

Vim Cheatsheet

Itsuki David Fukada, Jesus Christian Reyes

Format of command is: Verb Range Motion

- For example, if you use `d2w` , it will delete 2 words.
- If you do not use any range, it will automatically set it to 1. Also some verb can be used repeatedly.
- Also certain words allow `i` to be a range which `i` stands for "inner." This allows you to have your cursor located anywhere on a word and be able to edit the whole word instead of just your cursor to the end of the word. For example, if your cursor was in the middle of a word and you enter `dw`, you will delete only from your cursor to the end of the word. If you entered `diw`, you will delete the whole word.
- To exit insert or visual mode use `<Esc>`

Verb/Editing

<code>i</code>	Insert mode at current cursor
<code>I</code>	Insert mode at beginning of line
<code>a</code>	Insert mode after the cursor
<code>A</code>	Insert mode at end of line
<code>o</code>	Append a new line below the current line
<code>O</code>	Append a new line above the current line
<code>s</code>	Delete word and enter insert mode
<code>S</code>	Delete line and enter insert mode
<code>f+CHARACTER</code>	Finds first instance of CHARACTER
<code>r</code>	Replace a single character
<code>J</code>	Join the line below with the current
<code>c</code>	Replace (combine with Motion and/or Range)
<code>x</code>	Deletes the character at current cursor (cut)
<code>d</code>	Delete (cut)
<code>D</code>	Deletes from cursor to end of line
<code>y</code>	Yank the highlighted text (copy)
<code>p</code>	Paste the clipboard
<code>u</code>	Undo
<code>U</code>	Undos entire line
	original state
<code>Ctrl+r</code>	Redo
<code>></code>	Indent to the right
<code><</code>	Indent to the left
<code>~</code>	Changes character's case
<code>Ctrl-u</code>	Scroll Up
<code>Ctrl-d</code>	Scroll down
<code>Ctrl-y</code> (Yore)	Scroll up one line
<code>Ctrl-e</code> (Efter)	Scroll down one line

Motion/Navigation

<code>h</code>	Left
<code>j</code>	Down
<code>k</code>	Up
<code>l</code>	Right
<code>w</code>	Jump forward to the start of a word
<code>W</code>	Jump forward to the start of a word (can contain punctuation)
<code>e</code>	Jump forward to the end of a word
<code>E</code>	Jump forward to the end of a word (can contain punctuation)
<code>b</code>	Jump backward to the start of a word
<code>B</code>	Jump backward to the start of a word (can contain punctuation)
<code>0</code>	Jump to the start of the line
<code>\$</code>	Jump to the end of the line
<code>%</code>	Jump to matching bracket
<code>G</code>	Go to end of file
<code>H</code>	(High)Go to top of visible page
<code>M</code>	(Middle)Go to middle of visible page
<code>L</code>	(Low)Go to bottom of visible page
<code>(</code>	Jump forward one sentence
<code>)</code>	Jump backward one sentence
<code>{</code>	Jump forward one paragraph
<code>}</code>	Jump backward one paragraph
<code>``</code>	Return to the line where cursor was before last movement

Other Shortcuts

<code>gg</code>	Go to (e.g. 20gg goes to line 20)
<code>NUMBER+G</code>	to line NUMBER (e.g. 25G)
<code>dd</code>	Delete current line
<code>yy</code>	Copy current line
<code>cc</code>	Change/Replace an entire line

Commands

Save/Quit

:q	Quit
:q!	Quit (Ignores changes made)
:w	Save
:wq (:x)	Save and quit
:w FILENAME	Saves into FILENAME

Substitution/Searching

:s/OLD/NEW/g	Replaces all instances of OLD in current line with NEW
:%s/OLD/NEW/g	Replaces all instances of OLD with NEW
:%s/OLD/NEW/gc	Replaces all instances of OLD with NEW, but asks for confirmation
/TEXT	Search for TEXT
?TEXT	Search for TEXT backwards
n]	Move to next word that matches
N	Move to previous word that matches
:NUMBER	to line NUMBER

Other Commands

:!COMMAND	Runs COMMAND (bash commands)
:r FILENAME	Copy FILENAME into your current file
:r!COMMAND]	Copy the output of COMMAND into current file

Tabs/Split

:tabe	Creates a new tab
:tabe FILENAME	Creates a new tab with name FILENAME
gt	Move to next tab
gT	Move to previous tab
#gt]	Move to tab number

/TEXT
?TEXT

n]

N

:NUMBER

Set Options

:set ic	Turn off case-sensitive search
:set autoindent	Automatically indents
:set cindent	Automatically indents in C style
:set nu	Show line number
:set incsearch	Begins searching when you type first character
:set hls	Matches to search are highlighted

Colorschemes

Visual Commands

v	Start visual mode from cursor(highlight)
Ctrl+v	Highlight blocks of text
V	Highlight lines of text
<	Shift text left
>	Shift text right

:color <TAB> Navigate through default colorschemes
*Add colorschemes in `~/.vim/colors`
or create your own at bytefluent.com/vivify/

Macros

q+CHARACTER	Start recording to register CHARACTER
q	Stop recording
@+CHARACTER	Execute Macro written to CHARACTER
@@	Execute the same macro again

m+LETTER	Set mark LETTER at cursor location (e.g. ma)
'LETTER	Jump to line of mark 'LETTER
`LETTER	Jump to position mark LETTER
d'LETTER	Delete from current line to line of mark LETTER
d`LETTER	Delete from current cursor position to position of mark LETTER
c'LETTER	Change text from current line to line of mark LETTER
y`LETTER	Yank text to buffer from cursor to position mark LETTER
:marks	list all current marks

Marks

Buffers

:e file	Edit existing or new file
:new	Creates new window displaying contents of new buffer
:ls	Lists all buffers
Ctrl-W w	Move cursor to other window
:vsp	Split window vertically
:bd	Delete the current buffer(fails if unsaved)

Line Folding

"Highlighted text" + zf	Folds highlighted text	zm	Increase fold level by one
zf "NUMBER" j	Creates folds from cursor down NUMBER lines	zM	Close all open folds
zf/STRING	Creates fold from cursor to STRING	zr	Decrease fold level by one
zj	Move to next fold	zR	Decrease fold level to 0
zk	Move to previous fold	zd	Delete fold at cursor
zo	Open fold at cursor	zD	Delete all folds
ZO	Open all folds	[z	Move to start of open fold
]z	Move to end of open fold
		:set=foldmethod=indent	Automatically creates folds at every indent