

# Software Carpentry Workshop

## Berkeley Institute for Data Science

June 11, 2018

**Instructors:** Diya Das, Rebecca Barter, Richard Barnes

**Helpers:** Jenna Baughman, Caroline Cypranowska

## Version Control with Git

**LINK TO INSTRUCTOR GUIDE:** <http://swcarpentry.github.io/git-novice/>

### Monday PM: Git

Mac Installation instructions: <https://sourceforge.net/projects/git-osx-installer/files/>

Windows installation instructions: already installed with bash

### **What is Git and Github?**

For version control.

Keeping track of all changes you make to your code.

Allows collaboration, and merging of changes by different collaborators.

Git repository is backup of project and history.

GitHub is the web-based collaboration service.

configure git (this is global so only needs to be done this one time)

**\$ git config --global core.autocrlf input on mac** ("true" instead of "input" for windows)

**\$ git config --global core.editor "nano -w"**

**git init** <-- will turn a directory into a git repository

- all subdirectories are included in the repository - should not git init for each one for simplicity

**ls -a** <-- see hidden files in a directory, can use to see that your directory was turned into a git repository (.git should appear)

**git status** <-- tells you some details about the the status of your commits and where you are on the branches

**git log** <-- log of all of your commits

**git log --oneline** <-- all of your commits in one line each

Don't put Git repositories in your subdirectories. Only the parent file should have a .git.

**rm -rf .git**

- To force remove git repository within a folder

Two-step process to track the changes: git add and git commit

**git add <file name>**

- telling repository to keep track of this file (places files in the "staging area") - have to git add after every change
- 

**git status** to see that your file is being tracked (and its status)

**git commit -m "Start notes on Mars as a base"**

- **-m** indicates "message"
- **--**these go into the **git log**
- **--** only need to do git commit once for everything currently added to the staging area
- 

**git diff**

- what has changed in a file from your previous commit
- this won't display anything if you've already added the file to your staging area unless you specify:
  - **git diff --staged** <--changes in files in your staging area

**git diff HEAD** <-- most recent commit

**git diff HEAD~1** <-- edit before that

**git diff HEAD~2** <-- etc

**git diff <commit number>** <-- to look at one specific past version's differences

e.g. two stages of adding then committing changes with message at commit:

**git add mars.txt**

**git commit -m "Added second line of text."**

**git add .**

- - adds all changes

•

### **git add mars.txt venus.txt**

- - two arguments to git add (will add both mars.txt and venus.txt files)

•

### **git checkout HEAD mars.txt**

- reverts to the previously added version?

•

'detached HEAD state' creates an alternate working version branch e.g. for if you want to test out some code (an alternate reality)

**git checkout master** takes you back to your main branch

**touch <filename>** makes an empty file

### **nano .gitignore**

- - makes a file to list the files we want git to ignore (e.g. because they're too big).
- - git knows what to do with this special file

•

### **Get yourself an educational**

**account!** [https://education.github.com/discount\\_requests/new](https://education.github.com/discount_requests/new)

- (Select individual account, hit next)
- Since Berkeley lets you keep your email address forever and our IDs don't expire, you'll need to select "Upload proof..." an upload an enrollment verification OR a copy of your employment letter on Berkeley letterhead (as an image file)
- They usually grant it within a few days

GitHub > sign in > new repository > "...or push an existing repository from the command line" instructions copy into terminal

- several options
- ...or push an existing repository from the command line
- git remote add origin <https://github.com/diyadas/test.git> (or whatever the remote URL is)
- git push -u origin master # -u = set upstream // your current branch is now tracking master branch of origin

•

- git help push

•

### **git clone <https://github.com/partnername/respository.git> partner**

- creates directory named "partner" and clones partner's repository onto your device
- can now make edits, add, and commit to their version of the repository, but it doesn't edit their repository until they give you access!

•

### **git push**

- publishes your *local* commits to the *shared* repository
- 

### **git blame <file name>**

- log of edits made by all users
-