

VIM QUICK REFERENCE CARD

Basic movement

character left, right; line up, down **h l k j**
word/token left, right **b w**
end of word/token left, right **ge e**
beginning of previous, next section **{ }**
beginning of previous, next sentence **()**
beginning, middle of line **0 gm**
first, last character of line **^ \$**
line *n*, default the last, first **nG ngg**
percentage *n* of the file **n%**
column *n* of current line **n|**
match of next brace, bracket, comment, **#define** **%**
line *n* from start, bottom of window **nH nL**
middle line of window **M**

Insertion & replace → insert mode

insert before, after cursor **i a**
insert at beginning, end of line **I A**
insert text in first column **gI**
open a new line below, above the current line **o O**
replace character under cursor with *c* **rc**
like **r**, but without affecting layout **grc**
replace characters starting at the cursor **R**
like **R**, but without affecting layout **gR**
change text of movement command *m* **cm**
change current line **cc** or **S**
change to the end of line **C**
change one character and insert **s**
switch case and advance cursor **~**
switch case of movement command *m* **g~m**
lowercase, uppercase text of movement *m* **gum gUm**
shift left, right text of movement *m* **<m >m**
shift *n* lines left, right **n<< n>>**

Deletion

delete character under, before cursor **x X**
delete text of movement command *m* **dm**
delete current line, to the end of line **dd D**
join current line with next, without space **J gJ**
delete range *r* lines **:rd**
delete range *r* lines into register *x* **:rdx**

Insert mode

insert char *c* literally, decimal value *n* **^Vc ^Vn**
insert previously inserted text **^A**
same as **^A** & command mode **^@**
insert content of register *x*, literally **^Rx ^R^Rx**
text completion before, after cursor **^N ^P**
delete word before cursor **^W**
delete all inserted characters in current line **^U**
shift left, right one shift width **^D ^T**
enter digraph *c₁c₂* **^Kc₁c₂** or **c₁←c₂**
execute *c* in temporary command mode **^Oc**
scroll up, down **^X^E ^X^Y**
abandon editing & command mode **<ESC>** or **^[**

Copying

use register *x* for next delete, yank, put **"x**
show the content of all registers **:reg**
show the content of registers *x* **:reg x**
yank the text of movement command *m* **ym**
yank current line into register **yy** or **Y**
put register after, before cursor position **p P**
like **p**, **P** with indent adjusted **]p [p**
like **p**, **P** leaving cursor after new text **gp gP**

Advanced insertion

perform rot13 encoding on movement *m* **g?m**
+n, -n to number under cursor **n^A n^X**
format lines of movement *m* to fixed width **gqm**
centre lines in range *r* to width *w* **:rce w**
left align lines in range *r* with indent *i* **:rle i**
right align lines in range *r* to width *w* **:rri w**
filter lines of movement *m* through command *c* **!mc**
filter *n* lines through command *c* **n!!c**
filter range *r* lines through command *c* **:r!c**

Visual mode

start/stop highlighting characters, lines, block .. **v V ^V**
exchange cursor position with start of highlighting **o**
start highlighting on previous visual area **gv**
select a word, a sentence, a section **aw as ap**
select a block (), a block { } **ab aB**

Undoing, repeating & registers

undo last command, restore last changed line **u U**
repeat last changes, redo last undo **^R**
repeat last changes with count replaced by *n* **n.**
record, append typed characters in register *c* **qc qC**
stop recording **q**
execute the content of register *c* **@c**
repeat previous **@** command **@@**
execute register *c* as an **ex** command **:@c**
execute **ex** command *c* on range *r* where pattern *p* matches **:rg/p/c**

Complex movement

line up, down on first non-blank character **- +**
space-separated word left, right **B W**
end of space-separated word left, right **gE E**
down *n* - 1 lines on first non-blank character **n_**
beginning of screen line **gO**
first, last character of screen line **g^ g\$**
screen line up, down **gk gj**
next, previous occurrence of character *c* **fc Fc**
before next, previous occurrence of *c* **tc Tc**
repeat last **fFtT**, in opposite direction **; ,**
start of section backward, forward **[[]]**
end of section backward, forward **[]]]**
unclosed (,) backward, forward **[()]**
unclosed {, } backward, forward **[{ }]**
start of backward, forward Java method **[m]m**
unclosed **#if, #else** etc backward, forward **[#]#**
start, end of **/* */** backward, forward **[*]***

Search & substitution

search forward, backward for *s* **/s ?s**
search fwd, bwd for *s* with offset *o* **/s/o ?s?o**
repeat forward last search **n** or **/**
repeat backward last search **N** or **?**
search backward, forward for word under cursor **# ***
same, but also find partial matches **g# g***
local, global definition of symbol under cursor ... **gd gD**
substitute *f* by *t* in range *r* **:rs/f/t/x**
x: **g**-all occurrences, **c**-confirm changes
repeat substitution with new *r* & *x* **:rs x**

