

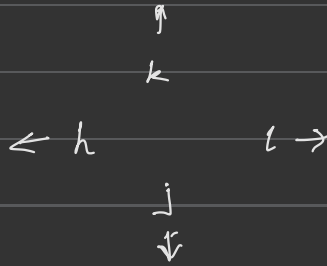
Video -4

Normal mode

Insert Mode

Command Line Mode

## Video - 8



ctrl + f → Page forward      ctrl + b - Page up.

W; W → move 1 word      b; B - Move word backward

Z + Enter → Reposition screen

(Zero) O → Move to start line      ^ - Start of line word.

\$ - End of line

gg - top of file

G - Bottom of page

2gg; 2G; ; 2 + Enter - goto end line in file

ctrl + g - stat of file

: set ruler - Row & Column info will show on right bottom

: set no ruler ; set ruler!

## Video - 11

x - delete current character

X - delete before character

d w - delete word. eg) dl, di, d\$  
operator ↗ motion

cll - delete line

[count] operator [count] 1 motion 3

3cll - delete 3 lines

5dw - delete 5 words

[count] operator [count] 1 motion 3

3d5w - 3 times delete 5 words = 15 words

• (dot) - repeat the last command.

:q - quit vim

q! - force quit      :!ls - Execute last command.

set ruler! - toggle on/off

## Video - 17 Cut, Copy, Paste

Cut & Copy text stored in Register in Vim

dd - Cut a line & store it in unnamed register or default register

p - paste below cursor line (p - put command)

P - paste Above

y - Copy or Yank and store it to unnamed register

yy - Copy Line

u - Undo change

Ctrl + r - Redo change

## Video - 18

### Types of Register

\*] Numbered — "1"

\*] Unnamed — "0" "1" ... "9"

\*] Named — "a", "b", "c" ... "z"

Registers are started with " Double quote

Unnamed register hold text from d, c, s, x and y operators

c - change s - substitute

"0 - hold recent text of yank

"1 - hold recent text of delete (d) or change (c)

Numbered register shift with each d or c

2 → 3, ... 8 → 9

"\_ - Black hole register

:reg Enter - to view content in register

"0 p - to paste the content from register

"\_ dd - delete line but will not affect register.

"A yy - append the text to append to named reg

"a yy - Delete old text place new text

:reg z - show content in particular register

[count] [register] operator

or

[register] [count] operator

2 "a p - paste content in a reg 2 times

## Video - 21

- (ai) I - first non blank character and Insert Mode
- (li) a - after current cursor go insert mode
- (fi) A - go end & insert mode  
↓  
append
- (ji) o - next line & insert mode
- (ki) O - above line & insert mode

8i <edit> <esc> - repeat the text 8 times.

8O <edit> <esc> - repeat in vertical line

R - replace Mode (overwrite existing text)

r - replace one character.

[register] c [motion]

cw - change word

cat → cat.  
auto delete <sup>v</sup> unlike replace.

delete the word and insert mode.

c\$ - delete current cursor to end of line and  
insert mode

3 cc - delete line and insert mode

v - Switch the case of current cursor

g v w - Switch casing of word.

g v \$ ; g v v - Switch casing in line

vv, dd, yy - same character twice operate on whole line

g V w - Upper case word

g V V

g u w - Lower Case word

J - \$ backspace space

g J - \$ backspace (no space)

z J



## Video - 22

f b - move cursor to first b in line forward

F b - search backward.

(Comma) , - search in opposite ; (semicolon) - next

t b - move cursor the letter before b

/ search - forward search ? search - backward search

n ; N - find next occurrence

: set is? - check the option is true or false

: set is / : set inc search - set incremental search

cursor moves while searching word itself.

