



start of line **0** first non-blank char **^** previous WORD **B** previous word **b** previous character **h**

**SEARCHING**

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

**mark-motions**

<b>mm</b> set mark <b>m</b> (a-z) in file	<b>mM</b> set mark <b>#</b> (A-Z) across files
<b>'m</b> jump to first char of line containing <b>m</b>	<b>~m</b> jump to exact character of <b>m</b>
<b>'[</b> jump to first char of just-changed text	<b>''</b> 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 <b>I</b>	before cursor <b>i</b>	after cursor <b>a</b>	end of line <b>A</b>
previous line <b>O</b>	next line <b>o</b>	substitute character <b>s</b>	substitute line <b>S</b>
			line from cursor <b>C</b>

## COOL INSERT MODE STUFF

<b>^W</b> delete word before cursor	<b>^U</b> delete line before cursor
<b>^Rr</b> insert the contents of register <b>r</b>	<b>^R=</b> use the expression register (try <b>^R=5+10</b> )
<b>^t</b> increase line indent by <b>shiftwidth</b>	<b>^d</b> decrease line indent by <b>shiftwidth</b>

<b>^x^l</b> line completion	<b>t^h</b> find next completion suggestion using <b>complete</b>
-----------------------------	--

## COMMAND-LINE MODE ONLY

edit using Normal mode <b>^f</b>	insert word under cursor <b>^R^W</b>	completion suggestions <b>^d</b>
----------------------------------	--------------------------------------	----------------------------------

<b>ZZ</b> Write current file, if modified, and quit	<b>ZQ</b> Quit without checking for changes (like <b>:q!</b> )
---	--

**:write** Write current file

**:wq** Write current file and quit

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

**:syntax** Configure syntax highlighting. Use **:sy sync fromstart** to redraw broken highlights

**:make** Run a compiler and enter quickfix mode

<b>:!</b> Execute external shell command	<b>!</b> 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

**j** down 1 line

**^d** down 1/2 page

**^f** down 1 page

**G** last line

**j** down 1 line

**^d** down 1/2 page

**^f** down 1 page

**G** last line

**j** down 1 line

**^d** down 1/2 page

**^f** down 1 page

**G** last line

**j** down 1 line

**^d** down 1/2 page

**^f** down 1 page

**G** last line

**j** down 1 line

**^d** down 1/2 page

**^f** down 1 page

**G** last line

**j** down 1 line

**^d** down 1/2 page

**^f** down 1 page

**G** last line

**j** down 1 line

**^d** down 1/2 page

**^f** down 1 page

**G** last line

**j** down 1 line

**^d** down 1/2 page

**^f** down 1 page

**G** last line

**j** down 1 line

**^d** down 1/2 page

**^f** down 1 page

**G** last line

**j** down 1 line

**^d** down 1/2 page

**^f** down 1 page

**G** last line

**j** down 1 line

**^d** down 1/2 page

**^f** down 1 page

**G** last line

**:h up-down-motions**

	<b>ts</b>	<b>sw</b>	<b>sts</b>	<b>et</b>	<b>tabstop</b>	<b>ts</b>	Columns per tabstop
use spaces only	<b>n</b>	<b>n</b>	<b>n</b>	on	<b>shiftwidth</b>	<b>sw</b>	Columns per <<
use tabs only	<b>n</b>	<b>n</b>	<b>0</b>	off	<b>softtabstop</b>	<b>sts</b>	Spaces per tab
Set <b>n</b> to desired tab width					<b>expandtab</b>	<b>et</b>	<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

**:h left-right-motions**

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

<b>p</b> paste after cursor	<b>P</b> paste before cursor	<b>^L</b> return to Normal mode
<b>u</b> undo	<b>^R</b> redo	<b>.</b> repeat
<b>gf</b> find file under cursor in <b>path</b> and jump to it	<b>dd</b> delete current line	<b>yy</b> yank current line
<b>x</b> delete character after cursor	<b>%</b> jump to matching paren	<b>r</b> replace char under cursor
<b>nG</b> jump to line <b>n</b>	<b>^O</b> jump back	<b>^i</b> jump forward
<b>zz</b> center screen on cursor	<b>zt</b> align top of screen with cursor	<b>zb</b> align bottom of screen with cursor
<b>==</b> auto-indent current line	<b>&lt;&lt;</b> shift current line left by <b>shiftwidth</b>	<b>&gt;&gt;</b> shift current line right by <b>shiftwidth</b>

Using **^L** 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!

ENTERING VISUAL (SELECT) MODE		
<b>v</b>	The most basic type. Use <b>Visual mode</b> to select characters within a line.	<b>V</b>
	Useful for moving chunks of a program around the file. Use <b>Visual Line mode</b> to select one or more lines.	<b>^v</b>
		Great for working with tables made of text, or anything that happens to be conveniently aligned. <b>Visual Block mode</b> can be used to select boxes across lines.

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

Put **cnoremap %<C-R>=expand('%:h')'/'<C-R>** in your **.vimrc** so you can type **%** in Command-line mode to refer to the directory of the current file, regardless of **pwd**.

