Kiley Huffman
QBIO-490x
Bash and GitHub HW
Due 9/19 (extension granted by Jonathan)

Bash and GitHub
Due Date: Thursday 9/19 @ 11:59 PM

Answer the following questions about Bash:

Type out definitions for the following commands and shortcuts in your own words. Refer to the slides for definitions of commands, and use google to look up the definitions of any unknown shortcuts.

a. **pwd**: This command stands for "print working directory." It tells you what directory you're currently in. This command is useful for when you need to identify your directory path and for reorienting yourself.
b. **mkdir**: This command stands for "make directory." It creates a new directory or folder.
c. **cd <dir>:** This command stands for "change directory." It moves you to a different directory. The directory you are moving to is entered following the "cd" command. This command accepts both absolute and relative paths. If you do not enter an input, cd takes you to your home directory.
d. **cd…:** This is the same as the cd<dir> command, but without an input. If you do not enter an input, cd takes you to your home directory.
e. **ls:** This command stands for "list." It lists the contents of your current directory (folders, files, etc.). This command can be modified to list contents of your desired directory.
f. **rm and the -r flag:** The rm command stands for "remove." It deletes files and/or directories. There is no undo button for this. The -r command allows you to operate on directories and their contents recursively.
g. **cat:** This command is short for "concatenate": it is used to concatenate and display the content of files. The cat command can display files, concatenate multiple files, and create new files.
h. **head:** This command displays the first 10 lines of a file, hence the head of the file.
i. **tail:** This command displays the first 10 lines of a file, hence the tail of the file.
j. **scp:** This command is short for "secure copy." This command copies one directory or file into another location. You can use the -r (recursive) when copying from directories.
k. **nano (including Ctrl+o and Ctrl+x):** Nano is a simple text editor for Bash-like systems that run in the terminal. The ctrl+ o command in nano saves the file. After pressing ctrl+o, you must confirm/enter the file name and then press enter to save the file.

l. **--help:** This shortcut provides a help message that lists out all of the command-line options in nano. It is helpful when you do not remember or command or need help figuring out what command to use.
m. **TAB:** This shortcut automatically completes texts or file names (depending on your terminal settings.
n. **Ctrl+a:** This shortcut moves your cursor to the beginning of the current line.
o. **Ctrl+e:** This shortcut moves the cursor to the end of the current line.
p. **Ctrl+r:** This shortcut opens a file for insertion at the cursor's position.
q. **Ctrl+k:** This shortcut cuts the current list and stores it as a buffer.
r. **Ctrl+u:** This shortcut pastes the line you last cut from the buffer at the current cursor.
s. **Ctrl+l:** This shortcut refreshes the screen.

What command would you use to navigate to your Desktop from /Users/ using an absolute path? Relative path?

Absolute Path (Mac):

cd /Users/kileyhuffman/Desktop

where my username is "kileyhuffman"

Relative Path (Mac):

cd /kileyhuffman//Desktop

How would you copy /Desktop/Example Folder/ with multiple documents inside to /Documents/?

cp -r ~/Desktop/Example Folder/ ~/Documents/

If you didn't know which folder you were in, how would you navigate back to /Documents/?

Before navigating, I would check what directory I'm currently in using pwd. Then, I would navigate back to /Documents/ depending on what my current directory is.

Absolute Path: cd /Users/kileyhuffman/Documents/

Relative Path: cd ./Documents (if I was already in /Users/kileyhuffman)

Fill in the blank:

To push your local changes to GitHub, use the following sequence of commands:

1. git status **to view any unsaved changes.**
2. git add. **to save all files, or to save a specific file/folder.**
3. git commit **to commit files for saving. Use '-m' to include a message.**
4. git push **to push your changes to GitHub.**