ENTERING vi

vi (visual) is a display-oriented text editor based on an underlying line editor ex. When using vi, changes you make to the file are reflected in what you see on your terminal screen. The position of the cursor on the screen indicates the position within the file. Many commands may be preceded by an integer, n, indicating that the command is to be repeated n times at the current cursor position. Typing the ESC (escape) character takes you out of input mode and into command mode. A partial command may be cancelled by typing ESC. To enter the vi editor type:

vi -options *filename*

OPTIONS

-l	lisp mode: indents appropriately for lisp code, the () { } [[and]] commands in vi are modified to have meaning for lisp
-r *filename*	recover filename after an editor or system crash. If file is not specified a list of all saved files will be printed.
-R	read only mode
+/*string*	search for string
vi *filenamelist*	enter vi, read in first file in filenamelist
:n	edit next file in filenamelist
:n *filenamelist*	specify new filenamelist
:e filename	
:e#	return to original file
:e! *filename*	edit filename, discard previous buffer
:r *filename*	place copy of filename below the current line
CONTROL-g	display the current file name and line number of current line

Modes of Operation

Command	normal and initial mode; other modes return to command mode upon completion. **ESC** (escape) is used to cancel partial command.
Input	To enter the Input mode, type one of the following: **a, i, A, I, o O, c, C, s, S, or R.** Text may then be entered. Input mode is terminated with **ESC** character.

UNDOING/CANCELING COMMANDS

ESC	cancel unexecuted vi command
Control-C	stop in-progress command
u	counteract last command that changed the buffer
U	counteract changes to current line

REPEATING A COMMAND

.	repeat the last command that changed the buffer

SAVING WORK/EDITING

:w	write file under original name
ZZ or **:wq**	write file under original name, exit vi
:q	exit vi, no changes are saved
:w *filename*	write the file under *filename*
:w! *filename*	write the file under *filename*, overwrite existing filename
:x,y w *filename*	write lines *x* through *y* under *filename*
:x,y w!	filename write lines *x* through *y* under *filename*, overwrite existing filename

DISPLAYING LINE NUMBERS

CONTROL-g	display line number of current line
:set number	display all line numbers, to the left of each line
:set nonumber	remove all displayed line numbers

INSERTING TEXT

Inserting Text

i	insert before the current character
a	insert after the current character
A	insert at end of current line
I	insert at beginning of current line
o	insert below current line (opens new line)
O	insert above current line (opens new line)
ESC	end text insertion

Replacing Text

r	replace the current character, no ESC necessary
s	replace the current character
cw	replace the current word
S	replace the current line
C	replace the current cursor position to end of line
R	replace from current cursor position to end of line, does not change characters typed over
ESC	end text insertion

Erasing During Insertion

Control-h or **DELETE**	erase character, move left
Control-w	erase word, move left
Control-u	erase the current line

Setting Margins

set wrapmargin=*x*	set right margin to *x*, automatic word wrap

Setting Automatic Indentation

:set autoindent	set automatic indentation
:set noai	turn off automatic indentation
Control-d	move left one tabstop
Control-t	move right one tabstop

JOINING LINES

J	join the current line and the line below it
#J	join # lines, starting with the current line

DELETING TEXT

x	delete the current character
#x	delete # characters, starting with current character
dw	delete current word
#dw	delete # words, starting with the current word
dd	delete the current line
#dd	delete # lines, starting with the current line
D	delete from current cursor position to end of line
:x,y d	delete lines x through y
d'/	delete from current line to line with marker /

COPYING TEXT

Making a Copy

y	yank current character
#y	yank # characters, starting with current character
yw	yank current word
#yw	yank # words, starting with current word
yy	yank current line
#yy	yank # lines, starting with current line
:x,y y	yank lines x through y
y'/	yank from current line to line with marker /

RETRIEVING TEXT

p place most recently deleted or yanked character(s) or word(s) after the current character, place line(s) below current line

P place most recently deleted or yanked character(s) or word(s) before the current character, place line(s) above the current line

