

VI Commands Cheat Sheets

Before doing anything to a document, type the following command followed by a carriage return:

:set showmode

GOOD PRACTICE NOTE ESPECIALLY FOR BEGINNERS: WHEN USING VI, HIT [ESC] TWICE BEFORE EVERY NEW COMMAND. THIS MAKES SURE YOU AREN'T IN THE WRONG MODE.

When you open VI, you can't just start typing a thesis because VI has different modes such as APPEND MODE, INSERT MODE, REPLACE MODE, and COMMAND MODE. So to do certain actions, you must first enter the appropriate mode. See "MODES" for more in-depth information.

The final page of this document is a list of VI core commands. These are the bare essentials of VI compressed onto one page.

STARTING VI (page 1/1 total) (VI is CaSe SEnsItiVe!!! So make sure Caps Lock is OFF.)

| <u>Command</u> | <u>Result</u> |
|------------------------------------|---|
| <code>vi filename</code> | Edits filename. |
| <code>vi -r filename</code> | Edits last saved version of <i>filename</i> after a crash. |
| <code>vi + n filename</code> | Edits filename and places the cursor at line n. |
| <code>vi + filename</code> | Edits filename and places cursor on the last line. |
| <code>vi filename file2 ...</code> | Edits filename and then edits file2 and so on. After saving <i>file1</i> enter :n for the next file. |
| <code>vi +/string file</code> | Edits file and places cursor at the first line containing string. |

ENDING VI (1/1) (Saving, exiting, etc.)

| <u>Command</u> | <u>Result</u> |
|--|--|
| <code>ZZ</code> or <code>:wq</code> or <code>:x</code> | Saves and exits VI. |
| <code>:w</code> | Saves current file but does not exit. |
| <code>:w file</code> | Saves current as file but does not exit. |
| <code>:w! file</code> | Saves file overriding normal checking. |
| <code>:n,mw file</code> | Saves lines n through m to a file named <i>file</i> . |
| <code>:n,mw>>file</code> | Appends lines n through m to the end of a file named <i>file</i> . |
| <code>:q</code> | Quits VI, saving changes before leaving. (You may be prompted to save.) |
| <code>:q!</code> | Quits VI without saving changes. |
| <code>Q</code> | Escapes VI into ex editor with the same file; :vi returns. |
| <code>:e!</code> | Re-edits current file disregarding changes since last save. |
| <code>:we!</code> | Re-edits current file saving changes. |

STATUS (1/1) (Line numbers, etc.)

| <u>Command</u> | <u>Result</u> |
|-----------------|---|
| := | Shows current line number . |
| := | Shows number of lines in file. |
| CTRL-g | Shows filename, current line number, total lines in file, and % of file location . |
| :l (letter "l") | Displays tab (^I) backslash (\) backspace (^H) newline (\$) bell (^G) formfeed (^L) of current line. |

MOVING (1/2) (These will tell you how to get the cursor where you want it—fast.)

MOVING THE CURSOR

Command (ESC exits all modes except the initial Command Mode.)

Result

| | |
|-------------|--|
| arrow keys | These <i>do</i> work, but they may be too slow on big files. |
| h j k l | Left, down, up, and right , respectively. |
| CTRL-d | Moves forward ½ screenful . |
| CTRL-f | Moves forward one entire screenful . |
| CTRL-u | Moves backward ½ screenful . |
| CTRL-b | Moves backward one entire screenful . |
| nG | Moves to line n in the file . |
| G | Moves to the end of file . |
| H | Moves to the top of the screen. |
| nH | Moves to line n from the top of the screen. |
| M | Moves to the middle of the screen. |
| L | Moves to the bottom of the screen. |
| nL | Moves to line n from the bottom of the screen. |
| w or W | Moves to the start of the next word in the line; W ignores punctuation. |
| b or B | Moves to the start of the previous word in the line; B ignores punctuation. |
| e or E | Moves to the end of the next word in the line; E ignores punctuation. |
| 0 (zero) or | Moves to the first column in the current line. |
| n | Moves to the column n in the current line. |
| ^ | Moves to the first non-blank character in the current line. |
| \$ | Moves to the last character in the current line. |
| + or <CR> | Moves to the first character in the next line . |
| - | Moves to the first non-blank character in the previous line . |
| (| Moves back to the beginning of sentence . |
|) | Moves forward to the beginning of the next sentence . |
| { | Moves back to the beginning of the paragraph . |
| } | Moves forward to the beginning of the next paragraph . |

MOVING (2/2) (These will tell you how to get the cursor where you want it—fast.)

