vi Command Summary

Movement Commands			Editing Commands		
Character			**Insert**		
h, j, k, l	Left, down, up, right ($\leftarrow$, $\downarrow$, $\uparrow$, $\rightarrow$)		i, a	Insert text before, after cursor	
			I, A	Insert text at beginning, end of current line	
Text			o, O	Open a new line for text below, above cursor	
w, W, b, B	Forward, backward by word		**Change**		
e, E	End of word				
), (	Beginning of next, previous sentence		r	Replace with next typed character	
			~	Change case of character under cursor	
			c*m*	Change text block defined by movement command *m* (cw or cL for word or line)	
Lines					
0, $	First, last position of current line		cc	Change current line	
^	First character of current line (ignore spaces)		C	Change to end of line	
+. -	First character of next, previous line		R	Type over characters	
n	Column *n* of current line		s	Delete character and substitute text	
H	Top line of screen		S	Delete current line and substitute text	
M	Middle line of screen				
L	Bottom line of screen			Note: r and ~ leave you in command mode, all other commands put you in insert mode	
*n*H	*n* (number) of lines after top of screen				
*n*L	*n* (number) of lines before bottom of screen				
Screens			**Delete, Move**		
Ctrl-F, Ctrl-B	Scroll forward, backward one screen		x	Delete character	
Ctrl-D, Ctrl-U	Scroll down, up one-half screen		X	Delete character before cursor	
Ctrl-E, Ctrl-Y	Show one more line at bottom, top of screen		d*m*	Delete text block defined by movement command *m* (dw or d) for word or sentence)	
z RETURN	Reposition current line to top of screen				
z.	Reposition current line to middle of screen		dd	Delete current line	
z-	Reposition current line to bottom of screen		D	Delete to end of current line	
			p, P	Put last delete after, before cursor	
			"*n*p	Put text from delete buffer *n* after cursor	
Searches			**Yank (Copy)**		
/*expr*	Search forward for *expr*		y*m*	Yank (copy) text block defined by movement command *m* (yw or y) or y} for word, sentence or paragraph)	
?*expr*	Search backwards for *expr*				
n, N	Repeat last search in same, opposite direction				
/, ?	Repeat previous search forward, backwards		yy, Y	Yank current line	
f*x*	Search forward for character *x* in current line		"*a*yy	Yank current line to named buffer *a*	
F*x*	Search backwards for character *x* in current line		p, P	Put yanked text after, before cursor	
; (*semicolon*)	Repeat previous current-line search		"*a*P	Put yanked text from named buffer a before cursor	
, (*comma*)	Repeat previous current-line search in opposite direction				
Line Number			**Other Commands**		
Ctrl-G	Show current line number		. (period)	Repeat last edit command	
*n*G	Position cursor on line *n*		u, U	Undo last edit, restore current line	
G	Position cursor on last line of file		J	Join two adjacent lines	
:*n*	Position cursor on line *n*		Ctrl-L, Ctrl-R	Redraw screen	
Marking			Other ex Commands		
m*x*	Mark current position as *x*		**ex Commands**		
`*x*	Return cursor to *x*		:	Invoke *ex* command from *vi*	
``	Return to previous mark		:sh	Invoke Unix shell	
'*x*	Move to beginning of line containing *x*		Ctrl-D	Return to editor from shell	
'' (2 apostrophes)	Return to beginning of line containing previous mark		:! *command*	Execute Unix *command*	
			:r *newfile*	Read *newfile* into current file	
			:r !*command*	Read output of *command* into current file	
Command Line Options			**Abbreviations**		
vi *file*	Start vi and open *file*		:ab *abbr phrase*	Abbreviate *phrase* as *abbr*, when *abbr* is typed in insert mode, it expands to *phrase*	
vi *file1 file2* …	Start vi end edit *files* sequentially				
vi -R *file*, view *file*	Start vi and open *file* read-only		:unab *abbr*	Disable abbreviation for *abbr*	
vi -r *file*	Attempt to recover *file* and recent edits after a system crash		:map *x sequence*	Define keystroke *x* as a command *sequence*	
			:unmap *x*	Disable the definition for *x*	
vi + *file*	Start vi and open *file* at last line		:map! *x sequence*	Define *x* as a command *sequence* in insert mode	
vi +*n file*	Start vi and open *file* at line *n*				
vi +/*expr file*	Start vi and open *file* at first line containing *expr*		:unmap! *x*	Disable insert mode mapping of *x*	
Exit Commands			**Setting Options**		
ZZ	Write (save) and quit		:set *option*	Activate option	
:x	Write (save) and quit		:set *option=value*	Assign value to option	
:wq	Write (save) and quit		:set no*option*	Deactivate option	
:q	Quit		:set	Display options set by user	
:q!	Quit and abandon edits since last write		:set all	Display current list of options, both default and those set by user	
Q	Quit to *ex*				
:e *file2*	Exit current file and begin editing *file2*		:set *option*?	Display value of option	
:e #	Edit alternate file				
:n	Edit next file (when starting with *vi file1 file2* …)			Note: when used in *.exrc* file, omit the leading colon for set commands	
:e!	Return to version of current file at last save				