Getting Started with git

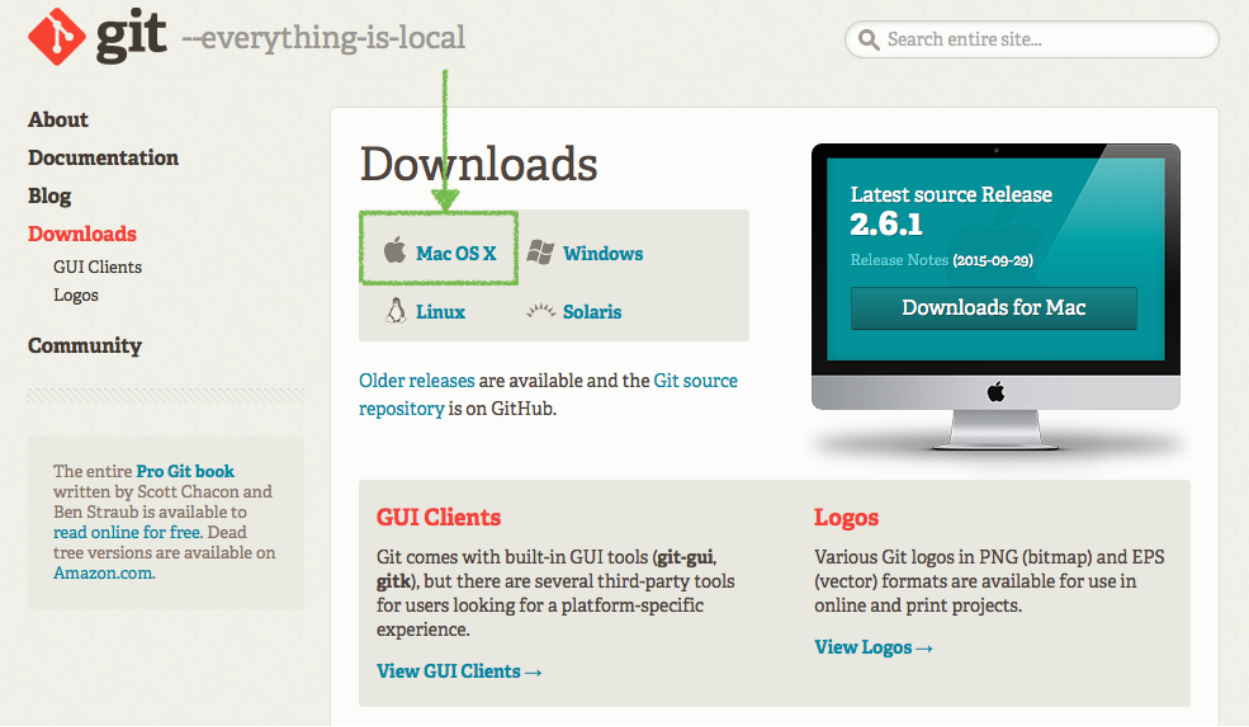
Daniel Anderson, University of Oregon - College of Education

Getting started with *git*

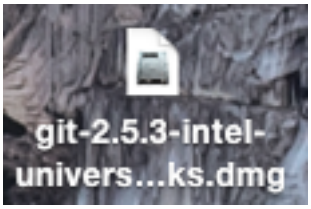
git is a version control system for collaborating on code. The largest git hosting engine is github. However, github is separate from *git* itself. There are multiple git host sites. For example Bitbucket is another popular host site that has some advantages over github like free private repositories (github repositories are all public, unless you pay a monthly fee). For this course, we'll be using *git* with github. You will need to only understand the very basics of *git* for the purposes of this class, but if you are going to go on to do any serious coding, you'll want to become more familiar with it. The first two chapters of the *Pro Git* book, available here will provide you with most all the background you need. The purpose of this document is to provide you an overview of installing *git* and cloning the course repository.

Installing *git*

Installing *git* is straight forward. Simply go to the downloads portion of the *git* website and click on the Mac OS X logo.

