Notes 8

cat

- Usage
    - The **cat** command is used for displaying the content of a file.
- Formula
    - `cat + option + file(s) to display`
- Example
    - Display the content of a file located in ~/Documents/sample_files/
        - `cat ~/Documents/sample_files/Code/helloWorld.py`
    - Display the content of a file with line numbers.
        - `cat -n ~/Documents/sample_files/Code/helloWorld.py`
    - Display the content of a file including non printing characters and line endings
        - `cat -A ~/Documents/sample_files/Code/helloWorld.py`

---

tac

- Usage

The **tac** command is used for displaying the content of a file in reverse order.

Just like cat, tac concatenates files and displays the output of the concatenates

- Formula

`tac + option + files(s) to display`

- Example
    - Display the content of a file located in
        - `tac ~/Documents/sample_files/Code/helloworld.py`
    - Display the content of multiple files in reverse order
        - `tac ~/Documents/sample_files/Code/helloWorld.py ~/Documents/sample_files/Code/helloWorld.sh`

---

head

- Usage

The **head** command displays the top N number of lines of given file. By default, it prints the first 10 lines. If more than one file name is provided then data from each file is preceded by its file name

- Formula

`head + option + file(s)`

- Example
    - Display the first 10 lines of a file
        - `head ~/Documents/Book/dracula.txt`
    - Display the first 5 lines of a file
        - `head -5 ~/Documents/Book/dracula.txt`

tail

- Usage

The **tail** command displays the last N number of lines of a given file. By default, it prints the last 10 lines.If more than one file name is provided then data from each file is preceded by its file name.

- Formula
    - `tail + option + file(s)`
- Example
    - Display the last 10 lines of a file
        - `tail ~/Documents/sample_files/`
    - Display the last 5 line of a file
        - `tail -5 ~/Documents/sample_files/`
    - Display the first 5 lines of multiple files
        - `tail -n 5 Txt/{dracula,war-and-peace}.txt`
    - Display the first line of multiple files using wildcards
        - `tail -n 1 Csv/*.csv Code/*.py`

---

cut

- Usage

The **cut** command is used to extract a specific section of each line of a file and display it to the screen.

- Formula
  - `cut + option + file(s)`

- Example
  - Display a list of all the users in yours system
    - `cut -d ':' -f1 /etc/passwd`
  - Display a list of all the users in your system with their login shell
    - `cut -d ':' -f1,7 /etc/passwd`

Remember: cut grabs columns

Fields start at 1 Must know what separates the fields(delimiter)

Options - `-d` - delimiter (what separates fields) `-f` - field number(s) `--output-delimiter` - change separator in output

More examples:

`cut -d': -f1 /etc/passwd` usersname only

`cut -d':' -f1,7 /etc/passwd` username and login shell

`cut -d' ' -f3,4 file.txt` fields 3 and 4 using space as delimiter

`cut -d':' -f1,7 --output-delimiter=' -> ' /etc/passwd` change how output looks

sort

- Usage

The **sort** command is used for sorting files. The sort command supports sorting: alphabetically, in reverse order, by number, and by month.

The sort command follows this order unless specified otherwise:

Line starting with a number will appear earlier in the alphabet will appear before lines starting with a letter that appears later in the alphabet

Lines starting with a lowercase will appear before lines starting with the same letter in uppercase.

- Formula
  - sort + option + file

- Example
  - Sort a file in alphabetical order

        - `sort users.lst`
    - Sort a file in reverse order
        - `sort -r users.txt`
    - Sort a file in numeric
        - `sort -n [file]`
    - Check if a file is sorted
        - `sort -c sorted.lst`
    - Sort and remove duplicate entries
        - `sort -u users.lst`
    - Sort by column 2
        - `sort -k2 file.txt`
    - CSV file, sort by column 3
        - `sort -t',' -k3 cereal.csv`
    - Save sorted output to a new file
        - `sort -o sorted.txt file.txt`
    - To check if a file is already sorted
        - `sort -c file.txt`

- If I need it ordered - `sort`
- If I need to save it - `-o`
- Specific column? - `-k`

Memory trick: sort = order `k` = column

`-t` = separator

`-o` = output

wc

- ### Usage

The **wc** command is used for printing the number of lines,characters and bytes in a file.

- ### Formula

`wc + option + file(s)`

- ### Example

    - Display the number of characters in a file
        - `wc -m users.txt`
    - Display the number of lines in a file
        - `wc -l users.txt`
    - Display the number words in a file
        - `wc -w users.txt`
    - Count everything at once
        - `wc file.txt`
            - Shows line,words, bytes