Reference Docs

Git and GitHub Desktop Intro

Updated: February 6, 2020. Version: GH Desktop 2.3.1

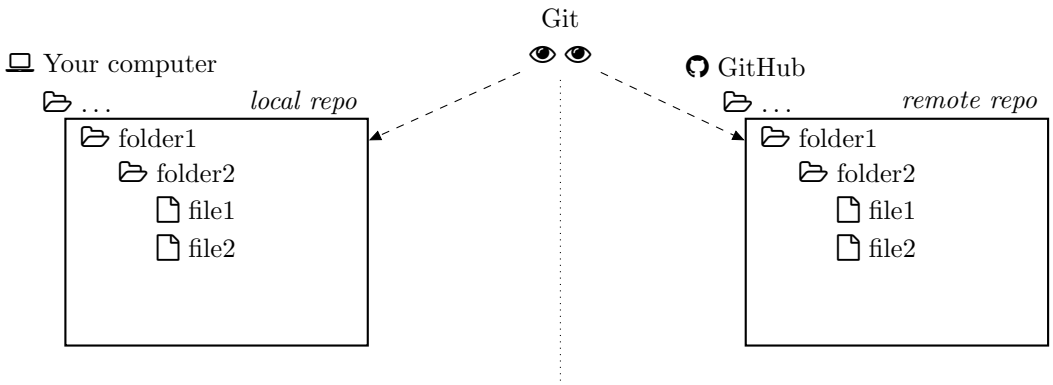
We will use Git, GitHub, and GitHub Desktop to manage files and keep them synchronized between you, your partner, and the instructor(s).

Git: Big Idea

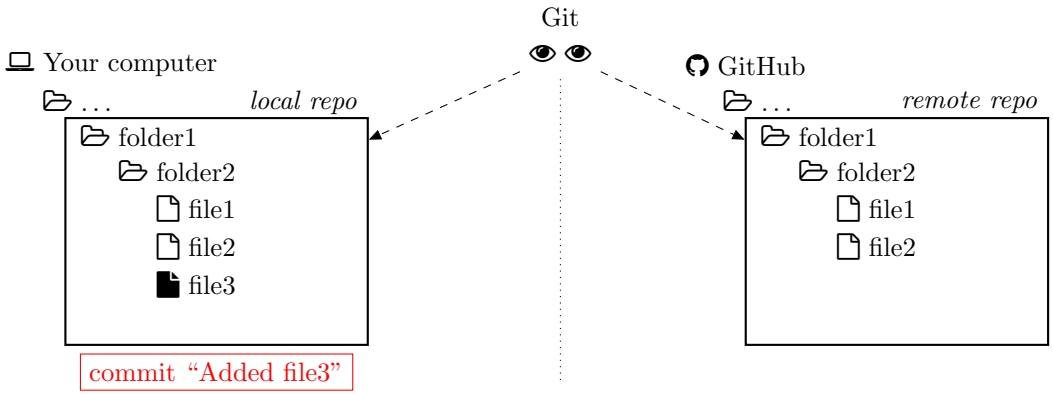
Git is a version control software. Think of it as a "wrapper" around the files in a certain folder on your computer. It watches these files and tracks all of the changes that you make. The diagram below shows how this might look.

GitHub is a cloud storage service, just like Box, Dropbox, Google Drive, Microsoft OneDrive, etc. The only difference is that GitHub allows you to track your files using Git (i.e. it supports the Git software). It also provides some additional features through their web interface (e.g. the ability to create "Issues" and collaborate to resolve them).

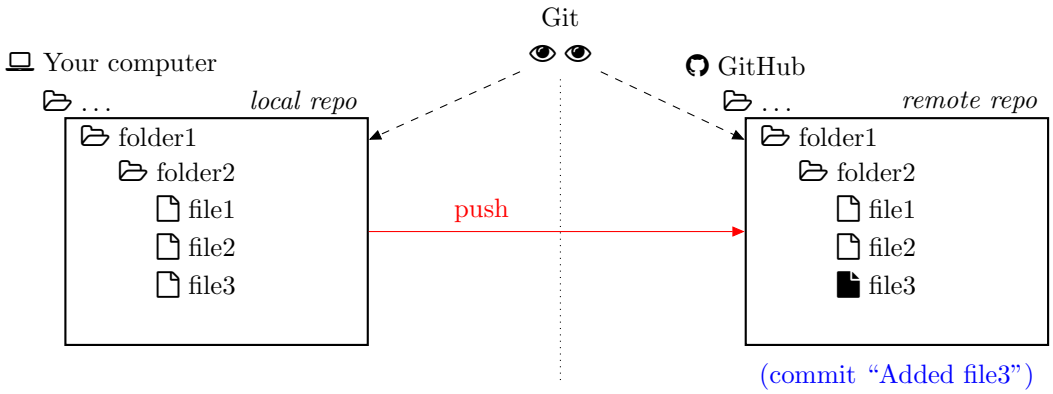
Git refers to a tracked folder as a *repository*, or **repo** for short. The repo on your computer is called your *local* repo, and the duplicate repo on the cloud server is called the *remote* repo.



