Version Control with Git

Matt Critchlow - UC San Diego Library

Reminders

- **Git installation** and **Github account**
- Sticky Notes!
- Etherpad https://public.etherpad-mozilla.org/p/library-carpentry-ucsd
- Ask questions at any time!

What are we covering?

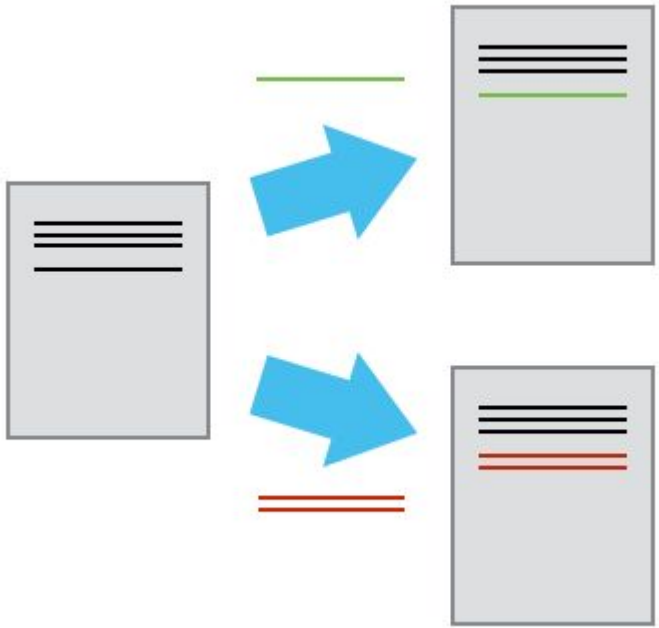
- What is version control?
- Why should I use it to manage my work?
- Ok, I’m sold. How do I use it?
- Discussion: Using Git/Github at work

"FINAL".doc

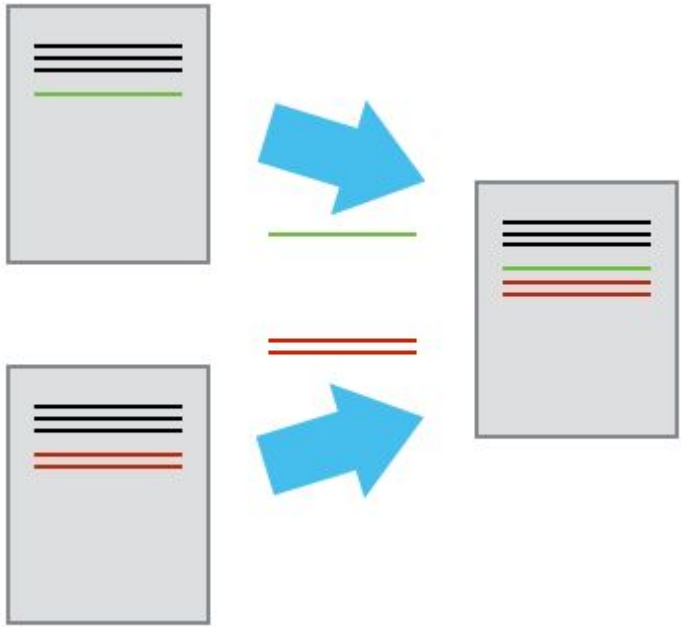

FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.
CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.
corrections9.MORE.30.doc
FINAL_rev.22.comments49.
corrections.10.#@$%WHYDID
ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012

Changes Saved Sequentially

Divergent Versions



Merging



Brief Glossary

- **Version Control** - A tool for managing changes to a set of files. Each set of changes creates a new commit of the files; the version control system allows users to recover old commits reliably, and helps manage conflicting changes made by different users.
- **Commit** - To record the current state of a set of files (a change set) in a version control repository
- **Repository** - A storage area where a version control system stores the full history of commits of a project and information about who changed what, when.
- **Merge** - To reconcile two sets of changes to a repository

