

Mission: Training

vi cheat sheet

vi mode COMMANDS

note: commands may be preceded by a numeric count

Scrolling/Cursor Movement

^U, ^D up, down 1/2 page
^F, ^B forward, back 1 page
j, ^N, ^J Down 1 line
k, ^P Up 1 line
l, <SP> 1 char right
h, <BS> 1 char left
nG go to line *n* (0 goes to last line)
0, \$ beginning, end of line
w, b beginning of next, previous word
/RE beginning of RegEx (search forward)
?RE beginning of RegEx (search backward)
n, N next, previous match of / or ?
H, M, L go to first, middle, last line on screen
% go to matching {}|[]<>

Text Entry/Change

a append after cursor
A append at end of line
i append before cursor
I insert before first nonblank
o open new line after current
O open new line before current
r replace 1 character
R replace chars until <ESC>
D delete to end of line
C change to end of line
cw change to end of word
s change 1 character
S change entire line
x delete char under cursor
dw delete word
dd delete line
D delete to end of line
yy yank (copy) line
yw yank word
p paste last yanked or deleted item after cursor
P same, but before cursor

Miscellaneous

u undo last change
U undo all changes to current line
. repeat last command
ZZ save & exit (same as :wq in ex mode)
^G print buffer name and current line number

Named Buffers

"xCMD put result of CMD in buffer *x*, where *x* is any letter
"xp paste contents of buffer *x* after cursor
"xP same, but before cursor

Note: <ESC> escape key
<BS> backspace key
<SP> space bar

ex mode commands

syntax: **:x,yCommand** where *x* is first line # of range and *y* is second (optional) line # of range. *x* and *y* may be REs, placed inside //. **\$** is shorthand for last line of file. ex commands must be followed by carriage return (ENTER).

s/RE/NEW/g	substitute NEW for RegEx RE. Optional g replaces all occurrences of RE on each specified line; otherwise, only 1st occurrence is replaced.
!COMMAND	execute UNIX shell command
w	write current buffer
w!	write current buffer file; overwrite existing file without prompting
w file	write buffer to file named <i>file</i>
wq	write file and quit
q	quit vi
q!	quit vi even if buffer changed (abandon edit)
e file	edit <i>file</i>
e! file	edit <i>file</i> ; abandon changes to current buffer
e#	edit previous buffer
e!#	edit previous buffer; abandon current
n	go to next file on command line
r file	read <i>file</i> into current buffer
number	go to line <i>number</i>

Editor Settings

syntax: **:set label** or **:set label=value**
examples: **:set scroll=3 set number**
note: prepending **no** to a label turns it off

label	what it does
ignorecase	ignore case in searches & substitutions [ic]
number	display line numbers on screen (not in file) [nu]
showmode	indicate text-entry modes
tabstop	number of spaces for tab [ts]
terse	make error/info messages briefer (!)

[abbreviations are in square brackets following descriptions]

Regular Expressions

may be used in / (search) command, s// (substitute) command, or as line specifier (inside //) for ex commands

.	match any 1 character
*	match 0 or more of preceding 1-character RE
+	match 1 or more of preceding 1-character RE
^	match beginning of line
\$	match end of line
\<	match beginning of word
\>	match end of word
[xyz]	match any 1 of chars x , y , or z
[^xyz]	match any 1 character <i>not</i> x , y , or z
[a-z]	match any 1 character in range a through z
\	escape (turn off special meaning of) characters

Compliments of:

Mission: Training
920 Urban Avenue
Durham, NC 27701
919-682-3401 **voice** 919-682-3369 **fax**

ksh/bash/tcsh Editing Commands¹

Emacs-mode Command	vi-mode Command²	Function
^P	k	Previous command
^N	j	Next command
^F	l	Forward 1 character
^B	h	Back 1 character
^A	0	Beginning of line
^E	\$	End of line
^D	x	Delete character under cursor
^K	D	Delete to end of line

¹ In *ksh* or *bash*, use **set -o emacs** or **set -o vi** to enter editing mode; in *tcsh* use **set edit = emacs** or **set edit = vi**

² Press <ESC> to enter commands when editing in vi-mode