cdWelcome to The Carpentries Etherpad!

This pad is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents.

Use of this service is restricted to members of The Carpentries community; this is not for general purpose use (for that, try https://etherpad.wikimedia.org).

Users are expected to follow our code of conduct: https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

---

Please enter your name and campus affiliation:
How are you hoping to bring version control into your life?
Qing Huang, communication, tbh I don't know exactly how version control works
Dan Ralston, math, data science projects
Isaac Park, data science projects
Sigrid Van Den Abbeele, geography department, I hope to use it to make collaborative projects with large data sets more efficient
jonjab. Librarian. Workflow management and cross GFW document sharing
Kristi Liu, Library staff,
Tyree Byrd, political science/black studies, research projects
Kat Le, Bren School, supporting students
Kayla Kauffman, EEMB, disease ecology/large collaborative projects
Nicola Paul, Marine Science, organization and collaborating
Ilda Cardenas, Electronic Resources Librarian, CSUF
XIkun Liu Chemitry UCSB

Day One

git verb --parameters

Bash tip to shorten the prompt: PS1="$ "

git config --global

Making Changes

1) git add <filenames> # to stage files. you can use wildcard!
2) git commit -m "message in quotes" # commit works on all staged files

If you get a "aborting commit due to empty commit message" means you have to make sure you missed step 2 above.

Tip: In real-life projects, these messages should be meaningful, and represent your edits. Also, we don’t advise too many edits in a bulk cause it makes harder to track them, especially in projects your are

collaborating with multiple people.

You can also add multiple files together: git add -A

git reset --hard HEAD can help you to reset to a know state

A resource that is worth bookmarking: https://github.blog/2015-06-08-how-to-undo-almost-anything-with-git/

Day Two
.ignore:
an invisible file in the root of your repo directory.
add it and commit it to make it active.

use touch to create empty files

When ignoring things

Git won't constantly say that you have untracked files in your repo unless you use flag those using:
`git status -u`