

## < The Vim commands cheat sheet - 1.2 >

A Vim cheat sheet, listing some useful, essential and most often used Vim commands.

Author: Nana Långstedt < nana.langstedt at gmail.com >

tuXfile created: 18 January 2003

Last modified: 5 September 2009

### contents

- ✳ Working with files
- ✳ Moving around in the file
- ✳ Inserting and overwriting text
- ✳ Deleting text
- ✳ Entering visual mode
- ✳ Editing blocks of text
- ✳ Undo and redo
- ✳ Search
- ✳ Replace

## < The list of Vim commands >

<b>Working with files</b>	
<i>Vim command</i>	<i>Action</i>
:e filename	Open a new file. You can use the Tab key for automatic file name completion, just like at the shell command prompt.
:w filename	Save changes to a file. If you don't specify a file name, Vim saves as the file name you were editing. For saving the file under a different name, specify the file name.
:q	Quit Vim. If you have unsaved changes, Vim refuses to exit.
:q!	Exit Vim without saving changes.
:wq	Write the file and exit.
:x	Almost the same as :wq, write the file and exit if you've made changes to the file. If you haven't made any changes to the file, Vim exits without writing the file.
<b>Moving around in the file</b>	
These Vim commands and keys work both in command mode and visual mode.	
<i>Vim command</i>	<i>Action</i>
j or Up Arrow	Move the cursor up one line.
k or Down Arrow	Down one line.
h or Left Arrow	Left one character.
l or Right Arrow	Right one character.
e	To the end of a word.
E	To the end of a whitespace-delimited word.
b	To the beginning of a word.
B	To the beginning of a whitespace-delimited word.

0	To the beginning of a line.
^	To the first non-whitespace character of a line.
\$	To the end of a line.
H	To the first line of the screen.
M	To the middle line of the screen.
L	To the the last line of the screen.
:n	Jump to line number <i>n</i> . For example, to jump to line 42, you'd type :42
<b>Inserting and overwriting text</b>	
<i>Vim command</i>	<i>Action</i>
i	Insert before cursor.
I	Insert to the start of the current line.
a	Append after cursor.
A	Append to the end of the current line.
o	Open a new line below and insert.
O	Open a new line above and insert.
C	Change the rest of the current line.
r	Overwrite one character. After overwriting the single character, go back to command mode.
R	Enter insert mode but replace characters rather than inserting.
The ESC key	Exit insert/overwrite mode and go back to command mode.
<b>Deleting text</b>	
<i>Vim command</i>	<i>Action</i>
x	Delete characters under the cursor.
X	Delete characters before the cursor.
dd or :d	Delete the current line.
<b>Entering visual mode</b>	
<i>Vim command</i>	<i>Action</i>
v	Start highlighting characters. Use the normal <a href="#">movement keys and commands</a> to select text for highlighting.
V	Start highlighting lines.
The ESC key	Exit visual mode and return to command mode.
<b>Editing blocks of text</b>	
Note: the Vim commands marked with (V) work in visual mode, when you've selected some text. The other commands work in the command mode, when you haven't selected any text.	
<i>Vim command</i>	<i>Action</i>
~	Change the case of characters. This works both in visual and command mode. In visual mode, change the case of highlighted characters. In command mode, change the case of the character uder cursor.
	Shift right (indent).

> (V)	
< (V)	Shift left (de-indent).
c (V)	Change the highlighted text.
y (V)	Yank the highlighted text. In Windows terms, "copy the selected text to clipboard."
d (V)	Delete the highlighted text. In Windows terms, "cut the selected text to clipboard."
yy or :y or Y	Yank the current line. You don't need to highlight it first.
dd or :d	Delete the current line. Again, you don't need to highlight it first.
p	Put the text you yanked or deleted. In Windows terms, "paste the contents of the clipboard". Put characters after the cursor. Put lines below the current line.
P	Put characters before the cursor. Put lines above the current line.
<b>Undo and redo</b>	
<i>Vim command</i>	<i>Action</i>
u	Undo the last action.
U	Undo all the latest changes that were made to the current line.
Ctrl + r	Redo.
<b>Search</b>	
<i>Vim command</i>	<i>Action</i>
/pattern	Search the file for <i>pattern</i> .
n	Scan for next search match in the same direction.
N	Scan for next search match but opposite direction.
<b>Replace</b>	
<i>Vim command</i>	<i>Action</i>
:r/foo/bar/a	Substitute <i>foo</i> with <i>bar</i> . <i>r</i> determines the range and <i>a</i> determines the arguments.
The range ( <i>r</i> ) can be	
nothing	Work on current line only.
number	Work on the line whose number you give.
%	The whole file.
Arguments ( <i>a</i> ) can be	
g	Replace all occurrences in the line. Without this, Vim replaces only the first occurrences in each line.
i	Ignore case for the search pattern.
I	Don't ignore case.
c	Confirm each substitution. You can type y to substitute this match, n to skip this match, a to substitute this and all the remaining matches ("Yes to all"), and q to quit substitution.
<b>Examples</b>	
:452s/foo/bar/	Replace the first occurrence of the word <i>foo</i> with <i>bar</i> on line number 452.
:s/foo/bar/g	Replace every occurrence of the word <i>foo</i> with <i>bar</i> on current line.
:%s/foo/bar/g	Replace every occurrence of the word <i>foo</i> with <i>bar</i> in the whole file.

<code>:%s/fool/bar/gi</code>	The same as above, but ignore the case of the pattern you want to substitute. This replaces <i>foo</i> , <i>FOO</i> , <i>Foo</i> , and so on.
<code>:%s/fool/bar/gc</code>	Confirm every substitution.
<code>:%s/fool/bar/c</code>	For each line on the file, replace the first occurrence of <i>foo</i> with <i>bar</i> and confirm every substitution.

[Linux help](#) > [Tips and cheat sheets](#) > **The Vim commands cheat sheet**

# 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
dnw	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).

: ., .+10 s/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/.\$//