

Lesson 1

h ← j ↓ k ↑ l → cursor move either the arrow or hjkl keys

vim FILENAME <ENTER> To start Vim from shell

<ESC> :q! <ENTER> to trash all

<ESC> :wq <ENTER> changes. to save the changes.

x To delete the character at the cursor

i INSERT MODE before the cursor

A INSERT MODE after the line

Lesson 2

Lesson 3

p put the deleted text **AFTER** the cursor

r [char] replace the character under the cursor

c change from the cursor where the motion take you. Eg. c\$ ce

cw

c [number] motion it is the format for changing

Lesson 4

CTRL-G show your location and the file status

G move to the end of the file

gg move to that line

number

go to the first line

/foo <ENTER> search foo **FORWARD** of the phrase

?foo <ENTER> search foo **BACKWARD** of the phrase

n find the **next**

CTRL-O occurrence

CTRL-I find the **previous** occurrence
Takes you to the **PREVIOUS** position
Takes you to the **NEWER** position

Lesson 4 (cont)

% type it on (,) , [,] , { or } and goes to its match

:s/old/new substitute **first** old for new

:#, #s/old/new/g substitute all old for new **in the line**

:%s/old/new/g substitute all matches between the lines #

:%s/old/new/g substitute all matches **in the file** to **check** each time add c

Lesson 5

:!command execute an external command
Eg. win: !dir or in Unix: !ls

:w FILENAME write the current VIM file to the disk with the name FILENAME

v motion :w FILENAME saves visually selected lines in the file FILENAME.

:r FILENAME retrieves disk file FILENAME and puts belong cursor position

:r !command reads the output command and puts belong the cursor Eg. win: !dir or in Unix: !ls

dw del **from**
the
cursor to
the **next**
word

d\$ del **from**
the
cursor to
the **end**
of line

dd del
whole
line

2w repeat
the
motion
with a
number

operator	[number]	motion	where:
operator			- is what
[number]			to do d
motion			-
			[optional]
			repeat
			motion 2
			- move
			to the
			text to
			operate
			2w

0 move to
the **start**
of the
line

u To undo
U **previus**
action
Ctrl -R To undo
all the
actions
on a line
To undo
the
undo's



By **im_Tavo** (gamejia90)
cheatography.com/gamejia90/
www.github.com/Athesto

Published 3rd July, 2020.
Last updated 6th May, 2020.
Page 1 of 3.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Lesson 6

- o insert a line **BELOW** the cursor
- O and start INSERT MODE.
Open a line **ABOVE** the cursor

- a insert text after the **cursor**
- A insert text after the **end of the line**

- e move to the **end of a word**

- y copy the text and paste it

- R Enter in REPLACE MODE until <ESC> is pressed

`:set foo` sets the option `foo`
Eg. `:set ic`

`ic` or `ignorecase` ignore upper/lower case when
`is` or `insearch` searching
`hls` or `hlsearch` show partial matches for a search phrase
highlight all matching phrases.

`:set nofoo` preset `no` to switch an option **off**
Eg. `:set noic`

Lesson 7

<HELP KEY> Vim has a comprehensive on-line help system. To get started, try one of these three

`:help <ENTER>` to find help on `cmd`

`CTRL-W CTRL-W` to jump to another window

`:q` to close help window

`~/ .vmrc` a startup script to keep your preferred settings.

`:cmd CTRL-d` to see possible completions,

`:cmd <TAB>` to autocomplete



By **im_Tavo** (gamejia90)
cheatography.com/gamejia90/
www.github.com/Athesto

Published 3rd July, 2020.
Last updated 6th May, 2020.
Page 2 of 3.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>