A Brief Introduction to git and GitHub

Cliff Johnson
Northwestern University and Adler Planetarium

git

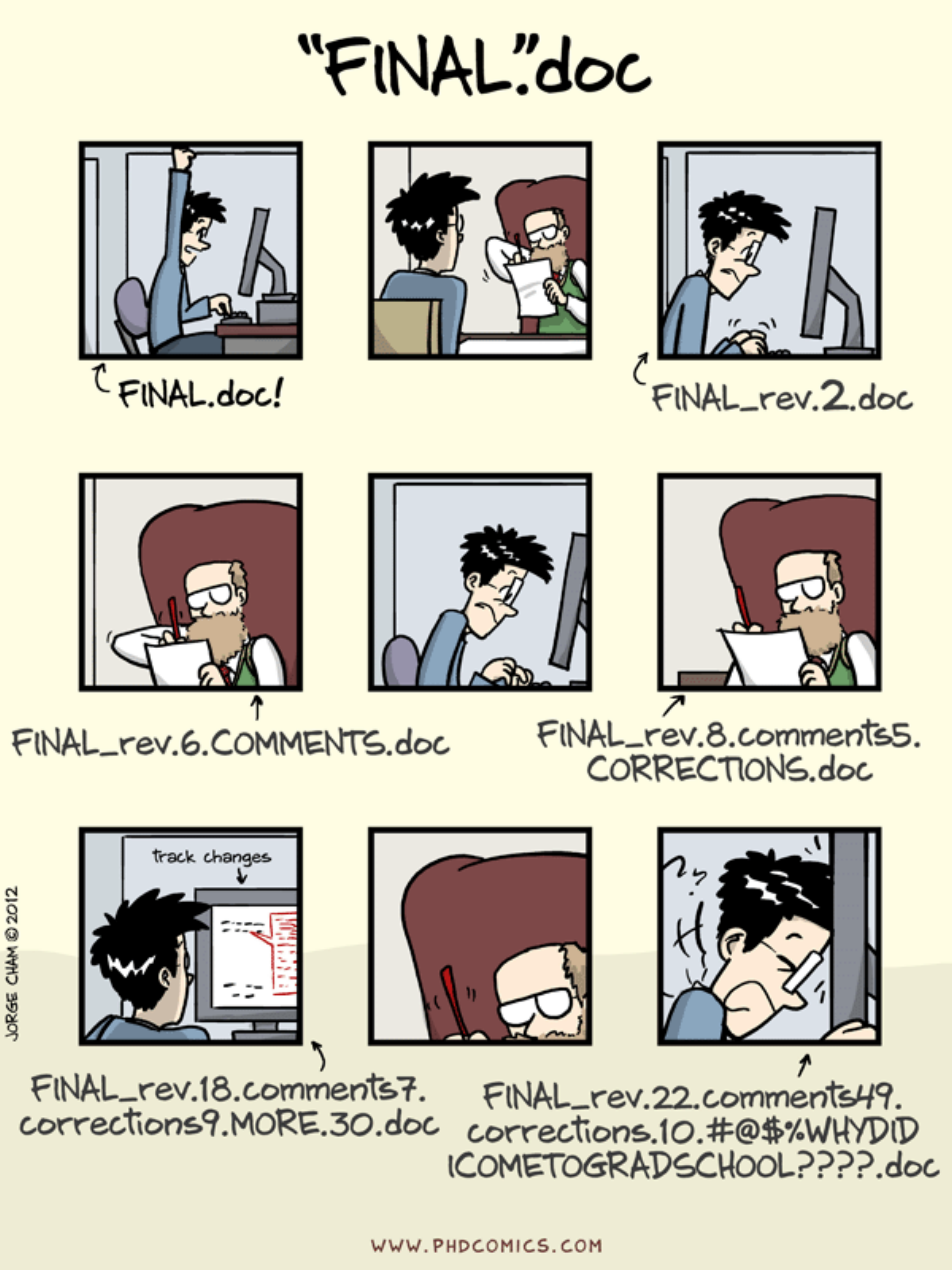
Version Control

Alts: Subversion/svn, Mercurial

GitHub

Repository Hosting Service
repository = “repo”

