Basic movement

h l k j.............. character left, right; line up, down
b w................. word/token left, right
g e.............. beginning of word/token left, right
{ }.............. beginning of previous, next paragraph
( ).............. beginning of previous, next sentence
0 gm.............. beginning, middle of line
- $.............. first, last character of line
nG ngg............. 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
nH nL............. line n from start, bottom of window
M........... middle line of window

Insertion & replace → insert mode

i a............. insert before, after cursor
I A............. insert at beginning, end of line
Gl............. insert in first column
o 0........ open a new line below, above the current line
rc........... replace character under cursor with c
grc........ like r, but without affecting layout
R........... replace characters starting at the cursor
gR........... like R, but without affecting layout
cm........... change text of movement command
cc=S........... change current line
C........... change to the end of line
s........... change one character and insert
-.............. switch case and advance cursor
g`m........... switch case of movement command
gum gUm........ lower case, uppercase text of movement m
<m >m............ shift left, right text of movement m
n<< n>>............ shift n lines left, right

Deletion

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

Insert mode

`c `vn........... insert char c literally, decimal value n
`A ................ insert previously inserted text
`0 ........ same as `A and stop insert → command mode
`R `R `x........... insert content of register x, literally
`N `P................ text completion before, after cursor
`W............. delete word before cursor
`U............. delete all inserted character in current line
`D `T........... shift left, right one shift width
`e1 e2 = f1 f2 ............... enter digraph {e1, e2} to f1 f2
`c ........... execute c in temporary command mode
`E `Y ........... scroll up, down
\langle esc \rangle = `I ........... abandon edition → command mode

Copying

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

Advanced insertion

g?m ........... perform rot13 encoding on movement m
n A nX ............. +n, -n to number under cursor
gqm ........... format lines of movement m to fixed width
:rc w=\langle x \rangle .......... center lines in range r to width w
:rl i=\langle x \rangle .......... left align lines in range r with indent i
:rr i=\langle x \rangle .......... right align lines in range r to width w
!ne=\langle x \rangle .......... filter lines of movement m through command c
n!!c=\langle x \rangle .......... filter n lines through command c
:r!c=\langle x \rangle .......... filter range r lines through command c

Visual mode

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

Undoing, repeating & registers

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

Complex movement

- +........ line up, down on first non-blank character
B W........ space-separated word left, right
\langle gE \rangle .......... end of space-separated word left, right
n........ down n - 1 line on first non-blank character
g0 ........ beginning of screen line
\langle gS \rangle .......... first, last character of screen line
g\langle k \rangle g\langle j \rangle .......... screen line up, down
f c F e ........ next, previous occurrence of character c
t e T e ........ before next, previous occurrence of c
; ........ repeat last ftTt, 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
[* ] * ........ end of /* */ backward, forward

Search & substitution

\langle /s\rangle .......... search forward, backward for s
\langle /s/o\rangle .......... search fwd, bwd for s with offset o
\langle n w/\rangle .......... repeat forward last search
N w/\langle ?\rangle .......... repeat backward last search
# * ........ search backward, forward for word under cursor
\langle g# \rangle g\langle d \rangle .......... same, but also find partial matches
gd gD .......... local, global definition of symbol under cursor
:ra/ft \langle \langle x \rangle \rangle ........... substitute f by t in range r
   \langle x \rangle : g ........ all occurrences, c ........ confirm changes
:ra x\langle \langle x \rangle \rangle .......... repeat substitution with new \langle x \rangle & x
Special characters in search patterns
 - \$ \ldots any single character, start, end of line
 - \< \\> \ldots start, end of word
 - [c₁-c₂] \ldots a single character in range c₁-c₂
 - [c₁-c₂] \ldots a single character not in range
 - c \| \K \ldots an identifier, keyword; excl. digits
 - f \ combining \ \p \ \V \ldots a file name, printable char.; excl. digits
 - s \= \\= \ldots a white space, a non-white space
 - e \non \ldots group patterns into an atom
 - \n \ldots the whole matched pattern, n\textsuperscript{th} () group
 - c \e \ldots ignore, match case on next pattern

