

Note: Do NOT use the parentheses listed below in the actual code. But do use the quotation marks as shown (like with grep).

Command_name	Usage	Stands for	Definition	Tips
cat	cat (in_file1) (in_file2) > (out_file)	Concatenate	Concatenates the content of the files by adding lines	
cd	cd (folder_name)	Choose directory	Move around directories/folders	
cp	cp (file or folder_name) (new_location)	Copy	Copy a file	Use cp -r for folders
grep	grep '(pattern_search)' (in_file)	Globally search for a Regular Expression and Print matching lines	Search patterns in a file; Use > (new_search_file) to put your search into a file	Use grep 'pattern' (in_file) wc -l to count the # of lines that contains the search pattern
head	head (file)	Head	Enables you to see the top of a file	Use head -n # to define the number of lines to show
less	less (file)		Read a file	Use less -N to add the line number
ls	ls (location)	List	Lists the contents of a directory	
mkdir	mkdir (new_folder_name)	Make directory	Create new folder	
mv	mv (in_file) (new_location or out_file)	Move	Move files or rename files	
pwd	pwd	Print working directory	Shows your current location	
rm	rm (file)	remove	delete files and folders	Use rm -r to delete folders. Be careful there is NO way to recover what you have removed
sed	sed '(script)' (in_file) > (out_file)	stream editor	Find and replace/insert and delete patterns	Use sed 's/(find)/(replace)/' to find and replace in the

				whole file; 's/(find)/(replace) ' is an example of a script!
tail	tail (file)	Tail	Enables you to see the bottom of a file	Use tail -n # to define the number of lines to show; ex: tail -n (#) (file)
vi	vi (file)	VIM	VIM text editor	Press "i" to write, :wq to quit and save, :q! to quit and NOT save
wc	wc (file)	Word count	Count number of lines, word count, and character count	Use wc -l to count only lines

Fun facts and tips:

- 1) Use “Tab” to automatically complete words. e.g: If the name of your file is “Pokemon.csv”, write P, then press Tab to autocomplete. Be aware that coding differentiates between lowercase letters vs. capital letters.
- 2) We can use pipes, the long straight line (|), to link commands together. It takes the output from one command and feeds it directly to the next command. It helps make our command line work more efficient.
- 3) We use “>” to denote our output. Look at the example for using cat above.
- 4) Dots are used to signal a location in your computer. One dot (.) means “this present location”, (..) one folder back, (../..) two folders back. e.g: If you were >fall2024>Lab_01 to go one folder back to >fall2024 you will need to use “cd ..”
- 5) Use the up arrow to go to a previous command you have written to edit it and run it again.