*Git config –global user.name* <name>

Changes the username

*Git config –global user.email* <name>2 dashes

Changes the user email

*Cd* change directory

You can drag a file after typing *cd* to quickly enter the directory

*Cd directoryName*

Accesses a directory inside the current directory

*Pwd* print working directory

Shows what directory you are accessing

*Mkdir* make directory

Creates a new folder directory in the current directory. Does not change directory

*Git init* git initialize

Creates a new repository (repo) in the current directory

*Touch <*filename*>*

Creates a new file in the current directory with a given name

*.gitignore* ignores certain files highlighted in a file in a directory

*Git status*

Prints changes to current directory

Checks status of the working tree

*Git add <file>*

Adds a file to the staging area (index)

*Git add .* the period means all, adds all files

*Git commit*

Commits a file to a repo to a local repository

*Git commit -m “*message*”*

Commits a file to a repo and attaches a message

*Git push origin master*

Pushes a local repo changes to the origin repository.

Github will track these changes

*Git checkout -- .*

Restores files to previous commit. (its 2 dashes, a SPACE and a period)

*Git clone* url

Clones a directory and its files into the current directory. This can be done with a url

*Git remote -v*

Shows which directory a file would push to, including onto a server

*Git remote set-url origin* url

This changes where a file will push to, including onto a server like a GitHub directory

*Git branch <*branchName*>*

Creates a new branch with a given name

*Git checkout* <branchName>

Selects which branch to access

Changing branches will change the file to whichever version of it is currently saved in a branch

*Git merge* <branchName>

Merges a branch directory with a given branch