Offsets in search commands
 - n \ldots n line downward in column 1
 - n \ldots n line upward in column 1
 - e+n \ldots e-n characters right, to left to end of match
 - s+n s-n \ldots n characters right, left to start of match
 - sc \ldots execute search command sc next

Marks and motions
 - mc \ldots mark current position with mark c \in [a..Z]
 - c 'c 'C \ldots go to mark c in current, C in any file
 - o \ldots go to last exit position
 - r \ldots go to position before jump, at last edit
 - [ ] \ldots go to start, end of previously operated text
 - \marks \ldots print the active marks list
 - \jumps \ldots print the jump list
 - n'O \ldots go to n\textsuperscript{th} older position in jump list
 - n'T \ldots go to n\textsuperscript{th} newer position in jump list

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

Tags
 - :ta t \ldots jump to tag t
 - :nta t \ldots jump to n\textsuperscript{th} newer tag in list
 - :ja t \ldots jump to the tag under cursor, return from tag
 - :ts t \ldots list matching tags and select one for jump
 - :tj t \ldots jump to tag or select one if multiple matches
 - :tags \ldots print tag list
 - :npo n T \ldots jump back from, to n\textsuperscript{th} older tag
 - :tl \ldots jump to last matching tag
 - :W pt t \ldots preview tag under cursor, tag \W \ldots split window and show tag under cursor
 - :Wz \ldots close tag preview window

Scrolling & multi-windowing
 - E Y \ldots scroll line up, down
 - D U \ldots scroll half a page up, down
 - F B \ldots scroll page up, down
 - zt z \ldots set current line at top of window
 - zz z \ldots set current line at center of window
 - zb z \ldots set current line at bottom of window
 - zh zl \ldots scroll one character to the right, left
 - zH zL \ldots scroll half a screen to the right, left
 - Ws sp \ldots split window in two
 - Wn \ldots create new empty window
 - Wo \ldots make current window one on screen
 - Wj Wk \ldots move to window below, above
 - Ww Wh \ldots move to window below, above (wrap)

Ex commands (!)
 - :e f \ldots edit file f, unless changes have been made
 - :e! f \ldots edit file f always (by default reload current)
 - :vn \ lv \ldots write file and edit next, previous one
 - :n N \ldots edit next, previous file in list
 - :rw \ldots write range r to current file
 - :rw f \ldots write range r to file f
 - :rw>ff \ldots append range r to file f
 - :q :q! \ldots quit and confirm, quit and discard changes
 - :q :q < up > down > recall commands starting with current
 - :r f \ldots insert content of file f below cursor
 - :r! c \ldots insert output of command c below cursor
 - :args \ldots display the argument list
 - :rc a :rm a \ldots copy, move range r below line a

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

Folding
 - zfm \ldots create fold of movement m
 - :rf \ldots create fold for range r
 - zd zE \ldots delete fold at cursor, all in window
 - zo zC z0 zC \ldots open, close one fold; recursively
 - [z] z \ldots move to start, end of current open fold
 - zj zk \ldots move down, up to start, end of next fold

Miscellaneous
 - :sh \ldots start shell, execute command c in shell
 - K \ldots lookup keyword under cursor with man
 - :make \ldots start make, read errors and jump to first
 - :cn \ldots display the next, previous error
 - :cl \ldots list all errors, read errors from file
 - 'L 'G \ldots redraw screen, show filename and position
 - gG \ldots show cursor column, line, and position
 - ga \ldots show ASCII value of character under cursor
 - gf \ldots open file which filename is under cursor
 - :redir>ff \ldots redirect output to file f
 - :mkview [f] \ldots save view configuration [to file f]
 - :loadview [f] \ldots load view configuration [from file f]

This card may be freely distributed under the terms of the GNU general public licence — Copyright © 2003 by Laurent Grégoire
(laurent.gregoire@ian.fr) — v1.6 — The author assumes no responsibility for any errors on this card. The latest version can be found at http://nerual.eriogerg.free.fr/