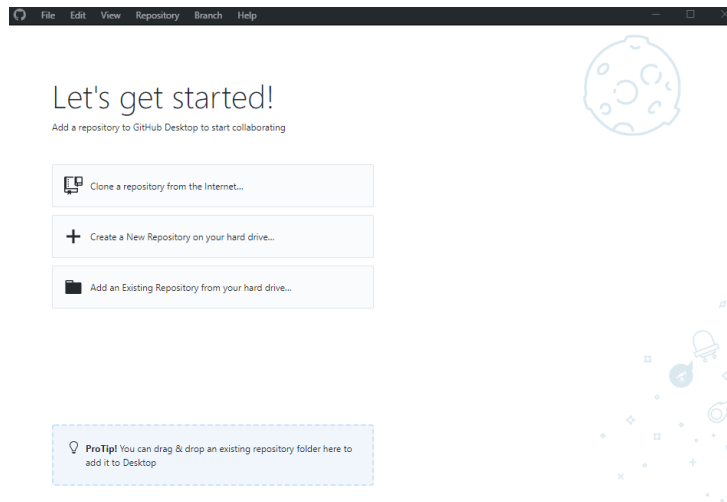


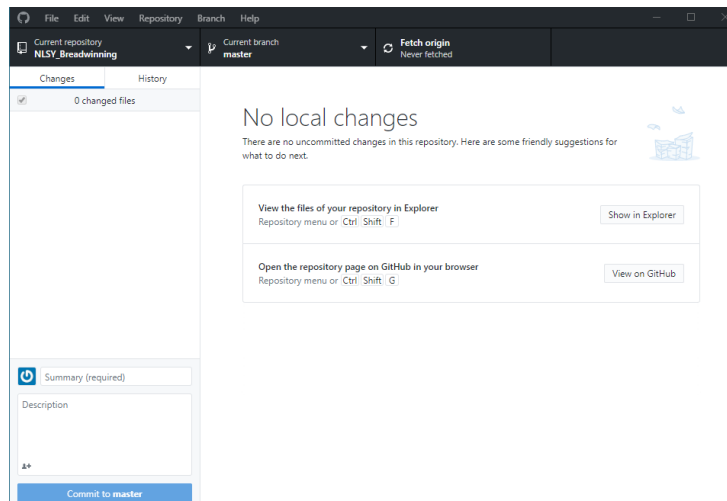
GitHub Desktop

It is possible to manage creating, cloning and syncing your repositories using line commands, but a program like GitHub Desktop can make it easier to see specific differences between versions of files (see 3.e below) and some aren't comfortable issuing line commands. A specific nice aspect of GitHub Desktop is that you can see your files on multiple repositories including GitHub and github.austin.utexas.edu all together.

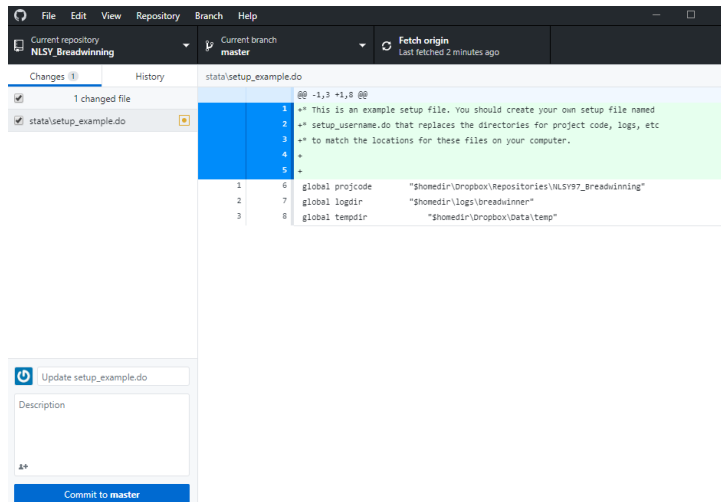
1. Install GitHub Desktop on your computer by going to desktop.github.com, downloading and installing.
2. When you first launch GitHub Desktop it will look like:



