

How to use version control with Github: Command line routine reminders using the terminal application on Mac OS

By: Jordan Wylie

This is a helper document for a few things that I do often in Github. It assumes that you have already created a repository on Github ([this link](#) explains how to do that) and have the basics set up on your local device (more on that [here](#) and a great tutorial on git and Github [here](#)). These are written in the first person.

I've made a change on my local drive and want to connect it (or "push" it in git language) to my remote repository for my collaborators to see/use. I should:

1. Open the terminal app
2. Type the command `git status` to see where you are in your folder system
3. Change your working directory to the correct folder (if needed) using `cd filepath` working
4. Type `git status` to make sure it worked
5. Type `git remote -v` to make sure you are linked to the correct remote repository. If you are not, use `git remote set-url origin https://github.com/username/location` to connect to the correct remote repository. Then use `git remote -v` to test the change
6. Type `git add filename` for whatever file/files you want to push
7. Type `git commit -m "short description of change"` to commit the change so that you can push it
8. Type `git push origin main`
9. Now if you use `git status` again, you should see there are no changes waiting to be pushed in the file/files you had changed.

My collaborators made changes and now I want to update my local copy (or "pull" them) so that I can make changes and then push those. I should:

1. Repeat steps 1 through 5 from above
2. Type `git fetch origin main`
3. Type `git pull origin main`
4. Now if you use `git status`, you should see there are no changes waiting, and you can look in your local file to see the added files

I've made a mistake in creating something, or I pulled data that was test data and now I want to delete it. DO NOT delete manually. This is a nightmare. Instead, I should:

1. Use `git status` to confirm what folder you are currently working from. Use `cd` to change if needed.
2. Type `git rm foldername/filename` (or just filename if you are already in the correct folder. If you have the same root name for the files, but different dates or subject numbers, you can use `*` to select all of them. So, for example, you may have files named "experiment1_participants_112", "experiment1_participants_113", "experiment1_participants_114", etc. Using `rm experiment1_participants_*` will select all of the files)
3. Type `git commit -a` (commit the changes; if you changed your `cd` to the location of the files to delete, you will need to navigate back to the main folder to commit the changes. Use `pwd` to see where you are.)
4. Type `git push origin main` (push the changes to remote)