

Notes 7

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`