: set hls - highlight the search

# ; \* - get the word and search word itself

d / search - to delete from current to search word

## Video-22

- : s / find / replace / - find & Replace in that line, 1st
- : s / find / replace / g - find & Replace all occurances in line
- : [range] s / find / replace / [flag]
- : 1 s - first line only
- : 1, s s - 1st to s<sup>th</sup> line only
- : . s - current line
- : \$ s - end line
- : ., \$ s - current to last line
- : % s - all line [1, \$]
- : / from /, / to / s / find / replace / g
- : s # find # replace # g
- : set nu - turn on line number

## Video - 28

daw — delete full word which cursor present.  
↓

text object (a word)

diw — delete full word except last whitespace  
↓  
delete inner word

{operator} {a} {object} {operator} {i} {object}

das; dis — delete a sentence

dap; dip — delete a paragraph

eg)  $a = [1, 2, 3]$

di[; di] — delete things inside brackets

da[; da] ; dab; dib

eg)  $\langle \text{hello} \rangle$

ci>; ci< — delete things inside  $\langle \rangle$  & go insert mode

eg)  $\langle p \rangle$  hello there  $\langle /p \rangle$

cit — delete things inside the tag & go insert mode  
↓  
tag option

da1; da9

daB  $\leftarrow$  Block

ci1; ci9 - delete current position and put 1 9

di"; da" - "hello there"

## Video - 31

### Macro

qa - record series of steps in a register

q/ - to exit macro

:reg a - to view macro

@a - replay the macro

@@ - replay the recent macro

5@a - replay 5 times the macro

q@a - Append to register

:rs, s2 normal @a - rs to s2 execute macro

"ap - paste the content to vim

"ay& - save the content to a reg

- viminfo file - store custom macros if overwritten Macro will be deleted

- vimrc - Save macro

let @b = 'Contents of Macro'

ctrl+v esc - to insert special buttons to custom  
(create macro eg) esc, backspace, enter

## Video - 34

### Visual Mode Types

Character wise	v
Line wise	V
block wise	Ctrl+v

### Some Command to use in Visual Mode

v ; c ; d ; y ; r ; x ; I ; A ; J ; u ; U ;

> - shift right ; < - shift left

O ; o - to move opposite end of selection

gv - select previously selected

: set shiftwidth=4

: set tabstop=8

: set expandtab

: set list - to easily spot tab character

### Select Visual and go Command mode

: '<', '>' center - to align center

center 40 - to align 40 character in center

left ; left 5 ; right ; right 50

## Video - 28 Vim RC

rc = run command

: version - to view all the vimrc options set

set ruler is same as : set ruler

: set - to view all non default options

: h hls - help in vim

: set history & - Reset to default value

: h option-list - to view all the set options

: options - list of options with description

" Comment in vim rc file - to comment in Vimrc

: set slowcmd - slow in couple command in bottom

: set wildmenu

## Video - 39

Set Macro to Keys.

map key keystrokes

eg) map <F2> iHello there <CR> <ESC> - to trigger F<sub>2</sub>

map <leader> w :w! <CR> - to trigger \w

let mapleader = " , " - change leader key

:map - to view all mappings

vmap - Works only in Visual Mode

nmap - Works only in Normal Mode



## Video - 42 Buffer

Temp memory area data is stored while its processed or transferred.

- :e file name - open file in vim (edit Command)
- :buffers;:ls - list all open file in vim
- :buffer 2;:b 2 - open file from buffer
- :b file.txt;:b cttld
- :b next;bn - goto next buffer
- :bprevious;bp - goto previous buffer
- :bn! - not save and switch
- :set hidden
- :qall! - exit all file not save to file
- :wall - exit all file save to file
- :badl - add file to buffer but not switch
- :bd;:bdls - remove file from memory
- :1,3 bd; :% bd
- :**butdo** set nu - perform action on all buffers
- :bufdo %s/find/replace/g | w  
Command separator ↩ ↪ write to file as the change done to current file

## Video-45

:E - file explorer window

vim is a view of a buffer

when vim start with no file name; it loads unnamed buffer in that window

ctrl+w ctrl+w - switch between windows

:sp; :split - split window horizontally

ctrl+w s ;:sp filename

:vs - split vertically

ctrl+W V

ctrl+w q = :q

:on; :only; ctrl+w O - to close other and open only that window

ctrl+w h|j|k|l - to switch windows

ctrl+w +; ctrl+w -; ctrl+w >; ctrl+w <

ctrl+w \_; ctrl+w |; ctrl+w = - change size of window

ctrl+w r - rotate window eg) 1 to 2, 2 to 3 window

ctrl+w H|K|J|L

:ball; :ba - open all buffer in window

:windo %s/#|+/g - execute command on open windows