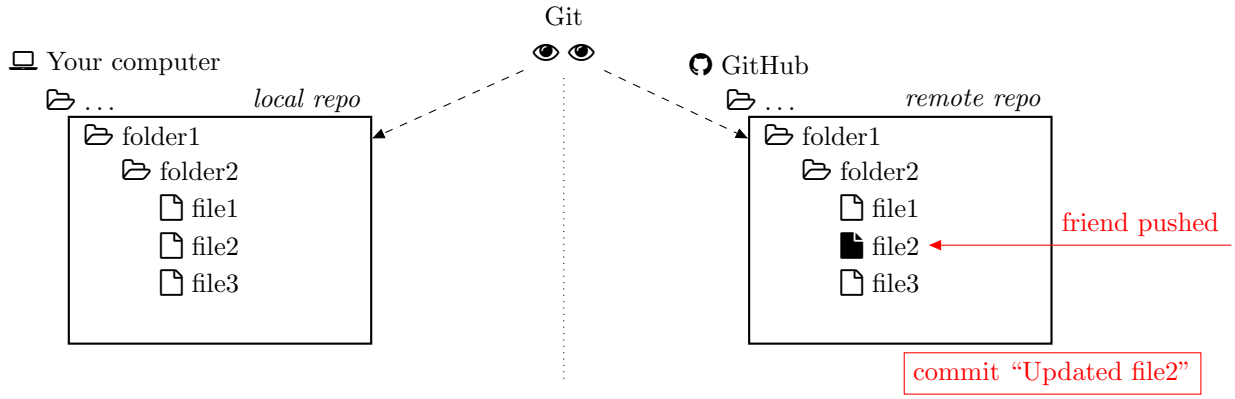
Suppose you create a new file, "file3". You would then **commit** that file to your local repo. Think of this like "committing" to the change you made. You're just telling Git, "Yes, I want to make this change." And you need to provide a message for you and your collaborators to know what change was made.



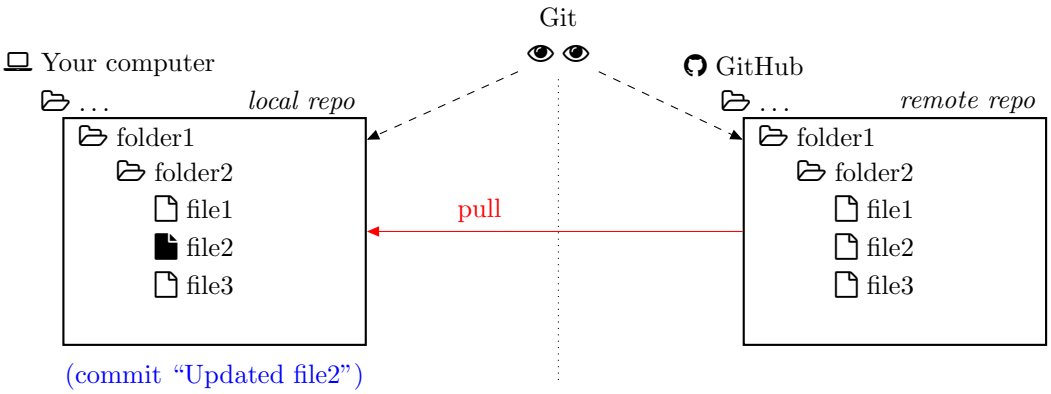
You then **push** your changes (your commit) to the remote repo on the cloud server (GitHub). The cloud server now has not only your file but also your commit message and all of the history of what has happened in your repo.



Now suppose your friend who is working on the same project pushes a change she made to file2.



You need to get this change, so you **pull** from the remote repo. You now have the commit she made, and your local repo is up-to-date.



That's the basic process you will follow as you make changes to your files.

GitHub

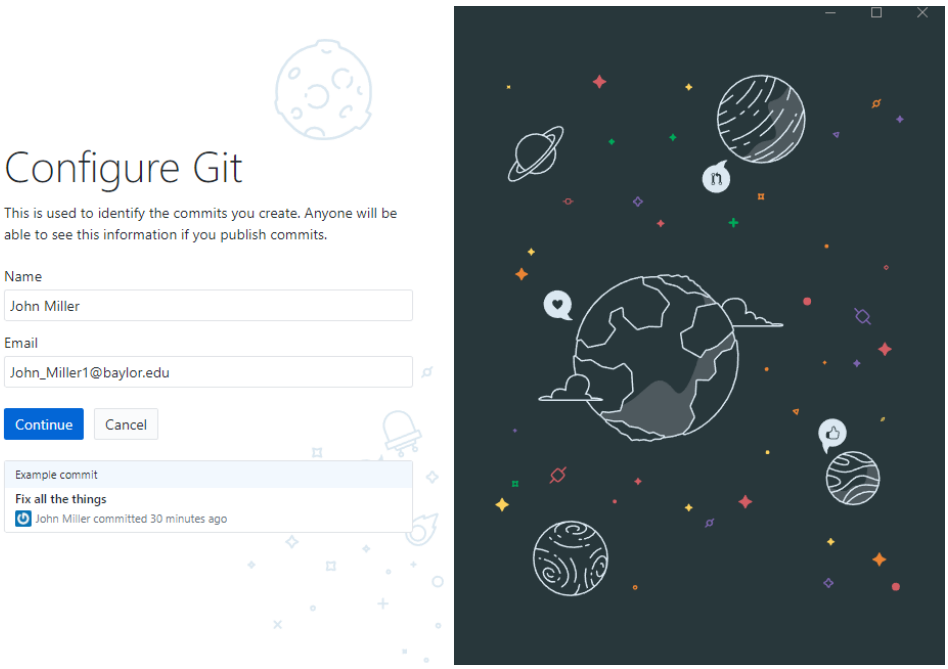
Your first step is to create a GitHub account. Go to https://www.github.com/ and sign up for a free account. Use your Baylor email, since you will be using this service for course work. But you are free to choose any user name that you like.

