

Vim Reference Card

a insert text after the cursor position	A insert text at the end of the line (same as \$a)	CTRL-A increment the number under the cursor
b go back a word	B go back a WORD	CTRL-B Move one screen backward
c <i>{motion}</i> change text till <i>motion</i>	C change till end of the line	cc change the whole line
d <i>{motion}</i> delete text till <i>motion</i>	D delete till end of the line	dd delete the whole line
e move to end of the word	E move to end of the WORD	:e filename open <i>filename</i>
f find a char in the same line	F find a char in the same line, backwards	CTRL-F Move one screen forward
gg go to first line	G go to last line	gu <i>{motion}</i> change to lowercase gu <i>{motion}</i> uppercase
gd jump to declaration in the current function	gD jump to global declaration	gf open the file under cursor
h move cursor to the left	H go to the top of the screen	:h online help
i insert text in the cursor position	I insert text at the start of the line (same as ^i)	CTRL-I go to next cursor position
j move down the cursor a line	J join two lines	:%j join all lines into single line
k move up the cursor a line	K jump to hyperlink	CTRL-X CTRL-K complete from dictionary
l move cursor to the right	L go to bottom of the screen	CTRL-X CTRL-L complete whole lines
ma create a mark <i>a</i>	M go to middle of the screen	:m {address} move the lines to specific <i>{address}</i>
n find the next match (used in search)	N find the previous match (used in search)	CTRL-N complete the next word
o create a new line below the cursor	O create a new line above the cursor	CTRL-O go to the previous cursor position
p put the yanked text(copied text)	P put the yanked text <i>before</i> cursor	CTRL-P complete the previous word
qa create a macro named <i>a</i> , press q again to stop recording	Q <i>{motion}</i> format the text till <i>motion</i>	QQ format the current line
r replace a char	R replace several chars; ends with ESC	:r filename read the file content to this file
s delete the char under cursor and jump to insert mode	S delete the whole line and jump to insert mode (cc)	:%s/foo/bar/g replace all <i>foo</i> with <i>bar</i>
t go <i>till</i> the char (similar to f)	T go <i>till</i> the char, backwards	:t same as :m
u undo	U undo all the changes in the line	CTRL-R redo
v go to visual mode (char-wise)	V go to visual mode (line-wise)	:v/foo/d execute d (delete) on lines that <i>not</i> contain <i>foo</i>
w go to the next word	W go to the next WORD	:w write to file
x delete a char under the cursor	X delete a char before the cursor	CTRL-X CTRL-F complete the file name
y <i>{motion}</i> yank text(copy) till <i>motion</i>	Y yank the whole line	yy yank the whole line (same as Y)
zf <i>{motion}</i> create a fold till <i>motion</i>	zF <i>{motion}</i> create a fold till <i>motion</i>	ZZ (uppercase) save and exit

Special Characters

~ change case	xp exchanges two chars	[[go to previous function	[m go to start of block
`` jump to previous location	dwp exchange two words]] go to next function]m go to end of block
! <i>{motion}</i> redirect the output	!! redirects output current line	{ goto previous paragraph	} go to next paragraph
@a play the macro <i>a</i>	@@ play the previous macro	[i find the previous occurrence of word under cursor]i find the next occurrence of word
# search backwards		: start an ex command	
\$ jump to end of line	`` jumps to last cursor position	'a go to mark <i>a</i> at the start of line	`a go to mark <i>a</i>
% jump to the matching (, [, {	:q! really quit, no questions	; find next f or t search (see f or t)	, find previous f or t
^ jump to start of line	:wq write and quit	. repeat the previous change	
& perform the previous substitute	:ls lists buffers	< <i>{motion}</i> shift left till <i>motion</i>	> <i>{motion}</i> shift right till <i>motion</i>
* search the char under cursor	:e# edit alternate file	/ start a search	? start a search backwards
(jump to start of sentence		 goto first non blank char in line	0 (zero) goto first char in the line
) jump to end of sentence		CTRL-P complete previous word in the file(insert mode)	CTRL-N complete the next word in the file
- go to previous line		CTRL-A increase the number under cursor	CTRL-X decrease the number under cursor
+ go to next line		CTRL-O jump to previous location (in history)	CTRL-I jump to next location (in history)
= <i>{motion}</i> indent till <i>motion</i>	== indent the current line	CTRL-X CTRL-L complete the next line(insert mode)	

You can give count to most of the commands (for example **2dd**); it will be executed as many times. Ravi Shankar Sivaraman (ravishjunk@gmail.com); you can distribute freely!

get intellisense for vim <http://insenvim.sourceforge.net> for your favorite languages (java,c++,c#,xml,html, etc)

V 1.01