**Cheat Sheet - Courtesy of Kevin**

Kevin recently posted a cheat sheet for this section, you can refer to this for the final test! Thank you so much Kevin!

**GIT Cheat Sheet**

**init** — Create an empty GIT repository in your development directory

1. Go to directory
2. git init

**status** — Show the current state of the repository including un-added and un-committed files

* git status

**add** — Add a file to the repository staging area

1. Create a file, e.g. ***file.txt***
2. git add ***file.txt***

Add all new or changed files to the repository staging area (the period means all)

* git add **.**

**commit** — Commit all changes to the repository for first time (-m means message)

* git commit –m “***Initial commit***”

Commit all changes to the repository for later activities

* git commit –m “***Description of changes being made to project***”

**branch** — Create a new branch of the project

1. Choose a name for the new branch (original branch is ***master***), e.g. ***test***
2. git branch ***test***

List all branches (\* appears next to current branch)

* git branch

**checkout** — Switch branches and check-out all files (e.g. to ***test*** branch)

* git checkout ***test***

Create a new branch and check-out files in one command

* git checkout –b ***test***

**merge** — Merge two branches together (go to the destination branch first, e.g. ***master***)

1. git checkout ***master***
2. git merge ***test***

**(delete)** — Delete a branch that you no longer need (e.g. after a merge)

* git branch ***test*** –d

Or to force the delete:

* git branch ***test*** –D

**log** — View commit history (including long commit ID numbers)

* git log

**revert** — Revert all files back to a previous commit point

* git revert <long commit ID from the **log** command>

**rm -cache**

* Removes cache from a specific file so that it can be added to the .gitignore