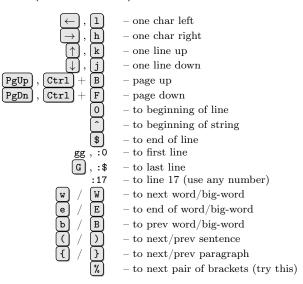
Vim Cheat Sheet

Walk (normal mode)

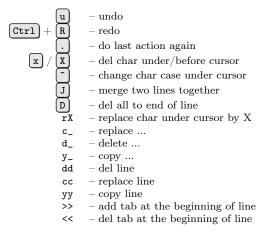


Walk marks (normal mode)

- set mark 'a' (use any char) - go to mark 'a' '. - go to last edit symbol :delm a - del mark 'a'

:delm! - del all marks

Commands (normal mode)



Here is the magic:

 replace next 3 words 40k - go 40 lines up

Insert mode

a - go edit before/after cursor Ī A - go edit beginning/end of text o O - add line under/upper and go edit Ctrl - while insert, switch input lang! Arrows - move cursor Del – del char under cursor Esc - leave insert mode (back to normal)

Find (normal mode)

- to G char in line (use any char) /substring - find substring in text - find word under cursor N - find prev/next

Find and Replace (normal mode)

:[%]s/<what>/<with>/[<mod>] wher: % - replace in all lines <mod> - replacement modifiers For example:

- replace 1st foo with bar in line :s/foo/bar/ :s/foo/bar/g - replace all foo with bar in line :%s/foo/bar/g - replace all foo with bar in all lines Modifiers:

- search all entries in a line

ask before replace

- case insensitive

case sensitive replacement

Modifiers can be combined:

- ignoring case replace all foo with bar :%s/foo/bar/gic in all lines and ask before replacement

Copy and Paste

First case is to use $c_/d_/y_$ commands and then paste data under cursor using [p] / [P] Second case is to use select mode:

1. [v] – go to select mode OR $\boxed{\mathtt{Ctrl}} + \boxed{\mathtt{v}} - \mathtt{box} \ \mathtt{select}.$

2. Use 'walk' keys to select a piece of a text.

3. Press y / d / c to copy/delete/replace that text. r and also work but not copy data to buffer.

4. Use p / P to paste data from buffer.

Named Registers

— named register's qualifier. Next letter after " specifies a name of register. For example: "add - remove curent line and put it to register 'a' copy 3 chars to register 'b' - paste from register 'c' Special named registers: :, ., %, # - read-only registers expression register

- selection and drop registers - last search pattern register

Macros (normal mode)

Macros – a sequence of actions that can be repeated.

1. qx - start record macro with name 'x'.Any char can be used.

2. Do something your need to repeat (be carefull).

3. Press [q] to stop recording.

4. Call macro:

Qx - call macro 'x' one time 100x - call macro 'x' ten times

Session (normal mode)

:q - force quit :!q

:w/:w filename - save/save to file 'filename'

 save and quit : x

:mks filename - save session to file 'filename'.

Use vim -S filename to restore a session.

:e ++enc=ENC set current buffer encoding to ENC

Windows and Tabs

:sp filename - split current window horizontally by two and show file 'filename' in second window

- same as previous but vertically :vs filename $[\mathtt{Ctrl}] + [\mathtt{w}]$ - move cursor from window to window

and Arrows :tabnew file - open file in new tab

 go to next/prev tab gt/gT :windo ...

- exec command in all windows :tabdo ... - exec command in all tabs

Spelling

:set spell/:set nospell - turn on/off spelling marks - turn on spelling using 'ru' lang :set spell spl=ru - open spell menu for a word under cursor

Folding

ss

zo/zc open/close fold zR/zM- open/close all folds Variables: :set foldlevel=... - integer value :set foldmethod=... - syntax, indent or manual For manual foldmethod:

1. v - select a part of text for folding.

2. **zf** – mark selected text as folding.

Useful variables

:set var - set boolean to True OR print current value :set novar - set boolean var to False set value to foo :set var=foo Variables: - boolean, search highlight on/off hls - integer, number of spaces to show TAB ts - boolean, copy current indent for new line ai - boolean, fixes ugly insert into vim-in-terminal paste - boolean, line numbers on/off nu - boolean, wrap long lines on/off wrap - integer, number of columns to scroll horizontally