awk

- **Definition**
- awk is used for processing and displaying text awk can work with a text file or a standard output
- **Usage**
- `awk + options + awk command + file + file to save`
- **Example**
- Print the first column of every line of a file
- `awk {print $1}' ~/Documents/Csv/cars.csv`
- Print first field of a file
- `awk -F: '{print $1}' /etc/passwd`
- Print the last field of a file
- `awk -F: '{print $NF}' /etc/passwd`
- print the first and last field of a file
- `awk -F: '{print $1," + " ,$NF}'`
- print the first and last 3 field with line numbers
- `awk -F: '{print NR,1$,$3}'(and file)`
- print first and fourth field with line numbers
- `awk -F: '{OFS="="}{PRINT $1,$4}'(file`
- start printing a file from a given line without the first 2 lines
- `awk 'NR > 3 { print }'(file)`
- Convert the first field to upper or lowercase
- `awk -F: '{print toupper($1)} (file)`
- prints the length of a line
- `awk '{print length($0)}'(file)`
- Print specific fields with a head of the /etc/passwd file
- `ls -1hF Document/ | awk 'Begin {printf "%s\t%s\n", "Size" , "Name"} {print $5,"\t" ,$9}'`
- Print specific fields with a head of the file (etc/passwd in this case)
- `awk -F: 'BEGIN { printf "%s\t\t%s\n","User", "Shell" } {print $1,"\t",$7}' /etc/passwd`

cat

- **Definition**
- The cat command is used for displaying the contents of a file
- **Usage**
- `cat + option +file(s) to display`
- **Example**
- 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`
- Display the content of a file with line numbers
- `cat -n ~/Documents/todo.md`

cp

- **Definition**
- copies files/directories from a source to a destination
- **Usage**
- `cp+option+argument`
- **Example**
- to copy directories
- `cp -r directory copied destination`
- to copy a file
- `cp filename destination`
- to copy a directory with absolute path
- `cp -r ~/directory destination`

cut

- **Definition**
- used to extract specific section of each line of a file and display it on the screen
- **Usage**
- `cut + option + files(s)`
- **Example**
- Display a list of all the users in your system
- `cut -d ';' -f1 /etc/passwd`
- Display the list of all users in your system with their login shell
- `cut -d ';' -f1 /etc/passwd`
- cut a range of bytes per line
- `cut -b 1-5 usernames.txt`
- cut a file using a delimiter but changing the delimiter in the output
- `cut -d ';' -f1,7 --output-delimiter=' => ' /etc/passwd`
- cut the permissions from the output of ls
- `ls -1 | cut -d ' ' --complement -s -f1`

grep

- **Definition**
- grep is used to search text in given files and works line by line
- **Usage**
- `grep + option + search criteria +file(s)`
- **Example**
- search any line that contains the word dracula
- `grep 'dracula' ~/Documents/dracula.txt`
- search any word regardless of the case
- `grep -i`
- remove this word from search
- `grep -v`
- display line number for every line matched
- `grep -n`

- search and display the word which matches
- `grep -o`

head

- **Definition**
- head displays the first lines of a file or standard input
- **Usage**
- `head + options + file`
- **Example**
- show first 10 lines of a file
- `head filename.txt`
- show first n lines 5 lines
- `head -n 5 filename.txt`
- first 10 lines of many files
- `head file1.txt file2.txt`
- read from standard output
- `cat filename.txt | head -n 3`

man

- **Definition**
- an interface to the system reference manuals
- **Usage**
- `man` + `option`
- **Example**
- Display the manual page for the program
- `man ls`
- Display all intro manual pages
- `man -a`
- Format the manual page for bash
- `man -t`

ls

- **Definition**
- list contents from directories
- **Usage**
- `ls + option + file`
- **Example**
- list all files
- `ls (filename)`
- list long format
- `ls -a`
- list long + hidden
- `ls -la`
- list human readable size
- `ld -lh`

mkdir

- **Definition**
- mkdir is used for creating a single or multiple directories
- **Usage**
- `mkdir +option+argument`
- **Example**
- create a directory with a parent directory together
- `mkdir -p website/docs`
- to create a directory in the pwd
- `mkdir folder`
- create a directory in a different directory using absolute path
- `mkdir ~/wallpaper/forest`

mv

- **Definition**
- used for moving and renaming both files and directories
- **Usage**
- `mv+option`
- **Example**
- to move a file to a different directory
- `mv file /path/to/destination/`
- to rename a file
- `mv old_filename new_filename`
- to move multiple files
- `mv file file2`

tac

- **Definition**
- The tac command is used for displaying the content of a file in reverse order
- **Usage**
- `tac + option + file(s) to display`
- **Example**
- Display the content of a file located in the pwd
- `tac todo.md`
- Display the content of the file using absolute path
- `tac ~/Documents/todo.md`
- reverse a files contents line by line
- `tac filename.txt`

tail

- **Definition**
- Prints the last 10 lines of a file
- **Usage**
- `tail + option + file`

- **Example**
- 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`
- Display the last lines using wildcard
- `tail -n 1 *.csv *.py`
- Display a given number of lines of the output of a given command
- `ls -1 ~/cis106/ | tail -n 2`
- Display the name of a file in the output
- `tail -v -n 7 ~/Documents/Books/dracula.txt`
- Display a given number of bytes instead of lines
- `tail -c 50 ~/Documents/Books/dracula.txt`

touch

- **Definition**
- touch is used for creating files
- **Usage**
- `touch`+`argument`
- **Example**
- to create a single file
- `touch filename`
- to create multiple files
- `touch file1 file2 file3`
- create a file using absolute path
- `touch ~/Downloads/file`

tr

- **Definition**
- translate substitute squeeze o delete characters from input
- **Usage**
- `tr + option`
- **Example**
- convert lowercase to uppercase
- `echo "hello" | tr 'a-z' 'A-Z'`
- delete specific characters
- `echo "hello 123" | tr -d '0-9'`
- replace spaces with underscores
- `echo a b c | tr ''' _'`

- squeeze repeated characters
- `tr -d '\n' < file.txt`

tree

- **Definition**
- displays directory structure in a tree like format
- **Usage**
- `tree + option + directory`
- **Example**
- display current directory as a tree
- `tree`
- include hidden files
- `tree -a`
- display in human readable form
- `tree -h`
- list only directories
- `tree -d`