Notes 3

echo

- definition:
  - Used to display text given on the screen
- Usage:
  - `echo` + `option` + `string`
- example:
  - display a line of text
    - `echo "hello"`
  - display text with a horizontal tab
    - `echo -e "\thello"`
  - display 2 lines of text on one command
    - `echo -e "hello\nworld"`

date

- definition:
  - print or set the systems date and time
- Usage:
  - `date`+`option`+`format`
- example:
  - display the current time and date
    - `date`
  - display the time in month/day/year
    - `date +%D`
  - display time in coordinated universal time
    - `date -u`

free

- definition:
  - display the amount of free and used memory in the system
- Usage:
  - `free`+`option`
- example:
  - display amount of memory
    - `free`
  - display amount of memory in bytes
    - `free -b`
  - display amount of memory in gigabytes
    - `free --giga`

uname

- definition:
  - Print system information
- Usage:
  - `uname+option`
- example:
  - display os system
    - `uname`
  - display all system info
    - `uname -a`
  - display just the kernal name
    - `uname -s`

history

- definition:
  - display all previous lines inputted
- Usage:
  - `history+option`
- example:
  - display history
    - `history`
  - clear history
    - `history -c`
  - write history t the history file
    - `history -w`

man

- definition:
  - an interface to the system reference manuals
- Usage:
  - `man+options+section+page`
- example:
  - find the reference for date
    - `man date`
  - Lookup the manual pages referenced by smail and print out the short descriptions of any found
    - `man -f smail`

apt

- definition:
  - command line interface for managing packages
- Usage:
  - `apt+options+command+package`
- example:
  - upgrade packages
    - `sudo apt upgrade`
  - installing a package

        - `sudo apt install vscode`
    - removing a package
        - `sudo apt remove vscode`

snap

- definition:
    - command lets you install, configure, refresh and remove packages that work across many different Linux distributions
- Usage:
    - `snap+options`
- example:
    - installing snaps
        - `sudo snap install vscode
    - run apps from snap
        - `snap run vlc`

flatpak

- definition:
    - Build, install and run applications and runtimes
- Usage:
    - `flatpak+option+command`
- example:
    - installing an application
        - `flatpak install "application name"`
    - removing an application
        - `flatpak uninstall "application name"`
    - updating an application
        - `flatpak update "application name"`

mkdir-

- definition:
    - Creates a directory
- usage:
    - `mkdir` + `options`
- examples:
    - making a directory in your current directory:
        - `mkdir hello`
    - creating a parent directory:
        - `mkdir -p parent/child/hw`
    - creating a directory in a different directory:
        - `mkdir /home/directory`

touch-

- definition:
  - creates a empty file
- usage:
  - `touch` + `options`+`filename`
- examples:
  - Creating a python file in your current directory:
    - `touch filename.py`
  - Creating multiple files of different types at the same time:
    - `touch file1.txt file2.py file3.jpeg`
  - Creating a file with a different modification time:
    - `touch -d "11 Mar 2001" file.txt`

rm-

- definition:
  - deletes a file
- usage:
  - `rm` + `options`+`filename`
- examples:
  - Deleting a file:
    - `rm file`
  - Forceing a deletion:
    - `rm -f file`
  - Promt the user to confirm before every deletion:
    - `rm -i file`

rmdir-

- definition:
  - removes a empty directory
- usage:
  - `rmdir` + `options`+`directory`
- examples:
  - deleting a directory:

        - `rmdir directory`
    - removing multiple directories:
        - `rmdir dir1 dir2 dir3`
    - deleting a empty parent directory :
        - `rmdir-p dir1/dir1.1/dir1.1.1`

mv-

- definition:
    - moves or renames file(s)
- usage:
    - `mv` + `options`+`filename`+`destination`
- examples:
    - moving a file to a different directory:
        - `mv file.txt /home`
    - preventing a old file from being changed :
        - `mv -n oldname nowname`
    - moving multiple files:
        - `mv file1.txt file3.png file2.py /home`

cp-

- definition:
    - copy files and directories
- usage:
    - `cp` + `options`+`source`+`destination`
- examples:
    - copying files:
        - `cp test.txt 2.sh`
    - copying to a certain directory:
        - `cp test.txt destination_directory`
    - copying a directory:
        - `cp -R copy_dirctory`

file-

- definition:
    - determines the type of a file

- usage:
    - `file` + `options`+`filename`
- examples:
    - show the file type:
        - `file test.pdf`
    - show just the file type:
        - `file -b test.txt`
    - show file type in a slash separated list :
        - `file --extension test.png`

* -

- definition:
    - matches 0 to any number of characters
- examples:
    - move all files in a given directory that's a pdf to home:
        - `mv givenDirec/*.pdf ~`
    - list all directories in a given directory without listing their contents:
        - `ls -1d givenDirec/*/`
    - list all directories in a given directory and their contents:
        - `ls givenDirec/*/`