Alts: BitBucket, GitLab


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.
CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.
corrections9.MORE.30.doc
FINAL_rev.22.comments49.
corrections.10.#@$%WHYDID
ICOMETOGRADSCHOOL????.doc
JORGE CHAM ©2012
WWW.PHDCOMICS.COM

Goals

- Keep track of changes to files
- Revert back to old versions of files
- Test changes to code/files without losing original
- Synchronizes code between different people, facilitating collaborative work

git commands:

```
add

commit

push
```

```
git add <file>

git commit -m
“<descriptive message>”
```

Note: [blah] is optional, <blah> means fill in with your own text, <<blah>> is a comment you should omit

```
mkdir test-2020
cd test-2020
echo "# test-2020" >> README.md

git init

git add README.md
git commit -m "first commit"
git status

git log [--oneline]
git reset

git reset --hard <hash>
git reset --soft <hash>
git reset --hard HEAD

(git revert)
```

git branch

```
git branch <new_branch>
git checkout <new_branch>
```

```
git checkout -b <new_branch>
```

```
git branch -D <branch>
```

`git merge`

Handling a dreaded merge conflict!

```
git push origin master
```

Remote Name Remote Branch

```
git remote add origin https://github.com/<user>/<repo>.git
git push --set-upstream origin master
```

In new directory:

```
git clone <repo url>
```

Or use GitHub Desktop

```
git clone https://github.com/<user>/<repo>.git
git clone https://github.com/ageller/CIERA_REU.git
```

More to come!

- Open source and collaborative workflows
- Intro to other useful GitHub features

Tangents

Markdown
- See Cheat Sheet = https://www.markdownguide.org/cheat-sheet/

Github Security Best Practices
- DON’T COMMIT PASSWORDS, KEYS, OR CREDENTIALS

Github Authentication
- Optional: 2FA and personal access tokens
- Desktop based OSX Keychain usage = https://desktop.github.com/
- git config --global credential.helper 'cache --timeout=3600'

GitHub Tutorial: Part 2

- Open source and collaborative workflows
  - Forks & Pull Requests
  - Development Patterns
- Intro to other useful GitHub features
  - Issues
  - GitHub Pages

GitHub Collaboration

- Two Methods:
  - Create a branch
  - Fork a repo

GitHub Collaboration

- Make a change, submit a pull request
  - Make changes on local copy of fork
  - Push to remote copy of fork
  - Use GitHub interface to create pull request to transfer changes back to original repo.

GitHub Collaboration

- Staying Up-to-date
  - Setup addition “upstream” remote

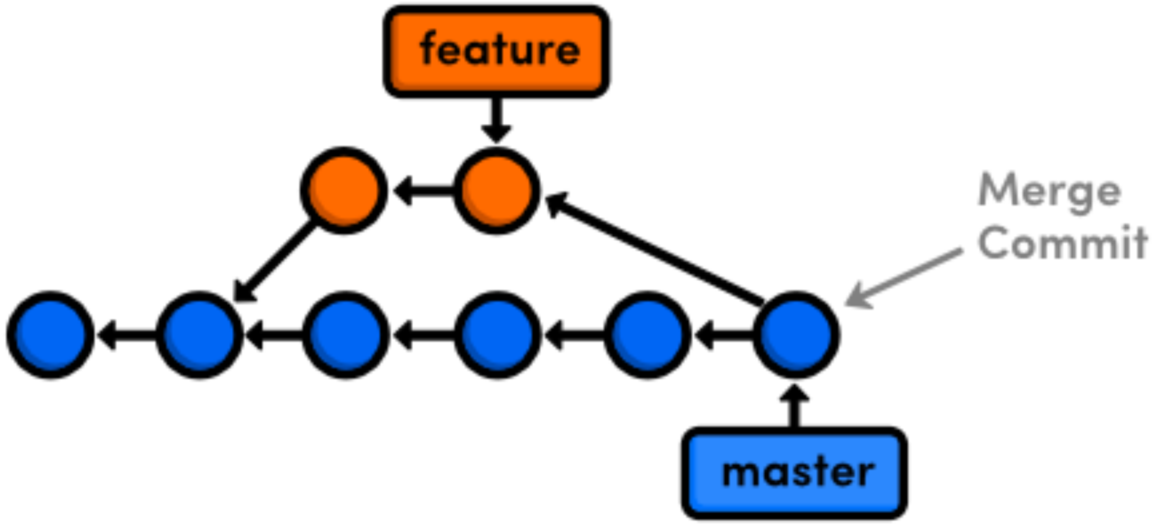
```
git remote add upstream <repo_url>
```

