Alphabetical Index

This file contains a list of all commands for each mode, with a tag and a short description. The list is sorted on ASCII value.

When looking for certain functionality use the search command. For example, to look for deleting something, use: Find "delete".

Sections:

For an overview of options see vim_help.html option_list. For a complete description of each option see vim_ref8.html options.

1. Insert mode

char	action
CTRL-@	insert previously inserted text and stop insert
CTRL-A	insert previously inserted text
CTRL-B	Only if compiled with TOGGLE_REVINS (which is not the default): toggle 'revins' (backwards insert) option.
CTRL-C	quit insert mode, without checking for abbreviation.
CTRL-D	delete one shiftwidth of indent in the current line
CTRL-E	insert the character which is below the cursor
CTRL-F	not used
CTRL-G	not used
<BS>	delete character before the cursor
{char1} <BS>{char2}	enter digraph (only when 'digraph' option set)
CTRL-H	same as <BS>
<Tab>	insert a <Tab> character
CTRL-I	same as <Tab>
<NL>	same as <CR>
CTRL-J	same as <CR>
CTRL-K {char1} {char2}	enter digraph
CTRL-L	not used
<CR>	begin new line
CTRL-M	same as <CR>
CTRL-N	find next match for keyword in front of the cursor

CTRL-O	execute a single command and return to insert mode
CTRL-P	find previous match for keyword in front of the cursor
CTRL-Q	same as CTRL-V (used for terminal control flow)
CTRL-R	<0-9a-z"%:> insert contents of register <0-9a-z"%:>
CTRL-S	(used for terminal control flow)
CTRL-T	insert one shiftwidth of indent in current line
CTRL-U	delete all entered characters in the current line
CTRL-V {char}	insert next non-digit literally
CTRL-V {number}	insert three digit decimal number as a single byte.
CTRL-W	delete word before the cursor
CTRL-X {mode}	enter CTRL-X sub mode, see below
CTRL-Y	insert the character which is above the cursor
CTRL-Z	not used
<Esc>	end insert mode
CTRL-[	same as <Esc>
CTRL-\	not used
CTRL-]	not used
CTRL-^	not used
CTRL-_	change languate (RIGHTLEFT)

<Space> to '~' not used, except '0' and '^' followed by CTRL-D

0 CTRL-D	delete all indent in the current line
^ CTRL-D	delete all indent in the current line, restore it in the next line
<Del>	delete character under the cursor

Meta characters (0x80 to 0xff, 128 to 255) not used

<Down>	cursor one line down
<End>	cursor past end of line
<C-End>	cursor past end of file
<F1>	same as <Help>
<Help>	stop insert mode and display help window
<Home>	cursor to start of line
<C-Home>	cursor to start of file
<Insert>	toggle Insert/Replace mode
<Left>	cursor one character left
<LeftMouse>	cursor at mouse click
<PageDown>	one screenfull forward
<PageUp>	one screenfull backward
<Right>	cursor one character right
<S-Down>	same as <PageDown>
<S-Left>	cursor one word left
<S-Right>	cursor one word right
<S-Up>	same as <PageUp>
<Up>	cursor one line up

commands in CTRL-X submode

CTRL-X CTRL-D complete defined identifiers

CTRL-X CTRL-E scroll up

CTRL-X CTRL-F	complete file names
CTRL-X CTRL-I	complete identifiers
CTRL-X CTRL-K	complete identifers from dictionary
CTRL-X CTRL-L	complete whole lines
CTRL-X CTRL-Y	scroll down
CTRL-X CTRL-]	complete tags

2. Normal mode

CHAR any non-blank chararacter WORD any sequences of non-blank characters N a number entered before the command {motion} a cursor movement command Nmove the text that is moved over with a {motion} SECTION a section that possibly starts with '}' instead of '{'

