TechRepublic

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**.)

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

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

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

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

Command	Result
:.=	**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** (^l) **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 *do* 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**.

TechRepublic

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

MOVING THE SCREEN

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

Command (**ESC** exits all modes except the initial Command Mode.)	**Result**
i*text* ESC	**insert mode**—You can start typing and it will insert *text* <u>before</u> the letter your cursor currently highlights until ESC is used.
I*text* ESC	**insert mode—**Same as (i) except it will insert *text* before the first nonblank character on the line until ESC is used.
a*text* ESC	**append mode—**Allows you to insert *text* <u>after</u> the letter your cursor currently highlights until ESC is used.
A*text* ESC	**append mode—**Same as (a) except it will insert *text* at the end of the line until ESC is used.
r*char* ESC	**replace mode—**Replaces the currently highlighted character with *char* until ESC is used.
R*text* 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 *char*	While **inserting, ignores special meaning of *char*** (e.g., for inserting characters like ESC and CTRL) until ESC is used.
:r *file*	**Reads** *file* and **inserts** it **after current line**.
:**n**r *file*	**Reads** *file* and **inserts** it **after line number n**.

TechRepublic

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

Characters	What they match...
^ (caret)	Matches beginning of line.
$	Matches end of line.
.	Matches any single character.
\<	Matches beginning of word.
\>	Matches end of word.
[*str*]	Matches any single character in *str*.
[^*str*]	Matches any character <u>not</u> in *str*.
[*a-n*]	Matches any character between *a* and *n*.
*	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.

Command (**ESC** exits all modes except the initial Command Mode.)	Result
%	**Searches to beginning of balancing ( ) [ ] or { }.**
f*char*	**Searches forward** in current line to ***char***.
F*char*	**Searches backward** in current line to ***char.***
t*char*	**Searches forward** in current line to **character before *char***.
T*char*	**Searches backward** in current line to **character before *char***.
/*str* <CR>	**Finds forward** to ***str***.
?*str* <CR>	**Finds** in **reverse** for ***str***.
:set ic	**Ignores case** when **searching**.
:set noic	**Pays attention to case** when **searching**.
:**n,m**s/*str1*/*str2*/*opt*	**Searches** from **n** to **m** for *str1*; **replaces** *str1* to *str2*; using *opt–opt* 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**.
:g/*str*/cmd	**Runs cmd** on **all** lines that **contain *str***.
:g/*str1*/s/*str2*/*str3*/	**Finds** the line containing *str1*, **replaces** *str2* with *str3*.
:v/*str*/cmd	**Executes** cmd on all lines that **do not match** *str*.

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

Command (**ESC** exits all modes except the initial Command Mode**.**)	Result
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,m**d	**Deletes lines n through m**.
D or d$	**Deletes from** the **cursor to** the **end** of the line.
d**cursor_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 **n**Y	**Places n lines in** the **buffer–copies**; omitting **n** copies only the current line to the buffer.
y**cursor_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)**n**yy or "(a-z)**n**dd	**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.
c**cursr_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.
"**n**p	**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)

Command	Result	Command	Result
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 *key* **cmd_seq**	Defines *key* 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)

Command	Result	Command	Result
:unmap *key*	Removes macro definition for key.	:ab *str* **string**	When *str* is input, replaces it with **string**.
:ab	Displays all abbreviations.	:una *str*	Unabbreviates *str*.

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

VI Core Commands (The minimum functions)

COMMAND	RESULT

STARTING VI   (VI is **CaSe SEnsItiVe**!!! So make sure Caps Lock is **OFF**.)

vi *file*	**edit** *file*

ENDING VI   (Saving, exiting, etc.)

ZZ or :wq or :x	**Saves** and **exits VI.**
:q	**Quits** VI**, saving changes** before leaving. (You may be prompted to save.)
:q!	**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 *do* 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 1 whole screenful**.
CTRL-u	Moves **backward ½ screenful**.
CTRL-b	Moves **backward 1 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.
M	Moves to the **middle** of the **screen**.
L	Moves to 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.
^	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**.

MODES   (Adding and/or replacing text)

i*text* ESC	**insert mode**—You can start typing, and it will insert *text* before the letter your cursor currently highlights until ESC is used.
a*text* ESC	**append mode**—This allows you to insert *text* <u>after</u> the letter your cursor currently highlights until ESC is used.
A*text* ESC	**append mode—**Same as (a) except it will insert *text* at the end of the line until ESC is used.
r*char* ESC	**replace mode**—Replaces the currently one highlighted character with *char* until ESC is used.
R*text* 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.

SEARCHING   (Find and/or replace what you need)

f*char* | F*char*	**Searches forward** in current line to *char*. | **Searches backward** in current line to *char*.
/*str* <CR> | ?*str* <CR>	**Finds forward** to *str*. | **Finds in reverse** for *str*.

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

CTRL–h <u>or</u> Backspace	**While inserting,** deletes **previous character**.
CTRL–w	**While inserting,** deletes **previous word**
nx	**Deletes n characters starting with current**; omitting **n** deletes current character only.
ndw	**Deletes** the **next n words starting with current**; omitting **n** deletes the current word only.
ndd	**Deletes n lines beginning with the current line**; omitting **n** deletes the current line only.
D or d$	**Deletes from** the **cursor to** the **end** of the line.
nyy or **n**Y	**Places n lines in** the **buffer–copies**; omitting **n** copies only the current line to the buffer.
p	**Pastes copied text after cursor**; also prints last deleted text.
nJ	**Joins** the **next n lines together**; omitting **n** joins the beginning of the next line to the end of the current line.
u	**Undoes last** command.
U	**Restores current line** to its original state.
n | N	**Repeats last / or ? search command**. | **Repeats,** in **reverse** order, **last / or ? search command**.
. (period)	**Repeats last** text **change command**.
CTRL–i or TAB	While inserting, **inserts one shift width**.

OPTIONS

Command	Result		Command	Result

Options given are default. To change them, enter type :set *option* to turn them on or :set no*option* 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.

Command	Result		Command	Result
:set all	Prints all options to the screen.		:set no*option*	Turns off *option*.
:set *ic*	Ignores case when searching.		:set *magic*	Allows pattern matching with special characters.
:set *nu*	Shows line numbers.		:set *ro*	Changes file type to "read only."
:set *scroll=n*	Sets *n* lines for CTRL–d and z.		:set *sw=n*	Sets shift width to **n** characters.
:set *showmode*	Indicates input or replace mode at bottom.		:set *window=n*	Sets number of lines in a text window to **n**.
:set *ts=n*	Sets tab stops to **n** for text input.		:set *wm=n*	Sets automatic wraparound **n** spaces from right margin.