Monday

Three steps to be a better learned:

1. Clear the stage – initializing the ram space and removing non-critical topics from interacting with the process of learning.

2. Clarify the what and the why – Clearly define what the goals of your time spent learning will be.

3. Prioritize process over product – Focus on the process of learning and how to better accomplish that rather than just pounding out the actual learning of the product.

Play:

1. Wrestle with it – look at it from multiple angles and really get to know - be vulnerable
2. Take risks – Don’t be afraid of failure – you learn from it if you are wise. Effort is better than intelligence. Avoid the fixed mind-set, pursue growth
3. Have Fun! - Play with it. Gain familiarity. Enjoy the process of problem solving. The body experience of apprehension won’t kill you – learn to appreciate embarrassment.

Reflect

1. Self assess
   1. What are my objectives?
   2. How far did I get?
   3. How much is left?
   4. How was the process?
   5. Did I set the stage well?
   6. Did I play well?
   7. What can I do better next time?
2. Ask new questions – what’s the next step.
3. Repeat – Continual learning is integral to the career of programming. This process will be repeated frequently. Get used to it and embrace it.

Exercise – Taking risks: I don’t think I this process will work. What will people think if I try this.

Request Response Cycles – Will be studied heavily all throughout the program

Tuesday

Command line:

atom . - to open the working directory in atom.

pwd - to print the working directory

touch - to create a blank file

open – to open a file or a URL

git add – to add files to the git staging area

git commit –m ‘text’ – to commit the staging area to git

Add a CNAME file to the project to avoid repetitively typing the domain name

touch CNAME – to create blank

Example: GITHUB-USERNAME.surge.sh – chadlatham.surge.sh – the domain address

surge – to deploy the project

whoami – to display the current account name

groups – to display the groups that the account name belongs to

ls and ll (fish command) – l to list the working directory in long format

la (fish command) – to list long format, human, and hidden

Permissions – read, write, execute triplets - owner, group, world

man {command} – to display the manual page (similar to /?) (q to quit)

ctrl-c – to kill a running program

id – to get the ID numbers of our user accounts

File System Navigation:

brew install tree - used to outline the directory structure

ctrl-u - to clear the current prompt from the cursor back to the prompt symbol

rm -rf - same as (del \*.\* /s/q) (remove files or directories)

rmdir - remove empty directories

mv - to move files (can rename at same time)

cp - to copy files

cat - to concatenate files (list them)

echo - (copy con)

sed -n 179p filename.txt - To return an individual line number. (Stream Editor)

> - file system redirect to another file

echo 'hello world' > test.html - will output hello world to test.html

>> - file system append redirect

| - the pipe to another application symbol

pbcopy - (paste board copy) ll | pbcopy will redirect the ll to be pasted later.

less - opens in a pager program (similar to cat but lists by page)

Project: Murder Mystery

Notes:

1. Collect the clues (crimescene file marked with CLUE)

CLUE: Footage from an ATM security camera is blurry but shows that the perpetrator is a tall male, at least 6'.

CLUE: Found a wallet believed to belong to the killer: no ID, just loose change, and membership cards for AAA, Delta SkyMiles, the local library, and the Museum of Bash History. The cards are totally untraceable and have no name, for some reason.

CLUE: Questioned the barista at the local coffee shop. He said a woman left right before they heard the shots. The name on her latte was Annabel, she had blond spiky hair and a New Zealand accent.

Wednesday

3 GIT stages

1. Unstaged
2. Staged
3. Committed

GIT Commands:

git config --global user.name 'chadlatham'

git config --global user.email 'chadlatham33@gmail.com'

git init - to initialize an empty git repository in the working directory.

# this creates a hidden .git folder in the working directory that is the repository. It also creates and empty staging and commits area.

git status - will describe what's going on with the staging area.

git rm --cached <file>..." to unstage

git commit -m 'Message' - to commit from staging area. The last 3 of create mode additive bit flags for the permissions. 4-read, 2-write, 1-execute.

Includes 5 pieces of info. UID, Changes, Message, Timestamp, Author

git clone url (from github) - clones repo from github to directory - must fork first from galvanize

git log - to show all commits - not so useful because of ll

git ll - an alias to log for compressed log statement - very useful

git la - an alias to log for lengthy log statement - outputs everything! in less.

rm -rf .git - to remove git repository status (deinitialize) just delete the .git folder.

git rm <filename> - to remove a file from staging area.

touch .gitignore - to ignore files from being included in the git repository. Then add files or folders to the list in the file using relative paths.

git add .gitignore - to stage .gitignore which acts as the bouncer.

To rename files:

Method 1: To rename a file, first perform the fish operation. Then Add the new file. Then remove the old file. Git will recognize it as a rename at this point with git status.

Method 2: (Simpler way) git mv <sourcePath> <destinationPath> - This renames the filesystem file and stages the change in one step.

GITHUB

when creating a new repository, only initialize if you have not already done so on your file system.

\*\*\* git remote add origin https://github.com/chadlatham/notebook.git - remote specifies sending the .git party to a remote party - in this case, a github repository. (origin is the key) (the url is the value)

git remote -v - to display the verbose remote connection (not used often)

All git repositories have a default branch called master

\*\*\* git push -u origin master - to send the master branch to origin (github) - only required once per repository.

git push is for subsequent upstream pushes

git st - compressed version of git status

git c 'Message' - compressed version of git commit -m 'Message'

When an issue is opened on GitHub, it can be auto-closed by adding

Close #5 (for issue #5) on the commit message.

To clone a repository:

1. grab the url from GitHub

2. go to a fresh directory

3. git close url

Friday

Template Strings in ES6 - Built with interpolation instead of concatenation.

Guard Clause - What I've been calling a Trap.

Delete - keyword to delete a key-value pair in an object

for in - for iterating over objects

for (var key in object) {

. . .

}

Object.keys(object) to return an array of the keys.

Object.hasOwnProperty - checks for a non-inherited property on an object.