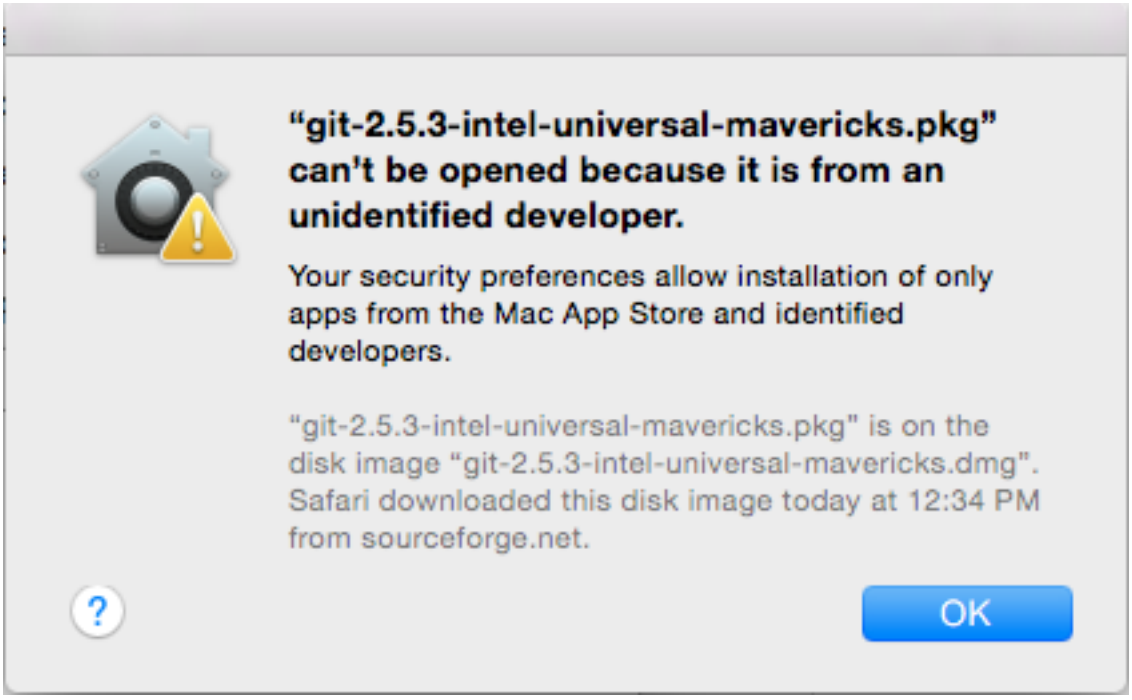


You should then get a .dmg file named something close to (or exactly) git-2.5.3-intel-universal-mavericks.dmg

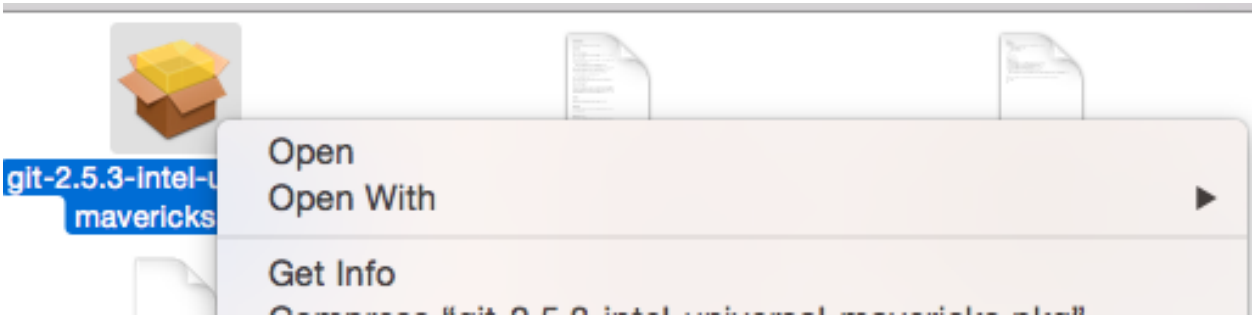


Double click the dmg file and you should see a .pkg installer

If you just double click the pkg installer you will likely be told it cannot open because it's from an unidentified developer.



To get around this you'll need to "right" click on the pkg installer (by either placing two fingers on the trackpad and clicking or holding control while clicking), and then selecting "open".



You will still be prompted by the unidentified developer warning. Just select "Open". You'll then be guided through the installer.

