VIM QUICK REFERENCE CARD

:viusage ..... Show a summary of all commands.
[optional], *{Visual} means in Visual mode*
Intialize vim settings in ~/.vimrc

Editing a File / Buffers

:e[dit] {file} ..... Edit {file}
:e[dit] ..... Edit current file. re-edit from outside changes
:e[dit]! {file} ..... Edit {file} force. Discard any changes
:e[dit]! ..... Edit current file force. Discard any changes
gf ..... Edit file whose name is under/after cursor. goto file
^Z :sus[pend][!] ..... Suspend Vim – fg bring to foreground
:buffers ..... Show list of buffers
:buffer n ..... Switch to buffer n
:badd file ..... Load file into new buffer
:bdelete n ..... Delete buffer n (also with filename)

Undo/Redo/Repeat & Registers

u[count] ..... Undo [count] changes.
:u[ndo] ..... Undo one change.
U ..... Undo all latest changes on one line
^R[count] ..... Redo [count] changes which were undone.
:red[o] ..... Redo one change which was undone.
. ..... Repeat last change, with count replaced with [count]
n. ..... Repeat last changes with count replaced by *n*

Exiting

:q[uit] ..... Quit Vim. This fails when changes have been made
:q[uit]! ..... Quit without writing
:cq[uit] ..... Quit always, without writing
:wq[!] ..... Write the current file and exit, [force]
:wq[!]{file} ..... Write to {file} and exit, [force]
:[range]wq[!][file] ... Same as above, but write lines in [range]
ZZ ..... Write current file, if modified, and exit
ZQ ..... Quit current file and exit (same as ":q!")

Basic Movement [count]command

h l k j ..... Character left, right; line up, down
b w ..... Word/token left, right
ge e ..... End of word/token left, right
{ } ..... Beginning of previous, next paragraph
( ) ..... Beginning of previous, next sentence
0 ^ $ ..... Beginning, first, last character of line
*n*G *n*gg ..... Line *n*, default the last, first
n% ..... Percentage *n* of the file *(n must be provided)*
n| ..... Column *n* of current line
% ..... Match of next brace, bracket, comment, #define

Movement Within Screen

*n*H *n*L ..... Line *n* from start, bottom of window
M ..... Middle line of window

Scrolling & Multi-Windowing

^E ^Y ..... Scroll line up, down
^D ^U ..... Scroll half a page up, down
^F ^B ..... Scroll page up, down

Repositioning

zt or z ..... Set current line at top of window
zz or z. ..... Set current line at center of window
zb or z- ..... Set current line at bottom of window
zh zl ..... Scroll one character to the right, left
zH zL ..... Scroll half a screen to the right, left
^Ws or :split↵ ..... Split window in two
^Wn or :new↵ ..... Create new empty window
^Wo or :on↵ ..... Make current window one on screen
^Wj ^Wk ..... Move to window below, above
^Ww ^W^W ..... Move to window below, above (wrap)
^Wn+ ^Wn- ..... Increase/decrease window size by n lines
^Wn > ^Wn < ..... Increase/decrease window width

Complex movement

- + ... Line up, down on first non-blank character
B W ..... Space-separated word left, right
gE E ..... Eend of space-separated word left, right
*n*_ ..... Down *n-1* line on first non-blank character
g0 ..... Beginning of *screen* line
g^ g$ ..... First, last character of *screen* line
gk gj ..... S*creen* line up, down
f*c* F*c* ..... *N*ext, previous occurence of character *c*
t*c* T*c* ..... *B*efore next, previous occurence of *c*
; , ... Repeat last fFtT, in opposite direction
[[ ]] ..... Start of section backward, forward
[]][..... End of section backward, forward
[( ]) ..... Unclosed (,) backward, forward
[{]} ..... Unclosed {, } backward, forward
[m]m ..... Start of backward, forward *Java* method
[#]# ..... Unclosed #if, #else, #endif backward, forward
[*]* ..... Start, end of /* */ backward, forward

Visual Mode (~ d c y > < ! = gq)

v V ^V ..... Start/stop highlighting characters, lines, block
o ..... Exchange cursor position with start of highlighting
gv ..... Start highlighting on previous visual area
aw as ap ..... Select a word, a sentence, a paragraph
n> n< = ... Indent/unindent n levels, reindent
<Esc> ..... Exit Visual Mode without making changes

Insertion & Replace → Insert Mode

i a ..... Insert before, after cursor
I A ..... Insert at beginning, end of line
gI ..... Insert text in first column
s ..... Change one character and insert
cc or S ..... Change current line
r*c* ..... Replace character under cursor with *c*
gr*c* ..... Like r, but without affecting layout
R ..... Replace characters starting at the cursor
gR ..... Like R, but without affecting layout
o O ..... Open a new line below, above the current line
c*m* ..... Change text of movement command *m*
C ..... Change to the end of line
:r[ead] [name] .....Insert the file [name] below the cursor
:r[ead] !{cmd} … Execute {cmd} & insert output below cursor.