MOVING THE SCREEN

| <u>Command</u> (ESC exits all modes except the initial Command Mode.) | <u>Result</u> |
|---|---|
| CTRL-e | Moves screen up one line . |
| CTRL-y | Moves screen down one line . |
| CTRL-u | Moves screen up ½ page . |
| CTRL-d | Moves screen down ½ page . |
| CTRL-b | Moves screen up one page . |
| CTRL-f | Moves screen down one page . |
| CTRL-I | Redraws screen . |
| z <CR> | z—carriage return makes the current line the top line on the page. |
| nz <CR> | Makes the line n the top line on the page. |
| z. | Makes the current line the middle line on the page. |
| nz. | Makes the line n the middle line on the page. |
| z- | Makes the current line the bottom line on the page. |
| nz- | Makes the line n the bottom line on the page. |

MODES (1/1) (Adding and/or replacing text)

| <u>Command</u> (ESC exits all modes except the initial Command Mode.) | <u>Result</u> |
|---|--|
| <i>itext</i> ESC | insert mode —You can start typing and it will insert <i>text</i> <u>before</u> the letter your cursor currently highlights until ESC is used. |
| <i>Itext</i> ESC | insert mode —Same as (i) except it will insert <i>text</i> before the first nonblank character on the line until ESC is used. |
| <i>atext</i> ESC | append mode —Allows you to insert <i>text</i> <u>after</u> the letter your cursor currently highlights until ESC is used. |
| <i>Atext</i> ESC | append mode —Same as (a) except it will insert <i>text</i> at the end of the line until ESC is used. |
| <i>rchar</i> ESC | replace mode —Replaces the currently highlighted character with <i>char</i> until ESC is used. |
| <i>Rtext</i> ESC | replace mode —Same as (r) except it will overwrite until ESC is used. |
| o | Opens new line below the current line and insert until ESC is used. |
| O (letter “O”) | Opens new line above the current line and insert until ESC is used. |
| CTRL-v <i>char</i> | While inserting, ignores special meaning of char (e.g., for inserting characters like ESC and CTRL) until ESC is used. |
| :r <i>file</i> | Reads file and inserts it after current line . |
| :nr <i>file</i> | Reads file and inserts it after line number n . |

SEARCHING (1/1) (Find and/or replace what you need)

| <u>Characters</u> | <u>What they match...</u> |
|-------------------|--|
| ^ (caret) | Matches beginning of line. |
| \$ | Matches end of line. |
| . | Matches any single character. |
| \< | Matches beginning of word. |
| \> | Matches end of word. |
| [<i>str</i>] | Matches any single character in <i>str</i> . |
| [^ <i>str</i>] | Matches any character <u>not</u> in <i>str</i> . |
| [<i>a-n</i>] | Matches any character between <i>a</i> and <i>n</i> . |
| * | Matches zero or more occurrences of previous character in expression. |
| \ | Escapes the meaning of the next character (e.g., \\$ allows you to search for \$). |
| \\ | Escapes the \ character. |

