VI Quick Reference

Entering/leaving vi

vi *name*	edit *name* at top
vi +n *name*	edit *name* at line n
vi + *name*	edit *name* at end
vi -r	list saved files
vi -r *name*	recover file *name*
vi *name1 name2 ...*	edit first file, rest via **:n**
vi -t *tag*	start at *tag*
vi +/*pat name*	search for *pat*
view *name*	read only mode
ZZ	exit from vi, saving changes
^Z	stop vi for later resumption

The display

Last line	Error message, echoing input to **: / ?** and **!**, feedback about i/o and large changes.
@ lines	On screen only, not in file.
~ lines	Lines past end of file.
^x	Control characters, ^? is delete.
tabs	Expand to spaces, cursor at last.

Vi states (or modes)

Command	Normal and initial state. Others return here. ESC (escape) cancels partial command.
Insert	Entered by **a A i I o O c C s S R**. Arbitrary text then terminates with ESC or abnormally with interrupt.
Last line	Reading input for : / ? or !; terminates with ESC or CR to execute, interrupt to cancel.

Counts before vi commands

line/column number	z G —
scroll amount	^D ^U
replicate insert	a i A I
repeat effect	most other commands

Simple commands

dw	delete a word
de	delete a word leaving punctuation
dd	delete a line
3dd	delete 3 lines
i*text*ESC	insert *text*
cw*new*ESC	Change word to *new*
ea*s*ESC	pluralize word
xp	transpose characters

Interrupting, cancelling

ESC	end insert or incomplate cmd
^?	(delete or rubout) interrupts
^L	reprint screen if **^?** scrambles it

File manipulation

:w	write back changes
:wq	write and quit
:q	quit
:q!	quit, discard changes
:e *name*	edit file *name*
:e!	reedit, discard changes
:e + *name*	edit, starting at end
:e + *n*	edit starting at line *n*
:e #	edit alternate file
^^	synonym for :e #
:w *name*	write file *name*
:w! *name*	overwrite file *name*
:sh	run shell
:!*cmd*	run *cmd*, then return
:n	edit next file in arglist
:n *args*	specify new arglist
:f	show current file and line
^G	synonym for **:f**
:ta *tag*	to tag file entry *tag*
^]	**:ta**, following word is *tag*

Positioning within file

^F	forward screenfull
^B	backward screenfull
^D	scroll down half screen
^U	scroll up half screen
G	goto line (end default)
/*pat*	next line matching *pat*
?*pat*	prev line matching *pat*
n	repeat last / or **?**
N	reverse last / or **?**
/*pat*/+n	n'th line after *pat*
?*pat*?-n	n'th line before *pat*
]]	next section/function
[[	previous section/function
%	find matching ( ) { } [or]

Adjusting the screen

^L	clear and redraw
^R	retype, eliminate @ lines
zCR	redraw, current at window top
z-	... at bottom
z.	... at center
/*pat*/z-	*pat* line at bottom
z*n*.	use *n* line window
^E	scroll window down 1 line
^Y	scroll window up 1 line

Marking and returning

‘‘	previous context
’’	... at first non-white in line
m*x*	mark position with letter *x*
‘*x*	to mark *x*
’*x*	... at first non-white in line

Line positioning

H	home window line
L	last window line
M	middle window line
+	next line, at first non-white
-	previous line, at first non-white
CR	return, same as **+**
j or ↓	next line, same column
k or ↑	previous line, same column

Character positioning

^	first non-white
0	beginning of line
$	end of line
l or →	right
h or ←	left
^[[	same as **h**
space	same as **l**
f*x*	find *x* forward
F*x*	find *x* backward
t*x*	find just before *x* forward
T*x*	find just before *x* backward
;	repeat last **f F t** or **T**
,	inverse of ;
n|	to column *n*
%	find matching ( ) { } [or]

Words, sentences, paragraphs

w	next word forward
b	back one word
e	end of word
)	to next sentence
}	to next paragraph
(	to next sentence
{	to prevoius paragraph
W	blank delimited word
B	back **W**
E	to end of **W**

Commands for LISP

)	Forward s-expression
}	... but don't stop at atoms
(	Back s-expression
}	... but don't stop at atoms

Corrections during insert

^H	erase last character
^W	erase last word
`erase`	your erase, same as **^H**
`kill`	your kill, erase this line
\	escapes **^H**, your erase and kill
ESC	ends insertion, back to command
^?	interrupt, terminates insert
^D	backtab over *autoindent*
↑^D	kill *autoindent*, save for next
0^D	... but at margin next also
^V	quote non-printing character

Insert and replace

a	append after cursor
i	insert before
A	append at end of line
I	insert before first non-blank
o	open line below
O	open line above
r*x*	replace single character with *x*
R	replaces characters

Operators (double to affect lines)

d	delete
c	change
<	left shift
>	right shift
!	filter through unix command
=	indent for LISP
y	yank lines to buffer

Miscellaneous operations

C	change rest of line
D	delete rest of line
s	substitute chars
S	substitute lines
J	join lines
x	delete characters
X	... before cursor
Y	yank lines

Yank and put

p	put back lines
P	put before
"*x*p	put from buffer *x*
"*x*y	yank to buffer *x*
"*x*d	delete into buffer *x*

Undo, redo, retrieve

u	undo last change
U	restore current line
.	repeat last change
"*d*p	retrieve *d*th last delete

Scanning pattern formation

^	beginning of line
$	end of line
.	any character
\<	beginning of word
\>	end of word
[*abc*]	any character in *abc*
[^*abc*]	any character not in *abc*
[*x* − *y*]	any character between *x* and *y*
*	any number of preceding

Useful options (:se option, :se noption)

autoindent	ai	apply indent
autowrite	aw	write before changing files
ignorecase	ic	in scanning
lisp		( ) { } are s-expressions
list		print ^I for tabs, $ at end
magic		. { * special in patterns
number	nu	show line numbers
paragraphs	para	macro names that start paragrap
redraw		simulate smart terminal
scroll		command mode lines
sections	sect	macro names for sections
shiftwidth	sw	for < >, and input **^D**
showmatch	sm	to) and] as typed
slowopen	slow	choke updates during insert
window		visual mode lines
wrapscan	ws	around end of buffer?
wrapmargin	wm	automatic line splitting

Aliasing and Mapping

:ab <*enter-str*> <*real-str*>	When <*enter-str*> is typed, replace by <*real-str*>
:ab *mroe more*	corrects spelling mistake
:ab *tse Transactions on Software Engineering*	expands tse
:map *char commands*	(re)defines *char*
:map ; :	**;** is now equivalent to **:**
:map g :1,$ s/	Frequent command
:map F ifor (i = 1; i <= N; i++)^M^MESCkkfN	a C `for` loop
:map B i{**ESCea}ESC**	Latex boldface a word
:map F !}fmt -70^M	Format a paragraph

Jeff Offutt, 1995