**Quick Git tutorial – actual terminal/cmd commands are italicized**

Git is a very robust version management software. For what we are using it for, we need to know about 3% of git’s actual functionality. Git is an industry standard for sharing and collaborating code and I highly encourage you to learn more. Here we will cover the absolute basics (and really unless you’re working on something big with a lot of other people, the basics are all probably all you’ll need to know).

If you don’t have git installed, follow this tutorial – <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

**Cloning** – cloning a repository is copying down all files from a repository on Github to a local directory on your physical machine.

* Go to <https://github.com/>
* Create an account
* Open a terminal (cmd for Windows)
  + Navigate to a directory of your choice
  + Initialize git in your terminal
    - *git init*
  + Clone the necessary repositories
    - *git clone https://github.com/neillyt/openstack.git*
    - *git clone https://github.com/neillyt/fun.git*
  + You will now see you have two new directories – one for each repository that you’ve cloned. The OpenStack repository will be for notes and documentation.

**Pulling files from a repository –** this assumes you have already cloned a repository

* Open a terminal (cmd for Windows)
* Navigate to your directory/repository that you’ve cloned
  + *git pull*

**Pushing a file(s) to a repository** – this assumes you have already cloned a repository

* Open a terminal (cmd for Windows)
* Navigate to your directory/repository that you’ve cloned
* Copy any files you want to push to the repository on github to this directory
* First we must **add** files to the staging area
  + *git add <filename> OR git add –all*
    - Example: *git add test.txt*
    - Example: *git add –all*
* Next we must **commit** those files to be pushed. Git commit requires a message of some sort which acts as an explanation to others as to why the file was pushed to the repository. If you do not add the **–m** switch, it will open VIM text editor and you can add a message there. If this happens and you are not familiar with VIM then see the **VIM** **section**.
  + *git commit <filename> -m <message>*
* Lastly we must **push** the file to the repository on Github.
  + *git push*

**VIM**

If you ever decide to use/learn Linux, becoming familiar with VIM is imperative. It’s an extremely powerful text editor with a steep learning curve, but after grasping VIM’s functionality it makes editing text without a mouse a breeze.

This is a quick and dirty crash course in case you get stuck when commiting a file using git. This is the sheer basics.

VIM has multiple modes, for this you’ll need to know 2: **Escape** mode and **Insert** mode. These can be accessed using the ***ESC*** key and the ***i*** key respectively.

Inserting text: We need to be in escape mode to enter insert mode, so first let’s make sure we are in escape mode. Press the ***ESC*** key a few times for good measure. Now press the ***i*** key. At the bottom of the screen you should now see –INSERT—. You are now free to type.

When you are finished typing, press the ***ESC*** key. You will see the bottom of the screen should now be blank. Press shift+semicolon ***:*** and you should see the colon appear at the bottom of the screen. We are now able to enter commands. Here is a list of commands:

* w = write (save)
* wq = write and quit
* ! = override
* wq! = write, quit, and override
* e = keep file open and refresh file to original state when opening

When you’ve entered your command, simply press enter.