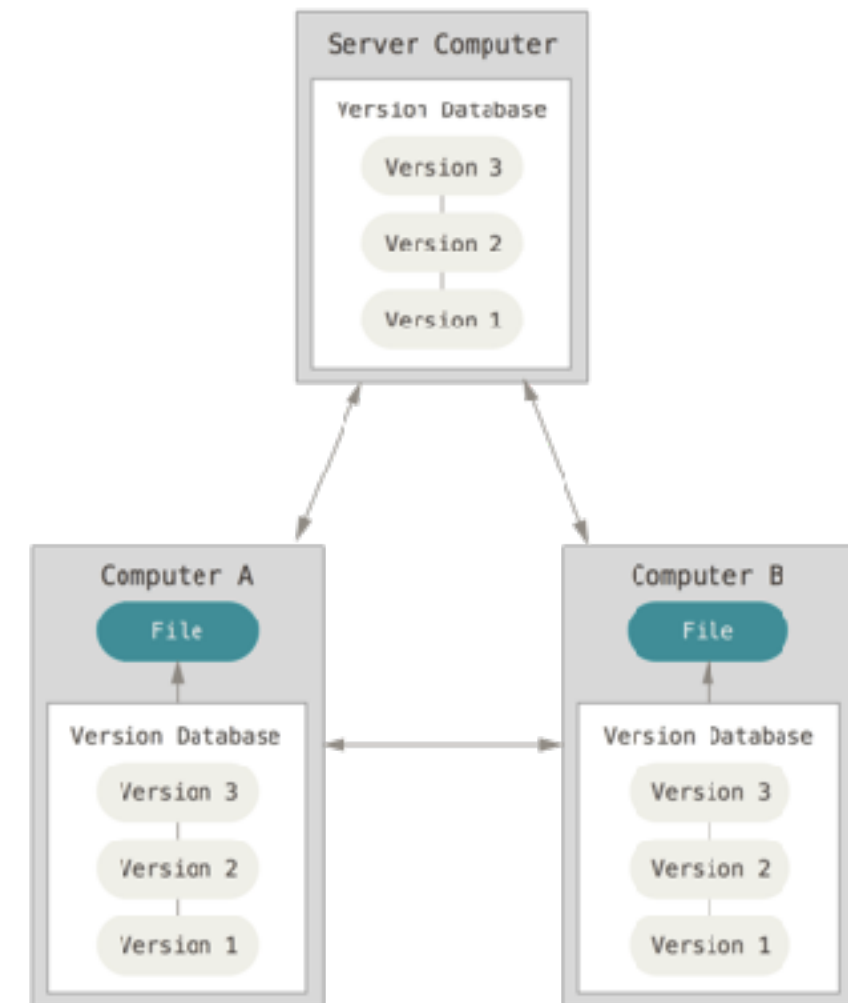
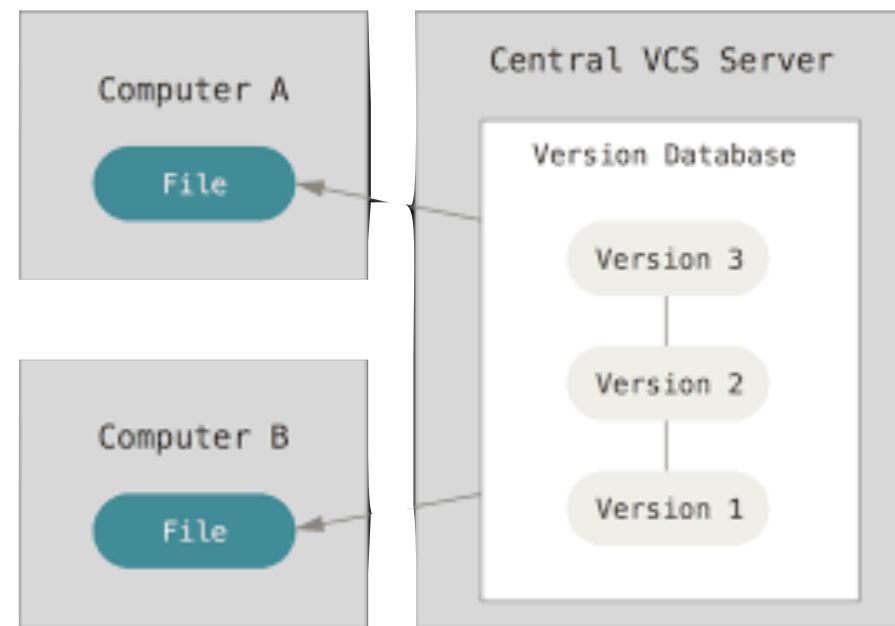