| <u>Command</u> (ESC exits all modes except the initial Command Mode.) | <u>Result</u> |
|---|--|
| % | Searches to beginning of balancing () [] or { }. |
| <i>fchar</i> | Searches forward in current line to <i>char</i> . |
| <i>Fchar</i> | Searches backward in current line to <i>char</i> . |
| <i>tchar</i> | Searches forward in current line to character before <i>char</i> . |
| <i>Tchar</i> | Searches backward in current line to character before <i>char</i> . |
| / <i>str</i> <CR> | Finds forward to <i>str</i> . |
| ? <i>str</i> <CR> | Finds in reverse for <i>str</i> . |
| :set ic | Ignores case when searching. |
| :set noic | Pays attention to case when searching. |
| : <i>n,ms/str1/str2/opt</i> | Searches from n to m for <i>str1</i> ; replaces <i>str1</i> to <i>str2</i> ; using <i>opt-opt</i> can be g for global change, c to confirm change (y to acknowledge, <CR> to suppress), and p to print changed lines. |
| & | Repeats last :s command. |
| : <i>g/str/cmd</i> | Runs cmd on all lines that contain <i>str</i> . |
| : <i>g/str1/s/str2/str3/</i> | Finds the line containing <i>str1</i> , replaces <i>str2</i> with <i>str3</i> . |
| : <i>v/str/cmd</i> | Executes cmd on all lines that do not match <i>str</i> . |

EDITING (1/3) (Deleting, copying, placing text, joining lines, changing text, repeat command, undo command)

| <u>Command</u> (ESC exits all modes except the initial Command Mode.) | <u>Result</u> |
|---|---|
| CTRL-h <u>or</u> Backspace | While inserting , deletes previous character . |
| CTRL-w | While inserting , deletes previous word |
| CTRL-x | While inserting , deletes to start of inserted text . |
| nx | Deletes n characters starting with current ; omitting n deletes current character only. |

EDITING (2/3) (Deleting, copying, placing text, joining lines, changing text, repeat command, undo command)

| Command (ESC exits all modes except the initial Command Mode.) | Result |
|---|---|
| nX | Deletes previous n characters; omitting n deletes previous character only. |
| xp | Switches character at cursor with following character. |
| ndw | Deletes the next n words starting with current; omitting n deletes the current word only. |
| ndb | Deletes the previous n words starting with current; omitting n deletes the previous word only. |
| ndd | Deletes n lines beginning with the current line; omitting n deletes the current line only. |
| :n,md | Deletes lines n through m. |
| D or d\$ | Deletes from the cursor to the end of the line. |
| dcursor_cmd | Deletes everything included in the cursor command (e.g., dG would delete from current position to the end of the file, and d4 would delete to the end of the fourth sentence). |
| nyy or nY | Places n lines in the buffer—copies; omitting n copies only the current line to the buffer. |
| ycursor_cmd | Copies from cursor to cursor_cmd (e.g., yG copies current line to the last line in the file to the buffer). |
| “(a-z)nyy or “(a-z)ndd | Copies or cuts (deletes) n lines into a named buffer a through z; omitting n works on current line. |
| p | Pastes copied text after cursor; also prints last deleted text. |
| P | Pastes copied text before the cursor; also prints last deleted text. |
| “(a-z)p or “(a-z)P | Pastes text from a named buffer a through z after or before the current line. |
| nJ | Joins the next n lines together; omitting n joins the beginning of the next line to the end of the current line. |
| s{text ESC | Substitutes text for the current character until ESC is used. |
| S or cc{text ESC | Substitutes text for the entire line until ESC is used. |
| cw{text ESC | Changes current word to text until ESC is used. |
| C{text ESC | Changes rest of the current line to text until ESC is used. |
| ccursr_cmd{text ESC | Changes to text from current position to cursr_cmd until ESC is used. |
| u | Undoes last command. |
| U | Restores current line to its original state. |
| “np | Retrieves the last nth delete (last 9 deletes are kept in a buffer). |
| “1pu.u. | Scrolls through the delete buffer until the desired delete is retrieved (repeat u.). |
| n | Repeats last / or ? search command. |
| N | Repeats, in reverse order, last / or ? search command. |
| ;(semi-colon) | Repeats last f F t or T search command. |
| ,(comma) | Repeats, in reverse direction, last / or ? search command. |
| .(period) | Repeats last text change command. |
| CTRL-i or TAB | While inserting, inserts one shift width. |

EDITING (3/3) (Deleting, copying, placing text, joining lines, changing text, repeat command, undo command)

Command (ESC exits all modes except the initial Command Mode)

Result

n<< or **n**>> Shifts **n** lines left or right (respectively) by **one shift width**; omitting **n** shifts one line.

