

HOW-TO make Vim not suck Out of the Box: `:help statusline` `:set nocompatible ruler laststatus=2 showcmd showmode number` Search `:set incsearch ignorecase smartcase hlsearch` Remove useless splash screen `:set shortmess+=!`

Best tips: <http://vim.wikia.com/> Best scripts: <http://www.vim.org/scripts/index.php> `:map <F9> :e $HOME/_vimrc<CR>` `:map <F6> :so $HOME/_vimrc<CR>`

	<code>~</code> toggle case	<code>!</code> extern filter	<code>@.</code> play macro	<code>#</code> prev identifier	<code>\$</code> →	<code>%</code> goto match	<code>^</code> soft ←	<code>&</code> repeat :s	<code>*</code> next identifier	<code>(</code> begin sentence	<code>)</code> end sentence	<code>_</code> cur line	<code>+</code> ↓
	<code>`</code> goto mark ¹³	<code>1</code>	<code>2</code>	<code>3</code>	<code>4</code>	<code>5</code>	<code>6</code>	<code>7</code>	<code>8</code>	<code>9</code>	<code>0</code> hard ←	<code>-</code> ↑	<code>=</code> auto-format ²
		<code>14</code> block select	<code>^w</code> window...	scroll line ↑	<code>12</code> :redo	<code>10</code> ctags return	scroll line ↓	half page ↑	<code>-Ctrl-+</code>	prev mark	<code>9</code> ↑	Normal	ctags identifier
	<code>Q</code> ex mode	<code>W</code> WORD ↘	<code>E</code> end WORD ↘	<code>R</code> Replace	<code>T</code> ← until char	<code>Y</code> copy line	<code>U</code> undo line	<code>I</code> insert ←	<code>O</code> open ↑	<code>P</code> paste ↑	<code>{</code> ↖ paragraph	<code>}</code> ↗ paragraph	
Tab	<code>q</code> record macro	<code>w</code> word ↘	<code>e</code> end word ↘	<code>r</code> replace char	<code>t</code> until char →	<code>y</code> copy ⁰	<code>u</code> undo	<code>i</code> ↵ insert	<code>o</code> open ↓	<code>p</code> paste ↓	<code>[</code> misc.	<code>]</code> misc.	
		<code>7</code> incr. #		<code>10</code> half page ↓	page ↓	file/cursor info	<code>-Ctrl-H</code>	<code>15</code> Ctrl-J		redraw	<code>-Ctrl-;</code>	<code>-Ctrl-'</code>	<code>-Ctrl-I</code>
Gaps	<code>A</code> append →	<code>S</code> subst line	<code>D</code> del →	<code>F</code> ← find char	<code>G</code> goto eof / goto line#	<code>H</code> Top screen	<code>J</code> Join lines	<code>K</code> man page identifier	<code>L</code> Bottom screen	<code>:</code> Ex cmd line	<code>"</code> register ⁰	<code> </code> goto col#	
	<code>a</code> append ↵	<code>s</code> subst char	<code>d</code> del ⁰	<code>f</code> find char →	<code>g</code> extra ⁶	<code>h</code> ←	<code>j</code> ↓	<code>k</code> ↑	<code>l</code> →	<code>;</code> "next" ¹⁶ f/F/t/T	<code>'</code> goto mark ←	<code> </code>	
Ctrl ^	<code>:suspend</code> ^{7,11}	<code>decr. #</code>	Normal / Cancel	block select	page ↑	<code>9,16</code> ↓	<code>15</code> <code>-Ctrl-M</code>	<code>-Ctrl-,</code>	<code>-Ctrl-.</code>	<code>-Ctrl-/</code>	Unused & Duplicate keys \ Ctrl-K Ctrl-S (free) ¹³ Ctrl-L (redraw) ¹³ ~ near dup of ' ¹⁴ Ctrl-Q = Ctrl-V ¹⁵ Ctrl-J = Ctrl-M = ^N		
Shift ↑	<code>Z</code> quit	<code>X</code> ← del char	<code>C</code> change →	<code>V</code> select lines	<code>B</code> ↖ WORD	<code>N</code> "prev" find	<code>M</code> Middle screen	<code><</code> undent ²	<code>></code> indent ²	<code>?</code> find ↖			
	<code>z</code> extra ⁵	<code>x</code> del char →	<code>c</code> change ⁰	<code>v</code> select chars	<code>b</code> ↖ word	<code>n</code> find "next"	<code>m</code> set mark	<code>,</code> "prev" f/F/t/T ¹⁶	<code>.</code> repeat cmd	<code>/</code> find ↘			

Legend: ¹⁶ The search direction is relative; **next** is the initial direction, **previous** is the opposite direction. `~` repeat same initial direction find. `~` repeat opposite initial direction find. **Note:** `;` only searches cursor line, `n` searches buffer.

