**Regex commands and VI shortcuts to use in VI EDITOR:**

**Rhea, Vikash, Aditi, Ananya, Drisana**

1. Remove multiple spaces

**:%s/\s\+/ /g**

% does the same thing as 1,$

Looks for one or more instances of blank spaces and replaces them with single space. -> To make sure that between words there’s only one space and not more than that.

1. Number each line

**:set nu** (:set nu! To remove numbering)

1. Removes special characters and only keeps alphanumeric characters in a file

**:1,$s/[[:punct:]]//g**

1. Sorting lines alphabetically

If there’s a file with like names or something, you can sort them in alphabetical order by using

**:sort**

Other sorting options:

: sort n does numerical sorting

:1,$sort! Does backwards z-a sorting

1. Delete all numbers in a text

**:1,$s/\d//g or :%s/[[:digit:]]//g**

Finds digits (\d) and deletes them

1. Convert a list to comma separated value format

Ex: like if there’s a file with names like

Ananya

Aditi

Rhea

Make it into Ananya, Aditi, Rhea

**:1,$s/\n/, /g**

1. Substitute new line with “, “ to make it into csv format.

Also, can make csv format into a list by replacing comma with a newline using:

**:1,$s/, /\r/g**

1. Capitalise Words

In a sentence, capitalises the first letter of the word after a full stop.

Ex: “Hello. world” - becomes “Hello. World”

**:1,$s/\.\s\*\([a-z]\)/. \U\1/g**

1,$ - from first to last line

s/ - substitute

\s\* - matches zero or more occurrences of whitespace characters

\([a-z]\) - matches a single lowercase character

Together \s\*\([a-z]\) -> matches zero or more occurrences of whitespace followed by lowercase character

Replaces that with “. “ + uppercased next character

\u - makes a character uppercase

\1 - referring to what was put in the variable \([a-z]\)

1. Capitalise first letter of every word

**:1,$s/\<[[:alpha:]]/\u&/g**

Substitutes one lowercase character

\< - start of a word

& - acts as a placeholder to hold the character that \u is working on

1. Changing the colorscheme (changes vi editor’s background and foreground color)

**:colorscheme desert** →default

**:colorscheme peachpuff/blue/ron**

1. Changing font color (all of the text)

**:highlight Normal ctermfg= white/blue/red**

(:highlight begins the command for applying a highlight rule)

(Normal defines the highlight group to apply the rule to)

ctermfg→affects text color

ctermbg→affects background color

:hi →shorthand for highlight

1. Underlining words

**:match Underlined /anyword\|anyword/**

**HIGHLIGHT GROUPS: Normal, Search, Underline- they change the appearance of the pattern matches in different ways.**

1. Highlighting words
2. Using :match

**:match Search /anyword/**

1. Using :highlight (gives option to change color of highlight)

**:highlight group1 ctermbg=red** (press enter)

**:match group1 /anyword/**

Assigning red to group 1 words,

**:match group2 /\(in\|world\)/** →mutiple words

**:match none** → removes all highlights

14. Abbreviating words

Example - whenever Isc is typed, it is automatically changes to Information Science

**:ab Isc Information Science**

:abbreviate / :aband :unabbreviate / :una

15. Gives total word count

**:w !wc -w**

16.

vi filename (open/edit file)

:w (save and continue editing)

:q / :q!(quit and don’t save changes)

:wq (save and quit)

**CMD MODE VS INSERT MODE:**

<esc> <i> (before cursor)

<a> (after cursor)

<o> (open line below

current line)

<r> (replace the current character)<s> (substitutes character)

**SIMPLE EDITS:**

<numeric arg> <cursor cmd> like ‘5j’ (cursor goes down 5 lines)

**Cursor/motion commands:-**

‘h’ (left), ‘j’ (down), ‘k’ (up), ‘l’ (right)

‘$’ (end of line), ‘0’ (start of line)

‘w’ (one word forward (beginning of word)), ‘b’ (one word back(beginning of word))

‘e’ (word forward (to the end of the word))

‘gg’ (start of file)

<shift> + ‘g’ (end of the file)

<line number> + ‘G’ (goes to that number line)

‘)’ → goes to the start of the next sentence (open bracket vice versa)

