

Vi Cheat Sheet

- [Modes](#)
- [Quitting](#)
- [Inserting Text](#)
- [Motion](#)
- [Deleting Text](#)
- [Yanking](#)
- [Changing Text](#)
- [Putting Text](#)
- [Buffers](#)
- [Markers](#)
- [Searching](#)
- [Replacing Text](#)
- [Regular Expressions](#)
- [Counts](#)
- [Ranges](#)
- [Files](#)
- [Other](#)

[Click here](#) for the Advanced VI Cheatsheet

Modes

Vi has two modes insertion mode and command mode. The editor begins in command mode, where the cursor movement and text deletion and pasting occur. Insertion mode begins upon entering an insertion or change command. [ESC] returns the editor to command mode (where you can quit, for example by typing :q!). Most commands execute as soon as you type them except for "colon" commands which execute when you press the return key.

Quitting

:x	Exit, saving changes
:q	Exit as long as there have been no changes
ZZ	Exit and save changes if any have been made
:q!	Exit and ignore any changes

Inserting Text

i	Insert before cursor
I	Insert before line
a	Append after cursor
A	Append after line
o	Open a new line after current line
O	Open a new line before current line
r	Replace one character
R	Replace many characters

Motion

h	Move left
j	Move down
k	Move up
l	Move right
w	Move to next word
W	Move to next blank delimited word
b	Move to the beginning of the word

B	Move to the beginning of blank delimited word
e	Move to the end of the word
E	Move to the end of Blank delimited word
(Move a sentence back
)	Move a sentence forward
{	Move a paragraph back
}	Move a paragraph forward
0	Move to the begining of the line
\$	Move to the end of the line
1G	Move to the first line of the file
G	Move to the last line of the file
nG	Move to nth line of the file
:n	Move to nth line of the file
fc	Move forward to c
Fc	Move back to c
H	Move to top of screen
M	Move to middle of screen
L	Move to botton of screen
%	Move to associated (), { }, []

Deleting Text

Almost all deletion commands are performed by typing d followed by a motion. For example, dw deletes a word. A few other deletes are:

x	Delete character to the right of cursor
X	Delete character to the left of cursor
D	Delete to the end of the line
dd	Delete current line
:d	Delete current line

Yanking Text

Like deletion, almost all yank commands are performed by typing y followed by a motion. For example, y\$ yanks to the end of the line. Two other yank commands are:

yy	Yank the current line
:y	Yank the current line

Changing text

The change command is a deletion command that leaves the editor in insert mode. It is performed by typing c followed by a motion. For wxample cw changes a word. A few other change commands are:

C	Change to the end of the line
cc	Change the whole line

Putting text

p	Put after the position or after the line
P	Put before the position or before the line

Buffers

Named buffers may be specified before any deletion, change, yank or put command. The general prefix has the form "c where c is any lowercase character. for example, "adw deletes a word into buffer a. It may thereafter be put back into text with an appropriate "ap.

Markers

Named markers may be set on any line in a file. Any lower case letter may be a marker name. Markers may also be used as limits for ranges.

mc	Set marker <i>c</i> on this line
`c	Go to beginning of marker <i>c</i> line.
'c	Go to first non-blank character of marker <i>c</i> line.

Search for strings

/string	Search forward for <i>string</i>
?string	Search back for <i>string</i>
n	Search for next instance of <i>string</i>
N	Search for previous instance of <i>string</i>

Replace

The search and replace function is accomplished with the :s command. It is commonly used in combination with ranges or the :g command (below).

:s/pattern/string/flags	Replace <i>pattern</i> with <i>string</i> according to <i>flags</i> .
g	Flag - Replace all occurrences of pattern
c	Flag - Confirm replaces.
&	Repeat last :s command

Regular Expressions

.	(dot) Any single character except newline
*	zero or more occurrences of any character
[...]	Any single character specified in the set
[^...]	Any single character not specified in the set
^	Anchor - beginning of the line
\$	Anchor - end of line
\<	Anchor - beginning of word

\>	Anchor - end of word
\(...\)	Grouping - usually used to group conditions
\n	Contents of n th grouping

[...] - Set Examples

[A-Z]	The SET from Capital A to Capital Z
[a-z]	The SET from lowercase a to lowercase z
[0-9]	The SET from 0 to 9 (All numerals)
[./=+]	The SET containing . (dot), / (slash), =, and +
[-A-F]	The SET from Capital A to Capital F and the dash (dashes must be specified first)
[0-9 A-Z]	The SET containing all capital letters and digits and a space
[A-Z][a-zA-Z]	In the first position, the SET from Capital A to Capital Z In the second character position, the SET containing all letters

Regular Expression Examples

/Hello/	Matches if the line contains the value Hello
/^TEST\$/	Matches if the line contains TEST by itself
/^[a-zA-Z]/	Matches if the line starts with any letter
/^[a-z].*/	Matches if the first character of the line is a-z and there is at least one more of any character following it
/2134\$/	Matches if line ends with 2134
/\ (21 35)/	Matches if the line contains 21 or 35 Note the use of () with the pipe symbol to specify the 'or' condition
/[0-9]*/	Matches if there are zero or more numbers in the line
/^[^#]/	Matches if the first character is not a # in the line
Notes: 1. Regular expressions are case sensitive 2. Regular expressions are to be used where <i>pattern</i> is specified	

Counts

Nearly every command may be preceded by a number that specifies how many times it is to be performed. For example, 5dw will delete 5 words and 3fe will move the cursor forward to the 3rd occurrence of the letter e. Even insertions may be repeated conveniently with this method, say to insert the same line 100 times.

Ranges

Ranges may precede most "colon" commands and cause them to be executed on a line or lines. For example :3,7d would delete lines 3-7. Ranges are commonly combined with the :s command to perform a replacement on several lines, as with :.,\$s/pattern/string/g to make a replacement from the current line to the end of the file.

:n,m	Range - Lines n-m
:.:	Range - Current line
:\$	Range - Last line
: 'c	Range - Marker c
:%	Range - All lines in file
:g/pattern/	Range - All lines that contain <i>pattern</i>

Files

:w file	Write to <i>file</i>
---------	----------------------

:r <i>file</i>	Read <i>file</i> in after line
:n	Go to next file
:p	Go to previos file
:e <i>file</i>	Edit <i>file</i>
!! <i>program</i>	Replace line with output from <i>program</i>

Other

~	Toggle upp and lower case
J	Join lines
.	Repeat last text-changing command
u	Undo last change
U	Undo all changes to line

[Return to the top](#)

Page produced by Lagmonster - Oct 2000