**Git and Github Notes 2**

**What is Git:**

git is a version control tool that

1. keeps track of changes to code
2. synchronize code between different people

- one version of the code will be in an online repository (a repository is a folder)

- two people can each get the same copy of the online repository

- both make changes so their copies are different

- push their changes to the online repository and merge them

1. test changes to our code without losing to previous working version
2. revert to old version of the code

**Initial Git Setup:**

<https://git-scm.com/download/win>

command prompt is for windows commands

git bash is for Linux/Mac commands (use this)

git --version (checks the version of the installed locally git)

git config --global user.name "Your Name" (sets up the name of the user)

git config --global user.email "yourname@somemail.eu" (sets up the mail of the user)

git config --list (lists all the git configurations)

Text

Description automatically generated

ls (lists all the folders)

ls -la (lists all the files)

cd .. (returns one dir back)

cd (enters a directory)

$ cd 'Folder Name With Space' (enters a directory that has spaces in the name)

Text

Description automatically generated

git help <verb> (shows documentation for verb in browser)



ctrl + insert (copy)

shift + insert (paste)

ctrl + l (clears screen and pushes previous code above)

clear (type to clear the screen)

ctrl + c (exit weird > stuff)

**Scenario 1: Existing project that we want to start tracking with git**

git init

(initializes the git repository to start tracking the folder and creates a .git folder inside the repository)

rm -rf .git

(removes the git repository to stop tracking the folder and deletes a .git folder inside the repository)

Text

Description automatically generated

touch .gitignore (creates a git ignore file which is a text file where we can add files that we want git to ignore)

atom . (opens the current folder in atom)

add files to .gitignore to ignore them

Graphical user interface, text

Description automatically generated

Timeline

Description automatically generated

staging area allows us to pick and choose what we want to commit, as well we want to make smaller more detailed commits as opposed to larger commits with vague messages

to add one file to a staging area, git add filename.fileextenstion

to add all files to a staging area, git add .

to remove one file to a staging area, git add filename.fileextenstion

to remove all files from a staging area, git reset .

If we are working with 10 different files, and there are only 3 I am happy with, I do not want to just type save and save all 10. I only want to save the 3 good files which is where git add comes into play. git add doesn’t actually save, it just takes it to the staging area.

to check the state of the files within a git repo, git status

to take all files in staging area into the git repository, git commit -m ‘message’

The message should describe the commit made so that it’s easy to revert back to this version if needed. Insertions mean number of lines added. Deletions mean number of lines deleted.

If you want to add all the files you have changed and then commit them all, git commit -am "message"

git log lists all the commit hash numbers, authors, and dates

git log --oneline lists a condensed version of all the commits

Text

Description automatically generated

**Scenario 2: Remote repository (another folder or Github)**

another folder:

to clone a remote repo that is another folder, git clone <url>

Text

Description automatically generated

Github:

Github is a website that stores git repositories

go to credentials manager and if Github account is linked, make sure it is the correct one

Graphical user interface, application

Description automatically generated

create Github account and a repo

Graphical user interface, text, application

Description automatically generated

to clone a remote repo that is on Github, copy link and type, git clone <url>

A screenshot of a computer

Description automatically generated

(do all your changes and then commit changes)

we can refresh the Github page, and our commits will not be there

to upload changes on Github, git push

now, we can refresh the Github page, and our commits will be there

if the Github version is more updated than our current version, we can do git pull <url>

Text

Description automatically generated

**merge conflicts:**

- when two or more people make changes, but git doesn’t know who’s changes to take

for example, if I try to git pull but their code could conflict with my local code, there’s going to be a problem

Graphical user interface, text, application

Description automatically generated with medium confidence

the long number is the hash of the commit

let’s say I make changes to my code

Text

Description automatically generated

Text

Description automatically generated

then someone online makes changes

Text

Description automatically generated

if I run git pull, I have an error

Text

Description automatically generated

open atom,

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Undo:**

undo wrong commit message:

git commit --amend -m 'updated message'

Text

Description automatically generated

suppose we made a commit to the master, and not the branch. we can go into the desired branch and use git cherry-pick commitHashNumber to copy the commit from the master to the branch. Note this does not remove the commit to the master branch

Text

Description automatically generated

removes all untracked files:

git clean -df

Text

Description automatically generated

deleting everything:

git rm -r \*

Text

Description automatically generated

going to old versions:

- checkout (not dangerous), just shows the code at a given time

git checkout commitHashNumber

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

go back to normal

Text

Description automatically generated

- revert (kind of dangerous, but still very safe)

Text

Description automatically generated

after we type git revert hashNumber,

press shift colon

then type wq and enter

if we get conflict error, go to atom and fix, then add and commit

Text

Description automatically generated

- reset (very dangerous)

git reset --soft commitHashNumber, puts changes in staging area

Text

Description automatically generated

git reset commitHashNumber, puts changes in working directory

Text

Description automatically generated

git reset --hard commitHashNumber, changes are all gone

Text

Description automatically generated

**Branching:**

Branching allows for working on different features at the same time

Graphical user interface

Description automatically generated with medium confidence

default branch is master branch

at any given time, you focus is on only 1 branch which is represented by the head

if your head is pointing at feature, local repo is working on the feature branch

then if we feel the feature is good, we can merge it back to the master branch

Graphical user interface

Description automatically generated

git branch tells you which branch you are currently on indicted by the star, and which branches exist

to create a new branch and auto go to that branch, use git checkout -b branchName

git checkout branchName changes your current branch

Text

Description automatically generated

now that we are on the style branch, any changes we do won’t mess up the master branch

Text

Description automatically generated

Text

Description automatically generated

but these changes are only made to the style branch

if you want to push these changes, do git push urlOfGithubRepo

do git merge branchName to merge master with style branch

Text

Description automatically generated

go to atom and fix the merge conflict

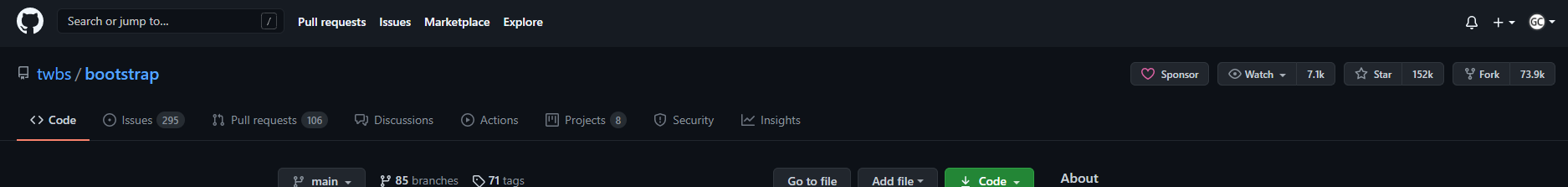
git branch --merged shows merged branches

git branch -d branchName deletes branch

Text

Description automatically generated

go to Github repo for bootstrap



forking a repo makes our own copy of a repo

A pull request is when we request for bootstrap to pull in our code and merge it, it’s sort of like asking bootstrap to do git pull