[1] = cursor movement command
[2] = can be undone/redone

char	action
CTRL-@	not used
CTRL-A [2]	add N to number at/after cursor
CTRL-B [1]	scroll N screens Backwards
CTRL-C	interrupt current (search) command
CTRL-D	scroll Down N lines (default: half a screen)
CTRL-E	scroll N lines upwards (N lines Extra)
CTRL-F [1]	scroll N screens Forward
CTRL-G	display current file name and position
<BS> [1]	same as "h"
CTRL-H [1]	same as "h"
<Tab> [1]	go to N newer entry in jump list
CTRL-I [1]	same as <Tab>
<NL> [1]	same as "j"
CTRL-J [1]	same as "j"
CTRL-K	not used
CTRL-L	redraw screen
<CR> [1]	cursor to the first CHAR N lines lower
CTRL-M> [1]	same as <CR>
CTRL-N [1]	same as "j"
CTRL-O [1]	go to N older entry in jump list
CTRL-P [1]	cursor N lines upward
CTRL-Q	(used for terminal control flow)
CTRL-R [2]	redo changes which were undone with 'u'
CTRL-S	(used for terminal control flow)
CTRL-T	jump to N older Tag in tag list
CTRL-U	scroll N lines Upwards (default: half a screen)
CTRL-V	start blockwise Visual
CTRL-W {char}	window commands
CTRL-X [2]	subtract N from number at/after cursor

CTRL-Y	scroll N lines downwards
CTRL-Z	suspend program (or start new shell)
CTRL-[<Esc>	not used
CTRL-\	not used
CTRL-]	:ta to ident under cursor
CTRL-^	edit Nth alternate file (equivalent to ":e #N")
CTRL-_	not used
<Space> 1	same as "l"
! {motion}{filter} 2	filter Nmove text through the {filter} command
!! {filter} 2	filter N lines through the {filter} command
" <a-zA-Z0-9.%:-">	use buffer <a-zA-Z0-9.%:-"> for next delete, yank or put (upper case to append) (<.%:> only work with put)
# 1	search backward for the Nth occurrence of the ident under the cursor
$ 1	cursor to the end of Nth next line
% 1	find the next (curly/square) bracket on this line and go to its match, or go to matching comment bracket, or go to matching preprocessor directive.
{count}% 1	go to N percentage in the file
& 2	repeat last :s
' <a-zA-Z0-9> 1	cursor to the first CHAR on the line with mark <a-zA-Z0-9>
'' 1	cursor to the first CHAR of the line where the cursor was before the latest jump.
'< 1	cursor to the first CHAR of the line where highlighted area starts/started
'> 1	cursor to the first CHAR of the line where highlighted area ends/ended
'[1	cursor to the first CHAR on the line of the start of last operated text or start of putted text
'] 1	cursor to the first CHAR on the line of the end of last operated text or end of putted text
(1	cursor N sentences backward
) 1	cursor N sentences forward
* 1	search forward for the Nth occurrence of the ident under the cursor
+ 1	cursor to the first CHAR N lines lower
, 1	repeat latest f, t, F or T in opposite direction N times
- 1	cursor to the first CHAR N lines higher
. 2	repeat last change with count replaced with N
/ {pattern}<CR> 1	search forward for the Nth occurrence of {pattern}
/<CR> 1	search forward for {pattern} of last search
0 1	cursor to the first char of the line
1	prepend to command to give a count
2	"
3	"

4	"
5	"
6	"
7	"
8	"
9	"
:	start entering an Ex command
{count}:	start entering an Ex command with range from current line to N lines down
; [1]	repeat latest f, t, F or T N times
< {motion} [2]	shift Nmove lines one 'shiftwidth' leftwards
<< [2]	shift N lines one 'shiftwidth' leftwards
= {motion} [2]	filter Nmove lines through "indent"
== [2]	filter N lines through "indent"
> {motion} [2]	shift Nmove lines one 'shiftwidth' rightwards
>> [2]	shift N lines one 'shiftwidth' rightwards
? <pattern><CR> [1]	search backward for the Nth previous occurrence of <pattern>
?<CR> [1]	search backward for {pattern} of last search
@ <a-z> [2]	execute the contents of named buffer <a-z> N times
@:	repeat the previous ":" command N times
@@ [2]	repeat the previous @<a-z> N times
A [2]	append text at the end of the line N times
B [1]	cursor N WORDS backward
["x]C [2]	change from the cursor position to the end of the line, and N-1 more lines [into buffer x]; synonym for "c$"
["x]D [2]	delete the characters under the cursor until the end of the line and N-1 more lines [into buffer x]; synonym for "d$"
E [1]	cursor forward to the end of WORD N
F {char} [1]	cursor to the Nth occurrence of {char} to the left
G [1]	cursor to line N, default last line
H [1]	cursor to line N from top of screen
I [2]	insert text before the first CHAR on the line N times
J [2]	Join N lines; default is [2]
K	lookup Keyword under the cursor with 'keywordprg'
L [1]	cursor to line N from bottom of screen
M [1]	cursor to middle line of screen
N [1]	repeat the latest '/' or '?' N times in opposite direction
O [2]	begin a new line above the cursor and insert text, repeat N times
["x]P [2]	put the text [from buffer x] before the cursor N times
Q {motion} [2]	format Nmove lines
R [2]	enter replace mode: overtype existing characters, repeat the entered text N-1 times