GitHub Desktop

Git is an open-source software, so there are multiple "clients" that can run it. Since we are using GitHub, it makes sense to use their client, GitHub Desktop (GHD), and it is fairly easy to use, which is nice. Start by downloading and installing it: https://desktop.github.com/

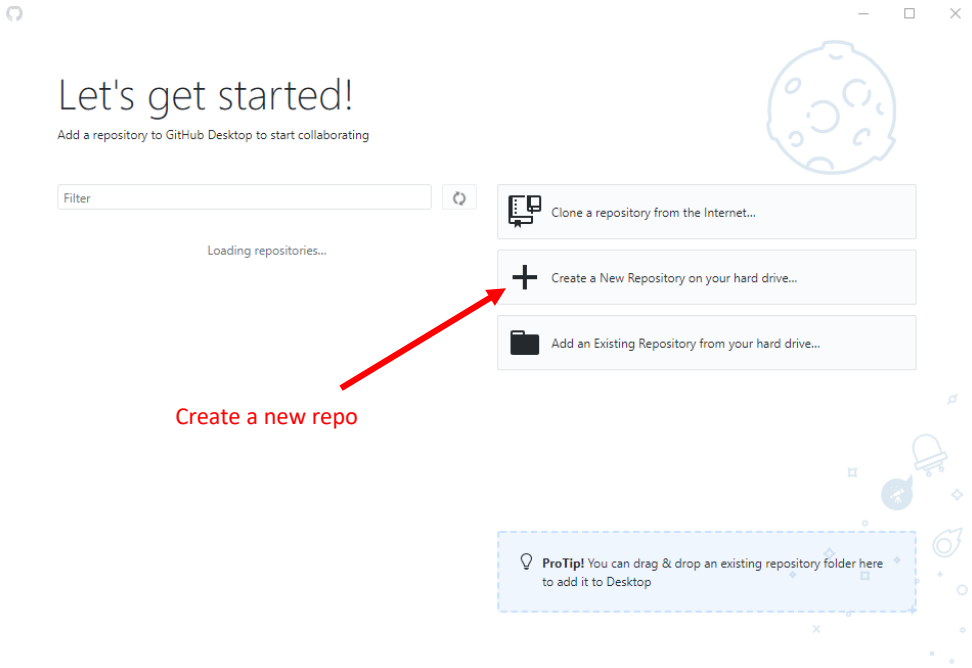
Setup

After you sign in with your GitHub account, you will see this Configure Git page:

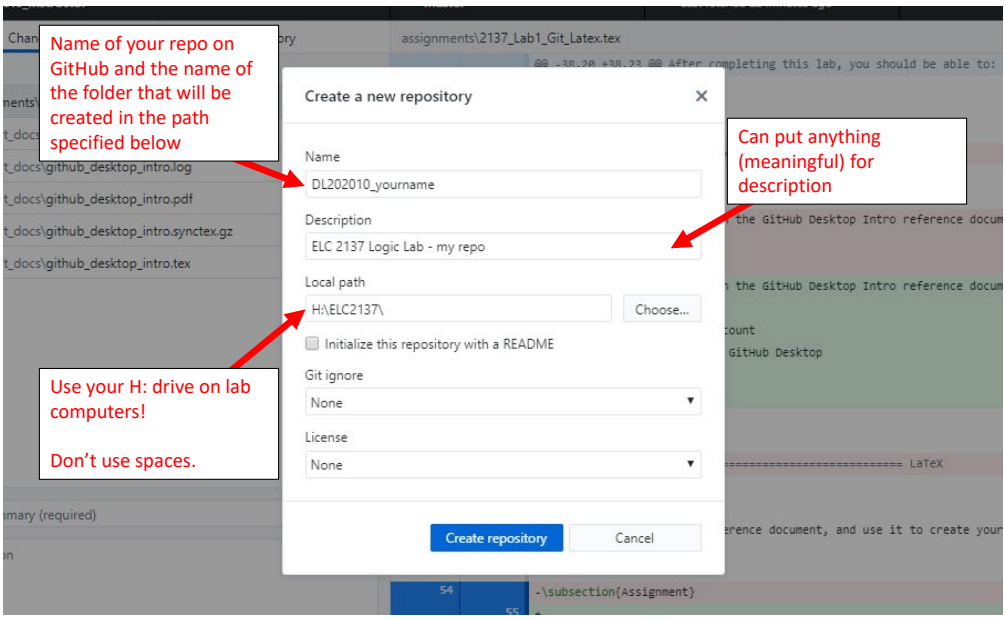


The name you type here will appear next to your commits (see the example they give you at the bottom of the dialog). It does not have to be your GitHub user name.

