2/26: Working with Git

- A remote repository is a copy of our project that is stored "in the cloud"
- It is where we back up our work and share it with others
- It is accessible anywhere there is an internet connection
- Shortcut to add all files to working directory is `git add .`
- `git push` tells git to upload all your changes
- Branches are smaller bits extending from a tree trunk that represent different versions of our code
- Branches allow us to work on code fixes and features without breaking what we already have working
- Fixes and new features should always start on a branch
- The `master` branch is the "trunk" of the code tree and should only contain clean code ready for deployment to a web server
- `git branch` returns a list of all available branches with an asterisk next to the one we are currently working on
- `git branch <name>` tells git to start a new branch with the given name
- `git checkout` tells git to switch our working folder to the branch name specified
- We use the `merge` command to combine branches
- A merge conflict is when a file has changed in both of the branches you are trying to combine and git can't automatically determine what you want to keep

From what we learned, git appears to be an incredibly useful tool for collaboration due to its version control and branching. The latter especially ensures that two people can work on their code at the same time without interfering with one another. Git also allows two people to work with the same files without having to send them back and forth to each other, and both can maintain a copy on their own machine as well as the cloud. Overall, I would rate my understanding of git at a 4/4.