DVCS vs VCS

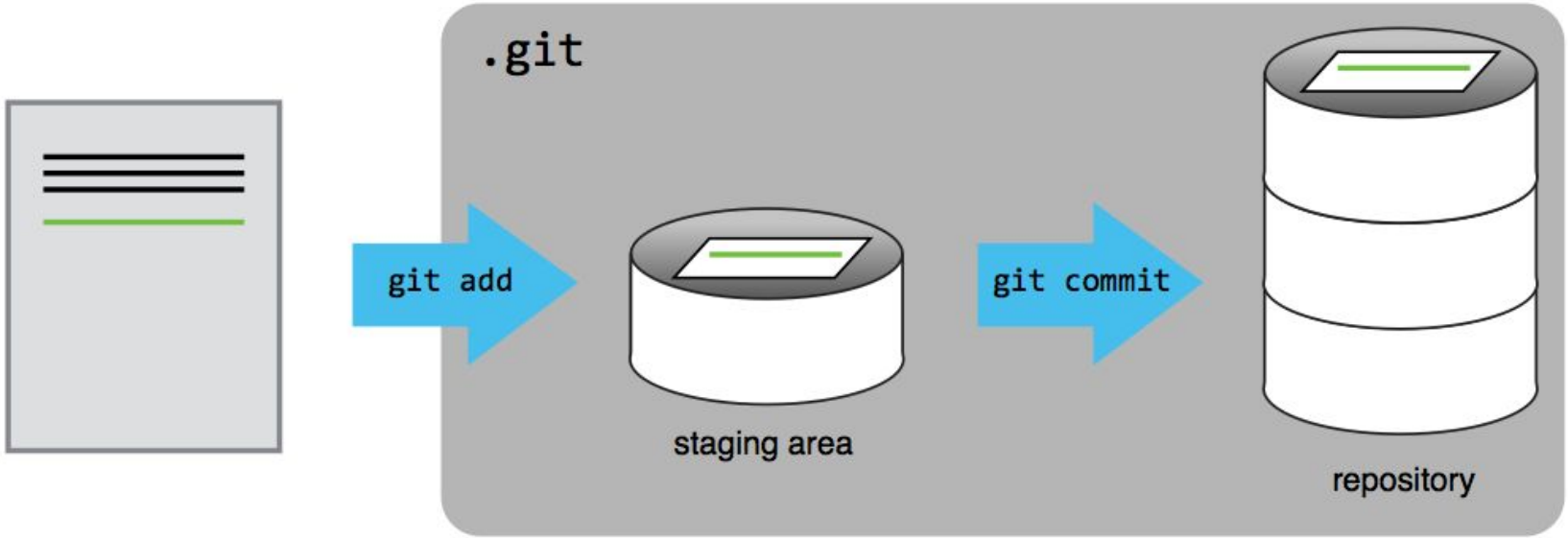
- VCS - Version Control System - CVS, Subversion, RCS
- DVCS - ***Distributed*** Version Control System - Git, Mercurial