Macro	Register name (0-9a-zA-Z) required	:help cmdline	:r file insert file	:help tags	<code>\</code> <code>map</code> \ :Explore<CR> manually type <C,R,>	:buffer #
Op	Motion req.; act between cursor & dst	<code>:w</code> save	<code>:gui</code> switch to GUI	<code>:ts</code> list active tags	<code>\$0</code> <code>"</code> before del/copy/paste to use register	:buffers list
Cmd	Command	<code>:q</code> quit	<code>:q!</code> quit w/o save	<code>^]</code> jump to tag under cursor	<code>"x</code> cut to system clipboard reg. '+'	:new blank file/buffer
Ins	Command and enter insert mode	<code>:e <file></code> edit file in new buffer	<code>:source %</code> exec cmds in cur file	<code>^t</code> restore cursor before tag jump	<code>+gp</code> paste from system clipboard	:bn next file
Move	Moves cursor or defines range for op	<code>:exec '...' do cmd</code>		<code>^p</code> complete word	¹ Number before any action repeats it	:bp prev file
Find	Search (<code>\</code> = reverse, <code>/</code> = forward)			<code>:ta Foo</code> manual jump to tag 'Foo'	<code>2p</code> paste twice ³ repeat thrice	:bd close file
tag	ctags / diffs / folding				² Repeat <code>op</code> to act on current line	:bd! force close
Code	Code formatting, whitespace, etc.				<code>yy</code> copy line <code>dd</code> del line	:set lines=#
Extra	Extended functionality; req. extra chars				<code><<</code> undent line <code>>></code> indent line	:set columns=#
•	Char arg req. <code>g z Z ^w ' " ` ...</code>				³ highlight words under cursor	:winpos # # GUI
Modes	:help modes				<code>4 ZZ</code> save & quit <code>ZQ</code> quit w/o save	Windows
n	Normal <code>Esc</code> <code>^</code> <code>^c</code>				<code>5 zz</code> center cursor line in window	:help windows
i	Insert <code>a i r s</code>				<code>zh</code> scroll left <code>zl</code> scroll right	<code>^w</code> or <code>:wincmd</code> •
v	Visual <code>v V ^v ^q</code>				<code>zt</code> scroll top <code>zb</code> scroll bottom	<code>w</code> :switch to next
o	Op pending <code>c d y < ></code>				<code>\$6 gg</code> top of file	<code>c</code> :close!
c	Command Line <code>:</code> <code>/</code> <code>?</code> <code>!</code>				<code>gf</code> open file under cursor	<code>n</code> :new
word	<code>Foo</code> <code>(</code> <code>src</code> <code>,</code> <code>dst</code> <code>,</code> <code>len</code> <code>)</code> ;				⁷ <code>^a</code> incr # under cursor (Dec / Hex)	<code>s</code> :split horz.
Note: There is no whitespace in-between 'Foo(src,' but before/after 'dst.'					<code>^x</code> decr # under cursor (Dec / Hex)	<code>v</code> :vsplit vertical
WORD	<code>Foo</code> <code>(</code> <code>src</code> <code>,</code> <code>dst</code> <code>,</code> <code>len</code> <code>)</code> ;				⁸ <code>*</code> start a "new" search	<code>o</code> :only maximize
Startup					⁹ <code>^p</code> prev auto-complete <code>^n</code> next	<code>=</code> all same size
vim <filename> +123 goto line 123					¹⁰ <code>^d</code> undent <code>^t</code> indent	<code>h</code> move to win ←
vim <file> ... -t Foo edit at tag 'Foo'					¹¹ <code>^x</code> <code>^f</code> filename completion	<code>j</code> move to win ↓
vim <file> ... -c "/Foo" cmd: find 'Foo' & edit					<code>^s</code> spelling <code>:set spell!</code>	<code>k</code> move to win ↑
GUI vim -g or gvim start GUI ver.					<code>^k</code> dictionary <code>:help c_CTRL-R</code>	<code>l</code> move to win →
GUI Linux :set guifont=ProggyTinyTT\ 12					<code>^t</code> thesaurus <code>:help spell</code>	<code>:sp [<filename>]</code>
GUI OSX :set guifont=ProggyTiny\ :h11					¹² <code>^r</code> paste register 0-9a-zA-Z or ...	edit in split window
diff gvimdiff <file1> <file2> [<file3>]					<code>+</code> clipboard (or '*') <code>:help c_CTRL-R</code>	Cursor Bookmarks
bug					<code>"</code> last del/copy % filename	<code>:marks</code> <code>^o</code>
Broken Keys -Ctrl-I=Tab, -Ctrl-!=ESC						<code>ma</code> mark local 'a'
Vim is still unable to map certain keys for your own use...						<code>'A</code> goto global 'A'
\$ Caps, Ctrl-1, Ctrl-Shift-1, Ctrl-I, Ctrl-!, etc.						<code>'</code> prev location
⁰ See: src/ops.c -c "/valid_yank_reg" for ^ reg. names						File / Directory
⁶ See: src/normal.c -c "/nv_cmds" for g* extra cmds						:Explore Or :e .
¹¹ See: src/edit.c -c "/ctrl_x_msgs" for ^x* insert cmds						:set browseDir=...
						one of buffer last