Basic movement

h  k  j  ....... character left, right; line up, down
b  w  .............. word/token left, right
gw  e  .......... beginning of word/token left, right
{ }  .......... beginning of previous, next paragraph
( )  .......... beginning of previous, next sentence
0  g0  .......... beginning, middle of line
$  .......... first, last character of line
nG  nng  .......... line n, default the last, first
n%  .......... percentage n of the file (m 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
gI  .......... insert in first column
o O  .......... open a new line below, above the current line
r c  .......... 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  glm  .......... lower case, uppercase text of movement
<m>  .......... shift left, right text of movement
n<<  .......... shift n lines left, right

Deletion

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

Insert mode

‘c ‘h h  .......... insert char c literally, decimal value n
‘A  .......... insert previously inserted text
’0  .......... same as ‘A and stop insert → command mode
’r  .......... insert content of register r, 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
Kc2c2  .......... enter digraph {c1, c2}
’c  .......... execute c in temporary command mode
‘X ‘E ‘Y  .......... scroll up, down
〈esc〉  .......... abandon edition → command mode

Copying

"x  .......... use register x for next delete, yank, put
:reg=x  .......... show the content of all registers
:reg x-y  .......... 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
l p [p]  .......... 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< <n> >  .......... +n, –n to number under cursor
gqm  .......... format lines of movement m to fixed width
:rc=  .......... center lines in range r to width w
:re i<  .......... left align lines in range r with indent i
:ri=  .......... right align lines in range r to width w
!n=  .......... filter lines of movement m through command c
!!=  .......... filter n lines through command c
:r!c=  .......... filter range r lines through command c

Visual mode

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
ab aB  .......... select a block ( ), a block { }

Undoing, repeating & registers

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

Complex movement

- +  .......... line up, down on first non-blank character
B W  .......... space-separated word left, right
gE E  .......... end of space-separated word left, right
n-  .......... down n → 1 line on first non-blank character
g0  .......... beginning of screen line
> gE  .......... first, last character of screen line
gk g]  .......... screen line up, down
fc Fe  .......... next, previous occurrence of character c
tc Tc  .......... before next, previous occurrence of c
;  .......... repeat last fFfT, in opposite direction
[ ]  .......... start of section backward, forward
[ ]  .......... end of section backward, forward
[ ]  .......... unclosed ( ) backward, forward
[ ]  .......... unclosed { } backward, forward
[ ]  .......... start, end of /* */ backward, forward

Search & substitution

/is<  .......... search forward, backward for s
/s<oe<  .......... search fwd, bwd for s with offset o
n< /<  .......... repeat forward last search
N< ?<  .......... repeat backward last search
* #  .......... search backward, forward for word under cursor
g# g<  .......... same, but also find partial matches
gd gD  .......... local, global definition of symbol under cursor
:ra/f/t/x←  .......... substitute f by t in range r
[ x : g]  .......... all occurrences, c−confirm changes
:ra x←  .......... repeat substitution with new r & x
Offsets in search commands
- \(n\) \(\rightarrow\) \(n\) line downward in column 1
- \(-n\) \(\rightarrow\) \(n\) line upward in column 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
- \(;sc\) \(\rightarrow\) execute search command \(sc\) next

Marks and motions
- \(m\) \(c\) \(C\) \(\rightarrow\) mark current position with mark \(c\) \(\in\) \([a..z]\)
- \(t\) \(C\) \(\rightarrow\) go to mark \(c\) in current, \(C\) in any file
- \(0\) \(9\) \(\rightarrow\) go to last exit position
- \(\langle\rangle\) \(\rightarrow\) go to position before jump, at last edit
- \(\{\}\) \(\rightarrow\) go to start, end of previously operated text
- \(;marks\) \(\rightarrow\) print the active marks list
- \(;jumps\) \(\rightarrow\) print the jump list
- \(n\) \(\rightarrow\) go to \(n\) \(th\) older position in jump list
- \(n\) \(\rightarrow\) go to \(n\) \(th\) newer position in jump list

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

Tags
- \(;ta\) \(t\) \(\rightarrow\) jump to tag \(t\)
- \(;nta\) \(\rightarrow\) jump to \(n\)\(th\) newer tag in list
- \(;t\) \(T\) \(\rightarrow\) jump to the tag under cursor, return from tag
- \(;ts\) \(t\) \(\rightarrow\) list matching tags and select one for jump
- \(;tj\) \(t\) \(\rightarrow\) jump to tag or select one if multiple matches
- \(;tags\) \(\rightarrow\) print tag list
- \(;npo\) \(n\) \(\rightarrow\) jump back from, to \(n\)\(th\) older tag
- \(;tl\) \(\rightarrow\) jump to last matching tag
- \(;W\) \(p\) \(t\) \(\rightarrow\) preview tag under cursor, tag \(t\)
- \(;W\) \(\rightarrow\) split window and show tag under cursor
- \(;Wz\) \(pc\) \(\rightarrow\) close tag preview window

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

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

This card may be freely distributed under the terms of the GNU general public licence — Copyright © 2003 by Laurent Grégoire (laurent.gregoire@icam.fr) — v1.7 — The author assumes no responsibility for any errors on this card. The latest version can be found at http://tnerual.eriogerg.free.fr/