["x]S [2]	delete N lines [into buffer x] and start insert; synonym for "^cc" or "0cc", depending on autoindent
T {char} [1]	cursor till after Nth occurrence of {char} to the left
U [2]	undo all latest changes on one line
V	start linewise Visual mode
W [1]	cursor N WORDS forward
["x]X [2]	delete N characters before the cursor [into buffer x]
["x]Y	yank N lines [into buffer x]; synonym for "yy"
ZZ	store current file if modified, and exit
[{char}	square bracket command (see below)
\	not used
] {char}	square bracket command (see below)
^ [1]	cursor to the first CHAR of the line
_ [1]	cursor to the first CHAR N - [1] lines lower
` <a-zA-Z0-9> [1]	cursor to the mark <a-zA-Z0-9>
`< [1]	cursor to the start of the highlighted area
`> [1]	cursor to the end of the highlighted area
`[[1]	cursor to the start of last operated text or start of putted text
`] [1]	cursor to the end of last operated text or end of putted text
`` [1]	cursor to the position before latest jump
a [2]	append text after the cursor N times
b [1]	cursor N words backward
["x]c{motion} [2]	delete Nmove text [into buffer x] and start insert
["x]cc [2]	delete N lines [into buffer x] and start insert
["x]d{motion} [2]	delete Nmove text [into buffer x]
["x]dd [2]	delete N lines [into buffer x]
e [1]	cursor forward to the end of word N
f {char} [1]	cursor to Nth occurrence of {char} to the right
g {char}	extended commands, see below
h [1]	cursor N chars to the left
i [2]	insert text before the cursor N times
j [1]	cursor N lines downward
k [1]	cursor N lines upward
l [1]	cursor N chars to the right
m <A-Za-z>	set mark <A-Za-z> at cursor position
n [1]	repeat the latest '/' or '?' N times
o [2]	begin a new line below the cursor and insert text, repeat N times
["x]p [2]	put the text [from register x] after the cursor N times
q <0-9a-zA-Z">	record typed characters into named register <0-9a-zA-Z"> (upper case to append)
q	(while recording) stops recording
r {char} [2]	replace N chars with {char}

["x]s [2]	(substitute) delete N characters [into buffer x] and start insert	
t {char} [1]	cursor till before Nth occurrence of {char} to the right	
u [2]	undo changes	
v	start characterwise Visual mode	
w [1]	cursor N words forward	
["x]x [2]	delete N characters under and after the cursor [into buffer x]	
["x]y{motion}	yank Nmove text [into buffer x]	
["x]yy	yank N lines [into buffer x]	
z<CR>	redraw, cursor line to top of window, cursor on first non-blank	
z{height}<CR>	redraw, make window {height} lines high	
z.	redraw, cursor line to center of window, cursor on first non-blank	
z-	redraw, cursor line at bottom of window, cursor on first non-blank	
zb	redraw, cursor line at bottom of window	
ze	when 'wrap' off scroll horizontally to position the cursor at the end (right side) of the screen	
zh	when 'wrap' off scroll screen N characters to the right	
zl	when 'wrap' off scroll screen N characters to the left	
zs	when 'wrap' off scroll horizontally to position the cursor at the start (left side) of the screen	
zt	redraw, cursor line at top of window	
zz	redraw, cursor line at center of window	
z<Left>	same as "zh"	
z<Right>	same as "zl"	
{ [1]	cursor N paragraphs backward	
| [1]	cursor to column N	
} [1]	cursor N paragraphs forward	
~ [2]	'tildeop' off: switch case of N characters under cursor and move the cursor N characters to the right	
~ {motion}	'tildeop' on: switch case of Nmove text	
<C-LeftMouse>	":ta" to the keyword at the mouse click	
<C-RightMouse>	same as "CTRL-T"	
["x]<Del> [2]	same as "x"	
{count}<Del>	remove the last digit from {count}	
<Down> [1]	same as "j"	
<End> <End> [1]	same as "$" <C-End> [1]	same as "G"
<F1>	same as <Help>	
<Help>	open a help window	
<Home> [1]	same as "0" <C-Home> [1]	same as "gg"
<Insert> [2]	same as "i"	
<Left> [1]	same as "h"	
<LeftMouse> [1]	move cursor to the mouse click position	
<MiddleMouse> [2]	same as "P" at the mouse click position	
<PageDown>	same as CTRL-F	

