



Advanced Online Media

Dr. Cindy Royal

Texas State University - San Marcos

School of Journalism and Mass Communication

Git and GitHub

Before going through this handout and the Git and GitHub tutorial, please watch the Command Line Basics video to understand how to get around in the Terminal or other command line application.

Git is a program that handles version control for your applications. It can help you keep track of your own versions of projects and files (even non-coding files like Word documents) and can keep track of versions in a multi-user environment.

GitHub.com is a great website that serves as a social network for coders. There are lots of projects you can review and download the code so you can tinker and modify.

Installing Git

We will be working with the Git application in the Terminal as well as the GitHub application. And we will be using the GitHub.com website to push our projects to the Web. Go there and set up an account. Make sure you respond to the verification email from GitHub before going forward.

1. Check to see if your computer has git. Go to the terminal and type `git --version`.
2. If you don't see a version, your computer will probably ask you to install XCode. Choose Install. This may take a few minutes. This works for a Mac. .
3. You need to close the terminal, then open it back up. Then do `git -version` again and you should see the version coming up.

Note: This is slightly different than the video shows. The XCode installation installs Git into the proper folder now.

Adding Repositories

1. First go to GitHub and create a new repository for these files to go into there. Choose the New Repository +. Make this one WITHOUT including a readme. This will allow you to see the instructions for

uploading your files. Make it public.

2. Next, let's create a local repository from one of our folders. Find a project you want to post on GitHub. Use Terminal to navigate to that folder (i.e. `cd Documents`, `cd coding`, etc) until the folder name appears before the `username$` in the Terminal.
3. You will do these commands to initialize the folder as a repository.

```
$ git init
$ git add .
$ git commit -m "First commit"
```

These three commands initialize the repository, adds all the files (instead of a ".", you can use a filename to add a single file) and commit it to the repository with a message of your choosing.

4. Now we want to create the remote and associate it with the term "origin". Find that line from your GitHub.com repository, it should look something like:

```
$ git remote add origin
https://github.com/cindyroyal/codingforcomm.git
```

Except it will have your repository's url in it.

5. Finally, we want to push this to GitHub.com.
- ```
$ git push -u origin master
```

The `-u` tells it to prompt for your GitHub username and password. Once you do this, you don't have to do the `-u` as long as you remain logged in.

So, give this a try. Get a few of your projects posted to GitHub. It seems a little confusing at first, but after a few experiences with it, it will come a bit more naturally.

You can create a repository called `<username>.github.io`, and push files to it, and it will actually be a publicly hosted website, so free hosting! Feel free to check that out.

If for some reason, you want to remove a git repository, you can delete it via GitHub. When you are in a repository, click on Settings. At the bottom of that page, you can Delete This Repository. It asks you to type in the name of the repository for security before it removes it. Then you need to remove

the tracking on the folder on your computer. You need to do this if you want to re-upload these files to another repository. Run this in the Terminal in the folder you want to remove the tracking.

```
rm -rf .git
```