< or > Use with cursor command to **shift multiple lines** left or right.

SHELL WORKS (1/1) (Commands that execute outside of the VI program.)

Command (ESC exits all modes except the initial Command Mode)

Result

! cmd Executes shell command **cmd**; you can add these special characters to indicate:
 % name of current file
 # name of last file edited

!! cmd Executes shell command **cmd**, **places output** in file starting at current line.

:!! Executes **last** shell command.

:r! cmd Reads and **inserts output** from **cmd**.

:f file Renames current file to **file**.

:w !cmd Sends currently edited file to **cmd** as standard input and **execute cmd**.

:cd dir Changes current working **directory to dir**.

:sh Starts a **sub-shell** (CTRL-d returns to editor).

:so file Reads and **executes** commands in **file** (*file* is a shell script).

!cursor_cmd cmd Sends **text** from current position to **cursor_cmd** to **shell command cmd**.

!}sort <CR> **Sorts** from current position to end of paragraph and replaces text with sorted text.

MACROS, SUBS, ABBREVIATIONS (1/2) (Macro functions, substitution strings, and abbreviations)

Command (ESC exits all modes except the initial Command Mode.)

Result

:[address]s/search-string/replace-string/[/g] This is the format of a substitute command—description below.

address One line number or two line numbers separated by a comma. A. represents the current line, \$ represents the last line, and % represents the entire file.

search-string A regular expression that can be a simple string of characters.

replace-string The replacement string.

g Indicates a global replacement (more than one replacement per line).

Setting Options (1/2)

| <u>Command</u> | <u>Result</u> | <u>Command</u> | <u>Result</u> |
|---|--|----------------|---|
| NOTE: Map allows you to define strings of VI commands. If you create a file called ".exrc" in your home directory, any map or set command you place inside this file will be executed every time you run VI. To imbed control characters like ESC in the macro, you need to precede them with CTRL-v. If you need to include quotes ("), precede them with a \ (backslash). Unused keys in vi are: K V g q v * = and the function keys. | | | |
| EX: :map v /I CTRL-v ESC dwiYou CTRL-v ESC ESC | | | |
| when v is pressed, search for "I" (/I ESC), delete word (dw), and insert "You" (iYou ESC). CTRL-v allows ESC to be inserted. | | | |
| :map <i>key</i> cmd_seq | Defines <i>key</i> to run cmd_seq when pressed. | :map | Displays all created macros on status line. |

MACROS, SUBS, ABBREVIATIONS (2/2) (Macro functions, substitution strings, and abbreviations)

Setting Options (2/2)

| <u>Command</u> | <u>Result</u> | <u>Command</u> | <u>Result</u> |
|-------------------|---|------------------------------|--|
| :unmap <i>key</i> | Removes macro definition for <i>key</i> . | :ab <i>str</i> string | When <i>str</i> is input, replaces it with string . |
| :ab | Displays all abbreviations. | :una <i>str</i> | Unabbreviates <i>str</i> . |

Options given are default. To change them, enter type :set *option* to turn them on or :set *nooption* to turn them off.