<PageUp>	same as CTRL-B
<Right>[1]	same as "l"
<RightMouse>	start Visual mode, move cursor to the mouse click position
<S-Down>[1]	same as CTRL-F
<S-Left>[1]	same as "b"
<S-LeftMouse>	same as "*" at the mouse click position
<S-Right>[1]	same as "w"
<S-RightMouse>	same as "#" at the mouse click position
<S-Up>[1]	same as CTRL-B
<Undo>[2]	same as "u"
<Up>[1]	same as "k"

2.1 text objects

These can be used after an operator or in Visual mode to select an object
a a word (with white space)
A a WORD (with white space)
p a paragraph
s a sentence
P a block from "[{" to "]}"
S a block from "[(" and "])"

2.2 window commands

command	action
CTRL-W CTRL-B	same as "CTRL-W b"
CTRL-W CTRL-C	same as "CTRL-W c"
CTRL-W CTRL-D	same as "CTRL-W d"
CTRL-W CTRL-F	same as "CTRL-W f"
CTRL-W CTRL-I	same as "CTRL-W i"
CTRL-W CTRL-J	same as "CTRL-W j"
CTRL-W CTRL-K	same as "CTRL-W k"
CTRL-W CTRL-N	same as "CTRL-W n"
CTRL-W CTRL-O	same as "CTRL-W o"
CTRL-W CTRL-P	same as "CTRL-W p"
CTRL-W CTRL-Q	same as "CTRL-W q"
CTRL-W CTRL-R	same as "CTRL-W r"
CTRL-W CTRL-S	same as "CTRL-W s"
CTRL-W CTRL-T	same as "CTRL-W t"
CTRL-W CTRL-W	same as "CTRL-W w"
CTRL-W CTRL-X	same as "CTRL-W x"
CTRL-W CTRL-]	same as "CTRL-W]"
CTRL-W CTRL-^	same as "CTRL-W ^"
CTRL-W CTRL-_	same as "CTRL-W _"
CTRL-W +	increase current window height N lines
CTRL-W -	decrease current window height N lines

CTRL-W =	make all windows the same height
CTRL-W R	rotate windows upwards N times
CTRL-W S	same as "CTRL-W s"
CTRL-W]	split window and jump to tag under cursor
CTRL-W ^	split current window and edit alternate file N
CTRL-W _	set current window height to N (default: very high)
CTRL-W b	go to bottom window
CTRL-W c	close current window (like ":close")
CTRL-W d	split window and jump to definition under the cursor
CTRL-W f	split window and edit filename under the cursor
CTRL-W i	split window and jump to declaration of identifier under the cursor
CTRL-W j	go to N next window (stop at last window)
CTRL-W k	go to N previous window (stop at first window)
CTRL-W n	open new window, N lines high
CTRL-W o	close all but current window (like ":only")
CTRL-W p	go to previous (last accessed) window
CTRL-W q	quit current window (like ":quit")
CTRL-W r	rotate windows downwards N times
CTRL-W s	split current window in two parts, new window N lines high
CTRL-W t	go to top window
CTRL-W w	go to N next window (wrap around)
CTRL-W x	exchange current window with window N (default: next window)
CTRL-W <Down>	same as "CTRL-W j"
CTRL-W <Up>	same as "CTRL-W k"

