

# VI Quick Reference

## Entering/leaving vi

*vi name*  
*vi +n name*  
*vi + name*  
*vi -r*  
*vi -r name*  
*vi name1 name2 ...*  
*vi -t tag*  
*vi +/pat name*  
*view name*  
*ZZ*  
*~Z*

edit *name* at top  
edit *name* at line *n*  
edit *name* at end  
list saved files  
recover file *name*  
edit first file, rest via **:n**  
start at *tag*  
search for *pat*  
read only mode  
exit from vi, saving changes  
stop vi for later resumption

## The display

Last line  
@ lines  
~ lines  
~x  
tabs

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

## Vi states (or modes)

Command  
Insert  
Last line

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

## Counts before vi commands

line/column number  
scroll amount  
replicate insert  
repeat effect

**Z** **G** —  
**~D** **~U**  
**a i A I**  
most other commands

## Simple commands

dw  
de  
dd  
3dd

delete a word  
delete a word leaving punctuation  
delete a line  
delete 3 lines

## Interrupting, cancelling

ESC  
~?  
~L

insert *text*  
Change word to *new*  
pluralize word  
transpose characters  
end insert or incomplete cmd (delete or rubout) interrupts  
reprint screen if ~? scrambles it

## File manipulation

:w  
:wq  
:q  
:q!  
:e *name*  
:e + *name*  
:e + *n*  
:e #  
:~  
:w *name*  
:w! *name*  
:sh  
:~:cmd  
:n  
:in *args*  
:if  
:~G  
:ta *tag*  
:]

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

## Positioning within file

~F  
~B  
~D  
~U  
G  
/pat  
?pat  
n  
N  
/pat/+n  
?pat?-n  
]]  
[[  
%

forward screenfull  
backward screenfull  
scroll down half screen  
scroll up half screen  
goto line (end default)  
next line matching *pat*  
prev line matching *pat*  
repeat last / or ?  
reverse last / or ?  
n<sup>th</sup> line after *pat*  
n<sup>th</sup> line before *pat*  
next section/function  
previous section/function  
find matching ( ) { } [ or ]

## Adjusting the screen

~L  
~R  
zCR  
z-  
z.  
/pat/z-  
zn.  
~E  
~Y

clear and redraw  
retype, eliminate @ lines  
redraw, current at window top  
... at bottom  
... at center  
*pat* line at bottom  
use *n* line window  
scroll window down 1 line  
scroll window up 1 line

## Marking and returning

‘  
‘  
’’  
mx  
‘x  
?x

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

## Line positioning

H  
L  
M  
+  
-  
CR  
j or ↓  
k or ↑

home window line  
last window line  
middle window line  
next line, at first non-white  
previous line, at first non-white  
return, same as +  
next line, same column  
previous line, same column

## Character positioning

~  
0  
\$  
l or →  
h or ←  
~[[  
space  
fx  
Fx  
tx  
Tx  
;  
,  
nl  
%

first non-white  
beginning of line  
end of line  
right  
left  
same as h  
same as l  
find *x* forward  
find *x* backward  
find just before *x* forward  
find just before *x* backward  
repeat last **F F** t or **T**  
inverse of ;  
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 previous 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  
tr 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  
"xp put from buffer x  
"xy yank to buffer x  
"xd delete into buffer x

## Undo, redo, retrieve

u undo last change  
U restore current line  
· repeat last change  
"dp retrieve dth 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 noop- tion)

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 show line numbers  
paragraphs para macro names that start paragraph  
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><read-str>  
When <enter-str> is typed,  
replace by <read-str>  
:ab move more corrects spelling mistake  
:ab use *Transactions on Software Engineering*  
expands tse  
:map char commands (re)defines char  
:map ; : is now equivalent to :  
:map g :l,\$ s/ frequent command  
:map F ifor (i = 1; i <= N; i++)~M~MESCkN  
a C for loop  
:map B i{ESCea}ESC LaTeX boldface a word  
:map F i!fnt -70~M Format a paragraph

Jeff Offutt, 1995