SHELL COMMANDS

:!*command*	execute one shell command
:sh	run subshell. Type exit and press RETURN to return to vi.
:r!*command*	insert output of shell command below current line

MOVING AROUND IN A FILE

Moving by characters

h or **backspace**	move left one character
j	move down one line
k	move up one line
l or **space bar**	move right one character

Moving by words

w	move right one word or group of punctuation marks
b	move left one word or group of punctuation marks
W	move right one word
B	move left one word
E	move to the end of a word

Moving by Lines

0	move to the beginning of current lune
$	move to the end of current line
RETURN	move to first non-white space character at beginning of next line
-(minus)	move to first non-whitespace character at beginning of previous line
H	move to top line on screen
L	move to bottom line on screen
M	move to middle line on screen
G	move to end of file
x G	moves to line number x
/	move to first character in line marked with /
	move to previous location (counteracts G or simple search)

SCROLLING

Control-e	scrollup one line
Control-y	scroll down one line
Control-f	scroll forward one screen
Control-b	scroll backward one screen
Control-d	scroll forward half screen
Control-u	scroll backward half screen
z<RETURN>	redraw screen, current line at top of screen
z-	redraw screen, current line at bottom of screen
z.	redraw screen, current line at center of screen.

TEXT BUFFERS

Automatic Numbered Buffers (1-9)

"# p	place text from buffer # below current line
"# P	place text from buffer # above current line

Named Buffers (a-z)

"b *command*	place text deleted or yanks with command into buffer b
"b p	place text from buffer b below current line
"b P	place text from buffer b above current line

MOVING BY LINE NUMBER

:x mz	move line x below line z
:x, y mz	move lines x through y below z

COPYING BY LINE NUMBER

:*x*t *z*	copy line *x* below line *z*
:*x,y* **m***z*	move lines *x* through *y* below line *z*

Recovering a File

vi -r *filename*	recover file lost during system crash or line disconnection
vi -r	list all recoverable files

SEARCHES

Searching in a Line

f c	move forward to find character c on current line, cursor rests on c
t c	move forward to find character c on current line, cursor rests on character before c
F c	move backwards to find character on current line, cursor rests on c
T c	move backwards to find character c on current line, cursor rests on character after c
;	repeat last **f**, **F**, **t**, **T**
,	inverse of last **f**, **F**, **t**, **T**

Simple Search

/*string*	search forward in the file for string
?*string*	search backward in the file for string
n	subsequent search in the same direction
N	subsequent search in the opposite direction
:set wrapscan	set wrapping during a search
:set nowrapscan	set no wrapping during a search

Global Search

:g/*string/command*	command affects lines containing string
:v/*string/command*	command affects lines not containing string

SEARCHES (CONTINUED)

Substitution (Search and Replace)

:*x,y***s**/*oldstring/newstring/flags*

x, y	line numbers in which the substitution is to occur. Give one line number, or two separated by a comma, which indicate a range of lines. Absence of line numbers indicates current line substitution command. Only the first occurrence of the string on the selected line(s) is replaces unless the g flag, described below, is used.
s	substitution command. Only the first occurrence of the string on the selected line(s) is replaced unless the **g** flag, described below, is used.
/oldstring	string to replace. If you do not specify an oldstring, vi searches for the oldstring specified in the last s command or search (if there was one).
/newstring	replacement string
/flags	optional, to specify the type of search:
	c — wait for confirmation before substituting. Type y or n and press RETURN.
	g — replace all occurrences of the string in specified lines

Special Characters in Searches
Character Classes

[cccc]	match any specified letter or number
[^cccc]	match all letters or numbers except those specified
[c1-c2]	match all letters or numbers in specified range

Character at Specified Locations

\<cc	match words that begin with cc
\>cc	match words that end in cc
^	match beginning of lines
^cccc	match lines that begin with cc
$	match end of lines
cccc$	match lines that end with cccc

Wildcard and Multiplier Characters

.	match any one character
*	match zero or more occurrences of the preceding character
.*	match any number of characters