Remote Repository

- Copy of project that’s stored in the cloud
- Where we backup work and share it with others
- Accessible anywhere with internet connection

Git push

  - Git remote add origin {link}
  - Git push -u origin master
- After first push you can use git push
- Git push tells git to upload changed to server
- Doesn’t need to push after every commit otherwise it’ll upload all commits
- Don't push everything, but commit everything

Branches

- Smaller bits that extend from tree trunk
- Represents different versions of our code
- Allows us to work on fixing code without breaking what’s already there
- Fixes & new features always start on a branch
- Master branch is trunk of code and should only have clean code ready to be uploaded on web servers

Git branch

- Git branch <name> tells git to maintain a new copy of code w/ given name
- Git branch lists branches available and displays * next to current one being worked on
- Git checkout <branch> tells git to switch working folder to branch name specified
- Add media queries to flexbox page
  - Git branch mobile
  - Git checkout mobile
  - Git branch
- Git merge <branch> combines the file changes in branch we name into current branch

Merge conflict

- When a file changed in both branches that’s trying to be combined
- Git can’t determine what is to be kept, so you need to clarify

Resolve conflict

- Remove tag and keep code needed

The topics of today can facilitate collaboration because my partner and I could collaborate on our Trouble project and any other projects in an easier way. We can also identify any errors or bring back any files before edits and change them if we want to go back to them with the use of remote repositories. On a scale of 1-4 my understanding of remote repos would be 3, branches would be 4, and merging would be 4. I don’t really have any questions about git.