**More cursor cmds:-**

‘z.’ (moves the line that the cursor is on to the center of screen)

‘z’ + <enter> (moves the line that the cursor is on to the top of screen)

‘z-’ (moves the line that the cursor is on to the bottom of screen)

‘n|’ where n takes a column value (moves cursor n columns/characters)

**Delete:-**

‘x’ (deletes character at cursor)

d is delete =>

‘dw’ (delete word), ‘dd’ (delete line), ‘d$’ (delete to end of line) , ‘d0’ (delete to beginning of line)

‘rm <file>’ (delete file)

**Changing Text:-**

c modifies existing text till <esc> => deletes the line the cursor is at and switches to insert mode

‘cw’ (changes words), ‘cc’ (changes lines)

**Replacing Characters:-**

‘r’ replaces a single character

‘R’ puts editor in overstrike mode, replacing till <esc>

**Moving and Copying text:-**

‘yy’ / ‘Y’ copies one line (/n to /n)

‘p’ paste line on the next line of cursor.

**Searches:-**

*movements=>*

‘/pattern’ (forward search for specific pattern - (finds the next one))-->after the cursor

‘?pattern’ (backward search for specific pattern- (finds the next one))-->before the cursor

‘N’ to repeat action backward; ‘n’ to repeat action forward

*current line searches=>*

*‘*f + <char>’ (moves to next occurrence of char)

*‘*F + <char>’ (moves to previous occurrence of char)

*‘*t + <char>’ (moves to char after next occurrence of char)

*‘*T + <char>’ (moves to char before next occurrence of char)

**Buffers in vi:- temp storage of text**

Unnamed buffers => used for yank and delete operations.

‘yy’/’Y’ uses unnamed buffers as a temporary store till ‘p’ is applied

Named buffers => identified by letters a-z

‘ “xyy ‘ / ‘ “xY ‘ (x is the name of the buffer where the line is yanked/yy/Y)

‘ “xdd ‘ (x is the name of the buffer where the line is yanked/yy/Y) →can be used across vi files

‘ “xp ‘ (takes the saved stuff in x buffer and adds to the next line)

Multiple line storage in a named buffer uses capitals ‘ “Xyy ‘ which opens buffer ‘x’ in append mode

*Default/Numbered register 0 contains the text from the most recent yank command.*

*Default/Numbered register 1 contains the text deleted by the most recent delete or change command*

**Indent:-**

**>>**

**<<**

**17. Alignment**

%right - makes the text right aligned

%left - makes the text left aligned

%centre makes the text centre aligned

**18. Working with multiple Files, Buffers, Tabs, and Windows:**

**Multiple files:**

**:e** -> Load new file in buffer for editing

**:edit** -> Same as :e

**:e <fileName> or :edit <fileName>** -> List the files for editing from current directory

**Multiple buffers:**

**:badd <fileName>** -> Add file into new buffer

**:bN** -> Switch to Nth buffer. For instance to switch to 3rd buffer use :b3

**:bnext** -> Move to the next buffer in buffer list

**:bprevious** -> Move to the previous buffer in buffer list

**:buffers** -> List all buffers

**:bfirst** -> Move to the first buffer

**:blast** -> Move to the last buffer

**:ball** -> Load all buffers

**Multiple tabs:**

**:tabnew** -> Open new tab

**:tabnew <file>** -> Open new file in tab

**:q** -> Close current tab

**:tabnext** -> Move to the next tab

**:tabprevious** -> Move to the previous tab

**:tabfirst** -> Move to the first tab

**:tablast** -> Move to the last tab

**Multiple windows:**

**:new <filenName>** -> Opens file in new window in same working windows

**Ctrl + ww** -> moves to next window

**:q** -> Close current window

**19. Bookmark Feature:**

We have some bookmarks which are default like numbers from 0 - 9 and some special char.

**Create bookmark (using only single char a - z or A - Z):**

**m{bookmark-name} [eg. ma]** -> Creates bookmark

**Jump to bookmark:**

**`{bookmark-name} [eg. `a]** -> Moves to the exact location of bookmark

**List all bookmarks:**

**:marks** -> List all bookmarks

**Delete bookmark:**

**:delmarks a** -> Deletes bookmark of char a