Special characters in search patterns

any single character, start, end of line `^ $`
start, end of word `\< \>`
a single character in range $c_1..c_2$ `[c_1-c_2]`
a single character not in range `[^c_1-c_2]`
an identifier, keyword; excl. digits `\i \k \I \K`
a file name, printable char.; excl. digits ... `\f \p \F \P`
a white space, a non-white space `\s \S`
<ESC>, <TAB>, <CR>, <BKSP> `\e \t \r \b`
match 0..1, 0..∞, 1..∞ of preceding atoms `\? * \+`
separate two branches `\|`
group patterns into an atom `\(\)`
the whole matched pattern, n -th () group `\& \n`
next character made upper, lowercase `\u \l`

Offsets in search commands

n line downward in column 1 `n` or `+n`
 n line upward in column 1 `-n`
 n characters right, left to end of match `e+n e-n`
 n characters right, left to start of match `s+n s-n`
execute search command `sc` next `;sc`

Marks and motions

mark current position with mark $c \in [a..Z]$ `mc`
go to mark c in current, C in any file `'c 'C`
go to last exit position `'0...9`
go to position before jump, at last edit `'' ''`
go to start, end of previously operated text `'[']`
print the active marks list `:marks`
print the jump list `:jumps`
go to n -th older position in jump list `n^0`
go to n -th newer position in jump list `n^I`

Key mapping & abbreviations

map $c \mapsto e$ in normal & visual mode `:map c e`
map $c \mapsto e$ in insert & cmd-line mode `:map! c e`
remove mapping c `:unmap c :unmap! c`
write current mappings, settings... to file f `:mk f`
add abbreviation for $c \mapsto e$ `:ab c e`
show abbreviations starting with c `:ab c`
remove abbreviation c `:una c`

Tags

jump to tag t `:tat`
jump to n -th newer tag in list `:nta`
jump to the tag under cursor, return from tag ... `] ^T`
list matching tags and select one for jump `:ts t`
jump to tag or select one if multiple matches `:tj t`
print tag list `:tags`
jump back from, to n -th older tag `:npo :n^T`
jump to last matching tag `:tl`
preview tag under cursor, tag t `^W}` `:pt t`
split window and show tag under cursor `^W]`
close tag preview window `^Wz` or `:pc`

Scrolling & multi-windowing

scroll line up, down `^E ^Y`
scroll half a page up, down `^D ^U`
scroll page up, down `^F ^B`
set current line at top of window `zt` or `z<CR>`
set current line at centre of window `zz` or `z.`
set current line at bottom of window `zb` or `z-`
scroll one character to the right, left `zh zl`
scroll half a screen to the right, left `zH zL`
split window in two `^Ws` or `:split`
create new empty window `^Wn` or `:new`
make current window one on screen `^Wo` or `:on`
move to window below, above `^Wj ^Wk`
move to window below, above (wrap) `^Ww ^W^W`

ex commands

edit file f , unless changes have been made `:e f`
edit file f always (by default reload current) `:e! f`
write file and edit next, previous one `:wn :wN`
edit next, previous file in list `:n :N`
write range r to current file `:rw`
write range r to file f `:rw f`
append range r to file f `:rw>>f`
quit and confirm, quit and discard changes `:q :q!`
write to current file and exit `:wq` or `:x` or `ZZ`
recall commands starting with current `<↑> <↓>`
insert content of file f below cursor `:r f`
insert output of command c below cursor `:r! c`
open a window for each file in the argument list ... `:all`
display the argument list `:args`
copy, move range r below line a `:rc a :rm a`

ex ranges

separates two lines numbers, set to first line `,` ;
an absolute line number n `n`
the current line, the last line in file `.$`
entire file, visual area `:% *`
position of mark t `'t`
the next, previous line where p matches `/p/ ?p?`
 $+n, -n$ to the preceding line number `+n -n`

Folding

create fold of movement m `zfm`
create fold for range r `zfo`
delete fold at cursor, all in window `zde`
open, close one fold; recursively `zo zc zO zC`
move to start, end of current open fold `[z]z`
move down, up to start, end of next fold `zj zk`

Miscellaneous

start shell, execute command c in shell `:sh !c`
lookup keyword under cursor with man `:K`
start make, read errors and jump to first `:make`
display the next, previous error `:cn :cp`
list all errors, read errors from file `:cl :cf`
redraw screen, show filename and position `^L ^G`
show cursor column, line, and character position ... `g^G`
show ASCII value of character under cursor `ga`
open file which filename is under cursor `gf`
redirect output to file f `:redir>f`
save view configuration [to file f] `:mkview [f]`
load view configuration [from file f] `:loadview [f]`
unmapped keys `^@ ^K ^_ \ Fn ^Fn`

This reference card may be freely distributed under the terms of the GNU general public license. The card is a (2008) much corrected and improved version from the L^AT_EX source of Sebastian Menge, which is a slight redoing of the original (<http://tnerual.eriogerg.free.fr/vim.html>) card of Laurent Grégoire.

For corrections or improvements to this version, e-mail me at boykobb@gmail.com (Boyko Bantchev).