```
git commit -m "Your Message Goes Here"

git push origin master # specifies the branch to push changes to as the master branch
git push origin branchOne # push changes to branchOne
git push origin branchTwo # push changes to branchTwo
git push # this is also valid if you are being lazy, git will default to the currently checked-out branch

# ----- BRANCHING -----
    # branches are used to isolate development of project features from the master branch until they are ready to be merged
    # DEFAULT branch is master branch (when you first create your repository)
    # OTHER branches can be created and used for seperate feature development and later merged with the master branch

git checkout -b branchWatermelon # CREATES a NEW branch named branchWatermelon and switches to it
git checkout master # switches back to master branch
git branch -d branchWatermelon # DELETES the EXISTING branch named branchWatermelon

git checkout -b branchPineapple # CREATES a NEW branch named branchPineapple and switches to it
git push origin branchPineapple # local branches ARE NOT AVAILABLE to others unless that branch is pushed to the remote repository

# ----- UPDATE & MERGE  -----

git pull # FETCHES and MERGES remote changes from the remote repository to your local working copy
git merge branchName # MERGES another branch's changes into your active branch
git diff sourceBranchName targetBranchName # PREVIEWS CHANGES between the branches that are to be merged, sourceBranchName and targetBranchName, to avoid merge conflicts
```

# More than the bare minimum

```
# ----- TAGGING -----

git tag 1.0.0 1b2e1d63ff # creates a new tag to accompany a given software release named 1.0.0, where the 1b2e1d63ff is the first 10 characters of the commit id as found with git log

# ----- GIT LOG -----

git log # displays repository history
git log --author=bob # only see commits of a certain author
git log --pretty=oneline # display a compressed log where each commit is one line
git log --graph --oneline --decorate --all # displays ASCII tree of all git branches with relevant tags
git log --name-status # shows only changed files
git log --help # see more parameters that augment viewing of git log

# ----- REPLACE LOCAL CHANGES -----

git checkout fileName3 # REPLACES LOCAL CHANGES in your working copy of the specified file with the last commited content in the HEAD (by default the master branch), whilst keeping changes already added to the index
git fetch origin # DROPS ALL LOCAL CHANGES and COMMITS, fetches the latest content from the remote repository and POINTS local master branch at it
git reset --hard origin/master

# ----- EXTRAS -----

gitk # calls the built-in git GUI
git config color.ui true # makes git output colorful
git config format.pretty oneline # displays the git log as just one line per commit
git add -i # enables interactive adding of files to the index
```

# Git clients