

VIM QUICK REFERENCE CARD

Normal mode

A.....append at end of line
a.....append after the cursor
B, $\langle ctrl \rangle + \langle left \rangle$space-separated word backward
b, $\langle shift \rangle + \langle left \rangle$word backward
C...delete from cursor to end of line, start insert mode
cm.....delete *m* text
cc.....delete line, start insert mode
cw.....delete word, start insert mode
D.....delete from cursor to end of line
dd.....delete line
dw.....delete word
E.....forward to end of space-separated word
e.....forward to end of word
Fc.....move cursor left to char *c*
fc.....move cursor right to char *c*
G, $\langle ctrl \rangle + \langle end \rangle$go to end of file
gg, $\langle ctrl \rangle + \langle home \rangle$go to beginning of file
[*n*]gg, :[*n*].....go to line number *n*
H.....first line of window
h, $\langle left \rangle$move left
I.....insert text before first non-blank
i.....insert text
J, gJ.....join lines
j, $\langle down \rangle$move down
k, $\langle up \rangle$move up
L.....last line of window
l, $\langle right \rangle$move right
M.....middle line of window
mm.....create mark *m* $\in \{0..9a..zA..Z\}$
'*m*.....jump to mark *m* $\in \{0..9a..zA..Z\}$
n.....repeat last search /?
N.....repeat last search /? in oposite direction
O.....begin new line above cursor
o.....begin new line under cursor
P.....put text before cursor
p.....put text after cursor
qc.....record macro *c* $\in \{0..9a..zA..Z\}$
q.....stop recording
R.....enter replace mode
rc.....replace char *c* under cursor
S.....delete line, start insert mode

s.....delete char, start insert mode
T/tc.....see F/f and place cursor left/right to char *c*
U.....undo latest changes on line
u.....undo latest changes
V, v, $\langle ctrl \rangle + V$start visual per character, linewise, blockwise
W, $\langle ctrl \rangle + \langle right \rangle$space-separated word forward
w, $\langle shift \rangle + \langle right \rangle$word forward
X.....delete character before cursor
x.....delete character under cursor
Y, yy.....yank line
ym.....yank motion text *m*
O, $\langle home \rangle$go to beginning of line
![!]pipe.....pipe selected region through external program
~.....change char's case
@[0-0a-zA-Z]-q.....execute macro {char}
#.....search forward word nearest to the cursor
\$, $\langle end \rangle$go to end of line
%.....match pair () { } []
^.....go to first char of line
*.....search backward word nearest to the cursor
:command..command for line selection, e.g. read/write filename
;.....repeat f, F, t, T
,.....repeat f, F, t, T in oposite direction
"x.....use register *x* $\in \{0..9a..zA..Z\}$ with y, d, p, etc...
.....repeat last change
/p.....search forward pattern *p*
?p.....search backward pattern *p*
 $\langle ctrl \rangle + A$increment number under cursor
 $\langle ctrl \rangle + B$, $\langle page up \rangle$, $\langle shift \rangle + \langle up \rangle$scroll up one screen
 $\langle ctrl \rangle + D$scroll down half a screen
 $\langle ctrl \rangle + E$scroll down one line
 $\langle ctrl \rangle + F$, $\langle page down \rangle$, $\langle shift \rangle + \langle down \rangle$scroll down one screen
 $\langle ctrl \rangle + G$display status line
 $\langle ctrl \rangle + I$go to newer cursor position in jump list
 $\langle ctrl \rangle + L$redraw the screen
 $\langle ctrl \rangle + O$go to older cursor position in jump list
 $\langle ctrl \rangle + R$redo one change which was undone
 $\langle ctrl \rangle + U$scroll up half a screen
 $\langle ctrl \rangle + X$decrement number under cursor
 $\langle ctrl \rangle + Y$scroll up one line

$\langle ctrl \rangle + \langle page up \rangle / \langle page down \rangle$next/previous tab

Insert mode

$\langle ctrl \rangle + V \langle non-digit \rangle$insert next $\langle non-digit \rangle$ key literally, e.g. insert Tab in expand tab mode
 $\langle ctrl \rangle + I$, $\langle ctrl \rangle + O$see above

Window

$\langle ctrl \rangle + W W$, $\langle ctrl \rangle + W w$...switch to the previous/next window
 $\langle ctrl \rangle + W c$close window