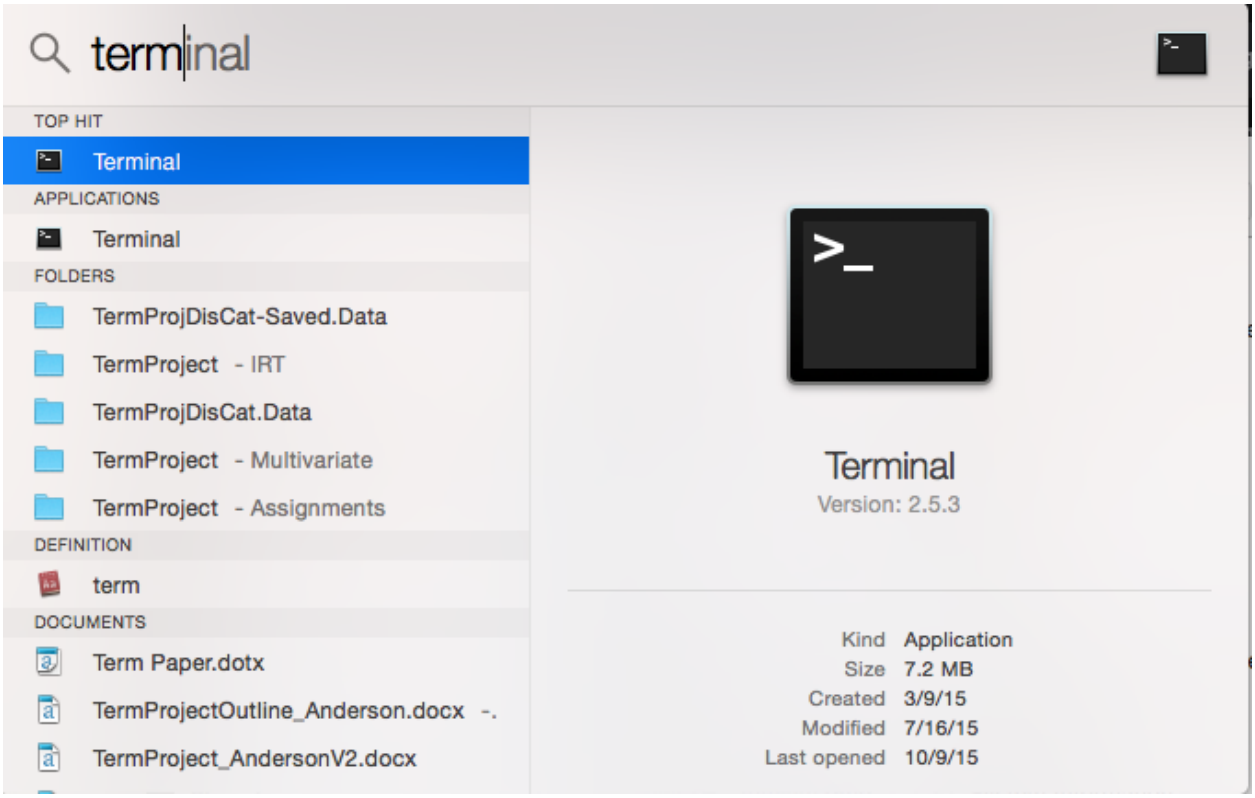
Note. After running the installer, *git* will be installed on your computer. However, you will not see it in your applications or anywhere else. Rather, it is installed as part of the command line tools, and you'll access *git* via terminal.

