

operator **[count]** **motion**

d delete/cut
y yank/copy
c change

gu **~**
uppercase swap case
< indent
shift left

Any motion can follow an operator. Marks and searches count as motions, too! **d/** **foo** will delete from the cursor to the next instance of "foo". **y3f** will yank from the cursor to the 3rd "f" on the line after it. Counts can also come before operators: **5dd** will delete five lines.

w word
W WORD
s sentence
[,] block
(,) block
<, > block
<, > XML/HTML tag
(,) block
" quoted string

(use text-objects)

i(**iW**

0 beginning of line
^ first non-blank character
^ previous WORD
B previous word
b previous character
h

SEARCHING

Prev	Next	Forward	Backward	Matches
N	n	/foo	?foo	foo
		*	#	word under cursor
		tx	Tx	upto x
		fx	Fx	find x

m set mark **#** (a-z) in file
m set mark **#** (A-Z) across files
'[jump to first char of just-changed text
'm jump to exact character of **#**
' jump back to last jump

Pass a directory to the **:edit** command to open a directory explorer. Instructions for usage are at the top of the screen.

ENTERING INSERT MODE

beginning of line **I** before cursor **i** after cursor **a** end of line **A**

previous line **O** next line **O** substitute character **s** substitute line **S** line from cursor **C**

ENTERING VISUAL (SELECT) MODE

v The most basic type. Use **Visual mode** to select characters within a line.

V Useful for moving chunks of a program around the file. Use **Visual mode** to select one or more lines.

Av Great for working with tables made of text, or anything that happens to be aligned. **Visual Block mode** can be used to select boxes across lines.

switch cursor to start/end **o** re-select previous area **gv** prepend to each Visual block line **I** jump to start of prior area **'<**

ZZ Write current file, if modified, and quit
ZQ Quit without checking for changes (like **q!**)

:write Write current file
:wq Write current file and quit

Use **:scriptnames** to list all files sourced during initialization.

:syntax Enable and configure syntax highlighting
Use **:sy sync fromstart** to redraw broken highlights

:make Run a compiler and enter quickfix mode

:!< Execute external shell command
!< Filter motion with shell command

Use **:earlier** and **:later** to quickly jump backward and forward in a file's history.

:read Read external program output into current file

gg first line
^b up 1 page
^u up 1/2 page
k up 1 line

ts **sw** **sts** **et**
use spaces only **n n n** on
use tabs only **n n 0** off

Set **n** to desired tab width (default 8)
expandtab **et** **<Tab>** inserts spaces

MIXING TABS AND SPACES IS RIGHT OUT. (that means don't do it.)

:retab Replace all tabs with spaces according to current **tabstop** setting

fileformat ff Try changing this if your line-endings are messed up

list Display whitespace visibly according to **listchars**

l next character
e end of word
w beginning of next word
E end of WORD
W beginning of next WORD
\$ end of line

p paste after cursor
P paste before cursor
^[] return to Normal mode

u undo
^r redo
. repeat

gf find file under cursor in path and jump to it
dd delete current line
yy yank current line

x delete character after cursor
% jump to matching paren
r replace char under cursor

nG jump to line **n**
^o jump back
^i jump forward

zz center screen on cursor
zt align top of screen with cursor
zb align bottom of screen with cursor

== auto-indent current line
<< shift current line left by **shiftwidth**
>> shift current line right by **shiftwidth**

Using **^[]** to return to Normal mode lets you keep your fingers on the home row. It's even easier if you map Caps Lock to Control.

COOL INSERT MODE STUFF

^w delete word before cursor
^u delete line before cursor

^rr insert the contents of register **r**
^r= use the expression register (try **^r=**)

^t increase line indent by **shiftwidth**
^d decrease line indent by **shiftwidth**

^x^l line completion
^n find next completion suggestion according to complete

COMMAND-LINE MODE ONLY

edit using Normal mode **^f** insert word under cursor **^r^w** completion suggestions **^d**

Put **onoremap << <C-R>:expand("%H"),/^/C<** in your **vimrc** so you can type **^f** in Command-line mode to refer to the directory of the current file, regardless of **pwd**.

Supply **%** as a range to the **:substitute** command to run it on every line in the file.

:%s/Scribble/Design/ "Scribble" -> "Designed"

Specify the **g** flag to apply the substitution to every match on a line.

:s/[dia]/g "badly" -> "by" **:h s_flags, :h [/]**

Vim supports many regular expression features.

:s/.k/ax/ "Mook" -> "Max" **:h usr_27, :h /.**

Use **_** instead of **_** if you want to search across multiple lines.

:%s/heat_.*Bungle/anto/ "Cheatsheet\hBunger" -> "Cantor" **:h _.**

Special escapes can be used to change the case of substitutions.

:s_.(f_)_U\1E_ "foobar" -> "FOObar" **:h sub-replace-special**

Use **:global** to perform a command on matching lines.

:g/foobar/delete Delete all lines containing "foobar"

If your pattern contains slashes, just use a different character as your delimiter.