Copying and Moving Text

"{a-zA-Z0-9.%#:-"} ..... Register for next delete, yank or put
:reg[isters]↵ ..... Display the contents of all named registers.
:reg[isters]{arg}↵ ..... Display the contents of {arg} registers
:di[splay][arg] ..... Same as :registers.
["x]y{motion} ..... Yank {motion} text [into register x].
["x]yy ..... Yank [count] lines [into register x]
["x]Y …. Yank [count] lines [into register x] (synonym for yy).
{Visual}["x]y ..... Yank highlighted text [into x]
{Visual}["x]Y ..... Yank the highlighted lines [into register x]
:[range]y[ank] [x].....Yank [range] lines [into register x].
:[range]y[ank] [x]{count} ....Yank {count} lines in [range] in x.
["x]p ..... Put the text [from x] after the cursor [count] times.
["x]P ..... Put the text [from x] before the cursor [count] times.
["x]gp ..... Like "p", but leave the cursor after the new text.
["x]gP ..... Like "P", but leave the cursor after the new text.
:[line]pu[t] [x] ..... Put the text [from x] after [line]
:[line]pu[t]! [x] ..... Put the text [from x] before [line]

Deletion

<Del> x X ..... Delete character under, before cursor
d{*m*} ..... *D*elete text of movement command *m*
dd D ..... Delete [count] lines, to the end of line
J gJ ..... Join current line with next, without space
:[*range*]d ..... Delete [range] lines
:[*range*]d{count} ..... Delete {count} lines starting [range]
:[range]*r*d*x*↵ ..... Delete range *r* lines into register *x*
{Visual}x or d ..... Delete the highlighted text
{Visual}CTRL-H ..... In Select mode: Delete highlighted text
{Visual}X or{Visual}D ..... Delete the highlighted lines

Macros/Recording

q*c* q*C* ..... Record, append typed characters in register *c*
q ..... stop recording
@*c* .....execute the content of register *c*
@@ ..... Repeat previous @ command
:@*c*↵ ..... Execute register *c* as an *Ex* command
:rg/p/c↵ Execute *Ex command c on range r where p matches*

Searching

/{pattern}[/] ..... Search for the [count]'th {pattern}
/{pattern}/{offset} ..... Search for [count]'th {pattern} and go {offset}
/↵ ..... Search for the [count]'th latest used pattern
//{offset}↵ ..... Search for [count]'th latest pattern with new
?{pattern}[?]↵ ..... Search back for count]'th {pattern}
?{pattern}?{offset}↵..... Search back for [count]'th {pattern}
?↵ ..... Search backward for [count]'th latest pattern
??{offset}↵ ..... Search backward [count]'th pattern {offset}
n or /↵ ..... Repeat the latest "/" or "?" [count] times.
N or ?↵ .... Repeat the last / or ? [count] times in opposite dir
/*s*↵ ?*s*↵ ..... Search forward, backward for *s*
/*s*/*o*↵ ?*s*?*o*↵ ..... Search fwd, bwd for *s* with offset *o*
* ..... Search backward, forward for word under cursor
g# g* ..... Same, but also find partial matches
gd gD ..... Local, global definition of symbol under cursor

Substitution

:[range]s[ubstitute]/{pattern}/{string}/[c][e][g][p][r][i][I] [count]↵ ..... For each line in [range] replace a match of {pattern} with {string}.

*:[range]**s**[ubstitute] [c][e][g][r][i][I] [count]* ***:[range]&**[c] [e][g][r][i][I] [count]* ..... *Repeat last :substitute with same search pattern and substitute string, but without the same flags. You may add extra flags*

***Substitute command arguments:**[c] Confirm, 'y', 'n' 'a' or 'q', [e] no error on failure, [g] replace all, [i] Ignore case, [I] Don't ignore case, [p] Print the line containing last substitute.*

Special Characters in Search Patterns

. ^ $ ..... Any single character, start, end of line
\< \> ..... *Start*, end of word
[*c1-c2*] ..... A single character in range *c1..c2*
[^*c1-c2*] ..... A single character not in range
\i \k \I \K ..... An identifier, keyword; excl. digits
\f \p \F \P ..... A file name, printable char.; excl. digits
\s \S ..... A white space, a non-white space
\e \t \r \b ..... <esc>, <tab>, <↵>, <←>
\= * \+ ..... Match *0..1*, *0..∞*, *1..∞* of preceding atoms
\| ..... Separate two branches (≡ *or*)
\(\) ..... Group patterns into an atom
\& *n* ..... The whole matched pattern, *nth* () group
\u \l ..... Next character made upper, lowercase
\c \C ..... Ignore, match case on next pattern

