

unzip -d vimclass vimclass.zip

++

unzip -d vimclass vimclass.zip

Vim is an advanced text editor

Vim = Vi + Improved

~~Vim is a text editor~~

Vim is Amazingly powerful

Vim's knowledge is transferable

cmd D-W → delete word

Once you start to think in vim you'll be able to easily start stringing together vim is just like you speak

Like language vim has the concept of verbs nouns or objects and adjectives for example to delete a word you type the command D-W. with D is an action and an object. if you want to perform a different action on that object then use a different action command for example to change text of word you would use CW to delete everything contained within a pair of quotes. You can use the command DI" quotation mark this means delete inside the quotes.

You have an action which is delete an adjective inside and an object. Quotation mark if you want to delete three words for example you use the command D3W. D 3W with literally means to delete

9) Vim is thoroughly documented

10) Vim is Fun!

Vim

+ Modes +

→ Normal mode = command Mode
→ Insert mode = ^{cmd} ^{case sensit} ~~edit~~ ~~loc~~

↳ i
↳ escape

→ Line mode = command line mode or keyboard mode

↳ :
↳ save file
↳ go to location in file
↳ : escape → normal

Vim Modes

↳ Normal / Command Mode
↳ Insert Mode
↳ Line Mode
↳ there are other modes, too

+ + Vim Quickstart + +

Vim ↵ // will start vim in Normal mode

exit : q! ↵ // quite Vim

// create file

Vim abc123 ↵ // new file

i // insert mode

escape // normal mode

: // cmd mode

: wq // write and quite

Vim myday.txt ↵ // create myday.txt file

i // go to insert mode

~~exit~~ ~~insert~~ // go to normal mode

esc ↵ // go to line mode

: // go to cmd mode

wq // write and quite

- 1: normal mode
- 2: insert mode
- 3: Line mode

mode	command	exercise	walk through	test	other
1	Vim myday.txt ↵				create new file
2	i				
2				today I woke up fed	
	↵			1. Family	next line
	↵			2. Health	
q	escape ↵			3. Sunshine	
1	esc ↵				
3	:				write and quit
	wq			save	
	Vim my day.txt				
	: q!			exist without change	

* Essential File Navigation *

⇒ open an existing file

```
cd Downloads
cd vim class
Vim nav.txt ↵
// normal mode
```

~~pressing end j to move the cursor~~

~~You can use this~~

To move down a line, press "j".

To move up a line, press "k".

To move to the right, press "l".

To move to the Left, press "h".

← H J K L →



ctrl -f page down // forward

ctrl -b page up // backward

@. ~ // end file

let's say that you want to move right one word

Just press \Rightarrow w

word in vim is sequence of non blank characters

\Rightarrow W ignores punctuation and uses white space as word boundaries to back up by word.

to back by word we'll use b key b, B same as w, W ignores punctuation

\Leftarrow z \Leftarrow Keep the cursor in the current position but move text up on your screen

0 // number zero jump to the begin of line
shift ^ // ^ represent the g begin in regular expression
shift \$ // end of line // same logic

go to line cmd

1st gg
2nd shift/g \Rightarrow G

go to a specific line the type that line number followed by either version of the go to line command

2 gg / gg | well if you do not specify a line number gg defaults to 1st line in the file

G / well if you do not specify a line number G default to last line in file

line number
\$ \leftarrow end of file

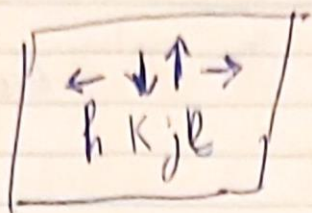
\Leftarrow : 1 \Leftarrow Jump to first line

if you know how many lines are in file you can press
G Ctrl & G

this displays the file name in the file status

You can even get more detailed information g ctrl g +

↳ character counts



sl status will display: line, column

: sl status ←

↳ turn on the rule

↳ : sl number ← // reverse

↳ : sl status ! // toggle

→ one word : w

→ one word and include punctuation : W

← one word : b

← one word and include punctuation : B

z ← : returns to cursor in current position
but move text up on the screen

0 // number zero jump to the begin of line
shift ^ // also jump to the begin of line
shift \$ // jump to the end of line

2 gg ⇒ goto line 2

2 shift g ⇒ 2 G ⇒ goto line 2

gg ⇒ goto 1st line in the file

G ⇒ goto Last line in the file

: 1 ← jump to 1. line ⇒ : line-number

: \$ ← jump to Last line in file

Ctrl+G ⇒ displays the file name and file status

example : "nav.txt" line 74 of 74 ~ 100% -- col 1

: "nav.txt"

cd Downloads

cd Vimclass

Vim mac.txt

j // x 10 // move down 10 lines

k // x 4 // move up 4 lines

| // x 8 // move

j // if cursor move down try imagine the same position of column

:\$ → Last line

:q! ← delete

+

+

Deleting text and thinking in Vim

~~cd Downloads~~

the simple of the deletion cmd is lowercase x

↳ delete the character at your current cursor position ~~on the~~

uppercase X ⇒ delete the character right before or to the left of your cursor

shift+x

to delete word ⇒ DW

operation {motion}

div

d = the delete operation

w = the word motion

dL delete character under the cursor

x is short cut to dL

X " " " dh

dJ that deletes the current line you're on and line below

dK like wise dK will delete the current line you're on and the above it because that is where the ~~cursor~~ cursor motion take you

so to delete all the way to the beginning of the line
from your current cursor position & just type the
D 0

also - You would use dollar sign so to delete to
the end of the line use the \$ sign \Rightarrow d\$

D is short cut to d\$

dd \Rightarrow delete the current line
3dd \Rightarrow delete 3 lines

[count] operation {motion}

5dw

5 = the count / how many times to repeat
dw = the command (delete word)

3w = Repeat word motion 3 times move 3 word
d3w = Delete the 3w motion. \Rightarrow [count] operation (count) {motion}
2d3w = Delete the 3w motion 2 times

DD cmd \Rightarrow DoD command repeat the previously
issued command to execute the doD cmd
just type of period

:w \leftarrow \Rightarrow // save and ~~return~~ return to normal mode

:q \leftarrow \Rightarrow // to quite

! \Rightarrow not an action