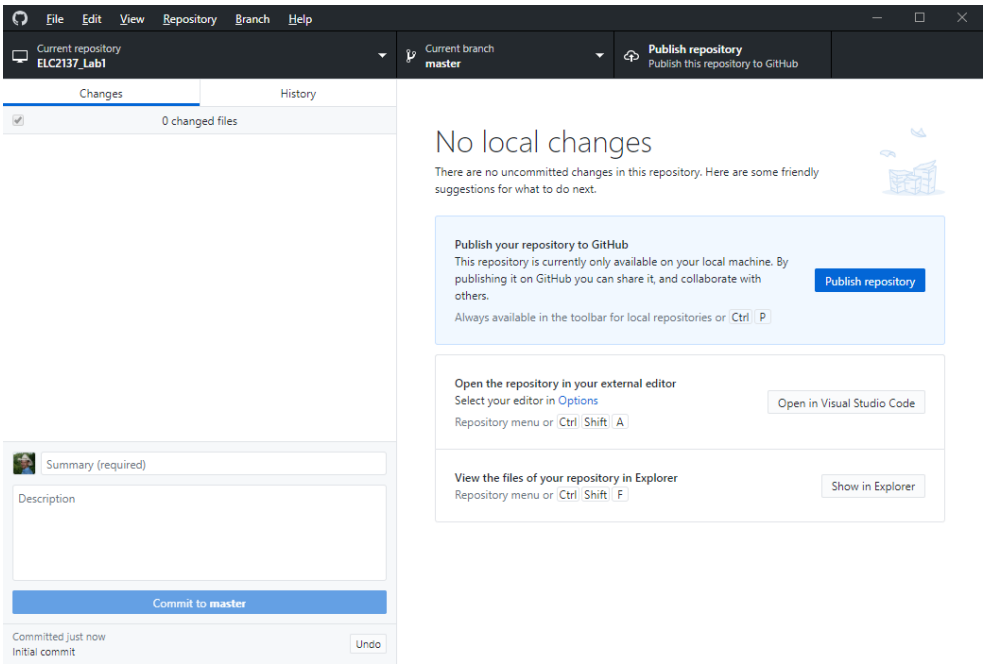
Next, you need to create a new repo on your computer.



Give your repo a name. This will also be the name of the main folder for your files. Put this repo in a folder on your H: drive with **no spaces in the name**.



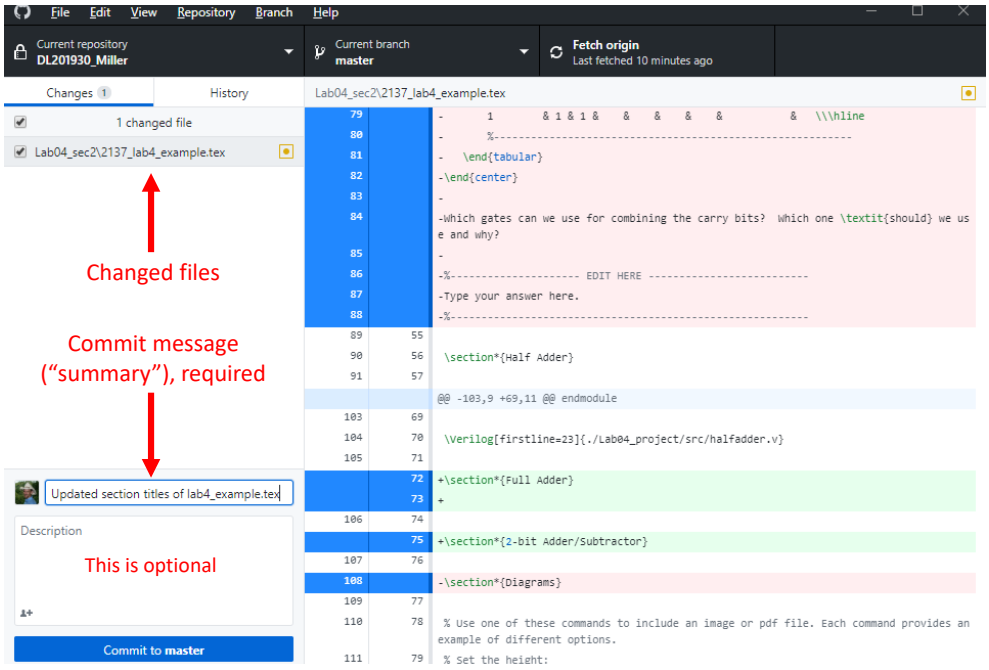
GHD will now look like this, and you are ready to get to work!



