

Module 1 - Open Notebook, Fail-Log

Exercise 1: Learning Markdown syntax with dillinger.io

- This Exercise was easy to understand and I encountered no major problems.
- Created a commentary on MarkyMarc's annotation (<https://hyp.is/ISCiyFiEeiJnbMIODYc7w/wcmnotebook-history.html>)
 - Used dillinger.io
 - Included 2 Creative Commons images 2 relevant website links
 - Very minor confusion regarding the upload of the images
 - Didn't realize at first that I needed to add "[alt text]" before the image urls, which needed to be in parenthesis
 - I made sure to include both the annotation's url in the commentary as well as to ensure ".md" ended the document name
 - Exported the file Markdown
 - Exported the file as both a pdf and html file

Exercise 2: Getting familiar with DH Box

- This Exercise was easy to maneuver and completed successfully without major issues.
- Created an account with DH Box under the username: dannyjkj
- Using the Command Line
 - Input command to copy of the Pandoc program
 - `$ wget https://github.com/jgm/pandoc/releases/download/1.19.2.1/pandoc-1.19.2.1-1-amd64.deb`
 - Unpacked the file
 - `$ sudo dpkg -i pandoc-1.19.2.1-1-amd64.deb`
 - created record of commands typed
 - `$ history`
 - Piped list into a new file
 - `$ history > dhbox-work-today.md`
 - Used text editor "Nano" to view inside the file
 - `$ nano dhbox-work-today.md`
 - did not need to reinstall nano
 - Created a new file with nano and titled it with "May 18, 2018" by inputting "#"
 - Minor confusion here, misunderstood the wording of the prompt to be saying I must open an already existing file in nano
 - Soon realized mistake and acted accordingly
 - Added text to explain exercise
 - Saved the new file and exited nano
 - Again ensured ".md" ended off the file name of "readme"
 - Entered the File Manager and saved copy of "todayscommands.md" onto my computer

- `$ dhbox-work-today.md`
- Converted the Markdown file into both .doc and .html files
 - .doc: `$ pandoc -o todayscommands.docx dhbox-work-today.md`
 - .html: `$ pandoc -o todayscommands.html dhbox-work-today.md`

Exercise 3: Setting up your GitHub space

- This Exercise was very straightforward and easily accomplished
- Created a Github account with username DannyJKJ (<https://github.com/DannyJKJ>)
 - Created new repository “hist3814o”
- Uploaded the “todayscommands.html” file, previously downloaded from DH Box into the new repo
 - Had to do this twice
 - I only later realized that the first upload was accidentally created as a Firefox HTML Document, rather than a Chrome version
 - In order for the file to open through Chrome in my repository, I needed to first change the file into Chrome and then reupload
- Uploaded Exercise 1’s commentary on MarkyMarc’s annotation to the new repo

Exercise 4: A detailed look at using git on the command line

- This Exercise involved a great amount of frustration at the end of 4.2, but was eventually rectified and ultimately successful
- 4.1: git init
 - Created a new directory
 - `$ mkdir first-repo`
 - changed new directory into my “first-repo”
 - `$ cd first-repo`
 - Created, and saved, a readme.md file in first-repo directory with nano
 - `$ nano readme.md`
 - Had program track the folder as a repository
 - `$ git init`
- 4.2: git status
 - Following the addition of more text to the readme.md file, checked status of the repository
 - `$ git status`
 - Committed the text addition to the file
 - `$ git add -A`
 - Took a snapshot of the committed git status
 - `$ git commit -m "My first commit"`
 - A Major issue occurred at this point

- I had originally done everything up to this point once before, but then realized I had made a mistake and thus tried to restart everything over. This caused a mass culmination of improper codes and nothing worked properly afterwards
 - I contacted Dr. Graham over Zulip, and provided snapshots of my screen to illustrate the issue. The entire conversation can be found here (<https://hist3814o.zulipchat.com/#narrow/stream/122217-Module-1/topic/Exercise.204.20redo.3F>) should anyone have similar issues
 - Dr. Graham provided me with a code that ultimately bypassed points 11 and 12 in the Exercise 4.2, and thereby set me back on track
 - `$ git commit -m "heres my message"`
- 4.3: git merge
 - Everything went smoothly from her on, after the aforementioned issue was rectified
 - Viewed my history of commits
 - `$ git log`
 - Created a new branch named “experiment” from one of my previous commits
 - `$ git checkout -b experiment <unique commit ID found in git log>`
 - Checked git status and saw the process was successful in moving the file into the master branch
- 4.4: git merge continued
 - Returned to the master branch to merge the files from the experiment branch
 - `$ git checkout master`
 - Merge successful with code
 - `$ git merge experiment`
- 4.5: git push
 - Created a non-readme and non-licensed GitHub repo titled “Git-Push”
 - Informed git of the new repo’s url
 - `$ git remote add origin https://github.com/DannyJKJ/Git-Push.git`
 - Pushed repo copy into GitHub version
 - `$ git push -u origin master`
- 4.6: git clone
 - Forked octocat’s “Spoon-Knife” repository into my own GitHub account
 - Cloned Spoon-Knife
 - `git clone https://github.com/DannyJKJ/Spoon-Knife.git`
 - Forked, then cloned into my account, Mimi Hailu’s repo “Historic”
 - `https://github.com/DannyJKJ/MimiHailu.git`
 - Created a new branch in DH Box titled “MimiHailu”
 - `$ git checkout -b MimiHailu <unique commit ID found in git log>`
 - Created a new file (“MimiHaiuluFile.md”) with nano inside new branch
 - Pushed repo copy into GitHub version
 - `$ git push -u origin MimiHailu`
 - Created and pushed a second file (“MimiHailuFile2.md”) for practice sake
- 4.7: pull request
 - Informed git to remember the original repo
 - `$ git remote add upstream https://github.com/MimiHailu/Historic.git`

- Sent New Pull request to MimiHaiulu from within her “Historic” GitHub repo
- 4.8: git merge again
 - No classmate has sent me a Git Pull request