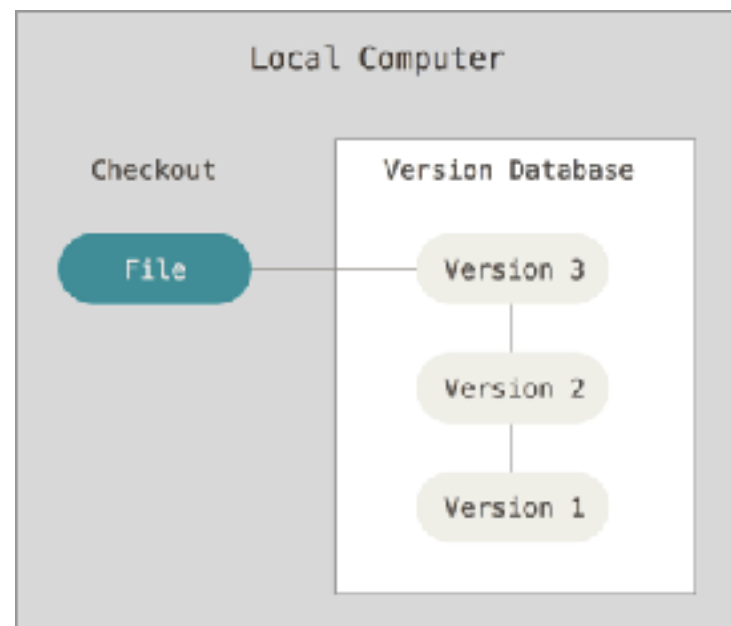
web-based hosting service for code

