Git Lecture Notes

Created	@Jan 2, 2021 7:08 PM
Tags	

Git & GitHub

Repo: where files and code are organized in a specific way

GitHub: website that allows you to upload Git repos online

Git: version control system

1. Helps minimize progress loss
    - By creating regular save points or **commits** and pushing them to GitHub
2. Iterating on different versions of the code easily
    - Your local commits (on your laptop) may be different than the ones online
    - Create **branches** off of different versions of the code usually named for features of the code
    - One branch is typically called the **master** branch
    - **Rebase:** where you move the base of the branch you are working on to the head of the other branch you are merging into
        - Not an easy operation, will require manual intervention to resolve conflicts
3. Collaboration is productive
    - How collaboration works:
        1. Make a copy of the main repository (**fork**)
        2. Make changes to the copy

3. Request to add your changes to the pain repository (**PR** or **pull request**)
4. Fetch changes that have been maid to master branch since you got your copy o the master branch (**bold**)

Demo: Committing Changes

> ❗ Remember to `git add` before `git commit`

> ❗
> 1. `git init` → initialize git repo
> 2. *create some files (e.g my-file.txt)*
> 3. `git status` → lists whether or not changes have been tracked
> 4. `git add` *`my-file.txt`* → adds file to staging area (`git add .` adds all files in current directory)
> 5. `git commit -m` *`message`* → commit changes and say what you did in your commit
> 6. `git status` → git status will say 'working tree is clean' if no changes have been made
> 7. `git diff` → displays exactly what has been changed
> 8. `git log` → shows initial commit, next commit, and unique ID for each change

Demo: Branching

1. `git branch` *`name-of-feature`* → create a branch
2. `git branch` → lists all of the branches
3. `git checkout` *`name-of-feature`* → jump to *name-of-feature* branch
4. `git status` → will list what branch you are on
5. *make changes to file in your repo*
6. `git add .` → add files with changes
7. `git commit -m message` → commit changes
8. `git status` → should say tree is clean if no changes have been made after commit
9. `git log`
10. `git push origin` *`name-of-feature`* → push changes to *name-of-feature* branch (not master branch)