Challenge: Committing Changes to Git

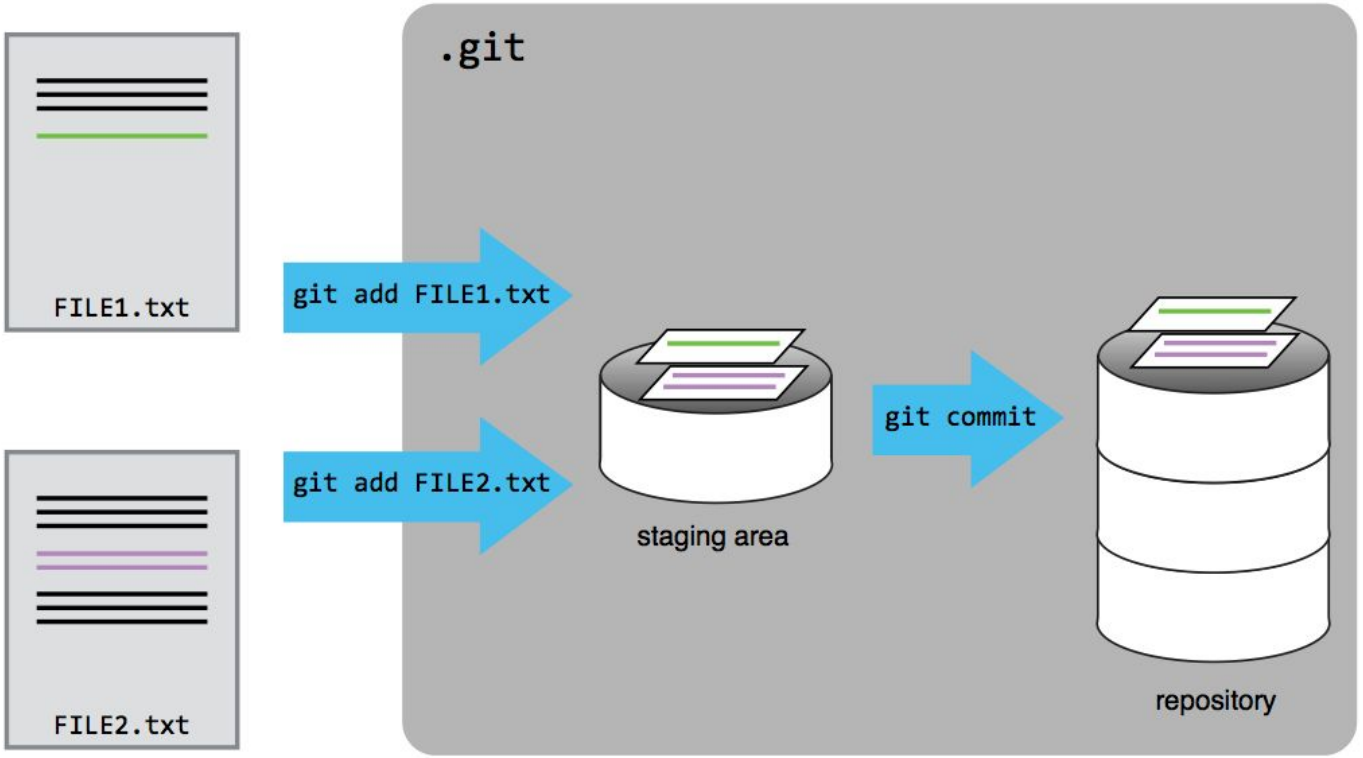
Which command(s) below would save the changes of myfile.txt to my local Git repository?

1. $ git commit -m “my recent changes”
2. $ git init myfile.txt -> $ git commit -m “my recent changes”
3. $ git add myfile.txt -> $ git commit -m “my recent changes”
4. $ git commit -m myfile.txt “my recent changes”

Git Staging Area



Git Staging Area (multiple files)



