Evelyn Zepeda

Command prompt

A low level of interface for executing command
Not as pretty as GUI

- Keyboard input

Configuring web servers
Used to run specific development tools

- Git
- Note.js
- Ruby
- SASS/SCSS

How do we access the Command prompt?

Start menu
Right click on start and select command prompt
Or Win+X then C
Information before > is working path= where commands you type will run
Aka prompt
Anything before a "/" is a file name

How to use?
Navigation commands:

C:= navigate to this © drive letter
cd= change directory
Cd ..= go back 1
Cd \= go to root directory
dir= list contents of directory (folders)
dir/ad= limit list to only directories

Switches
/w= column view
/p= pause after each screen

Md name= make specified directory

Rd name= remove specific empty directory
Del main.css= delete specified file

Open file by typing open and it’s name and extension: open main.css

Intro to git

Git removes the need to copy files to and from the class share and H drive as you collab on a project
Git is like using a camera to take snap of files so you can go back
Aka checkpoint
Git exists so you can modify and change code/ there are save points along the way
Git is a collab tool allowing different people to work on projects at the same time
Protects yourself and others

How to set it up
Create folder
Go to command prompt/ navigate to folder using “cd ______”
Write “git init”
Type “git status”
How to start a checkpoint: “git add index.html”

The local workflow

Git init creates a repository in the folder you ran the command on
Aka repo
Where checkpoints are stored
Modified- where 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- safely stored by git
Git add neatly packs a copy in a box for ever

Commit box to storage and note what it contains

Git commit -m “description goes here”
Until we commit, there is no checkpoint to save us from errors
Git physically moves the box of copies into long-term storage so make sure you describe what’s in the box just in case you need it

Does not move or remove files in the working directory

Add and commit the change
Git add index.html

Git commit -m “updated title”

Git log
Returns a list of all commits you have saved

Working with git

Goals:
Set up a remote repository
Push local files into remote server
Understand branches
How to merge branches
What is a merge conflict?

Remote repositories
A copy of a project that is saved in the cloud
Backup work/share with others
Accessible anywhere with a connection

Git push tells git to upload all changes to the server. It doesn’t need to be done after every commit, because it will upload all commits since the last path

Branches are like smaller bits extending from a tree
They represent different versions of code
Branches allow you to work on code fixes and features without breaking
Fixes and new features should ALWAYS start on a branch, so you don’t mess anything up on the master branch

The master branch is like the trunk and should contain clean code ready for use and deployment

Git branch= see all existing branches
Git checkout branchname tells git to switch working folder to the branch name specified

Working with branches

Create new branch
Git branch mobile
Navigate with
Git checkout mobile

Git tracks changes in files independently
Use the merge commands to combine branches
Make sure code is functional
Git merge branch combines the file changes into current branch
What is a merge conflict?

When a file has changed in both of the branches you are trying to combine and git can't automatically determine what to keep
Aka git is asking for help

How git tags

<<<<<<<<<<<<head- the branch you're working out

========== separates the conflicting changes in branches

Remove tagging to keep code, save. Add, comment

Reflection

Git can facilitate collaboration between people working on code, as one person can write something and propose their changes on a branch without breaking the entire mainframe of the code. If these changes are not wanted, the branch doesn't need to be merged to the master. Major changes can be committed and can be given descriptions to know who did what.

I rate my understanding of remote repositories a 4, branches a 3, and merging a 4.

Is there a less repetitive, more user-friendly/convenient way to work on code together instead of using Git? While I understand it, I feel like it's a tedious process to be constantly adding and committing, dealing with conflicts, etc.

The best part of thanksgiving break for me was probably the Overwatch free weekend. It was a lot of fun trying out a game I've always wanted, for free.