Intro to Git & Github

(Wed. 3.Nov.2021) by **Yadong(Adam) Liu**

What is Version Control

https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

(Example of why we want version control) (use project as example)

What is git

Git explaned: https://www.youtube.com/watch?v=2ReR1YJrNOM
GIt doc: https://git-scm.com/about/branching-and-merging

How to install git

(Show how to install git)
https://git-scm.com/downloads

First step after installing git

https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup

Git & Github

(Explain the idea of repository)
(CHECK MIT missing-semester -git for detail on how git internally works)

- Git is a version control tool. Github is an online platform for holding repositories
  (Mention) Github Student pack
  https://education.github.com/pack

SSH key

SSH key is used for accessing github without using the password.

- Generate SSH key
  Windows -> https://phoenixnap.com/kb/generate-ssh-key-windows-10
  MacOS/Linux -> https://www.siteground.co.uk/kb/how_to_generate_an_ssh_key_pair_in_mac_os/

(Show how to use ssh key)

Our repository

(Show repository created for PT)

Link: https://github.com/Adamliu1/ProgrammingTutor-Adam

- Everyone can contribute to the repository (eg. share resouces that you think are useful)

Git commands

Reference: https://missing.csail.mit.edu/2020/version-control/

Basics

```
    git help <command>: get help for a git command
    git init: creates a new git repo, with data stored in the .git directory
    git status: tells you what's going on
    git add <filename>: adds files to staging area
    git commit: creates a new commit
        Write good commit messages!
        Even more reasons to write good commit messages!
    git log: shows a flattened log of history
    git log --all --graph --decorate: visualizes history as a DAG
    git diff <filename>: show changes you made relative to the staging area
    git diff <revision> <filename>: shows differences in a file between snapshots
    git checkout <revision>: updates HEAD and current branch
```

Branching and merging

```
    git branch: shows branches
    git branch <name>: creates a branch
    git checkout -b <name>: creates a branch and switches to it
        same as git branch <name>; git checkout <name>
    git merge <revision>: merges into current branch
    git mergetool: use a fancy tool to help resolve merge conflicts
    git rebase: rebase set of patches onto a new base
```

Remotes

```
    git remote: list remotes
    git remote add <name> <url>: add a remote
    git push <remote> <local branch>:<remote branch>: send objects to remote, and
update remote reference
    git branch --set-upstream-to=<remote>/<remote branch>: set up correspondence
between local and remote branch
    git fetch: retrieve objects/references from a remote
    git pull: same as git fetch; git merge
    git clone: download repository from remote
```

Undo

```
    git commit --amend: edit a commit's contents/message
    git reset HEAD <file>: unstage a file
    git checkout -- <file>: discard changes
```

(ask for next topic)

- Markdown?

Recommend materials

- Pro git book:
  https://git-scm.com/book/en/v2
- MIT missing-semester Lecture - Git (Highly recommend)
  https://missing.csail.mit.edu/2020/version-control/