To make them execute every time you open VI, create a file in your HOME directory called .exrc and type the options without the colon (:) preceding the option.

| <u>Command</u> | <u>Result</u> | <u>Command</u> | <u>Result</u> |
|----------------------|--|---------------------------|---|
| :set all | Prints all options to the screen. | :set <i>nooption</i> | Turns off <i>option</i> . |
| :set <i>ai</i> | Turns on auto indentation. | :set <i>ap</i> | Prints line after <i>d c J m :s t u</i> commands. |
| :set <i>bf</i> | Discards control characters from input. | :set <i>eb</i> | Precedes error messages with a bell. |
| :set <i>ic</i> | Ignores case when searching. | :set <i>dir=tmp</i> | Sets directory or buffer file. |
| :set <i>lisp</i> | Modifies brackets for Lisp compatibility. | :set <i>magic</i> | Allows pattern matching with special characters. |
| :set <i>mesg</i> | Allows others to send messages. | :set <i>list</i> | Shows tabs (^I) and end of line (\$). |
| :set <i>nu</i> | Shows line numbers. | :set <i>opt</i> | Speeds output; eliminates automatic RETURN. |
| :set <i>prompt</i> | Prompts for command input with :. | :set <i>re</i> | Simulates smart terminal on dumb terminal. |
| :set <i>report</i> | Indicates largest size of changes reported on status line. | :set <i>ro</i> | Changes file type to "read only." |
| :set <i>scroll=n</i> | set <i>n</i> lines for CTRL-d and z | :set <i>sh=shell_path</i> | set shell escape (default is /bin/sh) |
| :set <i>showmode</i> | Indicates input or replace mode at bottom. | :set <i>sw=n</i> | Sets shift width to <i>n</i> characters. |
| :set <i>term</i> | Prints terminal type. | :set <i>terse</i> | Shorten messages with terse. |
| :set <i>timeout</i> | Eliminates one-second time limit for macros. | :set <i>tl=n</i> | Sets significance of tags beyond <i>n</i> characters (0 means all). |
| :set <i>ts=n</i> | Sets tab stops to <i>n</i> for text input. | :set <i>wa</i> | Inhibits normal checks before write commands. |
| :set <i>warn</i> | Warns "no write since last change." | :set <i>window=n</i> | Sets number of lines in a text window to <i>n</i> . |
| :set <i>wm=n</i> | Sets automatic wraparound <i>n</i> spaces from right margin. | | |

VI Core Commands (The minimum functions)

