**Git Notes-**

What is Git:

* Git removes the need to copy files to and from the class share
* Git is like using your camera to take a snapshot of your files at a specific point in time that you can magically go back to if terrible things happen
* (In gamers terms) Checkpoint for files
* Git exists so you can modify/change/break/improve your code, so you don't ruin your work to badly
* Git is a collaboration tool that allows different people to work on all parts of a project at the same time
* Git protects yourself and others from yourself and others

The Local Workflow:

1. Open file explorer
2. Create folder named practice in H drive
3. Type cmd in address bar
4. A command prompt should open to H:\practice
5. Tell Git to watch folder by using git init

Getting Started:

* Creates repository in the folder you ran the command on
* Often shortened to repo
* Hidden location where file checkpoints will be stored

Three Main “states”:

* Now git will track your working directory
* Files exist in three states
* Modified- files that are new or have changes not yet saved by Git
* Staged- the current version of a file, tagged to be included in the next commit
* Committed- files that are safely stored by Git

Git:

* We commit the box to storage and note what it contains
* Until we commit our work, there is no checkpoint to save us from errors
* Git commit physically moves the box of copies into long term storage
* It doesn’t move or remove files in your working directory
* Make sure to describe what's in the box just in case

Outcomes:

* Remote Repositories-
  + - * + Copy of our project that is stored in cloud or another computer elsewhere
        + Where we back up work and share with others
        + Accessible anywhere with internet connection
        + Git push tells git to upload all your changes to the server
        + Does not need to be done after every commit bc it will upload all commits since last punch
        + Branches are smaller bits extending from a tree trunk
        + Represent different versions of our code
        + Allow us to work on code fixes and features without breaking
        + fixes/features should always start on branch
        + Master branch should always contain clean code always ready for use
        + Git branch <name> tell git to maintain a new copy of our code
        + Git branch on own will list the branches available and display an asterisk
        + Git checkout <branch> will tell git to switch to another branch if defined
        + Git is tracking the files in branches independently
        + Merge command to combine branches
        + Git merge <branch> combines the file changes in branch we name into our current working branch
        + Merge conflict is when a file has changed in both of the branches you are trying to combine and git can’t automatically determine what you want to keep
        + Basically git is asking for help

I believe that these lessons learned today can facilitate collaboration because they can help you see the changes made between partners and their work. This is a nice feature in case you or your partner are not together and you can’t talk about what you changes you made or were going to make.

Scale: 3

No questions here.

Best part of my thanksgiving break was sleeping in.