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.(To exit insert or visual mode use <Esc>)
*If you do not use any range, it will automatically set it to 1. Also some verb can be used repeatedily.
*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.

Verb/Editing

`gg`	Go to (e.g. 20gg goes to line 20)
`f+character`	Finds first instance of character
`r`	Replace a single character
`J`	Join the line below with the current
`c`	Replace (combine with Motion and/or Range)
`x`	Deletes the character at current cursor (cut)
`d`	Delete (cut)
`D`	Deletes from cursor to end of line
`y`	Yank the highlighted text (copy)
`p`	Paste the clipboard
`u`	Undo
`U`	Undos entire line original state
`Ctrl+r`	Redo
>	Indent to the right
<	Indent to the left
`~`	Changes character’s case

Insertion/Deletion

`i`	Insert mode at current cursor
`I`	Insert mode at beginning of line
`a`	Insert mode after the cursor
`A`	Insert mode at end of line
`o`	Append a new line below the current line
`O`	Append a new line above the current line
`s`	Delete word and enter insert mode
`S`	Delete line and enter insert mode

Motion/Navigation

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

Other Navigations

`Ctrl-U`	Scroll Up
`Ctrl-D`	Scroll down
`Ctrl-E (Efter)`	Scroll down one line
`Ctrl-Y (Yore)`	croll up one line
`:NUMBER`	to line NUMBER
`NUMBER + G`	to line NUMBER (i.e.: 25G)

Commands

Save/Quit

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

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

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

Colorschemes

`:color <TAB >`	Navigate through default colorschemes

*Add colorschemes in `~/.vim/colors` or create your own at `bytefluent.com/vivify/`

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

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

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

Macros

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

Marks

`m+LETTER`	Set mark `LETTER at cursor` location
`’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’a`	Change text from current line to line of mark `LETTER`
`y‘a`	Yank text to buffer from cursor to position mark `LETTER`

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 (will fail 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
		`zr`	Decrease fold level by one
`zf/STRING`	Creates fold from cursor to `STRING`	`zR`	Decrease fold level to 0
		`zd`	Delete fold at cursor
`zj`	Move to next fold	`zD`	Delete all folds
`zk`	Move to previous fold	`[z`	Move to start of open fold
`zo`	Open fold at cursor	`]z`	Move to end of open fold
`ZO`	Open all folds	`:set=foldmethod=indent`	Automatically creats folds at every indent