? -

- definition:
    - matches only 1 character
- examples:
    - list all files in the current directory with j and a 3 character file extension:
        - `ls *j*.???`
    - list all files in the current directory that have a 2 character file extension:
        - `ls *.??`
    - list all files in current directory that names are only 5 characters:
        - `ls ?????.*`

[] -

- definition:
    - matches 1 character from a set
- examples:

    - list all files in the given directory that starts with a number from 0 to 9:
        - `ls givenDirec/[0-9]`
    - list all files that start with a uppercase letter
        - `ls [:upper:]*.*`
    - list all files in current directory that starts with 3 lowercase characters:
        - `ls [a-z][a-z][a-z]*.*`

cat

- definition:
    - View the contents of a file
- Usage:
    - `cat` + `option` + `file`
- example:
    - view the contents of the file file.txt:
        - `cat file.txt`
    - view file.txt contents and number all output lines:
        - `cat -n file.txt`
    - view file.txt with $ at the end of each line :
        - `cat -E file.txt`

tac

- definition:
    - view file contents in reverse order
- Usage:
    - `tac` + `option` + `file`
- example:
    - view the contents of the file file.txt:
        - `tac file.txt`
    - view file.txt contents and number all output lines:
        - `tac -n file.txt`
    - view file.txt with $ at the end of each line :
        - `tac -E file.txt`

head

- definition:
    - view the top number of lines in a file based on a given number(default = 10)
- Usage:
    - `head` + `option` + `file`
- example:
    - view the top 10 lines of file.txt:
        - `head file.txt`
    - view the first 5 lines of file.py:
        - `head -n 5 file.py`
    - view the first 10 lines of 2 different files:
    - `head file.txt sample.py`

tail

- definition:
  - view a number of lines starting from the bottom in a file based on a given number(default = 10)
- Usage:
  - `tail` + `option` + `file`
- example:
  - view the bottom 10 lines of file.txt:
    - `tail file.txt`
  - view the last 5 lines of file.py:
    - `tail -n 5 file.py`
  - view the contents of file.txt from a certain line to the end :
  - `tail +25 file.txt`

cut

- definition:
  - extract and display specific sections of files in each line
- Usage:
  - `cut` + `option` + `file(s)`
- example:
  - cut out the 2nd character on each line in file.txt:
    - `cut -c 2 file.txt`
  - cut out the 2nd byte on each line in file.txt:
    - `cut -b 2 file.txt`
  - cut the first column of file.txt using delimiter:
  - `cut -d ',' -f1 file.txt'

sort

- definition:
  - arrange the order of lines in alphabetical order
- Usage:
  - `sort` + `option` + `file`
- example:
  - sort file.txt in alphabetical order:
    - `sort file.txt`
  - sort file.txt. in reverse order:
    - `sort -r file.txt`
  - sort the file numerically:
  - `sort -n file.txt`

wc

- definition:
  - outputs the number of lines, word and character count, and file name
- Usage:
  - `wc` + `option` + `file`

- example:
  - output the lines, words, character count, and name of file.txt:
    - `wc file.txt`
  - display only word count and name of file.txt:
    - `wc -w file.txt`
  - display file name and character count:
  - `wc -m file.txt`

tr

- definition:
  - translating or deleting characters from output
- Usage:
  - standard output|`tr` + `option` + `set` + `set`
- example:
  - translating one character to another:
    - `cat file.txt | tr ',' to '.'`
  - translating spaces to tabs:
    - `cat file.txt | tr "[:space:}" to '\t"`
  - removing all e's in file.txt :
  - `cat file.txt |tr -d "e"`

diff

- definition:
  - compares and displays the differences between the two
- Usage:
  - `diff` + `option` + `file1`+`file2`
- example:
  - display the difference between two files:
    - `diff file.txt why-cars-deserves-7-tv-shows.docx`
  - display the difference between two files in column format:
    - `diff -y file.txt why-cars-deserves-7-tv-shows.docx`
  - display the difference between two files with case sensitivity:
    - `diff -I file.txt why-cars-deserves-7-tv-shows.docx`

grep

- definition:
  - searches for a defined criteria of word(s)
- Usage:
  - `grep` + `option` + `criteria`+`file`
- example: +searching for all mentions of Dracula in Dracula:
  - `grep 'Dracula' Dracula.txt`
  - searching for all mentions of Dracula in Dracula with case sensitivity:
  - `grep -I 'Dracula' Dracula.txt`
  - display the number of the line for all successful outputs:
  - `grep -n 'the war on terror' file.txt`

read

- def:
    - takes the given thing and turns it into a variable