Types of version control

local

centralized

distributed



ex: store copies of files in Google Drive, Dropbox, etc.

one machine holds master copy and additional computers checkout certain files

***each machine has a full mirror history of all files (Git/Github!)**

**How many of you have actively
used Git/Github or another
version control software?**

Github features

- **Open source software makes science reproducible!**
(Astronomy: LSST, DES, NASA)
- **Student pack** allows for unlimited public repositories and some private repositories (sign up with UA email)
- **Host web pages** (set up your personal webpage with pages.github.com)
- **Email notifications** about code changes made by collaborators
- **Issues:** Set reminders to yourself to fix bugs or report bugs for established software packages (astropy, numpy, etc.)
- **Documentation:** Set up all code repositories with README files detailing code specifications, dependencies, etc.

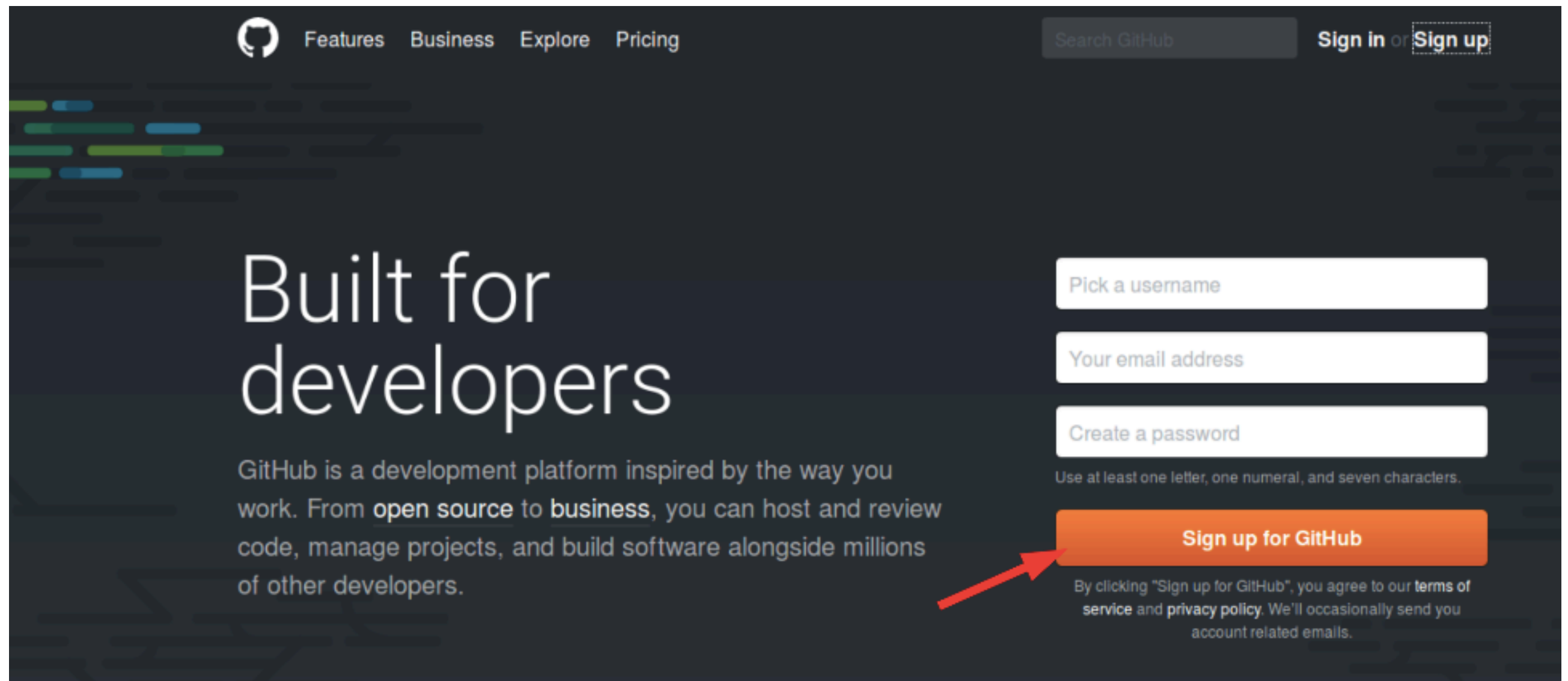
Git/Github lingo basics

- **repository (repo):** central location to store code for a *single project*
- **collaborator:** a person with read and write access to your repo
- **git [your command here]:** commands you will use on your computer
 - **clone:** create a local copy of a public repo
 - **pull:** merges any new changes to master branch into your local clone (IMPORTANT: if you have collaborators or have multiple clones of a repo, you should always 'git pull' before making new changes)
- **add:** specified files will be added to local tracked history
- **commit:** records changes to the repo (i.e. added files will be tracked globally) — message required
- **push:** updates versions of previously existing files to master repo and adds any new files established with the add+commit commands

order matters!

Questions?

Sign in on github.com

A screenshot of the GitHub website's sign-up page. The background is dark with a subtle pattern of horizontal lines in green and blue. At the top, there's a navigation bar with the GitHub logo, links for 'Features', 'Business', 'Explore', and 'Pricing', a search bar labeled 'Search GitHub', and links for 'Sign in' and 'Sign up'. The main heading 'Built for developers' is in large white text. Below it, a paragraph describes GitHub as a development platform. On the right, there's a sign-up form with three input fields: 'Pick a username', 'Your email address', and 'Create a password'. Below the password field is a note: 'Use at least one letter, one numeral, and seven characters.' At the bottom right is an orange button labeled 'Sign up for GitHub'. A red arrow points to this button. Below the button is a disclaimer: 'By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We'll occasionally send you account related emails.'