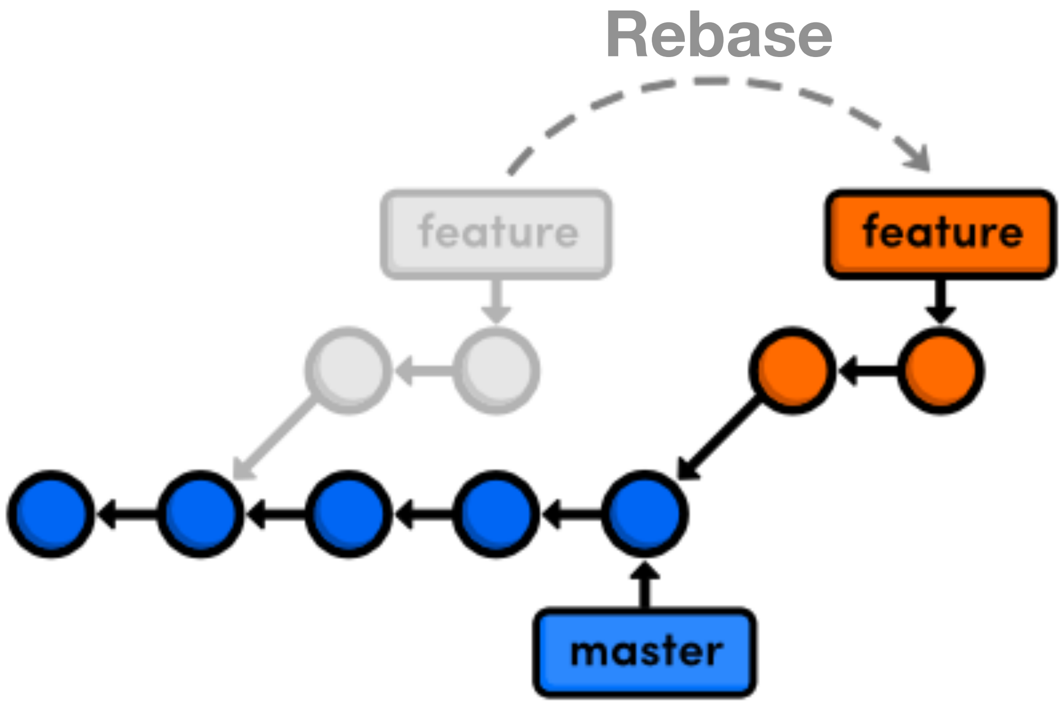
  - Sync with upstream repo

```
git fetch upstream master
git merge upstream/master
```

GitHub Collaboration

- Rebase vs. Merge





GitHub Collaboration

- Development Install: instead of installing a particular version of code/library, install
  - Use case: to stay up-to-date with latest code changes (between releases), to make local custom changes
  - From code directory (which contains setup.py file):

```
pip install -e .
```

See: https://docs.astropy.org/en/stable/development/workflow/development_workflow.html

GitHub Pages

- GitHub provides free hosting!
    - Visit the “Settings” page for a repo, and select from available hosting options: host contents of `master branch or /docs folder

Sources

An Introduction to Git and GitHub by Brian Yu
Link = https://www.youtube.com/watch?v=MJUJ4wbFm_A

NU EECS-330 Assignment Doc
Link = https://docs.google.com/document/d/17bt57nFiLo60ehKhSid5lSFofiXCXY5xKqZ2PSZb5g4/edit

Git Tower FAQ
https://www.git-tower.com/learn/git/faq

GitHub - Adding Existing Projects
https://help.github.com/en/github/importing-your-projects-to-github/adding-an-existing-project-to-github-using-the-command-line

Git Rebasing
https://code.tutsplus.com/tutorials/rewriting-history-with-git-rebase--cms-23191