vi basic commands

Summary of most useful commands

Entering command mode

[Esc]   Exit editing mode. Keyboard keys now interpreted as commands.

Moving the cursor

h   (or left arrow key) move the cursor left.
l   (or right arrow key) move the cursor right.
j   (or down arrow key) move the cursor down.
k   (or up arrow key) move the cursor up.
[Ctrl] f move the cursor one page **f**orward .
[Ctrl] b move the cursor one page **b**ackward.
^   move cursor to the first non-white character in the current line.
$ move the cursor to the end of the current line.
G   **g**o to the last line in the file.
*n*G   **g**o to line number *n.*
[Ctrl] G display the name of the current file and the cursor position in it.

Entering editing mode

i   **i**nsert new text before the cursor.
a   **a**ppend new text after the cursor.
o   start to edit a new line after the current one.
O   start to edit a new line before the current one.

Replacing characters, lines and words

r   **r**eplace the current character (does not enter edit mode).
s   enter edit mode and **s**ubstitute the current character by several ones.
cw   enter edit mode and **c**hange the **w**ord after the cursor.
C   enter edit mode and **c**hange the rest of the line after the cursor.

Copying and pasting

yy   copy (**y**ank) the current line to the copy/paste buffer.
p   **p**aste the copy/paste buffer after the current line.
P   **P**aste the copy/paste buffer before the current line.

Deleting characters, words and lines

All deleted characters, words and lines are copied to the copy/paste buffer.

x   delete the character at the cursor location.

dw   **d**elete the current **w**ord.
D   **d**elete the remainder of the line after the cursor.
dd   **d**elete the current line.

Repeating commands

.   repeat the last insertion, replacement or delete command.

Looking for strings

/*string* find the first occurrence of *string after the cursor*.
?*string* find the first occurrence of *string* before the cursor.
n   find the **n**ext occurrence in the last search.

Replacing strings

Can also be done manually, searching and replacing once, and then using n (next occurrence) and . (repeat last edit).

n,*p*s/*str1*/*str2*/g   between line numbers *n* and *p*, **s**ubstitute all (**g**: global) occurrences of *str1* by *str2*.
1,$s/*str1*/*str2*/g   in the whole file ($: last line), **s**ubstitute all occurrences of *str1* by *str2*.

Applying a command several times - Examples

5j   move the cursor 5 lines down.
30dd   **d**elete 30 lines.
4cw   **c**hange 4 **w**ords from the cursor.
1G   **g**o to the first line in the file.

Misc

[Ctrl] l redraw the screen.
J   **j**oin the current line with the next one
u   **u**ndo the last action

Exiting and saving

ZZ   save current file and exit vi.
:w   **w**rite (save) to the current file.
:w *file* **w**rite (save) to the *file* file.
:q!   **q**uit vi without saving changes.

Going further

vi has much more flexibility and many more commands for power users!
It can make you extremely productive in editing and creating text.

Learn more by taking the quick tutorial: just type vimtutor.

Free Electrons

Embedded
Linux
Experts