Github Commands.

* Commit all edited files and add a message

git commit -a -m "My commit"

* Add all new files ..

git add .

* Perform a pull operation

git pull REMOTENAME BRANCHNAME

* Perform a push operation

git push REMOTENAME BRANCHNAME

* Prune all stale remote tracking branches

git remote prune REMOTENAME

* Create a branch

git branch BRANCHNAME

* View branches

git branch

* Checkout a different branch

git checkout BRANCHNAME

* Checkout a remote branch

git checkout -b LOCALBRANCHNAME origin/REMOTEBRANCHNAME

* Merge the changes made in another branch in to the current branch

git merge BRANCHNAME

git merge BRANCHNAME

* Delete a local branch

git branch -d BRANCHNAME

* Delete a remote branch

git push origin :BRANCHNAME

* Delete a remote branch (sexier syntax)

git push origin --delete BRANCHNAME

* Scrap uncommitted state and return the working tree to the last committed state

git reset --hard HEAD

* Delete the latest commit, and return to the one previous (one before HEAD)

git reset --hard HEAD~1

* Return a single file to it's last committed state

git checkout -- FILENAME

git checkout HEAD FILENAME

* Git log

git log

git log --pretty=oneline

git log --pretty=short

* Cherry pick commits and apply them to another branch (first grab the commit ID from the branch with said commit, then checkout the branch you wish to apply the commit to)

git cherry-pick COMMIT-ID

* Stash uncommitted changes

git stash save "message"

* Apply stashed changes somewhere

git stash apply

* Stop a file being tracked (but do not delete it from the working directory, add to .gitignore etc after this)

git rm --cached <file/folder>

* Restore a file to a previous commit

git checkout <commitID> <file/to/restore>

* Restore a file to one before a commit (say you know the commitID where something went wrong, and want one before that point)

git checkout <commitID>~1 <file/to/restore>