Cloning the course repository

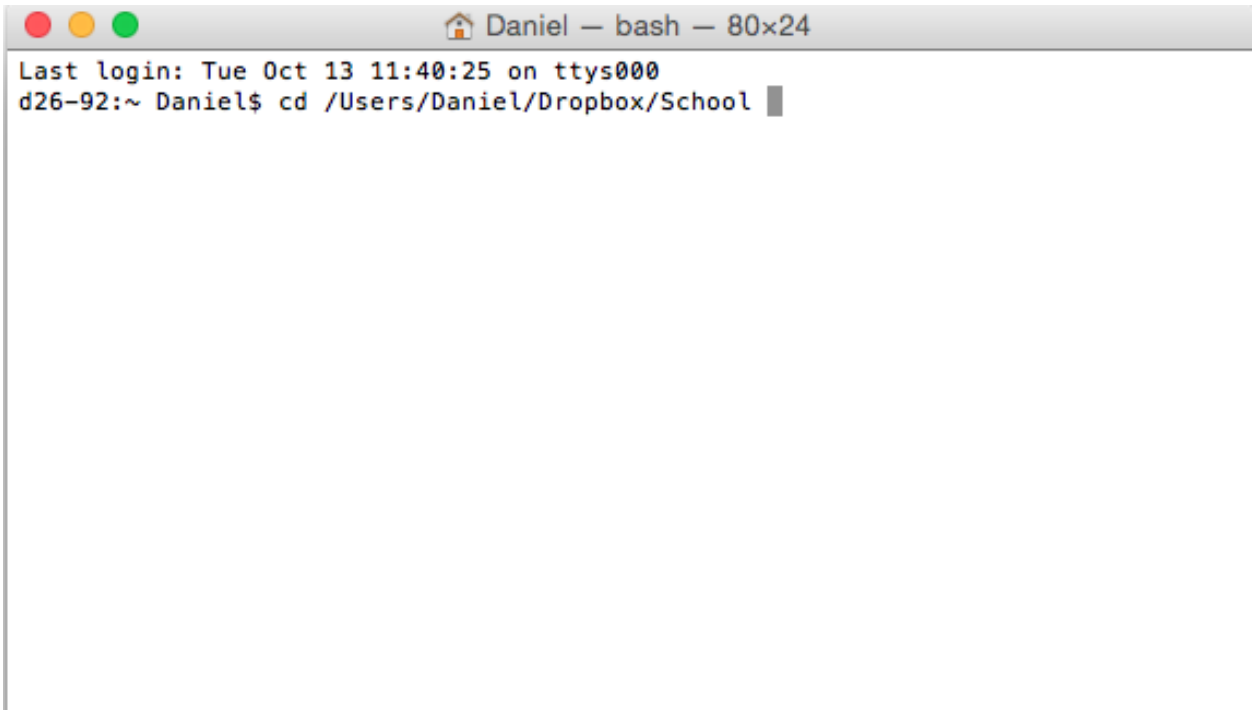
This tutorial describes how to clone the course repository through terminal. There are many free *git* graphic user interfaces (GUIs) that many people use. I don't know them, and prefer to just work with the code, but you can feel free to explore them.

Launch terminal

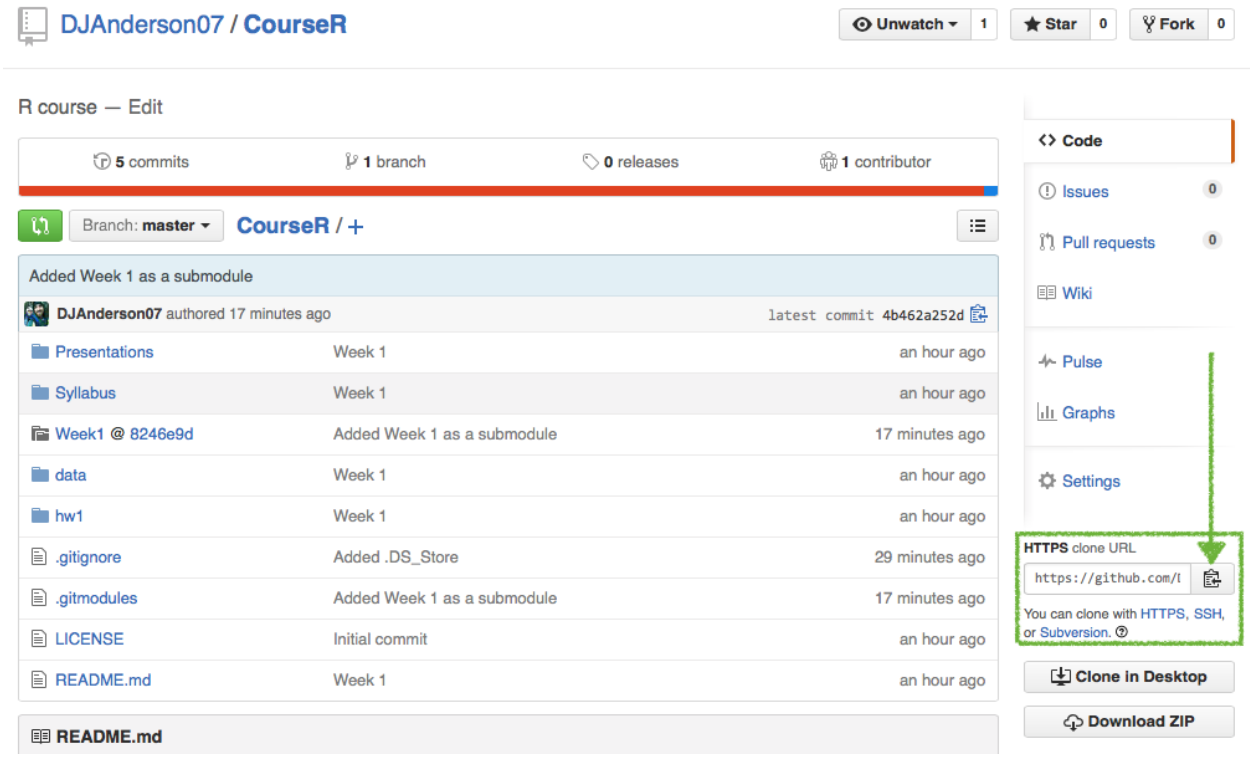
Terminal is a mac application available at /Applications/Utilities/Terminal. Alternatively, you can use Mac's spotlight feature by pressing command and space bar. Then, type terminal, and the application will come up.



