

Esc Normal

Revision 2.0  
Sept. 11, 2011

Vim 7.3+  
:version

# Vim Cheat Sheet for Programmers

Copyright © 2011  
May be freely distributed!  
Sharing is Caring.

<http://michael.PeopleOfHonorOnly.com/vim/>

**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+=I`

**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>`

<b>Ctrl-`</b>	<b>Ctrl-1</b>	<b>Ctrl-@</b>	<b>Ctrl-3</b>	<b>Ctrl-4</b>	<b>Ctrl-5</b>	<b>Ctrl-^</b>	<b>Ctrl-7</b>	<b>Ctrl-8</b>	<b>Ctrl-9</b>	<b>Ctrl-0</b>	<b>Ctrl-_-</b>	<b>Ctrl-=</b>
~ toggle case	! extern filter	@• play macro	# prev identifier	\$ → % goto match	^ soft →	& repeat :s	* next identifier	( begin sentence	) end sentence	_ cur line	+ ↓	
• goto mark	1	2	3	4	5	6	7	8	9	0 hard →	- ↑	= auto-format

<b>14</b>	<b>block select</b>	<b>^w</b> window...	<b>scroll line ↑</b>	<b>12</b>	<b>:redo</b>	<b>10</b>	<b>ctags return</b>	<b>scroll line ↓</b>	<b>half page</b>	<b>Ctrl-+</b>	<b>prev mark</b>	<b>9</b>	<b>↑</b>	<b>Normal</b>	<b>ctags identifier</b>
<b>Q</b> ex mode	<b>W</b> WORD ↘	<b>E</b> end WORD ↘	<b>R</b> Replace	<b>T</b> ← until char	<b>Y</b> copy line	<b>U</b> undo line	<b>I</b> insert ↵	<b>O</b> open ↑	<b>P</b> paste ↑	<b>{</b> paragraph	<b>}</b> paragraph				
<b>q•</b> record macro	<b>w</b> word ↘	<b>e</b> end word ↘	<b>r</b> replace char	<b>t</b> until char →	<b>y</b> copy	<b>u</b> undo	<b>i</b> ↵ insert	<b>o</b> open ↓	<b>p</b> paste ↓	<b>[</b> misc.	<b>]</b> misc.				

<b>7</b>	<b>incr. #</b>		<b>10</b>	<b>half page ↓</b>	<b>page ↓</b>	<b>file/cursor info</b>	<b>Ctrl-H</b>	<b>15</b>	<b>Ctrl-J</b>		<b>redraw</b>	<b>Ctrl-;</b>	<b>Ctrl-'</b>	<b>Ctrl-\</b>
<b>A</b> append →	<b>S</b> subst line	<b>D</b> del →	<b>F</b> ← find char	<b>G</b> goto eof / goto line#	<b>H</b> Top screen	<b>J</b> Join lines	<b>K</b> man page identifier	<b>L</b> Bottom screen	<b>:</b> cmd line	<b>"</b> register	<b>:</b> goto mark	<b>:</b> goto col#		
<b>a</b> append ↵	<b>s</b> subst char	<b>d</b> del	<b>f</b> find char →	<b>g</b> extra	<b>h</b> ←	<b>j</b> ↓	<b>k</b> ↑	<b>l</b> →	<b>:</b> "next" f/F/T	<b>:</b> "next" f/F/T	<b>:</b> "next" f/F/T	<b>:</b> "next" f/F/T		

<b>Ctrl-^</b>	<b>:suspend</b>	<b>7,11</b>	<b>decr. #</b>	<b>Normal / Cancel</b>	<b>block select</b>	<b>page ↑</b>	<b>9,16</b>	<b>↓</b>	<b>15</b>	<b>Ctrl-M</b>	<b>Ctrl-,</b>	<b>Ctrl-.</b>	<b>Ctrl-/</b>	<b>Unused &amp; Duplicate keys</b> \ Ctrl-L Ctrl-S (free) Ctrl-L (redraw) 13 ` near dup of 14 Ctrl-Q = Ctrl-V 15 Ctrl-J = Ctrl-M = ^N	
<b>Shift ↑</b>	<b>Z</b> quit	<b>X</b> ← del char	<b>C</b> change →	<b>V</b> select lines	<b>B</b> ↖ WORD	<b>N</b> "prev" find	<b>M</b> Middle screen	<b>&lt;</b> undent	<b>&gt;</b> indent	<b>?</b> find ↖					
	<b>z</b> extra	<b>x</b> del char →	<b>c</b> change	<b>v</b> select chars	<b>b</b> ↖ word	<b>n</b> find "next"	<b>m</b> set mark	<b>,</b> f/F/T	<b>.</b> repeat cmd	<b>/</b> find ↘					

<sup>16</sup> The search direction is relative; **next** is the initial direction, **previous** is the opposite direction. **n** repeat same initial direction find. **N** repeat opposite initial direction find. **Note**: **:** only searches cursor line, **n** searches buffer.

