A remote repository allows us to back up our work and share it with others.

Branches are exactly what they sound like, smaller bits extending from a tree trunk. Fixes and new features should always start on a branch. The master branch is the “trunk” of your code tree and should only contain clean code ready for use on the web.

Git branch <name> tells git to maintain a new copy of our code with the given name.

Git checkout <branch> tell git to switch our working folder to the branch name specified.

Git tracks the files in the branches independently.

Use the merge command to combine branches.

When a file has changed in both of the branches you are trying to combine and git can’t automatically fix itself.

On a scale from 1 - 4 on my understanding of what we did today, I’d give myself a 3. Looking around it seemed like a lot more people were lost than I was, so I guess that’s a good thing. As far as the repos and the branching goes, I’d say I have a pretty good understanding of it. The one thing that I got semi-lost on was when we got into the merging because towards the end of the lesson, it said everything was up to date, when it shouldn’t have. This lesson will allow us to collaborate with our partners, and on top of that will facilitate good collaboration because we can see the changes that were made to the files.