Syncing to the main course repository

Over the course of the course, I might make some additions or changes to the central course repository on GitHub. It would be useful for you to be able to keep track of those changes. It's actually pretty easy, but we just need to get a bit of terminology right, first.

A **Repository** is a collection of files and folders; it's like a project. Because we are livign in the era of the cloud, repositories can exist as a **local repository** on your machine, or a **remote repository** on GitHub.

Forking a repository gives you a copy of that repository that you can do with as you please.

We've already been through all this.

One more term I should introduce here is a **branch**. A branch is basically exactly like using the "Save As" function to save a Microsoft Word document. So you could have your master (or first) branch called "my paper", then in order to work on it a little bit you could save your document as "my paper 2". GitHub has a nice little bit of functionality that lets you very easily merge any changes in "my paper 2" back into "my paper".

Now, even though your forked repository and the central course repository are separate, GitHub is still very usefully tracking changes for you on the central course repository. That means that as I make changes to the central course repository (e.g. for example, adding some of the lecture notes, or that sample data set), here are the steps you can take to get them into your own version of your repository yourself. .

1. In GitHub Desktop, be sure you are signed in to GitHub, by clicking on "GitHub Desktop > Preferences"
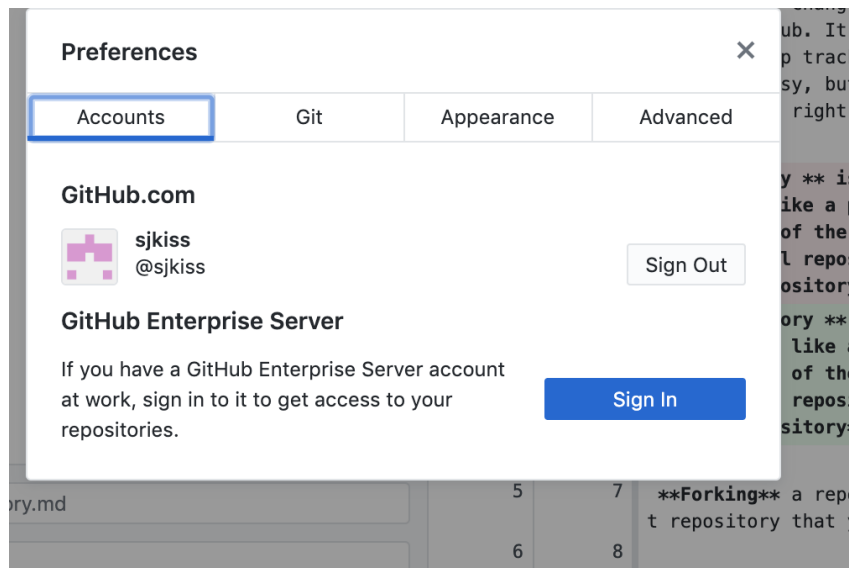
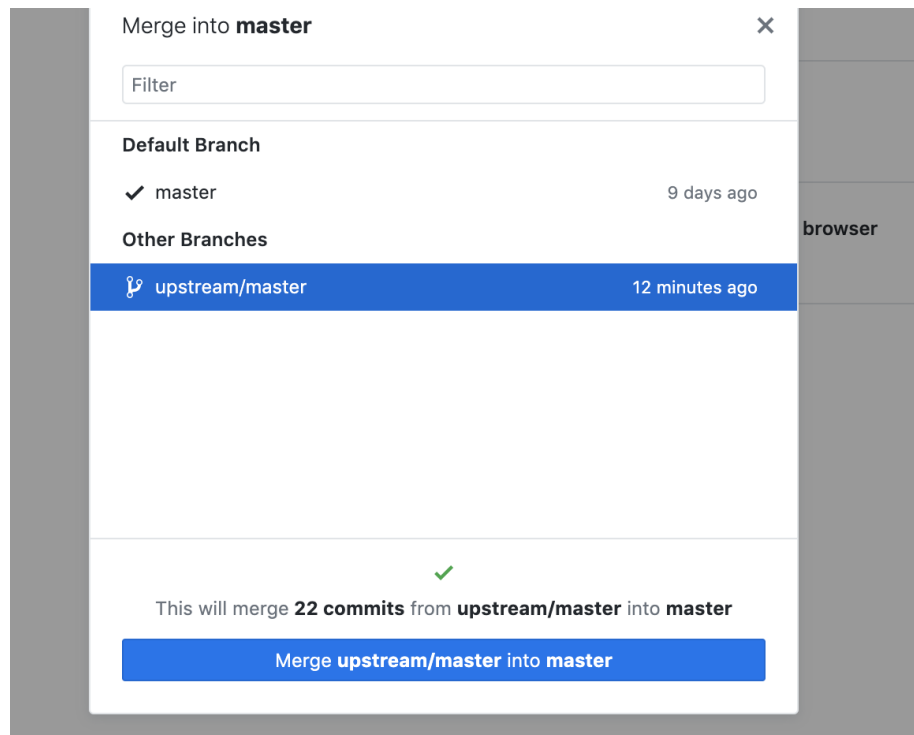2. Click on the "Current Branch - Master tab" in the top-center of your screen.
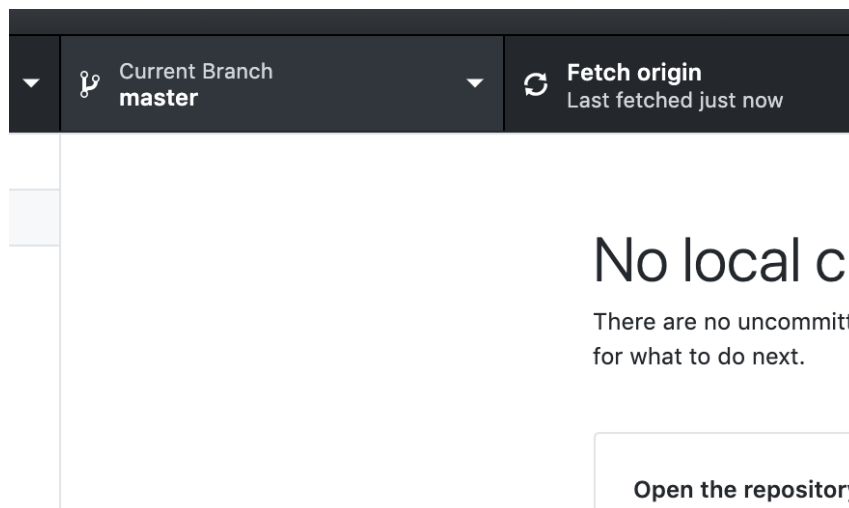
3. You will see two different "branches" a "Default Branch" and a branch called "upstream/master". The **default branch** is just your copy of your repository on your machine. But the branch **upstream/master** is the master branch on the repository that you forked your copy from.
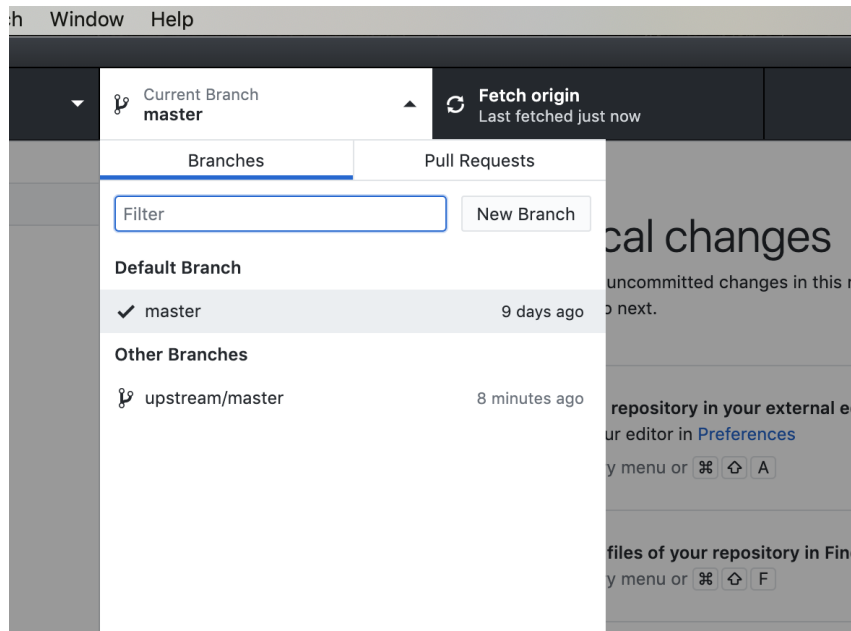
4. Click on "Choose a branch to merge into master" **at the bottom of your screen**.
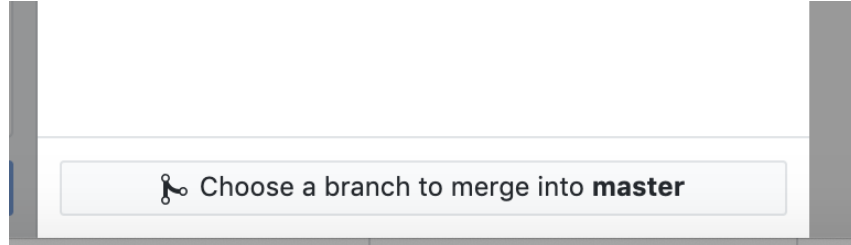
5. Then, select **upstream/master** and then click **Merge upstream/master into master** This will **pull** any changes I have made to the central repository (called **upstream/master**) down to your local, forked repository on your computer (just called **master**). Please note: You may get an error message here about some conflicts between files. AT this point, the best thing to do is to take note of the file names **and delete them from your course repository on your computer**. Don't worry about it.

