Note 8

awk

- AWK is a scripting language used for processing text files. AWK supports almost if not all of the features of a programming language. AWK can be tricky to learn however, every Linux user should be familiar with its basic functionality.

Usage

- `awk + aption + {awk command} + file + file to save

Examples

- print the first column of every line of a file.
- `awk '{print $1}' ~/Documents/Csv/cars.csv`
- print the last field of the /etc/passwd file
- `awk -F: '[print $NF]' /etc/passwd`
- print the first and 3 field with line number
- `awk -F: '{print NR,$1,$3}' /etc/passwd`

sed

- SED is stream editor that perform operations on files and standaed output. For instance it can search, find and replace, insert and deletion. By using SED you can edit files without opening them.

Usage

- `sed options + sed script + file`

Examples

- Replacing the number of occurrences of a pattern in a file
- `sed 's/pizza/rice/4' shopping-list.lst`
- replaceing string on a specific line number
- `sed '3 s/pizza/rice/' shopping-list.lst`
- Replancing string on a range of line
- `sed '1,2 s/pizza/rice/' shopping-list.lst`

less

- It's particularly useful for reading long file or outputs that don't fit entrirely in the terminal window.

Usage

- `less [option] filename`

Examples

- view a file
- `less file.txt`
- view command output combine
- `ls -1 | less`
- Display line numnber
- `less -N file.txt`

>

- > To redirect the output of a command to a file. Essentially saving the output of a command to a file

Usage

- `command output + > + file`

Examples

- save the output of command to a file
- `ls -lA ~ > all-files-in-home.txt`
- save the error generated by a command to a file
- `ls -la Downloads/ 2> error-of-ls`
- save the error and success to the same file
- `ls -la Downloads/ pictures &> allouput.txt`

>>

- > > to append (add) the output of a command to a file

Usage

- `ls -la >> filename`

Examples

- if we want to keep the old data, then we use
- `ls -la >> allmyfiles.lst`

|

- | to pass the output of a command to another

Usage

- `command_1 | command_2 | command_3 | ...| command_N`

Examples

- use grep to look for a string in a particular man page
- `man ls | grep "human-readable"`

- Display only the option of the of any command from its man page
- `man ls | grep "^[[:space:]]*[[:punct:]]"`
- Display only the 2nd line in file
- `head -2 file.lst | tail -1`