2.3 Square bracket commands

char	action
[_CTRL-D	jump to first #define found in current and included files matching the word under the cursor, start searching at beginning of current file
[_CTRL-I	jump to first line in current and included files that contains the word under the cursor, start searching at beginning of current file
[# [1]	cursor to N previous unmatched #if, #else or #ifdef
[([1]	cursor N times back to unmatched '('
[* [1]	same as "[/"
[/ [1]	cursor to N previous start of a C comment
[D	list all defines found in current and included files matching the word under the cursor, start searching at beginning of current file
[I	list all lines found in current and included files that contain the word under the cursor, start searching at beginning of current file
[P [2]	same as "[p"
[[[1]	cursor N sections backward
[] [1]	cursor N SECTIONS backward
[d	show first #define found in current and included files matching the word under the cursor, start searching at beginning of current file
[f	same as "gf"

[i	show first line found in current and included files that contains the word under the cursor, start searching at beginning of current file
[p [2]	like "P", but adjust indent to current line
[{ [1]	cursor N times back to unmatched '{'
[<MiddleMouse> [2]	same as "[p"
]_CTRL-D	jump to first #define found in current and included files matching the word under the cursor, start searching at cursor position
]_CTRL-I	jump to first line in current and included files that contains the word under the cursor, start searching at cursor position
]# [1]	cursor to N next unmatched #endif or #else
]) [1]	cursor N times forward to unmatched ')'
]* [1]	same as "]/"
]/ [1]	cursor to N next end of a C comment
]D	list all #defines found in current and included files matching the word under the cursor, start searching at cursor position
]I	list all lines found in current and included files that contain the word under the cursor, start searching at cursor position
]P [2]	same as "[p"
][[1]	cursor N SECTIONS forward
]] [1]	cursor N sections forward
]d	show first #define found in current and included files matching the word under the cursor, start searching at cursor position
]f	same as "gf"
]i	show first line found in current and included files that contains the word under the cursor, start searching at cursor position
]p [2]	like "p", but adjust indent to current line
]} [1]	cursor N times forward to unmatched '}'
]<MiddleMouse> [2]	same as "]p"

2.4 Commands starting with 'g'

char	action
g# [1]	like "#", but without using "\<" and "\>"
g$ [1]	when 'wrap' off go to rightmost character of the current line that is on the screen; when 'wrap' on go to the rightmost character of the current screen line
g* [1]	like "*", but without using "\<" and "\>"
g0 [1]	when 'wrap' off go to leftmost character of the current line that is on the screen; when 'wrap' on go to the leftmost character of the current screen line
gD [1]	goto definiton of word under the cursor in current file
gE [1]	go backwards to the end of the previous WORD
gI [2]	like "I", but always start in column 1
gU {motion} [2]	make Nmove text uppercase
g^ [1]	when 'wrap' off go to leftmost non-white character of the current line that is on the screen; when 'wrap' on go to the leftmost non-white character of the current screen line

g_CTRL-G	show information about current cursor position
ga	print ascii value of character under the cursor
gd [1]	goto definiton of word under the cursor in current function
ge [1]	go backwards to the end of the previous word
gf	start editing the file whose name is under the cursor
gg [1]	cursor to line N, default first line
gj [1]	like "j", but when 'wrap' on go N screen lines down
gk [1]	like "k", but when 'wrap' on go N screen lines up
gs	goto sleep for N seconds (default 1)
gu {motion} [2]	make Nmove text lowercase
gv	reselect the previous Visual area
g~ {motion} [2]	swap case for Nmove text
g<Down> [1]	same as "gj"
g<End> [1]	same as "g$"
g<Home> [1]	same as "g0"
g<LeftMouse>	same as <C-LeftMouse>
g<RightMouse>	same as <C-RightMouse>
g<Up> [1]	same as "gk"

3. Visual mode

Most commands in Visual mode are the same as in Normal mode. The ones listed here are those that are different.