Next, change your directory, via `cd` to where you would like all the course materials saved. One of the easiest ways to do this is to type `cd` and then a space, and then just drag the folder to terminal where you want the materials saved. The path will then be automatically copied into terminal.



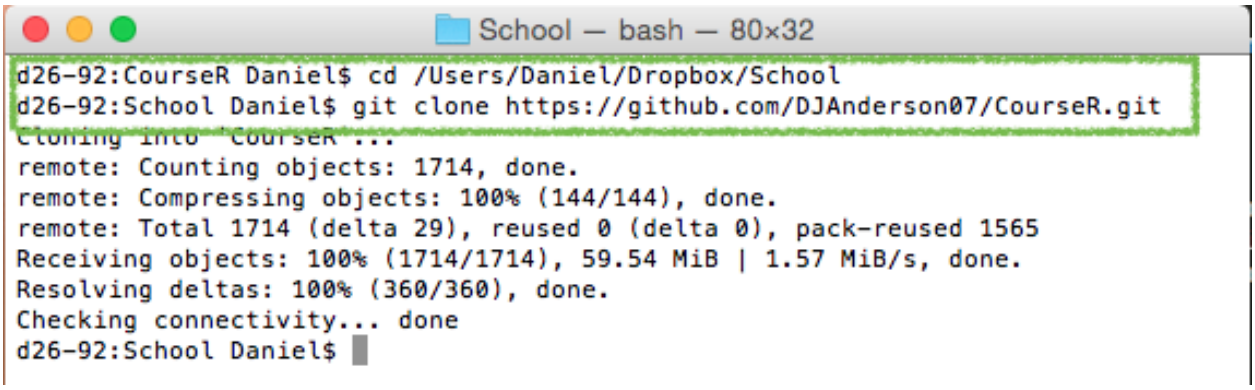
Finally, go to the course github repository and copy the link to clone the repository.



and type `git clone` into terminal, followed by the link (i.e., paste the link). It should like like this

```
git clone https://github.com/DJAnderson07/CourseR.git
```

After you hit enter, the course materials should all be cloned (copied) into your directory.

