Common git commands

---

starting up

git init

`git init` creates a new local git repo from the current directory. You can also use ‘git init <path to repo> to create a repo in another directory.

Note: ***It is almost alway easier to create a new repo on GitHub, and then clone it to make a corresponding repo on your local computer!***

git clone

Use `git clone \<link to GitHub repo\>` to clone a GitHub repository on your computer.

Generally, this is the best way to create a new repo.

git add

Add new files on your computer to a git repo.

`git add \<my new file\>` will make git aware that you want git to track this new file.

---

everyday local

git commit -m “*short descriptive message*”

Commit changes to git. This makes changes you have made to local files “official”. Use

```
git commit -a -m "*short descriptive message"
```

to make sure all (`-a`) files are committed.

git status

Use this freely. Lets you know if you are up to date, if anything needs to be committed, etc.

```
git status
```

---

everyday remote (GitHub)

git fetch origin <branch>

Fetches any changes that have been made on GitHub without messing with your local repo. The branch argument is optional.

```
git fetch origin
```

git log –oneline origin

Shows you the info from `git fetch` - what has happened to your GitHub repo.

```
git log --online origin
```

git merge origin

Merges the changes identified by `git fetch` into your local repo.

```
git merge origin
```

git pull origin

Incorporates the changes to your GitHub repo into your local repo with no questions asks. Make sure this is what you want to do before you do it!

```
git pull origin
```

git push origin <branch>

Incorporates any local changes that have been committed into the GitHub Version of your repo. Unless you know that nobody else (including past you) has tinkered with the origin repo since your last `pull` (or `fetch` & `merge`), best practice is to a `fetch`, `merge` before doing a `push`.

```
git push origin
```