Visual mode

\$.....in blockwise mode select all lines up to end
~, U, u.....change case, uppercase, lowercase
As $\langle esc \rangle$append string *s* to a blockwise selection
a<, i<.....select text in <>
a(, i(.....select text in ()
a[, i[.....select text in []
a{, i{.....select text in {}
D, d, x.....delete selection, see D, d
C, c.....change selection, see C, c
Is $\langle esc \rangle$...with a blockwise selection will insert string *s*
gv.....reselect the old visual
O, o.....move cursor to the other end of selection
y.....yank selection

Search and replace

/[*c*][*b*]p.....search forward pattern *p*
?[*c*][*b*]p.....search backward pattern *p*
c.....\c ignore, \C match case
b.....\< begin, \> end of word (for subst. too)
N, n, ?, *, #.....see above
[*r*]s/[$\%V$]p/s/[*f*].....replace pattern *p* with string *s*
r.....range, % whole document, '<', '>' inside visual area
f.....flags, c confirm, g all in line, i ignore case
p.....string or regular expression (like in sed)
 $\%V$match inside the visual area
s.....replacement string (matched subexpression)
\1..\9, matched pattern)

Search examples

`%s/\r//g` remove DOS-like newlines
`%s/\([a-z]\)/\U\1\E/g` .. convert the first character of line to uppercase
`%s/\<his\>/her/g` whole word substitution, i.e. doesn't produce ther from this
`%s/(good\nice)/awesome/g` translate any occurrence of either good or nice
`%s/\<(good\nice)\>/awesome/g` ... translate word good or nice
`%s/awesome/wonderful/gc` interactive find and replace
`%s/^\n=line(".")\u"."/g` substitute all lines with its line number
`%s!_HOME_!\= expand($HOME)!g` ... substitute special character with its equivalent value
4, `$s/\d\+/\=submatch(0)+1/` increment the first digit on lines 4 to end of file
`%s/\.\s*\w/\=toupper(submatch(0))/g` ... substitute the sentence beginnings with upper case
`/fred\|joe` search for fred or joe
`/fred\&.*joe` ... search for fred and joe in any order
`/\<fred\>` search for fred but not alfred or frederick
`/\<d{ 4}\>` search for exactly 4 digits
`/\([0-9]\|\^)\%.*%` ... search for absence of a digit or beginning of line
`/^\n{3}` find 3 empty lines
`/\(^str.*\n\){3}` find 3 successive lines starting with str
`/fred\s*joe` search for fred followed by joe, including newline
`/fred\(\.\)*joe` search for fred followed by joe anywhere in file
`?http://www.vim.org/` search fo a URL without backslashing, first seach *backwards*
`/^\{-}home.\{-}\zshome` match only the 2nd occurrence in a line of home
`%s/home.\{-}\zshome/alone` .. substitute only the 2nd occurrence of home in any line
`d/fred/` delete until fred
`y/fred/` yank until fred
`c/fred/e` change until fred end

For programmers

`:set [no]expandtab` use tabs or spaces
`:set tabstop=n` set tab size to n
`:retab[!]` change tabs to spaces or back
`:syntax on/off` syntax highlighting
`:set [no]autoindent, set ai` .. set autoindent mode
`:set [no]wrap` [no] wrap lines longer than the window width

Tabs

`:tabnew [file]` open/edit file in new tab
`:tabclose` close tab
`:tabm[ove] 0/N/\u` .. move tab to first/N/last position

My vimrc

```
syntax on
set nowrap
set list
set expandtab
set ai
"visible tab and end of line
set lcs=tab:→,eol:·
"remove all whitespaces char from all eol
map <s-del> :%s/\s*$//<cr>
"open quickfix window in new tab
map <c-f11> :tab copen<cr>
"how to switch between buff.(go to error line)
set switchbuf=useopen,usetab
"open new empty tab
map <c-tab> :tabnew<cr>
"next/prev error
map <f11> :cp<cr>
map <f12> :cn<cr>
"switch to next/prev tab more comfortable
map <c-up> <c-pageup>
imap <c-up> <c-pageup>
map <c-down> <c-pagedown>
imap <c-down> <c-pagedown>
```

Autors

Petr Olivka, petr.olivka@vsb.cz, <http://poli.cs.vsb.cz>
Roman Szturc, roman.szturc@gmail.com

This card has been compiled from several Vim sources, especially Vim Quick Reference Card by Laurent Grégoire, Best of Vim Tips by David Rayner (<http://www.rayninfo.co.uk/vimtips.html>), 12 Powerful Find and Replace Examples by Sathiya Moorthy (<http://www.thegeekstuff.com/2009/04/vi-vim-editor-search-and-replace-examples>) .

Last modification 28.3.2013