

# vi Editor "Cheat Sheet"

Invoking vi: `vi filename`

Format of vi commands: `[count][command]` (count repeats the effect of the command)

---

## Command mode versus input mode

Vi starts in command mode. The positioning commands operate only while vi is in command mode. You switch vi to input mode by entering any one of several vi input commands. (See next section.) Once in input mode, any character you type is taken to be text and is added to the file. You cannot execute any commands until you exit input mode. To exit input mode, press the escape (**Esc**) key.

---

## Input commands (end with Esc)

a	Append after cursor
i	Insert before cursor
o	Open line below
O	Open line above
:r <i>file</i>	Insert <i>file</i> after current line

Any of these commands leaves vi in input mode until you press **Esc**. Pressing the **RETURN** key will not take you out of input mode.

---

## Change commands (Input mode)

cw	Change word (Esc)
cc	Change line (Esc) - blanks line
c\$	Change to end of line
rc	Replace character with <i>c</i>
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

<ctrl>h	Back one character
<ctrl>w	Back one word
<ctrl>u	Back to beginning of insert

---

## File management commands

:w <i>name</i>	Write edit buffer to file <i>name</i>
:wq	Write to file and quit
:q!	Quit without saving changes
ZZ	Same as :wq
:sh	Execute shell commands (<ctrl>d)

---

## Window motions

<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
G	Go to last line
nG	Go to line <i>n</i>
:n	Go to line <i>n</i>
z<CR>	Reposition window: cursor at top
z.	Reposition window: cursor in middle
z-	Reposition window: cursor at bottom

---

## Cursor motions

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

---

## Deletion commands

---

dd or ndd	Delete <i>n</i> lines to general buffer
dw	Delete word to general buffer
d <i>n</i> w	Delete <i>n</i> words
d)	Delete to end of sentence
db	Delete previous word
D	Delete to end of line
x	Delete character

---

## Recovering deletions

---

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

---

## Undo commands

---

u	Undo last change
U	Undo all changes on line

---

## Rearrangement commands

---

yy or Y	Yank (copy) line to general buffer
“z6yy	Yank 6 lines to buffer <i>z</i>
yw	Yank word to general buffer
“a9dd	Delete 9 lines to buffer <i>a</i>
“A9dd	Delete 9 lines; Append to buffer <i>a</i>
“ap	Put text from buffer <i>a</i> after cursor
p	Put general buffer after cursor
P	Put general buffer before cursor
J	Join lines

---

## Parameters

---

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

---

## Move text from file *old* to file *new*

---

vi <i>old</i>	
“a10yy	yank 10 lines to buffer <i>a</i>
:w	write work buffer
:e <i>new</i>	edit new file
“ap	put text from <i>a</i> after cursor
:30,60w <i>new</i>	Write lines 30 to 60 in file <i>new</i>

---

## Regular expressions (search strings)

---

^	Matches beginning of line
\$	Matches end of line
.	Matches any single character
*	Matches any previous character
.*	Matches any character

---

## Search and replace commands

---

Syntax:

: [address]s/old\_text/new\_text/

Address components:

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

Examples:

The following example replaces only the **first** occurrence of Banana with Kumquat in each of 11 lines starting with the current line (.) and continuing for the 10 that follow (,+10).

:. . .+10s/Banana/Kumquat

The following example replaces **every** occurrence (caused by the *g* at the end of the command) of apple with pear.

:%s/apple/pear/g

The following example removes the last character from every line in the file. Use it if every line in the file ends with ^M as the result of a file transfer. Execute it when the cursor is on the first line of the file.