command	**action**
CTRL-V	make Visual mode blockwise or stop Visual
CTRL-]	jump to highlighted tag
! {filter} [2]	filter the highlighted lines through the external command {filter}
:	start a command line with the highlighted lines as a range
< [2]	shift the highlighted lines one 'shiftwidth' left
= [2]	filter the highlighted lines through the external program given with the 'equalprg' option
> [2]	shift the highlighted lines one 'shiftwidth' right
A	extend the highlighted area with a WORD
C [2]	delete the highlighted lines and start insert
D [2]	delete the highlighted lines
J [2]	join the highlighted lines
K	run 'keywordprg' on the highlighted text
P	extend the highlighted text with a {} block
Q [2]	format the highlighted lines
R [2]	delete the highlighted lines and start insert
S	extend the highlighted text with a () block
U [2]	make highlighted text uppercase
V	make Visual mode linewise or stop Visual
X [2]	delete the highlighted lines

Y	yank the highlighted lines
a	extend the highlighted area with a word
c[2]	delete the highlighted text and start insert
d[2]	delete the highlighted text
gv	exchange current and previous highlighted text
o	move cursor to other end of hightlighted text
p	extend the highlighted area with a paragraph
r[2]	delete the highlighted text and start insert
s	extend the highlighted area with a sentence
u[2]	make highlighted text lowercase
v	make Visual mode characterwise or stop Visual mode
x[2]	delete the highlighted text
y	yank the highlighted text
~[2]	swap case for the highlighted text

4. Command line editing

Get to the command line with the ':', '!', '/' or '?' commands. Normal characters are inserted at the current cursor position. "Completion" below refers to context-sensitive completion. It will complete filenames, tags, commands etc. as appropriate.

CTRL-@	not used
Command	**Action**
CTRL-A	do completion on the pattern in front of the cursor and insert all matches
CTRL-B	cursor to begin of command line
CTRL-C	same as <ESC>
CTRL-D	list completions that match the pattern in front of the cursor
CTRL-E	cursor to end of command line
CTRL-F	not used
CTRL-G	not used
<BS>	delete the character in front of the cursor
{char1} <BS> {char2}	enter digraph when 'digraph' is on
CTRL-H	same as <BS>
<Tab>	if 'wildchar' is <Tab>: Do completion on the pattern in front of the cursor
'wildchar'	Do completion on the pattern in front of the cursor (default: <Tab>)
CTRL-I	same as <Tab>
<NL>	same as <CR>
CTRL-J	same as <CR>
CTRL-K {char1} {char2}	enter digraph
CTRL-L	do completion on the pattern in front of the cursor and insert the longest common part
<CR>	execute entered command
CTRL-M	same as <CR>
CTRL-N	after using 'wildchar' with multiple matches: go to next match, otherwise: same as <Down>
CTRL-O	not used

CTRL-P	after using 'wildchar' with multiple matches: go to previous match, otherwise: same as <Up>
CTRL-Q	same as CTRL-V (used for terminal control flow)
CTRL-R <0-9a-z"%:>	insert contents of register <0-9a-z"%:>
CTRL-S	(used for terminal control flow)
CTRL-T	not used
CTRL-U	remove all characters
CTRL-V	insert next non-digit literally, insert three digit decimal number as a single byte.
CTRL-W	delete the word in front of the cursor
CTRL-X	not used
CTRL-Y	not used
CTRL-Z	not used
<Esc>	abandon command line without executing it
CTRL-[	same as <Esc>
CTRL-\	not used
CTRL-]	not used
CTRL-^	not used
CTRL-_	change languate (RIGHTLEFT)
<Del>	delete the character under the cursor
<Down>	recall next command line from history that matches pattern in front of the cursor
<End>	cursor to end of command line
<Home>	cursor to start of command line
<Insert>	toggle insert/overstrike mode
<Left>	cursor left
<LeftMouse>	cursor at mouse click
<PageDown>	same as <S-Down>
<PageUp>	same as <S-Up>
<Right>	cursor right
<S-Down>	recall next command line from history
<S-Left>	cursor one word left
<S-Right>	cursor one word right
<S-Up>	recall previous command line from history
<Up>	recall previous command line from history that matches pattern in front of the cursor

5. EX commands

This is a brief listing of all the ":" commands, without mentioning any arguments.