Offsets in Search Ccommands

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

Completion

^X^L ..... whole lines
^X^N ^X^I ..... Keywords in current file, plus included files
^X^K ^X^T..... Keywords in dictionary, thesaurus

Formatting & Filtering *(no m for visual mode)*

gq*m* *gqgq*..... *Format* lines of movement *m* to fixed width
:*r*ce *w*↵ ..... Center lines in range *r* to width *w*
:*r*le *i*↵ ..... Left align lines in range *r* with indent *i*
:*r*ri *w*↵ ..... Right align lines in range *r* to width *w*
!*mc*↵ ..... Filter lines of movement *m* through command *c*
n!!*c*↵ ..... Filter *n* lines through command *c*
:*r*!*c*↵ ..... Filter range *r* lines through command *c*
~ ..... Switch case and advance cursor
g~*m* ..... Switch case of movement command *m*
gu*m* gU*m* ..... Lowercase, uppercase text of movement *m*
<*m* >*m* ..... Shift left, right text of movement *m*
n<< *n*>> ..... *Shift n* lines left, right
^A ^X..... Increment/decrement number under cursor

Marks and Motions

m*c* ..... Mark current position with mark *c* ∈*[a..Z]*
`*c* `*C* ..... Go to mark *c* in current, *C* in any file
`*0..9* ..... Go to last exit position
`` `" ..... Go to position before jump, at last edit
`[ `] ..... Go to start, end of previously operated text
:marks↵ ..... Print the active marks list
:jumps↵ ..... Print the jump list
n^O ..... Go to *nth* older position in jump list
n^I ..... Go to *nth* newer position in jump list

Key Mapping & Abbreviations

:map *c e*↵ ..... Map *c* ↦ *e* in normal & visual mode
:map! *c e*↵ ..... Map *c* ↦ *e* in insert & cmd-line mode
:unmap *c*↵ :unmap! *c*↵ ..... Remove mapping *c*
:mk *f*↵ ..... Write current mappings, settings... to file *f*
:ab *c e*↵ ..... Add abbreviation for *c* ↦ *e*
:ab *c*↵ ..... Show abbreviations starting with *c*
:una *c*↵ ..... Remove abbreviation *c*

Tags

:ta *t*↵ ..... Jump to tag *t*
:*n*ta↵ ..... Jump to *nth* newer tag in list
^] ^T ..... Jump to the tag under cursor, return from tag
:ts *t*↵ ..... List matching tags and select one for jump
:tj *t*↵ ..... Jump to tag or select one if multiple matches
:tags↵ ..... Print tag list
:*n*po↵ :*n*^T↵ ..... Jump back from, to *nth* older tag
:tl↵ ..... Jump to last matching tag

Ex commands (↵)

:help hold-grail..... Show all Ex commands
:e *f* ..... Edit file *f*, unless changes have been made
:e! *f* ..... Edit file *f* force (by default reload current)
:wn :wN ..... Write file and edit next, previous one
:n :N ..... Edit next, previous file in list
:*r*w ..... Write range *r* to current file
:*r*w *f* ..... *W*rite range *r* to file *f*
:*r*w>>*f* ..... *A*ppend range *r* to file *f*
:q :q! ..... Quit and confirm, quit and discard changes
:wq or :x or ZZ ..... Write to current file and exit
<up> <down> ..... Recall commands starting with current
:r *f* ..... Insert content of file *f* below cursor
:r! *c* ..... Insert output of command *c* below cursor
:args ..... Display the argument list
:*r*co *a* :*r*m *a* ..... Copy, move range *r* below line *a*

Ex ranges

, ; ..... Separates two lines numbers, set to first line
n ..... *A*n absolute line number *n*
. $ ..... The current line, the last line in file
% * ..... Entire file, visual area
'*t* ..... *P*osition of mark *t*
/*p*/ ?*p*? ..... The next, previous line where *p* matches
+*n* -*n* ..... +*n*, -*n* to the preceding line number

Folding

:set fdm-indent ..... Indent-fold method
zf*m* ..... *Create* fold of movement *m*
:*r*fo ..... Create fold for range *r*
zd zE ..... Delete fold at cursor, all in window
zo zc zO zC ..... Open, close one fold; recursively
[z]z ..... Move to start, end of current open fold
zj zk ..... Move down, up to start, end of next fold

Miscellaneous

:sh↵ :!*c*↵ ..... Start shell, execute command *c* in shell
K ..... Lookup keyword under cursor with man
^L ^G ..... Redraw screen, show filename and position
: set cuc..... Show cursor colum visually
g^G ..... Show cursor column, line, and character position
ga ..... Show ASCII value of character under cursor
gf ..... Open file which filename is under cursor
:mkview *[f]* ..... Save view configuration [to file *f*]
:loadview *[f]* ..... Load view configuration [from file *f*]