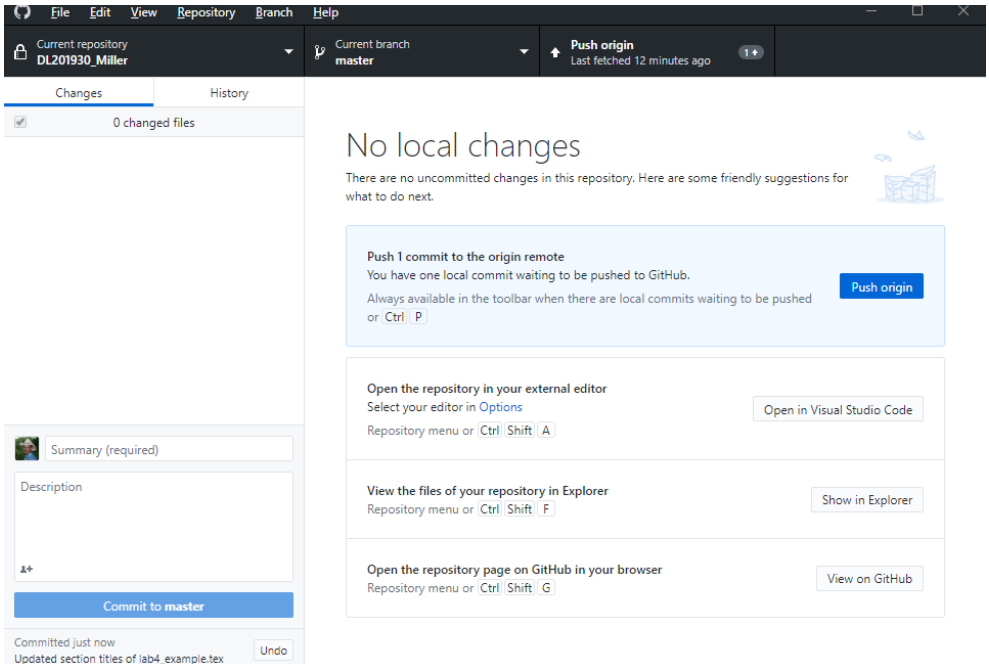
Workflow

You can work with the files in your local repo (folder) like any other files on your computer. Create new files, move them around, open, modify, save, etc. After you make some changes to your files, you will see the modified or added files in the left sidebar. If these are text-based files, then you can click on them to see a line-by-line comparison of the changes or differences. This is called the **diff** view. It shows deletions in red and additions in green (see below). If the file is not text-based, you will simply see a message that says "This binary file has changed."

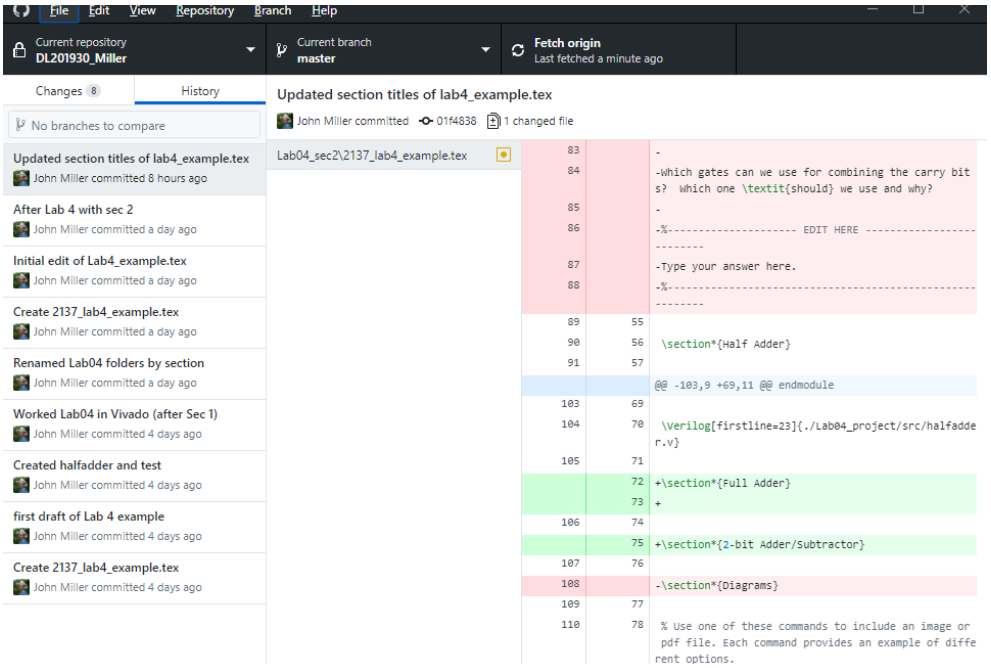
If you are ready to make a commit, you **must** type a message into the Summary field. This message should be both as short and as descriptive as possible. Yes, it's hard to do both, but that's your goal. The Description field is optional and allows you to provide more detail.



After clicking commit, you should see a message appear in the lower left saying that you made a commit and the option to *push* those changes to the remote repo. Note: If your repo is empty, this will say "Publish" the first time and then change to "Push" after that.



If you want to see all of your commits, click the History tab.



Notes

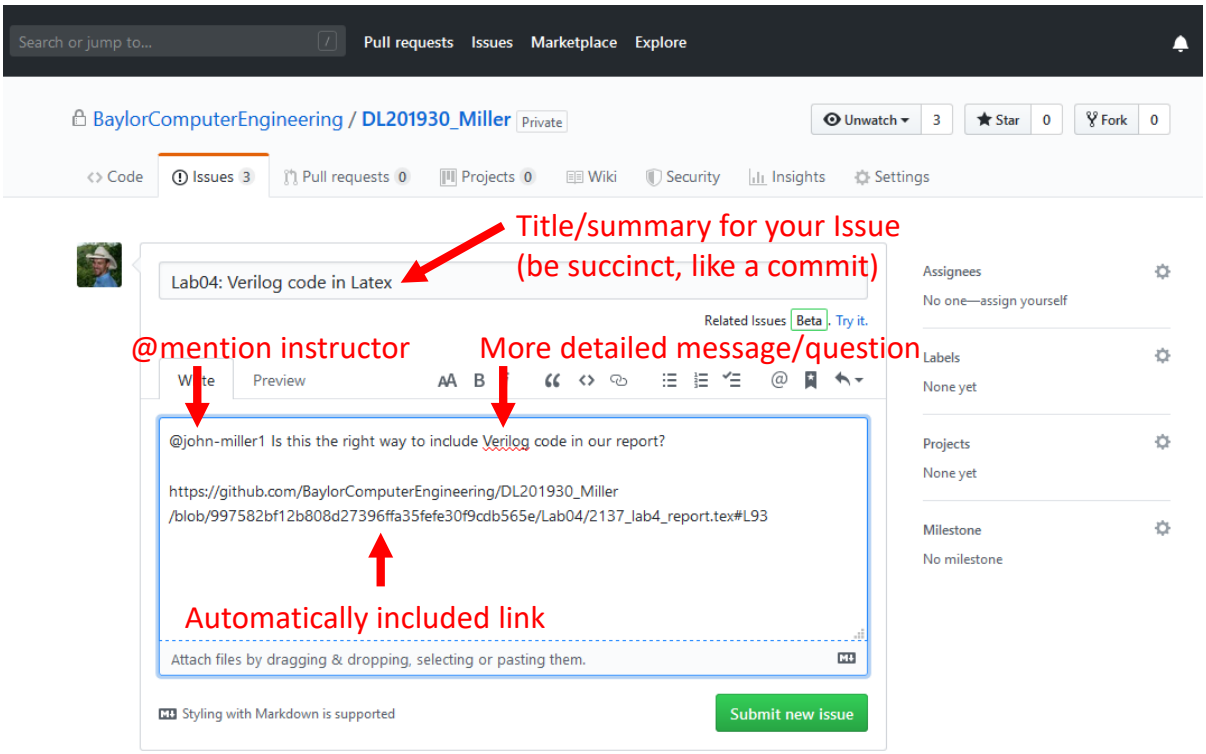
- Assuming you're working on your H: drive or personal computer, your files are saved, even if you don't commit/push them. Remember, Git is just a wrapper around the files. When you click save on your document/code file (in Vivado, TeXstudio, or other editing program), it is saved to your local disk. GHD simply makes a note of that and puts that into the list of changed files for you to commit and push later.
  - So still **save** often!
  - Commit and push less often, usually at natural breakpoints in your work, e.g. after you finish with a particular file or accomplish a certain task. This makes it easier to write a short, clear commit message as well.
  - Reminder: ECS computers are set up to *erase* files not stored on your H: drive. If you save files on the local computer (e.g. in Documents), be sure you sync them to your GH repo or copy them to your H: drive before logging off.