Command	**Action**
:!	filter lines or execute an external command
:!!	repeat last ":!" command
:#	same as ":number"
:&	repeat last ":substitute"
:<	shift lines one 'shiftwidth' left
:=	print the cursor line number
:>	shift lines one 'shiftwidth' right

:@	execute contents of a register
:@@	repeat the previous ":@"
:N[ext]	go to previous file in the argument list
:a[ppend]	append text (not implemented)
:ab[breviate]	enter abbreviation
:abc[lear]	remove all abbreviations
:al[l]	open a window for each file in the argument list
:ar[gs]	print the argument list
:argu[ment]	go to specific file in the argument list
:a[scii]	print ascii value of character under the cursor
:au[tocmd]	enter or show autocommands
:b[uffer]	go to specific buffer in the buffer list
:bN[ext]	go to next buffer in the buffer list
:ba[ll]	open a window for each file in the buffer list
:bd[elete]	delete specific files from the buffer list
:bl[ast]	go to last file in the buffer list
:bm[odified]	go to next file in the buffer list that has been modified
:bn[ext]	go to next file in the buffer list
:bp[revious]	go to previous file in the buffer list
:br[ewind]	go to last file in the buffer list
:buffers	list all files in the buffer list
:bun[load]	unload a specific buffer
:c[hange]	replace a line (not implemented)
:cN[ext]	go to previous error
:ca[bbrev]	like ":abbreviate" but for command line mode
:cabc[lear]	clear all abbreviations for command line mode
:cc	go to specific error
:cd	change directory
:ce[nter]	format lines at the center
:cf[ile]	read the file with error messages
:chd[ir]	change directory
:che[ckpath]	list included files
:cl[ist]	list all errors
:clo[se]	close current window
:cm[ap]	like ":map" but for command line mode
:cmapc[lear]	clear all mappings for command line mode
:cn[ext]	go to next error
:cno[remap]	like ":noremap" but for command line mode
:cnorea[bbrev]	like ":noreabbrev" but for command line mode
:co[py]	copy lines
:cp[revious]	go to previous error
:cq[uit]	quit Vim with an error code
:cu[nmap]	like ":unmap" but for command line mode
:cuna[bbrev]	like ":unabbrev" but for command line mode
:d[elete]	delete lines
:di[splay]	display registers

:dig[raphs]	show or enter digraphs
:dj[ump]	jump to #define
:dl[ist]	list #defines
:do[autocmd]	apply autocommands
:ds[earch]	list one #define
:dsp[lit]	split window and jump to #define
:e[dit]	edit a file
:ex	same as ":edit"
:exi[t]	same as ":xit"
:f[ile]	show or set the current file name
:files	list all files in the buffer list
:fix[del]	set key code of <Del>
:g[lobal]	execute commands for matching lines
:gu[i]	start the GUI
:gv[im]	start the GUI
:h[elp]	open a help window
:i[nsert]	insert text (not implemented)
:ia[bbrev]	like ":abbrev" but for Insert mode
:iabc[lear]	like ":abclear" but for Insert mode
:ij[ump]	jump to definition of identifier
:il[ist]	list lines where identifier matches
:im[ap]	like ":map" but for Insert mode
:imapc[lear]	like ":mapclear" but for Insert mode
:ino[remap]	like ":noremap" but for Insert mode
:inorea[bbrev]	like ":noreabbrev" but for Insert mode
:is[earch]	list one line where identifier matches
:isp[lit]	split window and jump to definition of identifier
:iu[nmap]	like ":unmap" but for Insert mode
:iuna[bbrev]	like ":unabbrev" but for Insert mode
:j[oin]	join lines
:ju[mps]	print the jump list
:k	set a mark
:l[ist]	print lines
:la[st]	go to the last file in the argument list
:le[ft]	left align lines
:ls	list all buffers
:m[ove]	move lines
:ma[rk]	set a mark
:mak[e]	execute external command 'makeprg' and parse error messages
:map	show or enter a mapping
:mapc[lear]	clear all mappings for Normal and Visual mode
:marks	list all marks
:me[nu]	enter a new menu item
:mk[exrc]	write current mappings and settings to a file
:mkv[imrc]	write current mappings and settings to a file
:mod[e]	show or change the screen mode

