Jordan Zehr
Github Tutorial

**** Anything in italics below is a command for the terminal ****

Welcome to the GitHub tutorial! Github is a great resource for version control, collaborative coding projects, and organization that yields to increased proficiency while programming. In this tutorial we are going to go over the basics of using this tool. As always, my disclaimer is:

> This is not to be used as a comprehensive guide, but rather, an introduction to how this tool can be used. It will be very helpful to explore this tool on your own, and as always, the internet is a great first place to look for answers.

Before we get started, check out this video to get a basic understanding of how git works. Once you are done with that, you are ready to get started!

1. Step one, we will need to install Git on our machines.
    a. Download Git
2. Step two, set up **git config** user name and email. Choose this wisely, as your username will be public to the whole coding community. Furthermore, it will be useful to note all of your settings, that way if you ever need to input this information again you will remember it.
    a. *git config --global user.name "User Name"*
    b. *git config --global user.email "email"*
    c. *git config --global color.ui true*

Now that that is taken care of, we can initialize our first project

-- *cd/to/a/directory and then*

3. *git init*
    a. At this point, git is tracking everything in this directory. Every time you add a file, git knows, every time you modify a file, git tracks it. Git is "big brother" now, watching your every move.
    b. This is the versioning control mentioned earlier.

Once you have made some changes, lets add them!

4. *git add <file_name>*
    a. This will tell git that you want to add this file to the tracked changes.
5. *git commit -m "my first commit!"*
    a. This will be your commit message that git will attached to the commit. Keep these messages short and sweet, but also informative.

6. *git push origin master*
    a. Ok so there is a bit going on here, so we will break it down.
        i. With *git push* we are saying “hey git, push this code to a remote repository”.
        ii. With *origin master* we are saying “push to our origin repo (where the repo started) and push those changes to the master branch.
7. One of the most useful commands is: *git status*
    a. This command gives you a quick synopsis of what branch you are working on and what repository you are working in, and which changes have and have not been tracked for changes logging.

This is just a crude, simple introduction, however, these are the commands you will use the most. Do not be afraid to make a mistake with git, as you can always fix your mistakes thanks to versioning control. Dive in and get dirty!