Git Curriculum

Basics

- Initializing a git repository
  - Git init
- Adding and committing
  - Git add
  - Git commit -m “Message”
- Branching
  - Creating new branches: git branch <branch_name>
  - Switching to branches: git checkout <branch_name>
- .gitignore
  - Ignore included files or patterns when tracking files

Git vs. GitHub

- Definitions
  - Git: local version control
  - GitHub: cloud storage for git repositories
    - Think of it like Google Drive for Git repositories
- Remotes
  - Check remote: git remote show **or** git remote -v
  - Adding remote: git remote add <shortname> <url>
- Pushing to and pulling from GitHub
  - Git push <remote> <branch>
  - Git pull <remote> <branch>
- Git fetch vs pull
  - Fetch: command that tells your local git to retrieve the latest meta-data info from the original (yet doesn’t do any file transferring. It’s more like just checking to see if there are any changes available)
  - Pull: does what fetch does AND brings (copy) those changes from the remote repository
- Git Fetch
  - Show log of new commits: git log <branch>..<remote>/<branch>
- Pushing branches to and pulling branches from GitHub
  - Set upstream branch: git push -u <remote> <branch>
- Recommended Workflow
  - Create repository on GitHub
  - Clone repository to local
  - Proceed with adding and committing files
- GitHub SSH Keys

    - Generate key: ssh-keygen -t ed25519 -C "your_email@example.com"
    - Accept defaults unless you want to change the file location/name or add a passphrase
    - Start ssh-agent: eval "$(ssh-agent -s)"
    - Add SSH private key: ssh-add ~/.ssh/id_ed25519
    - Copy public key to Settings->SSH and GPG Keys->Add SSH Key

Reverting back to commit

- Use this command to list all commits: git log --oneline
    - Also, you can look through the commit history in GitHub. Doing it this way allows you to see what code was in each commit
- Checkout commit ID: git checkout <commit-id> .
    - Don't forget to add the ".”! This will put you in detached head mode which will not save your commits
- Now you can make changes and commit
- To move the branch pointer: git reset <commit_id>
    - To clear out staging area: git reset --hard <last_commit_id>
- Revert changes: git revert <commit_id>
- Reset vs revert
    - Reset deletes commits
    - Revert adds another commit that undoes a previous commit
        - This makes it nicer when others have pulled down your commits