Issues on GitHub

These are good, not bad! They are a way for you to communicate with your instructor(s). The real advantage is that you can point them directly to the line in your code where you are having trouble. This article explains how to do that:

https://help.github.com/en/articles/opening-an-issue-from-code

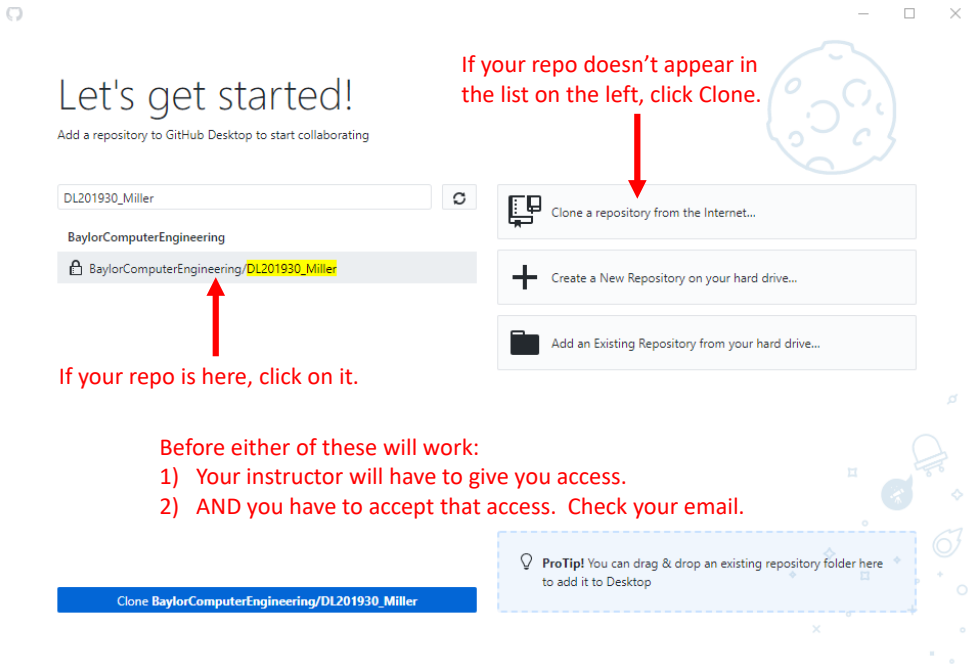
When you create an Issue following these instructions, GitHub will automatically include a link directly to the code you selected. **Don't delete that link**. Add an "@mention" of your instructor(s), so that they will get an email, and your comment/question. Below is an example.