Merge into **master**

Filter

Default Branch

✔ master   9 days ago

Other Branches

⑂ upstream/master   12 minutes ago

✓

This will merge **22 commits** from **upstream/master** into **master**

Merge **upstream/master** into **master**

6. **There is one last thing to do.** Any changes I have made are now merged into your on your **local, forked repository,** but they are not on **your** remote repository in the cloud. To finish that, you should **commit** your changes with a message and then **push** them up to the cloud.
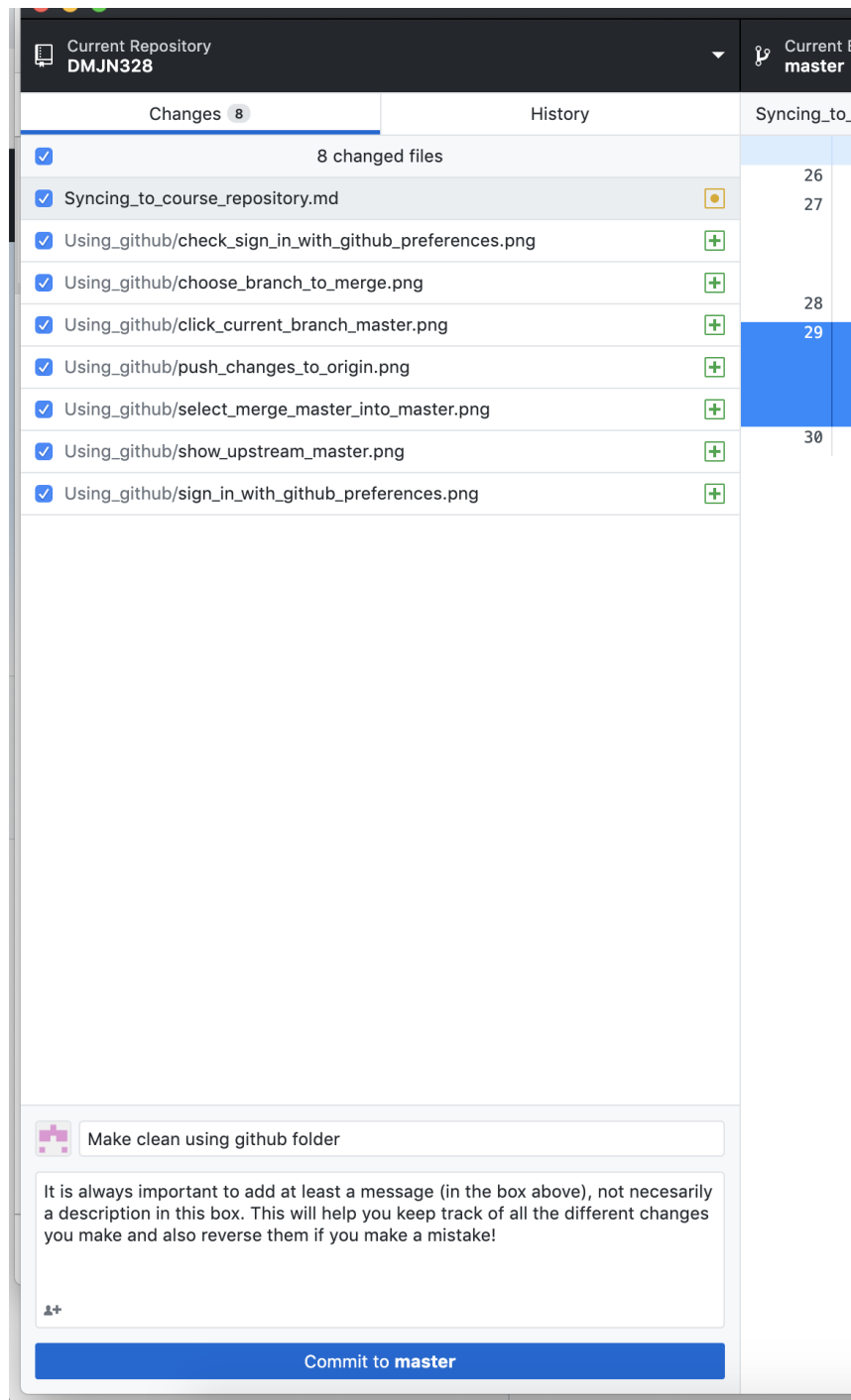
Then, after making the commits, push the changes up to your repository.

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Push 24 commits to the origin remote
You have local commits waiting to be pushed to GitHub.

Always available in the toolbar when there are local commits waiting to be pushed or ⌘ P

[Push origin]

Open the repository in your external editor