If you do not have an account yet, please make one now with your UA email!

Install git

- **Mac OSX:** <https://www.atlassian.com/git/tutorials/install-git>
- **Windows:** git-scm.com/download/win
- **Linux:**
 - `sudo yum install git-all`
 - `sudo apt-get install git-all`

If you already have git installed:

- type 'git' in a terminal
- you should see something like this:

```
[ekta [~] git
usage: git [--version] [--help] [-C <path>] [-c name=value]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
    clone      Clone a repository into a new directory
    init       Create an empty Git repository or reinitialize an existing one


work on the current change (see also: git help everyday)
    add        Add file contents to the index
    mv         Move or rename a file, a directory, or a symlink
    reset      Reset current HEAD to the specified state
    rm         Remove files from the working tree and from the index


examine the history and state (see also: git help revisions)
    bisect     Use binary search to find the commit that introduced a bug
    grep       Print lines matching a pattern
    log        Show commit logs
    show       Show various types of objects
    status     Show the working tree status


grow, mark and tweak your common history
    branch     List, create, or delete branches
    checkout   Switch branches or restore working tree files
    commit     Record changes to the repository
    diff       Show changes between commits, commit and working tree, etc
    merge      Join two or more development histories together
    rebase     Reapply commits on top of another base tip
    tag        Create, list, delete or verify a tag object signed with GPG
```

If you do not see this, go back to the previous slide and install!

Set up git

- Link your computer with your Github username and email
- Type the following into a terminal:
 - `git config --global user.name "myusername"`
 - `git config --global user.email "my@emailaddress"`
- you can also store your password: <https://help.github.com/articles/caching-your-github-password-in-git/#platform-mac>

Clone repo for this class

- Navigate to https://github.com/gurtina/ASTR400B_2018
- On the right side click the green “clone or download” button
- Copy the link in the drop down box
- In a terminal, navigate to wherever you would like to store the class repository
- Type `git clone https://github.com/gurtina/ASTR400B_2018.git`

If the clone was successful, you should see this in your terminal:

```
[ekta [Desktop] git clone https://github.com/gurtina/ASTR400B_2018.git
Cloning into 'ASTR400B_2018'...
remote: Counting objects: 24, done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 24 (delta 2), reused 17 (delta 2), pack-reused 0
Unpacking objects: 100% (24/24), done.
```

Locate/Create a repo

- Create a **public** repository: <https://help.github.com/articles/create-a-repo/>
- Your repository should be called `ASTR400B_Lastname`
- If you already did this, please locate the correct repo now.

Clone your class repo

- Navigate to your repository: https://github.com/ekta1224/ASTR400B_Patel
- On the right side click the green “clone or download” button
- Copy the link in the drop down box
- In a terminal, navigate to wherever you would like to store the class repository
- Type `git clone https://github.com/ekta1224/ASTR400B_Patel.git`

- **You now have a local copy of the class repository ASTR400B_2018:**
 - to receive updates, you should navigate to the directory on your computer and then type 'git pull'
 - you will never need to use any other commands for the purpose of this class with this specific repository
 - you can also view all files on the web interface
- **You also have a local copy of your own class repository ASTR400B_Lastname:**
 - in this repo you will add, commit, push files to submit your assignments
 - you will not need to pull changes unless you work on multiple computers
 - let's try an example!

Push to ASTR400B_Lastname

- Navigate to your directory in a terminal.
- Open your favorite text editor (emacs, vim, Text Wrangler, etc.) with a file called 'example.txt'
- In this file, type your favorite astronomical object, save, and close the file.
- Alternative: `echo 'favorite object' > example.txt`

Push to ASTR400B_Lastname

- Type `git add example.txt`
- Type `git commit -m`adding first file``
- Type `git push`
- Go to `github.com/username/ASTR400B_Lastname`
- You should now see `example.txt` in your master repo
- You will follow the exact steps above to submit Jupyter Notebooks or python scripts for your homework.

Questions?