Vi Reference Card

Modes

Vi has two modes: insertion mode, and command mode. The editor begins in command mode, where cursor movement and text deletion and pasting occur. Insertion mode begins upon entering an insertion or change command. [ESC] returns the editor to command mode (where you can quit, for example by typing `:q!`). Most commands execute as soon as you type them except for "colon" commands which execute when you press the return key.

Quitting

exit, saving changes	`:x`
quit (unless changes)	`:q`
quit (force, even if unsaved)	`:q!`

Inserting text

insert before cursor, before line	`i` , `I`
append after cursor, after line	`a` , `A`
open new line after, line before	`o` , `O`
replace one char, many chars	`r` , `R`

Motion

left, down, up, right	`h` , `j` , `k` , `l`
next word, blank delimited word	`w` , `W`
beginning of word, of blank delimited word	`b` , `B`
end of word, of blank delimited word	`e` , `E`
sentence back, forward	`(` , `)`
paragraph back, forward	`{` , `}`
beginning, end of line	`0` , `$`
beginning, end of file	`1G` , `G`
line *n*	*n*`G` or `:`*n*
forward, back to char *c*	`f`*c* , `F`*c*
top, middle, bottom of screen	`H` , `M` , `L`

Deleting text

Almost all deletion commands are performed by typing `d` followed by a *motion*. For example `dw` deletes a word. A few other deletions are:

character to right, left	`x` , `X`
to end of line	`D`
line	`dd`
line	`:d`

Yanking text

Like deletion, almost all yank commands are performed by typing `y` followed by a *motion*. For example `y$` yanks to the end of line. Two other yank commands are:

line	`yy`
line	`:y`

Changing text

The change command is a deletion command that leaves the editor in insert mode. It is performed by typing `c` followed by a *motion*. For example `cw` changes a word. A few other change commands are:

to end of line	`C`
line	`cc`

Putting text

put after position or after line	`p`
put before position or before line	`P`

Buffers

Named buffers may be specified before any deletion, change, yank, or put command. The general prefix has the form `"`*c* where *c* may be any lower case letter. For example, `"adw` deletes a word into buffer `a`. It may thereafter be put back into the text with an appropriate put command, for example `"ap`.

Markers

Named markers may be set on any line of a file. Any lower case letter may be a marker name. Markers may also be used as the limits for ranges.

set marker *c* on this line	`m`*c*
goto marker *c*	`` ` ``*c*
goto marker *c* first non-blank	`'`*c*

Search for Strings

search forward	`/`*string*
search backward	`?`*string*
repeat search in same, reverse direction	`n` , `N`

Replace

The search and replace function is accomplished with the `:s` command. It is commonly used in combination with ranges or the `:g` command (below).

replace pattern with string	`:s/`*pattern*`/`*string*`/`*flags*
flags: all on each line, confirm each	`g` , `c`
repeat last :s command	`&`

Regular Expressions

any single character except newline	`.` (dot)
zero or more repeats	`*`
any character in set	`[...]`
any character not in set	`[^ ...]`
beginning, end of line	`^` , `$`
beginning, end of word	`\<` , `\>`
grouping	`\(...\)`
contents of *n*th grouping	`\`*n*

Counts

Nearly every command may be preceded by a number that specifies how many times it is to be performed. For example `5dw` will delete 5 words and `3fe` will move the cursor forward to the 3rd occurance of the letter `e`. Even insertions may be repeated conveniently with this method, say to insert the same line 100 times.

Ranges

Ranges may precede most "colon" commands and cause them to be executed on a line or lines. For example `:3,7d` would delete lines 3–7. Ranges are commonly combined with the `:s` command to perform a replacement on several lines, as with `:.,$s/pattern/string/g` to make a replacement from the current line to the end of the file.

lines *n*-*m*	`:`*n*`,`*m*
current line	`:.`
last line	`:$`
marker *c*	`:'`*c*
all lines	`:%`
all matching lines	`:g/`*pattern*`/`

Files

write file (current file if no name given)	`:w` *file*
read file after line	`:r` *file*
next file	`:n`
previous file	`:p`
edit file	`:e` *file*
replace line with program output	`!!`*program*

Other

toggle upper/lower case	`~`
join lines	`J`
repeat last text-changing command	`.`
undo last change, all changes on line	`u` , `U`