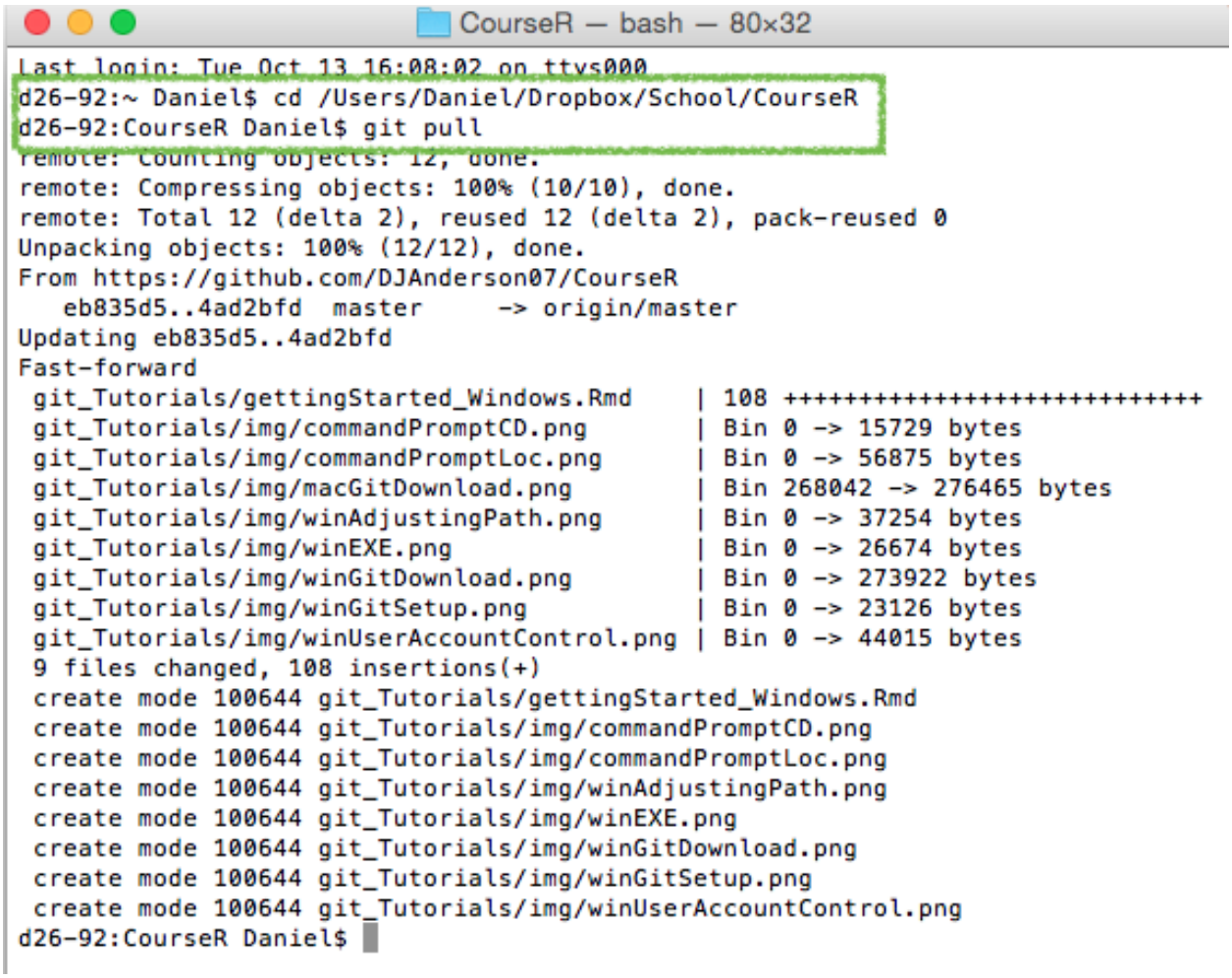


Note. If you do not have command line tools installed, you will be prompted to do so. You will need to install the command line tools to work with *git* via terminal.

Updating the repository

As mentioned in the opening, if you're planning on doing any serious coding, you should really dig into *git* a bit more. However, for the purposes of this class, you should only need one command: `git pull`. The process is pretty easy. Just follow the following steps

1. Launch terminal
2. type `cd` followed by a space, then drag the course folder (CourseR) into terminal to get the path. Hit enter.
3. Type `git pull` and hit enter. Assuming the repository has been updated, terminal should print some messages about some files.



4. Type `git status`. If you see the following message, you should be good to go. If you see something like "master ahead of branch by X commits", or some other message, you are not up to date. Make sure you're in the right directory and try `git pull` again. Otherwise, contact me and I can try to help you diagnose the problem.

```
# On branch master
nothing to commit, working directory clean
```