

remembers which mode you are in. *If you are in command mode, you cannot enter data into your file; if you are in input mode you cannot give vi commands.* So there has to be a way of switching back and forth between command and input modes.

CHANGING MODES (1): Vi always starts in command mode.

CHANGING MODES (2): To go from command mode to input mode: type **i** or **a**

CHANGING MODES (3): To go from input mode to command mode: hit the **ESC** key

NOTE: None of the commands in vi (like i a ESC) require hitting return after the command unless otherwise indicated.

INPUT MODE: Once you are in **input mode**, everything you type goes into the file you are making. There is no "wrap-around"— when you want to make new lines in vi, you have to hit the RETURN key, just like on a typewriter. If you make a mistake, you can use the backspace key to go backwards over it and fix it. When you want to do something else (like delete a line, save your work, or just go home) you have to hit ESC to leave input mode.

COMAND MODE: Here are a few of the commands that you can use in vi:

<u>command</u>	<u>what it does</u>	<u>Needs a RETURN?</u>
x	erases the character under the cursor	NO
dl	erases the entire line that the cursor is on	NO
ZZ	saves the file and exits the editor	NO
:q!	exits the editor WITHOUT saving the file	YES
i	goes into input mode	NO
a	goes into input mode	NO
space	moves forward one character	NO
h	moves backwards one character	NO
j	goes down one line	NO
k	goes up one line	NO
J	joins the line the cursor is on with the next line	NO
u	undoes the last command!	NO

SUMMARY: moving commands: h j k space
 erasing commands: dd x
 leaving commands: :q! ZZ
 join command: J
 oooops command: u

NOTES:

- case is significant: J≠j be sure that you don't set caps-lock.

BEEP: When vi is in command mode, if you type something it doesn't understand, it will make a beeping sound. For example, if you are at the end of a line and you type space (to move forward to the next character) it will beep because it can't go on, or if you hit ESC (when you are already in command mode).

Sometimes a vi user can feel lost and be unsure of which mode he or she is in. If that ever happens, hit ESC. After

doing that, you will *always* be in command mode. Then you can stay in that mode if you like or go into input mode by typing **i** or **a**.

Survival VI

General Characteristics:

- EXTREMELY powerful, flexible, convenient
- EXTREMELY unfriendly, bordering on hostile
- EXTREMELY redundant (10 ways to do the same thing)

Modes: input mode, command mode (no visible difference!)

Command-->input mode: i a

Input--->command mode: ESC

Moving Commands: (note:case is significant!)

small movements:	h j k SPACE
screen top/middle/bottom:	H M B
forward/backward one line:	+ -
beginning/end of same line:	^ \$
next/previous screen:	ctrl-F ctrl-B
next/previous word:	w b
end of current word:	e
scroll down/up:	ctrl-D ctrl-U
jump to line N:	NG (example: 178G takes you to line 178)
jump to last line:	\$G (note: in general "\$" means "last")
find pattern:	/pattern (example: /unix takes you to next instance of unix)

Deleting Commands:

delete current line:	dd
delete current word:	dw
delete current character:	x

Joining current and next lines: J

File Commands:

:f	(shows file being edited)
:f name	(changes name of file being edited)
:w	(save what you have under the name of the file being edited)
:e name	(edit a different file)
:q	(quit)

NOTE: ! after w,e or q means FORCE (i.e. changes will be lost)

Global Substitution:

:line#,line#/s/matchpattern/replacepattern/	(just first instance on each line)
:line#,line#/s/matchpattern/replacepattern/g	(every instance on each line)

Global Pattern Commands:

:g/matchpattern/p	print every line with that matchpattern
:g/matchpattern/d	delete every line with that matchpattern

Match patterns are regular expressions. Special Characters Are:

- matches any character
- * 0 or more repetitions of previous character
- [] matches 1 character within the brackets. Example: [abxy] will match a "b" or an "x" or a "y" or an
- "a"
- ^ matches beginning of line (not a character)
- \$ matches end of line (not a character)