**:h tags-and-searches**

<b>^]</b> Jump to tag under cursor, including [tags] in help files	<b>^t</b> Jump back up the tag-list
<b>g^]</b> Jump to tag if it's the only match; else list matching tags	<b>g]</b> Show tag list for word under cursor, even if it's the only one

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

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

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

**:%s/[dla]/g** "badly" -> "by" **:h s\_flags, :h [/]**

Vim supports many regular expression features.

**:%s/./a/** "Mook" -> "Max" **:h usr\_27, :h /.**

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

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

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

**:%s\\_(\f.\\_)\\_U\I\E** "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-=**

**:help**

<b>:h cmd</b>	Normal mode <b>cmd</b> help
<b>:h i_cmd</b>	Insert mode <b>cmd</b> help
<b>:h v_cmd</b>	Visual mode <b>cmd</b> help
<b>:h c_cmd</b>	Command-line editing <b>cmd</b> help
<b>:h :cmd</b>	Command-line <b>cmd</b> help
<b>:h 'option'</b>	<b>Option</b> help
<b>:helpgrep</b>	Search through all help docs!

**:h word-motions**

**7 words** **1 WORD**

<b>:set opt?</b>	View current value of <b>opt</b>
<b>:set noopt</b>	Turn off flag <b>opt</b>
<b>:set opt</b>	Turn on flag <b>opt</b>
<b>:set opt=val</b>	Overwrite value of <b>opt</b>
<b>:set opt+=val</b>	Append to value of <b>opt</b>
<b>:echo &amp;opt</b>	Access <b>opt</b> as a variable

**:h buffers**

<b>:ls</b>	List all open files
<b>:b path</b>	Jump to unique file matching <b>path</b> . Use <b>&lt;Tab&gt;</b> to scroll through available completions!
<b>:bn</b>	Jump to file <b>n</b> , number from first column of <b>ls</b>

**: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

**:h windows**

**:h buffers**

**:h registers**

**:h registers**

**:h registers**

**:h registers**

**:h registers**

**:h registers**

**:h registers**

**:h registers**

<b>&lt;CR&gt;</b>	<b>^m</b>	<b>\r</b>	Enter
<b>&lt;Tab&gt;</b>	<b>^i</b>	<b>\t</b>	Tab
<b>&lt;C-n&gt;</b>	<b>^n</b>		Ctrl- <b>n</b>
<b>&lt;M-n&gt;</b>			Alt- <b>n</b>
<b>&lt;Esc&gt;</b>	<b>^[</b>		Escape
<b>&lt;BS&gt;</b>	<b>^h</b>	<b>\b</b>	Backspace
<b>&lt;Del&gt;</b>			Delete

Use **^** instead of **l** when beginning text-object motions to include delimiters or surrounding whitespace. For example, **d1(** will change "(foo)" into "()", but **da(** will delete the parentheses as well.

Use **map** to view all current custom key mappings. Read **:h map-which-keys** for a guide on which keys are best for your own custom mappings. Get used to Vim's help system - it's a fantastic resource!

**:h word-motions**

**1 WORD**

<b>hidden</b>	<b>hid</b>	Lets you switch buffers without saving
<b>laststatus</b>	<b>ls</b>	Show status line never (0), always (2) or with 2+ windows (1)
<b>hlsearch</b>	<b>hls</b>	Highlight search matches. Also see 'highlight'
<b>number</b>	<b>nu</b>	Show line numbers
<b>showcmd</b>	<b>sc</b>	Show commands as you type them
<b>ruler</b>	<b>ru</b>	Show line and column number of the cursor
<b>backspace</b>	<b>bs</b>	Set to '2' to make backspace work like sane editors
<b>wrap</b>		Control line wrapping
<b>background</b>	<b>bg</b>	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".

**:h registers**

**: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

**0** Contains the last text you yanked

**1** Last big delete register

**1** Contains the last line(s) you deleted

**2-9** Big delete register stack

**2-9** 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

**a-z** 26 registers for you to play with

**A-Z** Append registers

**A-Z** Append to register it rather than overwrite it

**qr** Record

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

**@r** Playback

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

**@@** Repeat last playback

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



**FIRST EDITION funded by...**

Haraldur Tristan Gunnarsson	Travis McHenry
St. Louis Vin Geeks	Damon Jablons
Christer Edvartsen	Bandersnatch
Lin Qiu	Infonautic.de
David Macfarlane	@webjoe
Arya Reais-Parsi	grantok
Matthew Datcher	storgie
Michael Schmidt	David O.
Hurricane Labs	Digininja
Philip Ratzsch	Dave Reay
Thomas Nitsche	Ralph Bean
Jens O. Meiert	Arve Loken
Sam Napolitano	Rajat Ghai
Digital Onomes	GuY Bensky
Rouven Hernier	Kyle Verma
Ganesh Sugunan	Ross Timson

vim one-liner used to sort the list of names by length:  
**exe 'g"/let &x = len(getline(" ")) | normal <20% ^ | sort n | :g/normal de**