## Legend:

<b>Macro</b>	Register name (0-9a-zA-Z) required
<b>Op</b>	Motion req.; act between cursor & dst
<b>Cmd</b>	Command
<b>Ins</b>	Command and enter insert mode
<b>Move</b>	Moves cursor or defines range for op
<b>Find</b>	Search (↖ = reverse, ↘ = forward)
<b>tag</b>	ctags / diffs / folding
<b>Code</b>	Code formatting, whitespace, etc.
<b>Extra</b>	Extended functionality; req. extra chars
<b>Char arg req.</b>	<code>g z Z ^w ' ' ` ...</code>
<b>Modes</b>	<code>:help modes</code>
<b>n</b>	Normal <code>Esc [ ^ c</code>
<b>i</b>	Insert <code>a i r s</code>
<b>v</b>	Visual <code>v V ^v ^q</code>
<b>o</b>	Op pending <code>c d y &lt; &gt;</code>
<b>c</b>	Command Line <code>: / ? !</code>

**word** `Foo ( src , b dst , b len ) ;`

**WORD** `Foo ( src , b dst , b len ) ;`

**Startup**

<code>vim &lt;filename&gt; +123</code>	<b>goto line 123</b>
<code>vim &lt;file&gt; ... -t Foo</code>	<b>edit at tag 'Foo'</b>
<code>vim &lt;file&gt; ... -c "/Foo"</code>	<b>cmd: find 'Foo' &amp; edit</b>
<code>vim -g</code> or <code>gvim</code>	<b>start GUI ver.</b>
<code>GUI Linux :set guifont=ProggyTinyT\ 12</code>	
<code>GUI OSX :set guifont=ProggyTiny\h11</code>	
<code>diff gvimdiff &lt;file1&gt; &lt;file2&gt; [&lt;file3&gt;]</code>	

**Broken Keys** `Ctrl-I = Tab`, `Ctrl-[] = ESC`  
Vim is still unable to map certain keys for your own use...

**Caps, Ctrl-1, Ctrl-Shift-1, Ctrl-I, Ctrl-[], etc.**

**0 See:** `src/ops.c -c "/valid_yank_reg"` for `* reg. names`

**6 See:** `src/normal.c -c "/nv_cmds"` for `g* extra cmds`

**11 See:** `src/edit.c -c "/ctrl_x_msgs"` for `^x* insert cmds`

<code>:help cmdline</code>	<code>:r file</code> insert file
<code>:w save</code>	<code>:gui</code> switch to GUI
<code>:q quit</code>	<code>:q!</code> quit w/o save
<code>:e &lt;file&gt;</code>	edit file in new buffer
<code>:source %</code>	exec cmds in cur file
<code>:exec '...' do cmd</code>	

<code>:help movement</code>	
<code>soft ^ ← Start of Line</code>	1st non-whitespace
<code>hard 0 ← Start of Line</code>	column 0
<code>\$ → End of Line</code>	
<code> </code> move col 0	<code># </code> move col #
<code>~</code> page ↑	<code>^f</code> page ↓
<code>u</code> 1/2 page ↑	<code>^d</code> 1/2 page ↓
<code>e</code> scroll line ↑	<code>y</code> scroll line ↓
<code>lg</code> start of file	<code>og</code> end of file
<code>g</code> goto line #	<code>g</code> end of file
<code>[</code> begin this func {	
<code>]</code> begin next func {	
<code>:set matchpairs=(;):[,;],&lt;&gt;,:?</code>	
<code>%</code> goto matching { } <> [ ]	

<code>soft ^ ← Start of Line</code>	1st non-whitespace
<code>hard 0 ← Start of Line</code>	column 0
<code>\$ → End of Line</code>	
<code> </code> move col 0	<code># </code> move col #
<code>~</code> page ↑	<code>^f</code> page ↓
<code>u</code> 1/2 page ↑	<code>^d</code> 1/2 page ↓
<code>e</code> scroll line ↑	<code>y</code> scroll line ↓
<code>lg</code> start of file	<code>og</code> end of file
<code>g</code> goto line #	<code>g</code> end of file
<code>[</code> begin this func {	
<code>]</code> begin next func {	
<code>:set matchpairs=(;):[,;],&lt;&gt;,:?</code>	
<code>%</code> goto matching { } <> [ ]	

<code>:help range</code>	
<code>:s/Foo/Bar</code>	find Foo replace w/ Bar
<code>:s/Foo/Bar/g</code>	...all instances on line
<code>:%s/Foo/Bar</code>	apply to whole file
<code>.,.+ #</code>	cur line, cur line + # lines
<code>\$</code>	last line
<code>'&lt;</code>	start of select
<code>'&gt;</code>	end of select

