vim Quick Reference

Many keys allow a <count> before them, to execute the command multiple times

Many commands work line-wise by doubling them up (e.g. `d` = delete, `dd` = delete line)

Command line commands (starting with `:`) require `<cr>` to run

NW = non-whitespace, (I) = in insert mode

vi Command Line Flags

`vi` <file>	Edit <file>
`vi +`<num> <file>	Edit <file> at line <num>
`vi +` <file>	Edit <file> at last line

Mode Change Commands

`a`	Append text after highlighted character
`A`	Append text after last character on line
`i`	Insert text before highlighted character
`I`	Insert text before first NW character on line
`c`<motion>	Delete up to <motion>, then insert
`C`	Delete rest of line, then insert
`s`	Delete current character, then insert
`S`	Delete current line, then insert
`o`	Add new line after current, then insert
`O`	Add new line before current, then insert
`R`	Enter replace (overtype) mode
`Esc`	Return to normal mode
`v`	Enter / exit visual mode
`V`	Enter / exit linewise visual mode
`gv`	Enter visual mode, remember previous area

Movement Commands

```
  k
h   l
  j
```

Cursor keys

`gk`, `gj`	Up or down one *screen* line
`0`	To first character on line
`^`	To first NW character on line
`$`	To last character on line
`w`	To next word
`b`	To previous word
`e`	To space before next word
`W`	As w (space delimited only)
`B`	As b (space delimited only)
`E`	As e (space delimited only)
`+`, `<CR>`	To next line (first NW)
`-`	To previous line (first NW)
`)`	To next sentence
`(`	To previous sentence
`}`	To next paragraph (i.e. blank line)
`{`	To previous paragraph
`H`	To first line of window
`M`	To middle line of window
`L`	To last line of window
`f`<char>	To next <char> on line (inclusive)
`t`<char>	To next <char> on line(exclusive)
`F`<char>	To previous <char> on line(inc)
`T`<char>	To previous <char> on line (exc)
`;`	Repeat last f, F, t or T
`,`	Reverse of last f, F, t or T
`/`<string><cr>	To next occurrence of <string>
`?`<string><cr>	To previous occurrence of <string>
`n`	Repeat last / or ?
`N`	Reverse of last / or ?
`/`<string>`/e`<cr>	To end of next occurrence
`/`<string>`/e+1`<cr>	To after end of next occurrence
`/`<string>`/+`<num><cr>	Line <num> after <string>
`m`<char>	Mark position with letter <char>
`'`<char>	Jump to mark <char>
` `` `	To position before last mark jump / search
`''`	To line before last jump / search
<num>`G`	To line <num>
`G`	To last line
<num>`|`	To column <num>
`%`	To matching bracket

Window Scrolling

`C-f`	Scroll down (forwards) a page
`C-d`	Scroll down half a page
`C-e`	Scroll down one line
`C-b`	Scroll up (backwards) a page
`C-u`	Scroll up half a page
`C-y`	Scroll up one line
`z`<cr>	Scroll so current line is at top
`z.`	Scroll so current line is at middle
`z-`	Scroll so current line is at bottom

File commands (require <cr> to end)

`:w`	Write current file
`:wa`	Write all files
`:w` <file>	Write to <file>
`:w!` <file>	Write to <file>, force overwrite

`:q`	Quit current file (! forces)
`:qa`	Quit all files (! forces)
`:wq`	Writes file, then quits (! forces)
`:e`	Re-edit (load external changes, ! forces)
`:e` <file>	Edit (open) <file>
`:e` <file>	Edit <file>
`:e +`<num> <file>	Edit at line <num> (default last)
`:enew`	Edit a new file
`:r` <file>	Insert <file> at cursor
`:!` <cmd>	Run <cmd> in shell
`!!` <cmd>	Insert output of <cmd> at cursor

Buffers & Splits

`:ls`	List all open buffers
`:ls!`	List all (including unlisted)
`:b` <num>	Go to buffer <num>
`:sb` <num>	Split window & edit buffer <num>
`:bn`	Go to next buffer in list
`:bp`	Go to previous buffer in list
`:bf`	Go to first buffer in list
`:bl`	Go to last buffer in list
`:bm`	Go to next modified buffer in list
`:ba`	Show all listed buffers on screen in splits
`:new`	Create new file in new split
`:sp` <file>	Open <file> in new split
`:vsp` <file>	Open <file> in new vertical split
`C-w` <cursor key>	Change active split
`:clo`	Close split, hide buffer

Register (clipboard) commands

`d`<motion>	Cut (and delete) <motion>
`dd`	Cut line
`x`	Cut character under cursor
`X`	Cut character before cursor
`y`<motion>	Yank (copy) <motion>
`yy, Y`	Yank line
`p`	Put (paste) after cursor
`P`	Put before cursor
`gp, gP`	Put, leaving cursor after text
`]p, ]P`	Put, adjusting indent to current line
`"`<char><clipbrd action>	Use register <char>
`"_`	Do not store in any register
`:reg, :di`	Display contents of all registers

Macros

`q`<char>	Start recording, storing in register <char>
`q`	Stop recording
`@`<char>	Run macro contained in register <char>
`@@`	Repeat previous @ command

Undoing

`u`	Undo (multiple times)
`U`	Undo all changes to current line
`C-r` or `:redo`	Redo undone changes

Changing Case

`~`	Swap case of current character
`g~`<motion>	Swap case up to <motion>
`g~~`	Swap case of current line
`gu`	Make <motion> text lowercase
`guu`	Make current line lowercase
`gU`	Make <motion> text uppercase
`gUU`	Make current line uppercase

Other Commands

`>`<motion>	Indent up to <motion>
`<`<motion>	Unindent up to <motion>
`J`	Join line to next
`C-a`	Increase number / char under cursor (I)
`C-x`	Decrease number / char under cursor (I)
`*`	Find word under cursor downwards
`#`	Find word under cursor upwards
`g*, g#`	... including part words
`C-n`	Word completion (I)
`C-x C-l`	Line completion (I)
`C-r=7*6`	Insert 42 into document (I)
`:`<range>`s/`<string1>`/`<string2>	Find <string1> in <range>, replace with <string2>
`C-g`	Display current file information

If this key doesn't work:	Try:
Backspace	`C-h`
Tab	`C-i`
Esc	`C-[`

Ranges (two positions, with comma between)

<num>	Absolute line number
`.`	Current line
`$`	Last line in buffer
`%`	Entire buffer
`'`<char>	Position of mark <char>
`/`<string>	Next line where <string> occurs
`?`<string>	Previous line where <string> occurs
`\/`	Next line where previously used search matches
`\?`	As above, but previous line
`\&`	Next line where last substitute pattern matches