Git- Removes the need to copy files to and from the class share and your “h” drive

* Like a snapshot/checkpoint
* Exists so you can modify/change/break/improve your code, secure the knowledge that you can not ruin your work too badly because you created save point along the way.
* A collab. tool that allows different people to work on all the parts of a project at the same time.

3 main states:

* Modified - files that are new or have changes not yet saved by git
* Staged - current version of a file
* Committed - files saved in git
* Use “git init” for folder
* Git add (file)
  + Save as checkpoint
* Git commit -m “added (file)”
  + Commits to saving
  + Git commit -m “(adds description)”
* Git status
  + Sees status of git

Working with Git:

* Remote Repositories- A copy of our project that’s stored “in the cloud”
  + Where we backup our work and share it with others
  + Accessible anywhere with internet connection
  + Github.com
  + “Push” commands
  + “Git push”
* Branches- Smaller bits that expand
  + Fixes and new features should always start on a branch
  + Master branch is the trunk of our tree, should only contain clean code ready deployment
  + “Git branch ()” - tells git to maintain a new copy of code with the given name
  + “Git checkout <branch>” tells git to switch our working file to the branch name
  + Tracks files independently
  + Use “merge” command to combine
  + “Git merge <branch>”
  + Merge conflict - file has changed in both branches

In today’s lesson, I’ve learned how to use the “merge” command where you can combine branches. One can now collaborate more easily and efficiently without making mistakes like accidentally deleting a word, phrase, or number. While there can be many branches, individuals in a team can save each other's’ progress.

Remote Repos - 4

Branches - 4

Merging - 4

The best part of my Thanksgiving was obviously the food and seeing some of my family members that I don’t get to see a lot.