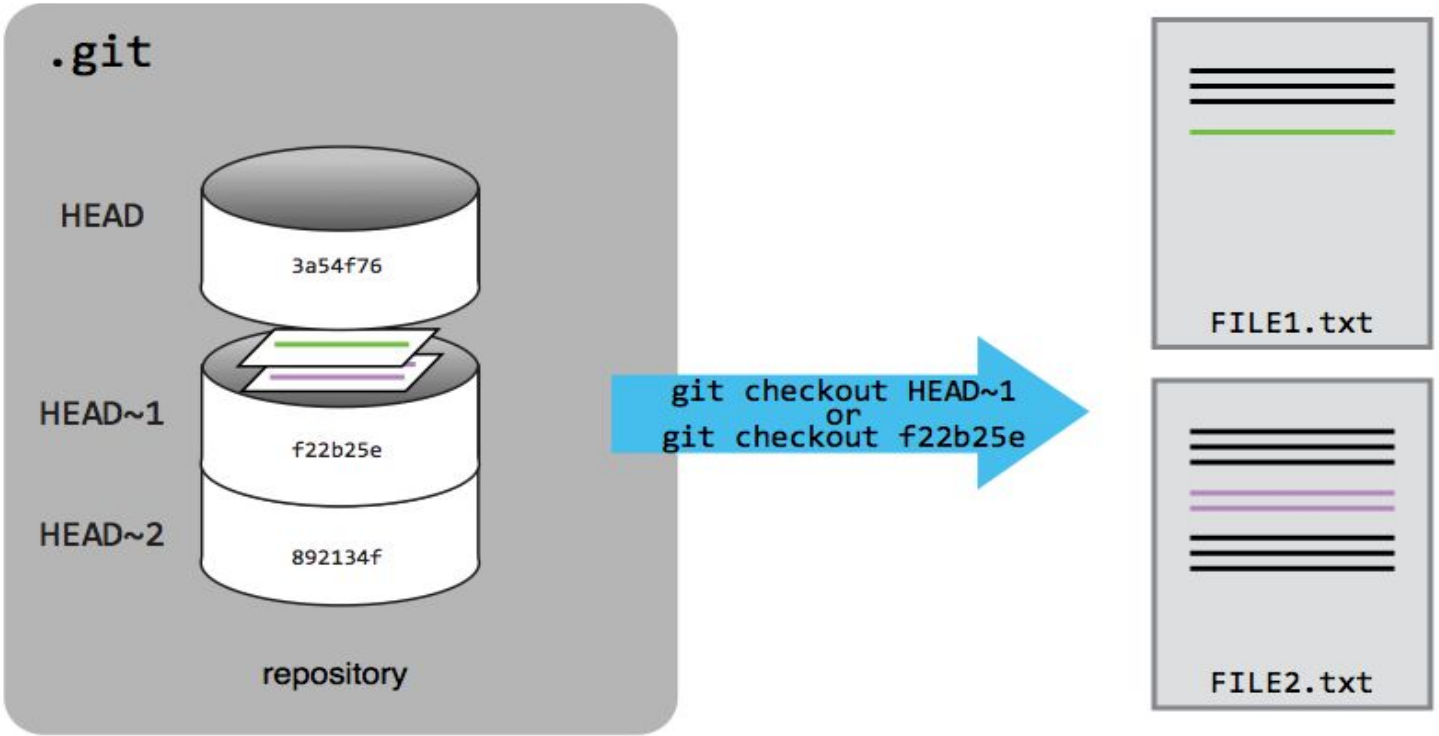
Challenge: Tracking Changes

- Add two additional git command descriptions that we've discussed to your basics.md
- There are three to choose from: **commit, diff, log**
- Commit your changes
- Modify the description of one of the commands, and add a description of the third
- Display the differences between its updated state and its original state
- Commit your changes
- View your version history to confirm

