Notes 7

Defintions of the following terms:

cat

- Definition:

    - Short for concatenate which is the command's intended use.

- Usage:

    - Used for displaying the content of a file

- Examples:

    - Basic Example:
        - `cat + option + file(s) to display`
    - Display the content of a file located in the pwd
        - `cat todo.lst`
    - Display the content of a file using absolute path
        - `cat ~/Documents/todo.lst`

---

tac

- Definition:

    - Tac concatenates files and displays the output of concatenation

- Usage:

    - Used for displaying the content of a file in reverse order

- Examples:

    - Basic Example:
        - `tac + option + files(s) to display`
    - Display the content of a file located in the pwd
        - `tac todo.md`
    - Display the content of a file using absolute path
        - `tac ~/Documents/todo.md`

---

head

- Definition:

    - Prints the first lines of one or more files (or piped data) to standard output

- Usage:

    - Used to display the top N number of lines of a given file.

- Examples:

    - Basic Example:
        - `head + option + file(s)`
    - Display the first 10 lines of a file
        - `head ~/Documents/Book/dracula.txt`
    - Displayu the first 5 lines of a file
        - `head -5 ~/Documents/Book/dracula.txt`

---

tail

- Definition:

    - Prints the last few number of lines of a certain file

- Usage:

    - Used to display the last N number of lines of a given file.

- Examples:

    - Basic Example:
        - `tail + option + file`
    - Display the last 10 lines of a file
        - `tail ~/Documents/Book/dracula.txt`
    - Display the last 5 lines of a file
        - `tail -5 ~/Documents/Book/dracula.txt`

---

cut

- Definition:

    - A command-line utility that extracts specific sections of a specified file or piped data and prints the result to standard output

- Usage:

    - Used to extract a specific section of each line of a file and display it to the screen

- Examples:

    - Basic Example:
        - `cut + option + files(s)`
    - Display a list of all the users in your 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`

---

sort

- Definition:

    - ○ Supports sorting: alphabetically, in reverse order, by number, and by month.

- Usage:

    - ○ Used for sorting files

- Examples:

    - ○ Basic Example:
        - ■ `sort + option + file`
    - ○ Sort a file
        - ■ `sort users.lst`
    - ○ Sort a file in reverse order
        - ■ `sort -r users.txt`

---

wc

- Definition:

    - ○ Counts the words, lines, and characters in a list of files

- Usage:

    - ○ Used for printing the number of lines, characters and bytes in a file

- Examples:

    - ○ Basic Example:
        - ■ `wc + option + file(s)`
    - ○ Display the number of characters in a file
        - ■ `wc -m users.txt`
    - ○ Display the number of lines in a file
        - ■ `wc -l users.txt`

---

tr

- Definition:

    - ○ Translates or deletes characters

- Usage:

    - Used for translating or deleting characters from standard output

- Examples:

    - Basic Example:
        - `Standard output | tr + option + set + set`
    - Translate one character to another (For example a period with a comma)
        - `cat file.txt | tr '.' ','`
    - Translate white space into tabs
        - `cat program.py | tr "[:space:]" '\t'`

---

diff

- Definition:

    - A command-line utility that allows you to compare two files line by line

- Usage:

    - Compares files and displays the differences between them

- Examples:

    - Basic Example:
        - `diff + option + file1 + file2`
    - Display the difference between two files
        - `diff cars.csv cars-backup.csv`
    - Display the difference between two files in a column format
        - `diff -y cars.csv cars-backup.csv`

---

grep

- Definition:

    - Works line by line basis (it matches the search criteria in a line by line basis)

- Usage:

    - Used to search text in given file

- Examples:

    - Basic Examples:
        - `grep + option + search criteria + file(s)`
    - Search any line that contains the word "dracula" in the given file:
        - `grep 'dracula' ~/Documents/dracula.txt`

- Search any line that contains the word 'dracula' regardless of the case
    - `grep -i 'dracula' ~/Documents/Books/dracula.txt`