Movement

h j k l	Move cursor (h: ← j: ↓ k: ↑ l: →)
$	Move to end of line
0	Move to beginning of line (including whitespace)
^	Move to first character on line
gg	Move to first line of file
G	Move to last line of file (EOF)
w	Move forward to next word, with cursor on first character (**W** to jump by whitespace only)
b	Move backward to next word, with cursor on first character (**B** to jump by whitespace only)
e	Move forward to next word, with cursor on last character (**E** to jump by whitespace only)
ge	Move backwards to next word, with cursor on last character (**gE** to jump by whitespace only)
(	Move to beginning of previous sentence. Use **)** to go to next sentence
{	Move to beginning of previous paragraph. Use **}** to go to next paragraph
+	Move forward to the first character on the next line
-	Move backwards to the first character on the previous line
<C>+u	Move up by half a page
<C>+d	Move down by half a page
<C>+b	Move up by a full page
<C>+f	Move down by a full page
H	Move cursor to header (top) line of current visible window
M	Move cursor to middle line of current visible window
L	Move cursor to last line of current visible window
f*c*	Move cursor to next occurrence of character *c* on the current line. Use **F***c* to move backwards
t*c*	Move cursor till next character *c* on the current line. Use **T***c* to move backwards
*	Search forward for word under cursor
#	Search backwards for word under cursor
/*word*	Search forward for *word*. Accepts regular expressions to search.
/\v*word*	Starts the search from the cursor position downwards and wrap around to the top.
/\c*xyz*	Case insensitive search (\c can appear anywhere in the pattern)
?*word*	Search backwards for word. Accepts regular expressions to search
:%s/x/y/z	:[range]s[ubstitute]/{pattern}/{string}/[c][e][g][p][r][i][I] [i] Ignore case for the pattern [g] Replace all occurrences in the line [c] Confirm each substitution ('a' substitute all occurrences, 'q' to quit) Ranges: % is short hand for 1,$ meaning the entire file .,$ is go from the current line to the End Of File (EOF)
n	Repeat the last / or ? command
N	Repeat the last / or ? command in the opposite direction
m*a*	Set a marker at cursor position to come back to later. a can be any character you choose
`*a*	Move cursor to exact position of the marker you set with ma
'*a*	Move cursor to the first character of the line marked with ma
%	Move cursor to next brace, bracket or comment paired to the current cursor location
<C>+z ; fg	Hit <C>+z to pause the VIM process, play around in the shell, then type 'fg' to go back to VIM

Deletion

x	Delete character forward (under cursor). use x do delete backwards (before cursor)
r	Replace single character under cursor, and remain in normal mode
s	Delete character under cursor, then switch to insert mode
dm	Delete in direction of movement m. For m, you can also use w, b, or any other variation
dd	Delete entire current line
D	Delete until end of line

Yank & Put

y	Yank (copy) highlighted text
yy	Yank current line
p	Put (paste) yanked text below current line
P	Put yanked text above current line
J	Join current line with the next line. Use gJ to exclude join-position space
"+yy	Copy to clipboard (can also use "*yy)
"+p	Paste from clipboard (can also use "*p)
"_d	Delete something without saving it in a register (_ for the "black hole register")
"_dP	Paste something and keep it available for further pasting

Normal (Command) Mode → Insert Mode

i	Enter insert mode to the left of the cursor
a	Enter insert mode to the right of the cursor
I	Enter insert mode at first character of current line
A	Enter insert mode at last character of current line
o	Insert line below current line and enter insert mode
O	Insert line above current line and enter insert mode
cm	Delete (change) the character or word (w) in motion m, then enter insert mode
cc	Delete current line and enter insert mode (unlike dd which leaves you in normal mode)
C	Delete (change) from cursor to end of line, and enter insert mode

Window Management

<C>+w s	Split current window horizontally
<C>+w v	Split current window vertically
<C>+w c	Close current window
<C>+w m	Move to window according to motion m
<C>+w o	Maximize current window (note: this overwrites your current window configuration)
:ls / :args	List all current open buffers
:bd	Buffer delete
:prev	Go to previous buffer
:n	Go to next buffer

