8/4/17

* Day 1
* Learn how to use command line to navigate directories and modify files
* Discussed commands:
* PWD – print working directory
* Ls – list directory contents
* .. = parent directory
* Clear
* Mkdir = make directory / create new folder
* Touch keyboard.txt – created a file called keyboard.txt
* Day 2
* Quiz – reviews command line interface commands, got a 11/11 on first try
* Setting up command line
* Windows has a different CLI called Command prompt
  + But Bash is more popular
  + Installed Windows 10 Founder’s update to use the Ubuntu bash thru command prompt
  + Go to command prompt, and just type bash then hit ENTER
* Learning about Git
* Git is a versioning software
  + Records changes made to a project
  + Version control allows you to create different versions of your code as you achieve each milestone
* In bash,
  + Turn a directory into a Git project… Navigate to the directory you want to turn into a Git project
    - Type: git init
* Git project can be thought of as having three parts:
  + Working Directory – make changes to files, create, edit, delete, and organize them
  + Staging Area – Where you list changes made in the working directory
  + Repository – Where Git permanently stores those changes as different versions of the project, as a “commit”
* Type ‘git status’ to get a log of the changes that have occurred in the current version of a project
* Untracked files are files that Git sees but has not begun to track yet
* To add tracked files (adding them to the staging area)
  + Type “git add filename”
  + OR, to add all files in a project by typing “git add .”
* Git commit
  + A commit commits all of the files in your staging area (all the ones that have been git added).
* “git commit –m “This is the commit message”
* Good commit messages
  + Are in quotation marks
  + Use present tense
  + 50 characters or less
  + First word is an imperative verb and ends with a period. Example: “Fix broken registration link.”
* GitHub can be used just like the above repository, but remote
* To specify the remote GitHub repository, we add the remote and label it as the origin as such
  + git remote add origin <https://github.com/your-username/your-repository-name>
    - The remote is the same as the URL of the repo
    - The origin is an alias – instead of referring to the long URL now we can just refer to it as “origin”
    - To confirm this works, use “git remote –v”
      * The “push” and “fetch” should both point to your repo URL
* To push new versions (aka commits) to the remote repo,
  + Use git push –u origin master
    - Git push tells Git we want to push the update from the local repo to the remote repo
    - Origin is the alias (like described above) for the remote repo
    - Master refers to the master branch. Will learn about branches more in later lessons
    - -u stands for “upstream”
  + Only need to use the “-u” option first time pushing to a remote repo.
  + Now if we wanted to add more files, we would add our changes, commit them , then push the changes with
    - Git push origin master
      * Not no “-u” option this time. The local and remote repos are already connected
    - Do this every time you want to back up
    - Good practice to push after every commit
* In summary
  + Create your files in your git tracked working directory
  + “git add” them to the staging area
  + “git commit” them to the local repo
  + “git push” them to the remote repo
* Day 3
* Quiz on basic git workflow, got a 13/13 on first try
* Day 4
* Bicycle world project: “practice command line with this real world scenario”
* 8/6/17
* Cycle world project was just a guided exercise on the CLI commands
* Will learn how to configure Git on local machine and use it w/ GitHub to deploy first real website!
* Git and GitHub on local machine
* Deploying a github site
* GitHub offers a free service called GitHub Pages
  + Easy to setup
  + Free hosting with 95% uptime
  + Live updates when updated on github
* Project: **Excursion**
  + Guided walkthrough of git and github with a site on local machine. On GitHub as “excursion”
* Project: **Broadway**
  + Building and deploying a website from scratch, making Git commands along the way
  + \*\*I should be committing on Git more often\*\*