| COMMAND | RESULT | | |
|---|--|----------------------------|--|
| STARTING VI (VI is CaSe SEnStiVe!!! So make sure Caps Lock is OFF.) | | | |
| <code>vi file</code> | edit file | | |
| ENDING VI (Saving, exiting, etc.) | | | |
| <code>ZZ</code> or <code>:wq</code> or <code>:x</code> | Saves and exits VI. | | |
| <code>:q</code> | Quits VI, saving changes before leaving. (You may be prompted to save.) | | |
| <code>:q!</code> | Quits VI without saving changes. | | |
| MOVING (These will tell you how to get the cursor where you want it—fast.) | | | |
| MOVING THE CURSOR | | | |
| arrow keys | These <i>do</i> work, but they may be too slow on big files. | | |
| <code>h j k l</code> | Left, down, up, and right , respectively. | | |
| <code>CTRL-d</code> | Moves forward ½ screenful . | | |
| <code>CTRL-f</code> | Moves forward 1 whole screenful . | | |
| <code>CTRL-u</code> | Moves backward ½ screenful . | | |
| <code>CTRL-b</code> | Moves backward 1 entire screenful . | | |
| <code>nG</code> | Moves to line n in the file . | | |
| <code>G</code> | Moves to the end of file . | | |
| <code>H</code> | Moves to the top of the screen. | | |
| <code>M</code> | Moves to the middle of the screen. | | |
| <code>L</code> | Moves to the bottom of the screen. | | |
| <code>w</code> or <code>W</code> | Moves to the start of the next word in the line; <code>W</code> ignores punctuation. | | |
| <code>b</code> or <code>B</code> | Moves to the start of the previous word in the line; <code>B</code> ignores punctuation. | | |
| <code>e</code> or <code>E</code> | Moves to the end of the next word in the line; <code>E</code> ignores punctuation. | | |
| <code>^</code> | Moves to the first non-blank character in the current line. | | |
| <code>\$</code> | Moves to the last character in the current line. | | |
| <code>+ or <CR></code> | Moves to the first character in the next line . | | |
| MODES (Adding and/or replacing text) | | | |
| <code>itext ESC</code> | insert mode —You can start typing, and it will insert <i>text</i> before the letter your cursor currently highlights until ESC is used. | | |
| <code>atext ESC</code> | append mode —This allows you to insert <i>text</i> after the letter your cursor currently highlights until ESC is used. | | |
| <code>Atext ESC</code> | append mode —Same as (a) except it will insert <i>text</i> at the end of the line until ESC is used. | | |
| <code>rchar ESC</code> | replace mode —Replaces the currently one highlighted character with <i>char</i> until ESC is used. | | |
| <code>Rtext ESC</code> | replace mode —Same as (r) except it will overwrite until ESC is used. | | |
| <code>o</code> | Opens new line below the current line and insert until ESC is used. | | |
| <code>O (letter "O")</code> | Opens new line above the current line and insert until ESC is used. | | |
| SEARCHING (Find and/or replace what you need) | | | |
| <code>fchar Fchar</code> | Searches forward in current line to <i>char</i> . Searches backward in current line to <i>char</i> . | | |
| <code>/str <CR> ?str <CR></code> | Finds forward to <i>str</i> . Finds in reverse for <i>str</i> . | | |
| EDITING (Deleting, copying, placing text, joining lines, changing text, repeat command, undo command) | | | |
| <code>CTRL-h</code> or <code>Backspace</code> | While inserting , deletes previous character . | | |
| <code>CTRL-w</code> | While inserting , deletes previous word | | |
| <code>nx</code> | Deletes n characters starting with current ; omitting <i>n</i> deletes current character only. | | |
| <code>ndw</code> | Deletes the next n words starting with current ; omitting <i>n</i> deletes the current word only. | | |
| <code>ndd</code> | Deletes n lines beginning with the current line ; omitting <i>n</i> deletes the current line only. | | |
| <code>D</code> or <code>d\$</code> | Deletes from the cursor to the end of the line. | | |
| <code>nyy</code> or <code>nY</code> | Places n lines in the buffer—copies ; omitting <i>n</i> copies only the current line to the buffer. | | |
| <code>p</code> | Pastes copied text after cursor ; also prints last deleted text. | | |
| <code>nJ</code> | Joins the next n lines together ; omitting <i>n</i> joins the beginning of the next line to the end of the current line. | | |
| <code>u</code> | Undoes last command . | | |
| <code>U</code> | Restores current line to its original state. | | |
| <code>n N</code> | Repeats last / or ? search command . Repeats, in reverse order, last / or ? search command . | | |
| <code>.</code> (period) | Repeats last text change command . | | |
| <code>CTRL-i</code> or <code>TAB</code> | While inserting, inserts one shift width . | | |
| OPTIONS | | | |
| Command | Result Command Result | | |
| Options given are default. To change them, enter type <code>:set option</code> to turn them on or <code>:set nooption</code> to turn them off. | | | |
| To make them execute every time you open VI, create a file in your home directory called <code>.exrc</code> and type the options without the colon (<code>:</code>) preceding the option. | | | |
| <code>:set all</code> | Prints all options to the screen. | <code>:set nooption</code> | Turns off <i>option</i> . |
| <code>:set ic</code> | Ignores case when searching. | <code>:set magic</code> | Allows pattern matching with special characters. |
| <code>:set nu</code> | Shows line numbers. | <code>:set ro</code> | Changes file type to “read only.” |
| <code>:set scroll=n</code> | Sets <i>n</i> lines for CTRL-d and z. | <code>:set sw=n</code> | Sets shift width to <i>n</i> characters. |
| <code>:set showmode</code> | Indicates input or replace mode at bottom. | <code>:set window=n</code> | Sets number of lines in a text window to <i>n</i> . |
| <code>:set ts=n</code> | Sets tab stops to <i>n</i> for text input. | <code>:set wm=n</code> | Sets automatic wraparound <i>n</i> spaces from right margin. |