- *Version control software* keeps track of and stages changes that are made to a project (often one consisting of source code to software) and allows for development from one party without having to make sure that no one else is working on the same files (and that if conflicts come up, they are able to be resolved or worked around in their own time).
- `git add` "stages" a file that has been changed to be committed to a branch in a repository, essentially meaning that it marks a file as ready to be included in some formalized notion of a "change".
- `git commit` takes the files that have been staged as ready to change and "commits to" those changes by writing them into the source code as a specific point in time (often a specific change). Commits usually have extra messages that explain what the change was.
- `git push` takes all commits that are local to your computer and *attempts* to sync them with the code present in a more "centralized" location, such as a repository on a farm like GitHub or GitLab.
- `git pull` syncs changes from others in a centralized location onto your computer. In this way, it is the exact opposite of `git push`, and is a very common command for making sure you are up-to-date *in advance of* pushing changes out.
- `git clone` replicates an existing repository from a "remote" location so that you are able to make changes to it locally. Often, contributing to open-source development involves cloning a repository, making some changes on a branch of your own (called a *fork*), and requesting that your changes be integrated into the main branch (or a future branch for release later).