Repo created by instructor

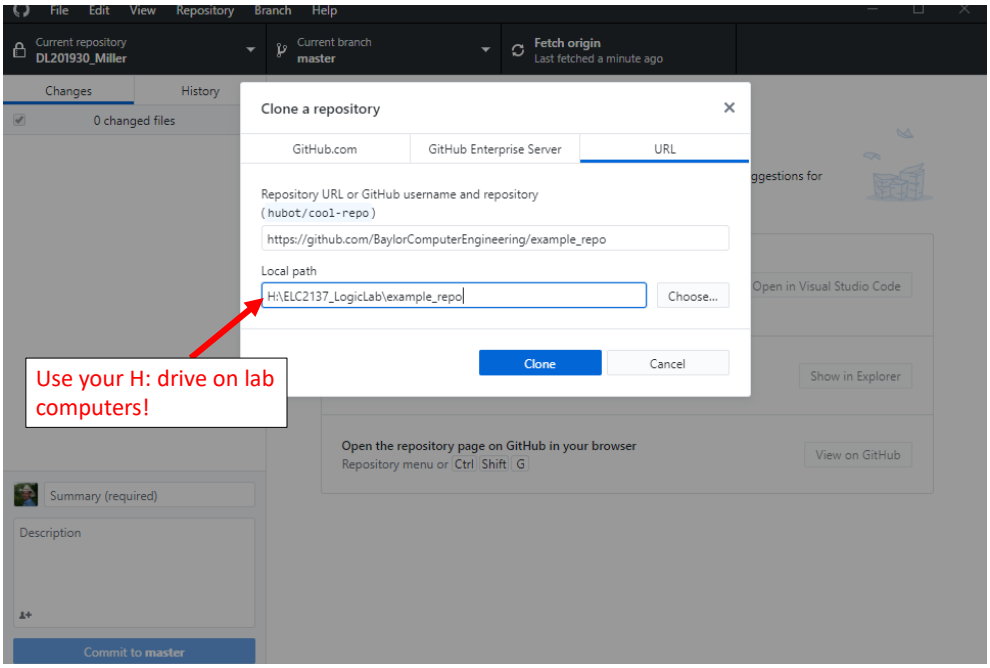
If your instructor provides you with a repo, **you will need to accept the invitation that will be sent to your email**.

Next, you need to **clone** the repo from GitHub to your computer. If it appears in the list, then you can simply click on it there (see below). If not, you may need to click the Clone button and select your repo.

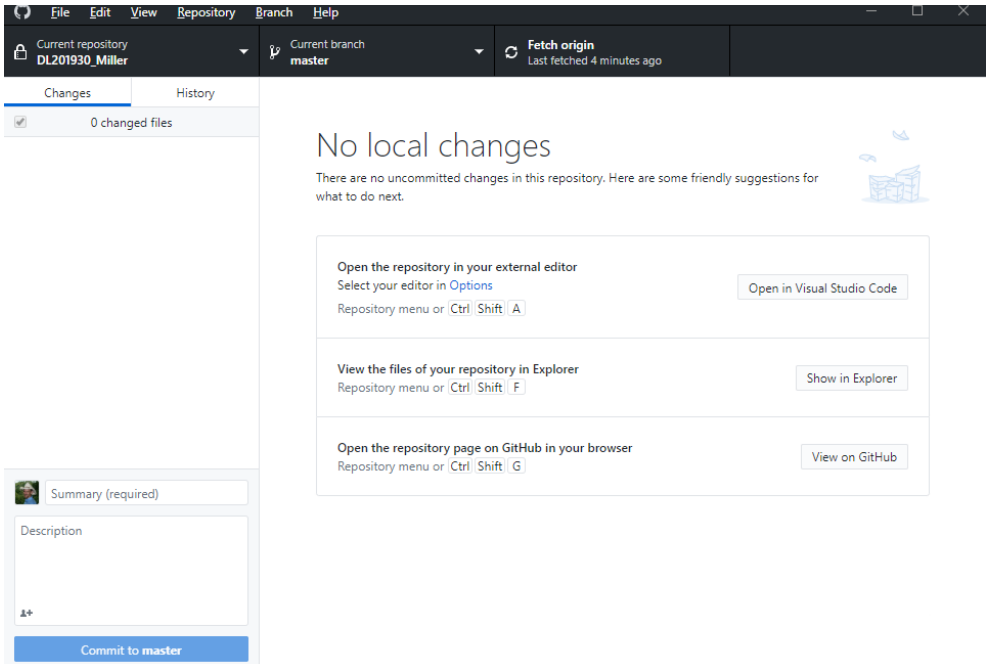


Important 1: You have to first *accept* the repo if it was created by someone else (instructor or teammate).

Important 2: If working on lab computers, save your local repo on your **H: drive**. You specify this in the dialog box when you clone the repo (below).



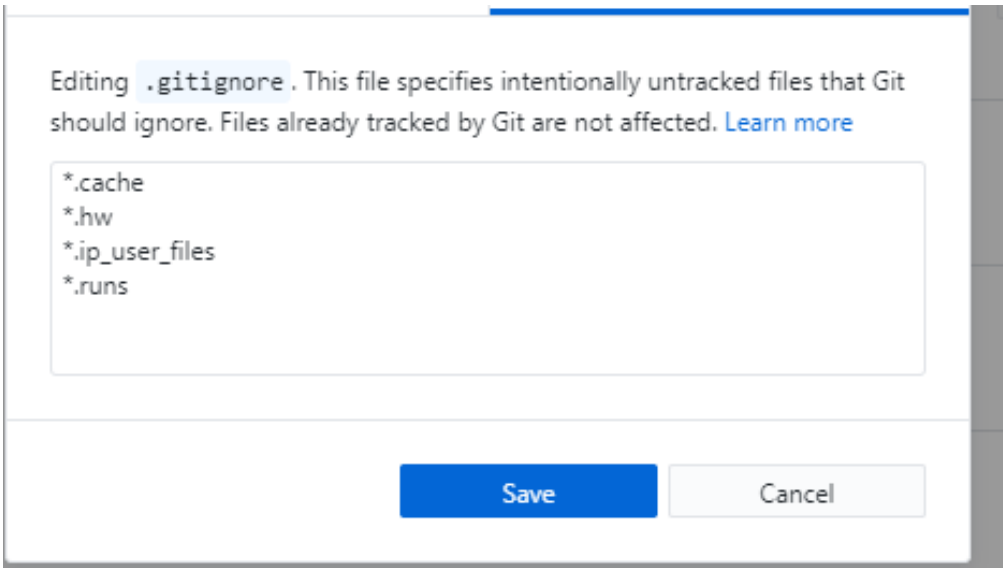
Once you connect GHD to your repo, it will download the files (if there are any) to your computer. You will then see a screen like below, and you are ready to get to work!