_s_Data/Lore_Brent_Spiner_

Use **=** to evaluate expressions with replacement groups.

:s_d_\\submatch(0) + 1_g "10 25" -> "21 36" **:h sub-replace-\=**

:h cmd Normal mode **cmd** help

:h i_cmd Insert mode **cmd** help

:h v_cmd Visual mode **cmd** help

:h c_cmd Command-line editing **cmd** help

:h :cmd Command-line **cmd** help

:h 'option' **Option** help

:helpgrep Search through all help docs!

7 words **:h word-motions**

http://www.vimcheatsheet.com

1 WORD

:set opt? View current value of **opt**

:set noopt Turn off flag **opt**

:set opt Turn on flag **opt**

:set opt=val Overwrite value of **opt**

:set opt+=val Append to value of **opt**

:echo &opt Access **opt** as a variable

:ls List all open files

:b path Jump to unique file matching **path**. Use **<Tab>** to scroll through available completions!

:bn Jump to file **n**, number from first column of **:ls**

:bnext Jump to next file

:bprev Jump to previous file

:bdelete Remove file from the buffer list

:edit Open a file for editing

:enew Open a blank new file for editing

:split Split current window horizontally

:vsplit Split current window vertically

^w hjkl Move cursor to window left, below, above or to the right of the current window

^w HJKL Move current window to left, bottom, top, or right of screen

^w r Rotate windows clockwise

^w +-<> Increase/decrease current window height/width

^w T Move current window to a new tab

:only Close all windows except current window

:bufdo Execute a command in each open file

vim

<CR> **^m** **\r** Enter

<Tab> **^i** **\t** Tab

<C-n> **^n** **Ctrl-n**

<M-n> **Alt-n**

<Esc> **^[** Escape

<BS> **^h** **\b** Backspace

**** Delete

^] **:h tags-and-searches**
Jump to tag under cursor, including [tags] in help files

^t Jump back up the tag-list

g^] Jump to tag if it's the only match; else list matching tags

7 words **:h word-motions**

http://www.vimcheatsheet.com

1 WORD

hidden **hid** Lets you switch buffers without saving

laststatus **ls** Show status line never (0), always (2) or with 2+ windows (1)

hlsearch **hls** Highlight search matches. Also see 'highlight'

number **nu** Show line numbers

showcmd **sc** Show commands as you type them

ruler **ru** Show line and column number of the cursor

backspace **bs** Set to '2' to make backspace work like sane editors

wrap Control line wrapping

background **bg** Set to 'dark' if you have a dark color scheme

REGISTERS are **CLIPBOARDS**

All commands that delete, copy, or paste text use registers. To change which register is used by a command, type the register before the command. The default register is called "the unnamed register", and it is invoked with a pair of double-quotes ("**''**"). Typing **dd** or **yy** is the same as typing "**''dd**" or "**''yy**". Think of the first **'** as a short way of saying "register", so **''** is pronounced "register **'**", and **''a** "register **a**".

:registers View all current registers

:echo @r Access register **r** as a variable

"/ Last search pattern register Contains the last pattern you searched for

"_ The black hole register Use this to delete without clobbering any register (**''dd**)

"0 Last yank register Contains the last text you yanked

"1 Last big delete register Contains the last line(s) you deleted

"2-"9 Big delete register stack Every time **"1** is written to, its content is pushed to **"2**, then **"2** to **"3**, and so on

"_ Small delete register Contains the last text you deleted within a single line

"+ System clipboard If the OS integration gods smile upon you, this register reads and writes to your system clipboard.

"a-"z Named registers 26 registers for you to play with

"A-"Z Append registers Using upper-case to refer to a register will append to it rather than overwrite it

qr Record Record into register **r**. Stop recording by hitting **q** again

@r Playback Execute the contents of register **r**

@@ Repeat last playback Repeat the last **@r**, this is particularly useful with a count

vim one-liner used to sort the list of names by length:
:sort |let @s = join getline(".*") | normal "a" | sort n | :g/more de

Last Edition
taken by...
M. Adam
jilene
Lin Du
Benjamin
graciele
George
David G.
Dimitris
Drew Remy
Rajeev Beem
Ayaz Lekan
Rajeev Shree
Paul Herli
Neri Eason
Nell Rocco
Kylie Verro
Roy Remy
Rosa Tinson
Andrii Dots
David Beck
David Smith
Katie Rella
David Chen
Pivotal Labs
Eric Bortone
M. Adam Price
David Ruppert
Bill Lafferty
Brandon Byrne
Laurence Ems
Dimitris de
Benedictus
Drew Remy
Tristan Hensley
Dennis Dugan
Rajeev Herli
Digital Omeas
Drew Remy
Jane O. Herli
Thomas Hensley
Philip Rella
Hartley Labs
Michael Smith
Nell Rocco
Ayaz Rella-Herli
David Ruppert
Christie Eason
B. Louis Via Beem
Harold Tristan Hensley
and others!