:%s/.\$//

# Vi Editor Cheat Sheet

## Movement Commands

### Character

**h, j, k, l** Left, down, up, right

### Text

**w, W, b, B** Forward, backward by word

**e, E** End of word

**(, )** Beginning of next, previous sentence

**{, }** Beginning of next, previous paragraph

**[, ]** Beginning of next, previous section

### Lines

**0, \$** First, last position of current line

**^** First non-blank character of current line

**+, -** First character of next, previous line

**H** Top line of screen

**M** Middle line of screen

**L** Last line of screen

**nH, nL** Line *n* from top, bottom of screen

### Scrolling

**[Ctrl]F, [Ctrl]B** Scroll forward, backward one screen

**[Ctrl]D, [Ctrl]U** Scroll down, up one-half screen

**[Ctrl]E, [Ctrl]Y** Show one more line at bottom, top of window

**z[Enter]** Scroll until line with cursor is at top of screen

**z.** Scroll until line with cursor is at middle of screen

**z-** Scroll until line with cursor is at bottom of screen

### Searches

**/pattern** Search forward for *pattern*

**?pattern** Search backward for *pattern*

**n, N** Repeat last search in same, opposite direction

**/, ?** Repeat previous search forward, backward

**fx** search forward for character *x* in current line

**Fx** search backward for character *x* in current line

**tx** search forward for character before *x* in current line

**Tx** search backward for character after *x* in current line

**;** Repeat previous current-line search

**,** Repeat previous current-line search in opposite direction

### Line Number

**[Ctrl]G** Display current line number

**nG** Move to line number *n*

**G** Move to last line in file

**:n** move to line number *n*

### Marking Position

**mx** Mark current position as *x*

**`x** Move cursor to *x*

**``** Return to previous mark or context

**'x** Move to beginning of line containing mark *x*

**''** Return to beginning of line containing previous mark

---

## Editing Commands

### Insert

**i, a** Insert text before, after cursor

**I, A** Insert text at beginning, end of line

**o, O** Open new line for text below, above cursor

### Change

**r** Replace with next typed character

<b>~</b>	Change between uppercase and lowercase
<b>cm</b>	Change text block defined by movement command <i>m</i> (e.g., cw changes next word)
<b>cc</b>	Change current line
<b>C</b>	Change to end of line
<b>R</b>	Type over characters
<b>s</b>	Delete character and continue typing
<b>S</b>	Delete current line and continue typing
<b>Delete, Move</b>	
<b>x</b>	Delete character
<b>X</b>	Delete character to the left of the cursor
<b>dm</b>	Delete text block defined by movement command <i>m</i> (e.g., dw deletes next word)
<b>dd</b>	Delete current line
<b>D</b>	Delete to end of line
<b>p, P</b>	Put deleted text before, after cursor
<b>"np</b>	Put text from delete buffer number <i>n</i> after cursor (for last nine deletions)
<b>Yank (copy)</b>	
<b>ym</b>	Yank (copy) text block defined by movement command <i>m</i> (e.g., yw yanks next word)
<b>yy, Y</b>	Yank current line
<b>"aYY</b>	Yank current line into named buffer <i>a</i>
<b>p, P</b>	Put yanked text before, after cursor
<b>"aP</b>	Put text from buffer <i>a</i> before cursor
<b>Other Commands</b>	
<b>.</b>	Repeat last edit command
<b>u</b>	Undo last edit
<b>U</b>	Undo changes to current line
<b>J</b>	Join two lines
<b>[Ctrl]L, [Ctrl]R</b>	Redraw screen

---



---

## Invoking vi

<b>vi <i>file</i></b>	Invoke vi editor on <i>file</i>
<b>vi <i>file1 file2</i></b>	Invoke vi editor on files sequentially
<b>view <i>file</i></b>	Invoke vi editor on <i>file</i> in read-only mode
<b>vi -R <i>file</i></b>	Invoke vi editor on <i>file</i> in read-only mode
<b>vi -r <i>file</i></b>	Recover <i>file</i> and recent edits after system crash
<b>vi + <i>file</i></b>	Open <i>file</i> at last line
<b>vi +n <i>file</i></b>	Open <i>file</i> at line number <i>n</i>
<b>vi +/<i>pattern file</i></b>	Open <i>file</i> at <i>pattern</i>

---



---

## Exit and Save Commands

<b>ZZ</b>	Save file and quit
<b>:x</b>	Save file and quit
<b>:wq</b>	Save ("write") file and quit
<b>:w</b>	Save file
<b>:w!</b>	Save file (overriding protection)
<b>:30,60w <i>newfile</i></b>	Save lines 30 through 60 as file <i>newfile</i>
<b>:30,60w&gt;&gt; <i>file</i></b>	Append lines 30 through 60 to file <i>file</i>
<b>:w <i>%</i>.<i>new</i></b>	Save current buffer named <i>file</i> as <i>file.new</i>
<b>:q</b>	Quit
<b>:q!</b>	Quit, discarding any changes
<b>Q</b>	Quit vi and invoke ex
<b>:e <i>file2</i></b>	Edit <i>file2</i> without leaving vi
<b>:e! <i>file2</i></b>	Discard changes to current file, then edit <i>file2</i> without leaving vi
<b>:n</b>	Edit next file
<b>:e!</b>	Discard all changes since last save
<b>:e#</b>	Edit alternate file

---



---