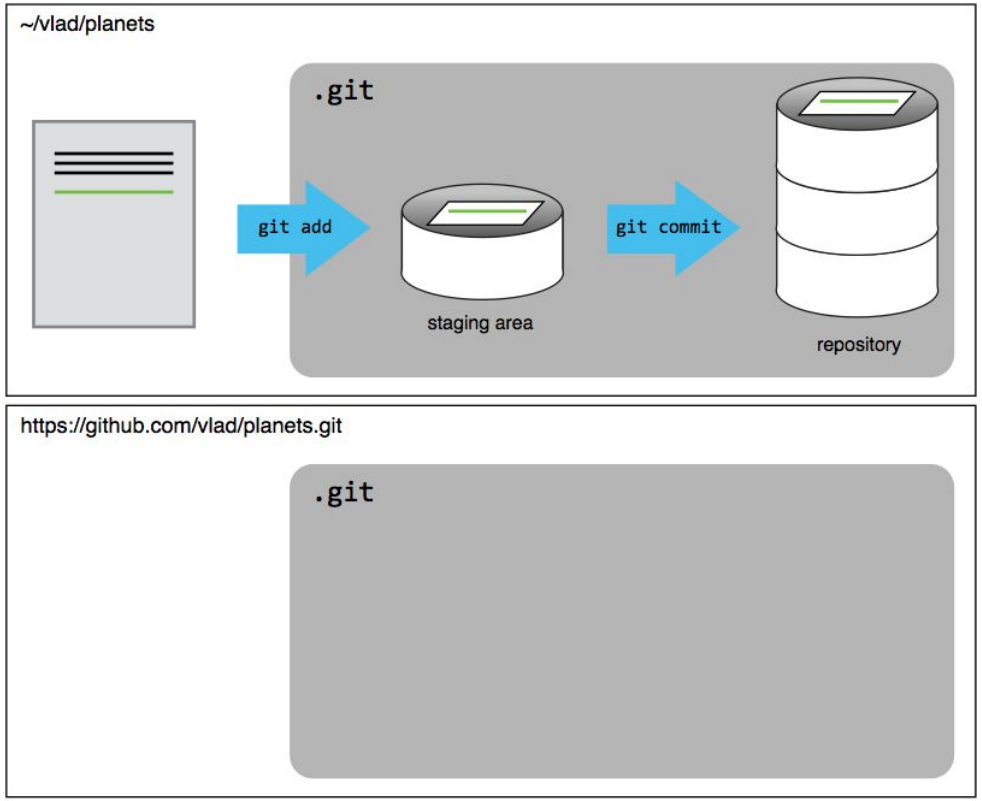
Challenge: Recovering Older Versions of a File

1. $ git checkout HEAD
2. $ git checkout HEAD data_cruncher.py
3. $ git checkout HEAD~1 data_cruncher.py
4. $ git checkout <unique ID of last commit> data_cruncher.py
5. Both 2 and 4

