

Notations and Conventions

Commands you type are in *fixed font*.
Italics = substitute desired value.
 <CR> means press RETURN key.
 ^X means press CONTROL and X keys together. Boxed commands switch to insert mode; press ESC key to end new text.

Beginning Your Edit Session

vi *file* <CR> edit or create *file*
 vi -r <CR> show rescued files
 vi -r *file* <CR> recover rescued *file*

Ending Your Edit Session

:q! <CR> quit and discard changes
 :wq <CR> or ZZ quit and save changes
 :wq *new*<CR> save as *new* and quit

Controlling Your Screen Display

^R Eliminate @ lines
 ^L Repaint screen after interruption
 :set wm=x <CR> Auto word wrap at *x* chars before line end
 :set nu <CR> Show line numbers on screen (not added to file)
 :set nonu <CR> Stop showing line numbers on screen
 (put set commands into .exrc file for automatic settings each time you start vi)

Moving the Cursor

h one position left
 k one line up
 j one line down
 l (letter "ell") one position right
 0 (zero) beginning of current line
 \$ end of current line
 w forward one word
 W ... including punctuation
 b backward one word
 B ... including punctuation
 e forward to end of current word
 E ... including punctuation

- up one line, 1st non-blank char
 + or <CR>down line, 1st non-blank char
 H beginning of first screen line
 M beginning of middle screen line
 L beginning of last screen line

Paging Through Text

^F forward one screen
 ^B backward one screen
 ^D scroll down half screen
 ^U scroll up half screen
 nG move screen to line number *n*
 G move screen to last line

Searching Through Text

/*pattern* forward search for *pattern*
 (regular expression syntax)
 ?*pattern* backward search for *pattern*
 (regular expression syntax)
 n repeat search for next occurrence
 N repeat search, reverse direction

Creating Text

Press ESC key to end new text. To enter a control character as text, precede it with ^V

a append text after cursor
 A append text at end of current line
 i insert text before cursor
 I insert text before first non-blank character on current line
 o open new line after current line
 O open new line before current line
 :r *file* <CR> insert contents of *file* after current line

Modifying Text -- Simple Changes

(* = can be preceded with a repeat count, e.g., 5x or 16dd. Count starts at cursor.)
 x delete character at cursor *
 dd delete line with cursor *
 dw delete current word *
 D delete to end of line.
 rc change character at cursor to *c*

cw	replace current word with new text *
cc	replace entire current line *
C	replace line from cursor to end
J	join current line with next
~	change case of current character
u	undo last text change
.	repeat last text-change command (could be at new location)

Modifying Text -- Operators

(Can precede with repeat count. Double to affect whole lines, e.g., 5yy. Follow with one of the cursor movement or searching commands to select affected text, e.g., dw or c5w or y/Geology<CR>).

d	delete
c	change (replace with new text)
Y	yank (copy) to buffer
!	filter selected text through command typed on status line, then replace with command output
<<	shift line(s) left one tab position
>>	shift line(s) right one tab position

Moving Text Around

(Use these commands to insert a copy of text from buffer. Deleted, replaced (old), or yanked text goes to unnamed buffer by default. Or use named buffer a thru z by prefixing command with " operator and buffer name, e.g., "a5yy and then "ap)
 p copy buffer text after cursor or line
 "ap ... using named buffer a (or b,c, etc)
 P copy buffer text before cursor or line
 "aP ... using named buffer a (or b,c, etc)
 xp transpose characters

Global Text Substitution

:n,m s/old/new/g<CR>
 Change all occurrences of regular expression *old* to text *new* on all lines *n* thru *m*. Can use symbolic line numbers "." (current line) or "\$" (last line).