Git ignore

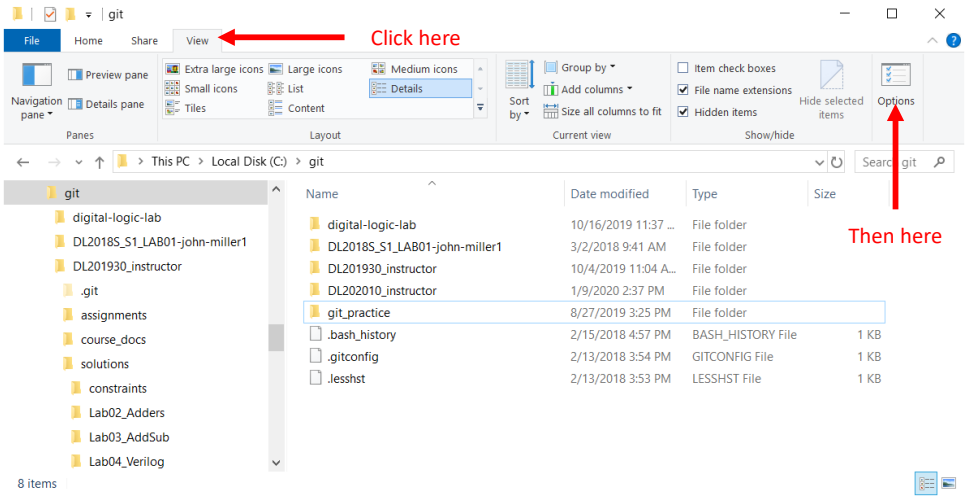
There are times when you do not want to sync all of the files in a repo. This increases the speed of transfer and makes your remote repo (on GitHub) smaller. We will make use of this to ignore files auto-generated by Vivado. You should change these settings before you have these files in your repo (otherwise some of them will get transferred and then won't be deleted because they are ignored).

In GHD, click the Repository menu, then Repository Settings. The dialog below will appear. Click the Ignored Files tab, and then type the list shown below. Click Save.

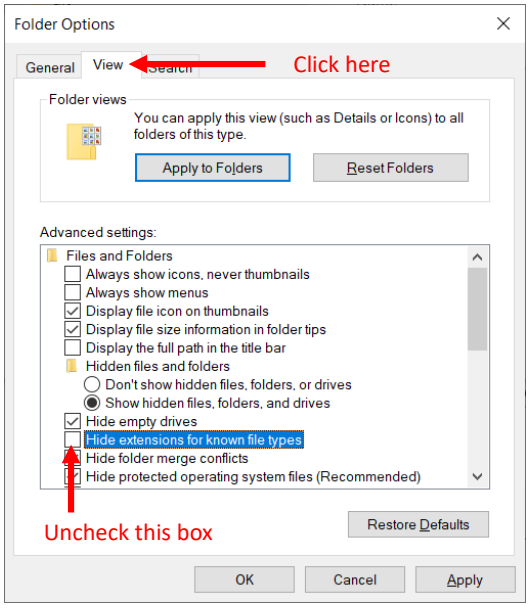


File extensions in Windows

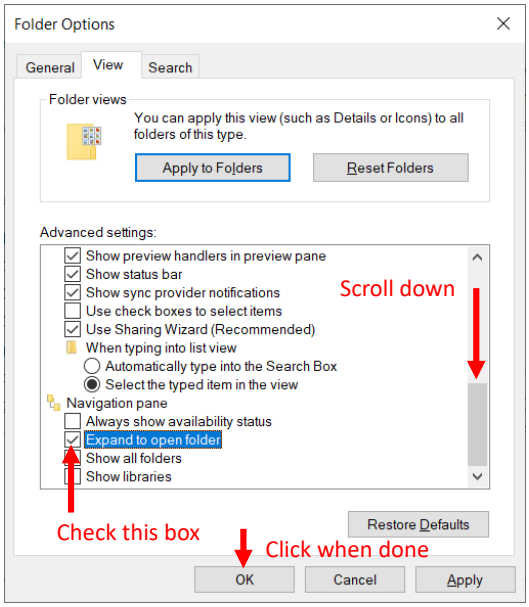
We will work with lots of different file types in this class. By default, Windows hides the file extensions that tell you what type of file each one is. This can be changed in the folder options from a Windows File Explorer window. Open File Explorer. Go to the View tab and then click Options.



Click the View tab again, and uncheck the "Hide extensions for known file types" item.



Scroll down and check the "Expand to open folder" item. This makes File Explorer open folders in the left-hand navigation pane as you move through them, which helps you keep track of the folder in which you are currently working.


Folder Options
General
View
Search
Folder views
You can apply this view (such as Details or Icons) to all folders of this type.
Apply to Folders
Reset Folders
Advanced settings:
Show preview handlers in preview pane
Show status bar
Show sync provider notifications
Use check boxes to select items
Use Sharing Wizard (Recommended)
When typing into list view
Automatically type into the Search Box
Select the typed item in the view
Navigation pane
Always show availability status
Expand to open folder
Show all folders
Show libraries
Scroll down
Check this box
Restore Defaults
Click when done
OK
Cancel
Apply