:n[ext]	go to next file in the argument list
:new	create a new empty window
:nm[ap]	like ":map" but for Normal mode
:nmapc[lear]	clear all mappings for Normal mode
:nn[oremap]	like ":noremap" but for Normal mode
:no[remap]	enter a mapping that will not be remapped
:norea[bbrev]	enter an abbreviation that will not be remapped
:norm[al]	execute Normal mode commands
:nu[mber]	print lines with line number
:nun[map]	like ":unmap" but for Normal mode
:o[pen]	start open mode (not implemented)
:on[ly]	close all windows except current one
:p[rint]	print lines
:po[p]	jump to older entry in tag stack
:pre[vious]	go to previous file in argument list
:pres[erve]	write all text to swap file
:pu[t]	insert contents of register in the text
:pw[d]	print current directory
:q[uit]	quit current window or Vim
:qa[ll]	quit Vim
:r[ead]	read file into the text
:rec[over]	recover a file from a swap file
:red[o]	redo one undone change
:reg[isters]	display the contents of registers
:res[ize]	change current window height
:ret[ab]	change tab size
:rew[ind]	go to the first file in the argument list
:ri[ght]	right align text
:rv[iminfo]	read from viminfo file
:s[ubstitute]	find and replace text
:sN[ext]	split window and go to previous file in argument list
:sa[rgument]	split window and go to specific file in argument list
:sal[l]	open a window for each file in argument list
:sb[uffer]	split window and go to specific file in the buffer list
:sbN[ext]	split window and go to previous file in the buffer list
:sba[ll]	open a window for each file in the buffer list
:sbl[ast]	split window and go to last file in buffer list
:sbm[odified]	split window and go to modified file in the buffer list
:sbn[ext]	split window and go to next file in the buffer list
:sbp[revious]	split window and go to previous file in the buffer list
:sbr[ewind]	split window and go to first file in the buffer list
:se[t]	show or set options
:sh[ell]	escape to a shell
:sl[eep]	do nothing for a few seconds
:sla[st]	split window and go to last file in the argument list
:sn[ext]	split window and go to next file in the argument list

:so[urce]	read Vim or Ex commands from a file
:sp[lit]	split current window
:spr[evious]	split window and go to previous file in the argument list
:sr[ewind]	split window and go to first file in the argument list
:st[op]	suspend the editor or escape to a shell
:sta[g]	split window and jump to a tag
:sun[hide]	same as ":unhide"
:sus[pend]	same as ":stop"
:sv[iew]	split window and edit file read-only
:sw[apname]	show the name of the current swap file
:t	same as ":copy"
:ta[g]	jump to tag
:tags	show the contents of the tag stack
:u[ndo]	undo last change(s)
:un[abbreviate]	remove abbreviation
:unh[ide]	open a window for each loaded file in the buffer list
:unm[ap]	remove mapping
:ve[rsion]	print version number and other info
:v[global]	execute commands for not matching lines
:vi[sual]	same as ":edit"
:vie[w]	edit a file read-only
:vm[ap]	like ":map" but for Visual mode
:vmapc[lear]	remove all mappings for Visual mode
:vn[oremap]	like ":noremap" but for Visual mode
:vu[nmap]	like ":unmap" but for Visual mode
:w[rite]	write to a file
:wN[ext]	write to a file and go to previous file in argument list
:wa[ll]	write all (changed) buffers
:wn[ext]	write to a file and go to next file in argument list
:wp[revious]	write to a file and go to previous file in argument list
:wq	write to a file and quit window or Vim
:wqa[ll]	write all changed buffers and quit Vim
:wv[iminfo]	write to viminfo file
:x[it]	write if buffer changed and quit window or Vim
:xa[ll]	same as ":wqall"
:y[ank]	yank lines into a register
:z	print some lines (not implemented)
:~	repeat last ":substitute"

Send feedback on this page to Rajesh Kallingal

For Vim version 3.24. Last modification: 1996 Apr 25