Control: <C> Command: <D> Shift: <S> Option/Alt: <A> Carriage Return: <CR>

Visual Mode

v	Enter visual mode and highlight characters
V	Enter visual mode and highlight lines
<C>+v	Enter visual block mode and highlight exactly where the cursor moves
o	Switch cursor from first & last character of highlighted block while in visual mode
vat	Highlight all text up to and including the parent element
vit	Highlight all text up to the parent element, excluding the element
vac	Highlight all text including the pair marked with c (like va<, va' or va")
vic	Highlight all text inside the pair marked with c
v%	Select to the next matching parenthesis: if the cursor is on the starting/ending parenthesis
vib	Select to the next matching parenthesis: if the cursor is inside the parenthesis block
vi"	Select text between quotes: for double quotes
vi’	Select text between quotes: for single quotes
viB	Select a curly brace block, very common on C-style languages (can also use: vi{)
ggVG	Select the entire file
gg=G	Auto-indent code

Miscellaneous

u	Undo
U	Undo all changes on current line
<C>+r	Redo
:red[o]	Redo one change which was undone
.	Repeat last change or delete
;	Repeat last f, t, F, or T command
,	Repeat last f, t, F, or T command in opposite direction
gg=G	Format HTML. Make sure FileType is set to html with :setf html
<C>+n	Press after typing part of a word. It scrolls down the list of all previously used words
<C>+p	Press after typing part of a word. It scrolls up the list of all previously used words
guu	Lowercase line
gUU	Uppercase line
~	Invert case (upper->lower; lower->upper) of current character
gf	Open file name under cursor (goto file)
=	(Re)indent the text on the current line or on the area selected
=%	(Re)indent the current braces { ... }
ga or ga8	Display HEX, ASCII value / HEX value of utf-8 character under cursor
<C>+X, <C>+L	Word completion in insert mode
:echo expand('%:p')	Full path of the file (%:t for name of file 'tail', %:p:h for directory containing file 'head')
“a, y, “A, y, “ap	Copy multiple lines into the same buffer: operate on the buffer a, yank it, mark the next section and press "A - capitalizing the buffer name means "do not overwrite the buffer, append to it instead", yank again using y and paste the accumulated buffer a using "ap

Control: <C> Command: <D> Shift: <S> Option/Alt: <A> Carriage Return: <CR>

Regular Use

vimtutor	Starts editing a copy of a tutorial file
i	Insert mode. Next keys typed are inserted into the file.
<Esc>	Go back to normal mode so you can navigate and use edit commands
A	Append at end of line
o	Insert at new line below
u	Undo last command, again and again
x	Delete character under cursor
dw	Delete everything right from the cursor to the start of next word and put in the default register
dd	Delete line and put in the default register
p	Paste the default register
/*myname*	Search forward for myname
:wq	Write and quit (can also use :x)
:w *f*	Write a copy of the file you are editing as filename
:q!	Quit without saving even if changes were made!
:help	Display help
<Tab>	Use tab completion to scroll through commands that start with what you typed
^ w e $	Beginning of line, word, end of word, end of line
10w	Enter a number before a command to repeat it, examples: 10w: skip forward 10 words | 10dd: delete 10 lines
<C>+v	Visual block selection allows you to select a rectangular area of text, you just have to press Ctrl-v/V to start it, and then select the text block you want and perform any type of operation such as yank, delete, paste, edit, etc. It's great to edit *column oriented* text.
:sp f	Horizontal split
:vs f	Vertical split (can also use :vsp f)
* # g* g#	Find word under cursor (forwards/backwards)
%	Match brackets {}[]()
:syntax on	Color syntax in C, Perl, HTML, PHP etc.
:set wrap	Set word-wrap (disable word wrap using :set nowrap)
<C>+o	Execute a normal command without leaving insert mode
yw/yb	Copy word under cursor (beginning)/(end)
"_d	Delete something without saving it in a register (_ for the "black hole register")
:! cmd	Run shell commands

How to Exit

:q[uit]	Quit Vim. This fails when changes have been made
:q[uit]!	Quit without writing
:cq[uit]	Quit always, without writing
:wq	Write the current file and exit
:wq!	Write the current file and exit always
:wq {f}	Write to {f} and exit if not editing the last
:wq! {f}	Write to {f} and exit always

Control: <C> Command: <D> Shift: <S> Option/Alt: <A> Carriage Return: <CR>