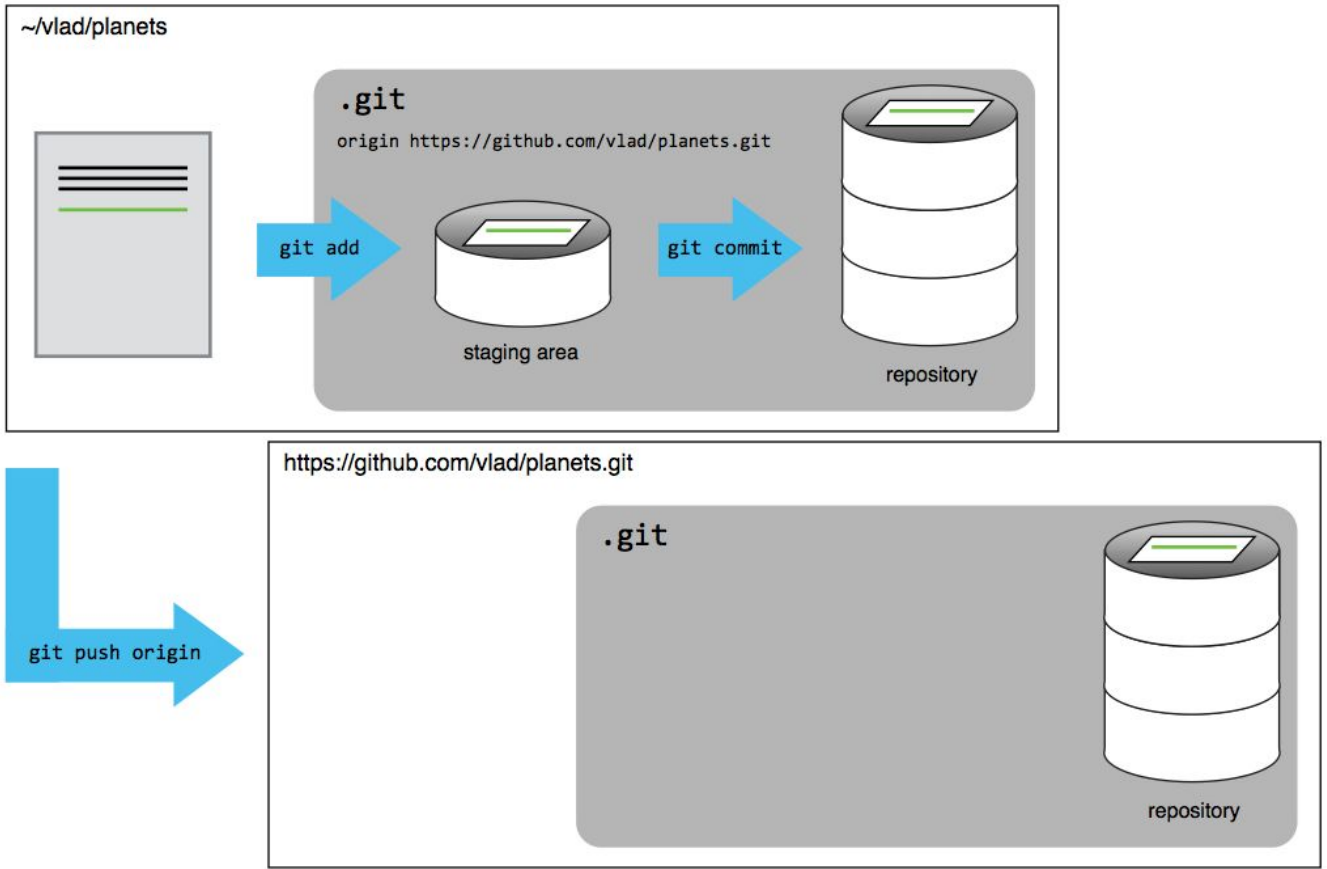
Exploring History



New Github Repository - Existing Project



Github Remote Repository Populated



Challenge: Github GUI

Browse to your git-basics repository on Github. Under the Code tab, find and click on the text that says “XX commits” (where XX is some number). Hover over, and click on, the three buttons to the right of each commit.

What information can you gather/explore from these buttons? Discuss with folks at your table.

Push vs. Commit

In this lesson, we introduced the ‘git push’ command.

How is ‘git push’ different from ‘git commit’?

Discussion: Git(hub) in Libraries

- Software Carpentry Courses!
- Software applications
- Development Team Guides
- Metadata?
- Other Projects?
- ….. ?

Challenge: Review changes

The **Owner** push commits to the repository without giving any information to the **Collaborator.**

How can the **Collaborator** find out what has changed with the command line? And on Github?

Challenge: Solving Conflicts that you Create

Switch roles for creating a conflict. If the Owner created the conflict last time, have the collaborator do so this time.

- Add a new file to the repo with content of your choice
- Both pull latest copy
- Switch role of who commits conflicting changes first
- Resolve merge conflict after git pull
- Push changes back to repository
- Have other teammate pull merged change and confirm

Conflicts on Non-textual files

What does Git do when there is a conflict in an image or some non-textual file that is stored in version control?

Resources

- http://swcarpentry.github.io/git-novice
- http://swcarpentry.github.io/git-novice/reference.html
- https://services.github.com/kit/downloads/github-git-cheat-sheet.pdf
- http://ndpsoftware.com/git-cheatsheet.html
- https://help.github.com/articles/dealing-with-line-endings/#platform-all
- https://help.github.com/articles/generating-an-ssh-key/ (SSH keys)
- https://www.youtube.com/watch?v=dBSHLb1B8sw (Deep Dive into Git)
- https://git-man-page-generator.lokaltog.net/ (Fun with Git Docs)
-