1. If there is a repository in the cloud that you would like to sync to your computer, select "**Clone a repository from the Internet...**"
 - a. If the repository is on [GitHub.com](https://github.com), then select [GitHub.com](https://github.com) and sign in. You'll need to have a (free) GitHub account. (See 5 below if you are pulling from an enterprise service like github.austin.utexas.edu)
 - b. After you log in you will have a list of your repositories that you can select to download. Select a repository and then click the "clone button at the bottom"
 - c. The repository URL will be filled in with the correct information, assuming you selected the right repository to download, but you might want to change the local path to which the repository will be downloaded.
 - d. Once you've downloaded the repository, you'll see a screen that looks like:



- e. If I edit one of the files in the repository the screen changes to:



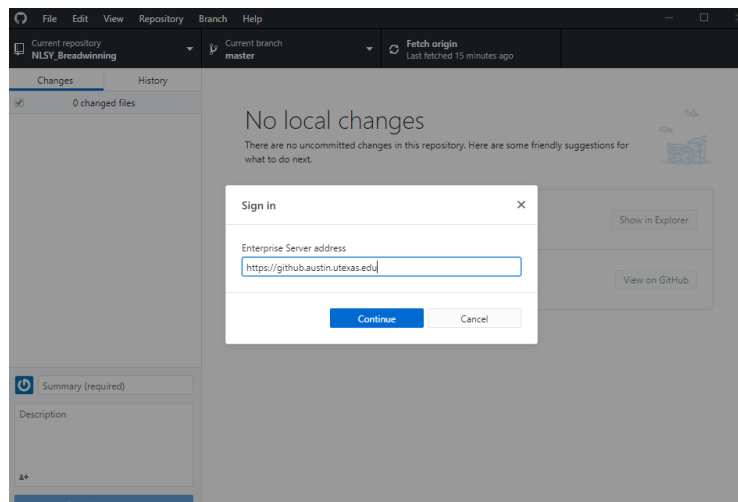
You can see above that I have changed 1 file. It's called `stata\setup_example`. The yellow square indicates that it is a changed file, as opposed to a new one. This means that there is a copy of the file in the cloud repository that does not match the version on my computer. In the box on the right, you can see the difference between the file on my computer and the one in the repository. Green indicates new text and red (not present) represents deleted text.

f. To push these changes to the repository, you should add a brief comment about what changes you made in the small window below the list of changed files. In this case, I might say, "added descriptive comment to top of script." Then I click "commit to master." This commits the new version of file to your local repository. You need to then "push origin" to put those changes in the cloud repository.

g. If you decide that you don't want to commit the changes to the repository, you can right click on the file and select discard changes. Confirm that you want to discard changes in the pop-up window. The file will change back to the original version of the file on disk. Note that this is often my solution when I find that I made changes locally without first pulling the changes to the files that my collaborators made. It's OK as long as I remember what changes I made, but it would probably be better for me to learn how to stash my version of the file, pull the repository changes, and then merge in my stashed file before pushing the result back up to the cloud. That's beyond the scope of this tutorial.

4. If you have files on your computer that you would like to turn into a repository and upload to the cloud, select **Create a New Repository** on your hard drive. A window will pop up. Give the repository a descriptive name, add a description, and identify the location of the files you want to deposit. You might want to add a License and initialize with a README. Then click create repository. This will create a bunch of hidden files in your directory that GitHub Desktop uses to manage your repositories. For example, there's a `.gitignore` file with a list of files you want GitHub not to sync to the repository or even list among the changed files. An example of a file you might want to ignore could be your personal setup file, described elsewhere in our discussion of workflow.

5. GitHub desktop is written for using [GitHub.com](https://github.com), but you can also use it to manage repositories hosted elsewhere, like UT-Austin. To clone a repository from github.austin.utexas.edu, select "GitHub Enterprise Server"



Enter <https://github.austin.utexas.edu> for the server address. It will note that you need to sign in with your browser. "Continue with browser." At this point I received an error message, but if I selected clone repository from the "file" dropdown menus and "GitHub Enterprise Server" the list of repositories I have on github.austin.utexas.edu were available. I don't know what will happen for you. It is different every time I do this. YMMV, as they say.