

vi Editor Cheat Sheet

ACNS Bulletin ED-03 — February 1995

Invoking vi: `vi filename`

Command format: `[count] [command]` — count repeats the effect of the command.

Command mode vs. input mode

vi starts in **command mode**. Positioning commands work only in command mode. Entering any input command (see below) switches vi to **input mode**, where every character you type is added to the file and no commands can be executed. Press **Esc** to return to command mode — pressing RETURN does *not* leave input mode.

File management

Command	Action
<code>:w name</code>	Write edit buffer to file name
<code>:wq</code>	Write to file and quit
<code>:q!</code>	Quit without saving changes
<code>ZZ</code>	Same as <code>:wq</code>
<code>:sh</code>	Execute shell commands (Ctrl-d to return)

Window motions

Command	Action
Ctrl-d	Scroll down (half a screen)
Ctrl-u	Scroll up (half a screen)
Ctrl-f	Page forward
Ctrl-b	Page backward
/string	Search forward
?string	Search backward
Ctrl-l	Redraw screen
Ctrl-g	Display current line number and file information
n	Repeat search
N	Repeat search, reverse direction
G	Go to last line
nG	Go to line n
:n	Go to line n
z<CR>	Reposition window: cursor at top
z.	Reposition window: cursor in middle
z-	Reposition window: cursor at bottom

Cursor motions

Command	Action
H	Upper left corner (home)
M	Middle line
L	Lower left corner
h	Back a character
j	Down a line
k	Up a line
l	Forward a character
^	Beginning of line
\$	End of line
w	One word forward
b	Back one word
f c	Find c
;	Repeat find (find next c)

Input commands (end with Esc)

Command	Action
a	Append after cursor
i	Insert before cursor
o	Open line below
O	Open line above
:r file	Insert file after current line

Change commands (input mode)

Command	Action
cw	Change word (Esc)
cc	Change line (Esc) — blanks line
c\$	Change to end of line
rc	Replace character with c
R	Replace (Esc) — typeover
s	Substitute (Esc) — 1 char with string
S	Substitute (Esc) — rest of line with text
.	Repeat last change

Changes during insert mode

Command	Action
Ctrl-h	Back one character
Ctrl-w	Back one word
Ctrl-u	Back to beginning of insert

Deletion

Command	Action
dd or ndd	Delete n lines to general buffer
dw	Delete word to general buffer
dnw	Delete n words
d)	Delete to end of sentence
db	Delete previous word
D	Delete to end of line
x	Delete character

Recovering deletions

Command	Action
p	Put general buffer after cursor
P	Put general buffer before cursor

Undo

Command	Action
u	Undo last change
U	Undo all changes on line

Rearrangement

Command	Action
yy or Y	Yank (copy) line to general buffer
"z6yy	Yank 6 lines to buffer z
yw	Yank word to general buffer
"a9dd	Delete 9 lines to buffer a
"A9dd	Delete 9 lines; append to buffer a
"ap	Put text from buffer a after cursor
p	Put general buffer after cursor
P	Put general buffer before cursor
J	Join lines

Parameters

Command	Action
<code>:set list</code>	Show invisible characters
<code>:set nolist</code>	Don't show invisible characters
<code>:set number</code>	Show line numbers
<code>:set nonumber</code>	Don't show line numbers
<code>:set autoindent</code>	Indent after carriage return
<code>:set noautoindent</code>	Turn off autoindent
<code>:set showmatch</code>	Show matching sets of parentheses as they are typed
<code>:set noshowmatch</code>	Turn off showmatch
<code>:set showmode</code>	Display mode on last line of screen
<code>:set noshowmode</code>	Turn off showmode
<code>:set all</code>	Show values of all possible parameters

Move text from file old to file new

```
vi old
"a10yy      yank 10 lines to buffer a
:w          write work buffer
:e new      edit new file
"ap         put text from a after cursor
:30,60w new write lines 30 to 60 in file new
```

Regular expressions (search strings)

Pattern	Matches
^	Beginning of line
\$	End of line
.	Any single character
*	Any number of the previous character
.*	Any character

Search and replace

Syntax: :[address]s/old_text/new_text/

Address components

Address	Meaning
.	Current line
n	Line number n
.+m	Current line plus m lines
\$	Last line
/string/	A line that contains string
%	Entire file
[addr1],[addr2]	Specifies a range

Examples

Replace only the first occurrence of Banana with Kumquat on each of 11 lines, starting with the current line (.) and continuing for the 10 that follow (.*+10):

```
:. ,.+10s/Banana/Kumquat/
```

Replace every occurrence (the trailing g) of apple with pear throughout the file:

```
:%s/apple/pear/g
```

Remove the last character from every line in the file — useful when every line ends with ^M after a file transfer. Run it with the cursor on the first line:

```
:%s/.$//
```