<code>Code</code>	<code>&lt; &gt; &lt;&lt; &gt;&gt;</code>
<code>:set backspace=indent,eol,start</code>	allow backspace join lines
<code>:set shiftwidth=#</code>	indent width for ai
<code>:set autoindent!</code>	toggle auto-indent
<code>:set lisp</code>	lisp indent mode

<b>Tags</b>	
<code>:ts</code>	list active tags
<code>^]</code>	jump to tag under cursor
<code>^t</code>	restore cursor before tag jump
<code>^p</code>	complete word
<code>:ta Foo</code>	manual jump to tag 'Foo'

<b>Diff</b>	
<code>[c</code> prev diff	<code>:hi DiffAdd guifg=#rrggbb</code>
<code>]c</code> next diff	<code>:hi DiffChange guifg=#rrggbb</code>
<code>:diffupdate</code>	<code>:hi DiffText gui=none</code>
<code>resync</code>	<code>:hi DiffDelete</code>

<b>Folding</b>	<b>Changes</b>
<code>zR</code> fold remove	<code>:changes</code>
<code>zo</code> fold open	<code>g</code> older change
<code>zc</code> fold close	<code>g</code> newer change
<code>zi</code> invert all	
<code>zr</code> fold reduce	
<code>zm</code> fold more	

<b>Syntax</b>	
<code>:syntax enable</code>	
<code>:set filetype=</code>	
<code>c cpp sh make perl python</code>	Note: chose only ONE type!
<code>Convert &lt;eol&gt;</code>	
<code>:set fileformat=</code>	unix or dos or mac
then <code>:w</code> to convert	

<code>:set tabstop=#</code>	set tab stop every #th col
<code>:set expandtab!</code>	toggle hard/soft tabs
<code>:set listchars=...</code>	toggle whitespace
<code>:set list!</code>	toggle right margin indicator
<code>:set colorcolumn=80</code>	block comment
<code>noremap + :s/^\\//&lt;CR&gt;</code>	

<code>\\ :map \\ :Explore&lt;CR&gt;</code>	manually type <C,R>
<code>\$0</code>	" before del/copy/paste to use register
<code>" +x</code>	cut to system clipboard reg. '+'
<code>" +gP</code>	paste from system clipboard
<code>1</code>	Number before any action repeats it
<code>2p</code>	paste twice
<code>3.</code>	repeat thrice
<code>2</code>	Repeat op to act on current line
<code>yy</code>	copy line
<code>dd</code>	del line
<code>&lt;&lt;</code>	undent line
<code>&gt;&gt;</code>	indent line
<code>3 #</code>	highlight words under cursor
<code>4 zz</code>	save & quit
<code>5 zz</code>	center cursor line in window
<code>zh</code>	scroll left
<code>zt</code>	scroll top
<code>zb</code>	scroll bottom
<code>\$6 gg</code>	top of file
<code>gf</code>	open file under cursor
<code>7 ^a</code>	incr # under cursor (Dec / Hex)
<code>^x</code>	decr # under cursor (Dec / Hex)
<code>8 *</code>	start a "new" search

<b>Insert mode</b>	
<code>9 ^p</code>	prev auto-complete ^n next
<code>10 ^d</code>	undent
<code>^t</code>	indent
<code>\$11 ^x* ^f</code>	filename completion
<code>^s</code>	spelling
<code>^k</code>	dictionary
<code>^t</code>	thesaurus
<code>12 ^r*</code>	paste register 0-9a-zA-Z or ...
<code>+</code>	clipboard (or '*')
<code>"</code>	last del/copy % filename
<code>:set numbers!</code>	toggle line numbers
<code>:set wrap!</code>	toggle linewidth display
<code>:set showmatch</code>	highlight matching ()
<code>noremap - :s/^\\//&lt;CR&gt;</code>	uncomment

<code>:buffer #</code>	
<code>:buffers</code>	list files
<code>:new</code>	blank file/buffer
<code>:bn</code>	next file
<code>:bp</code>	prev file
<code>:bd</code>	close file
<code>:bd!</code>	force close
<code>:set lines=#</code>	
<code>:set columns=#</code>	
<code>:winpos # #</code>	GUI
<b>Windows</b>	
<code>:help windows</code>	
<code>^w</code> or <code>:wincmd</code>	
<code>w</code>	switch to next
<code>c</code>	close!
<code>n</code>	new
<code>s</code>	split horz.
<code>v</code>	split vertical
<code>o</code>	only maximize
<code>=</code>	all same size
<code>h</code>	move to win ←
<code>j</code>	move to win ↓
<code>k</code>	move to win ↑
<code>l</code>	move to win →
<code>:sp [&lt;filename&gt;]</code>	edit in split window
<b>Cursor Bookmarks</b>	
<code>:marks</code>	^ ^ ^o
<code>ma</code>	mark local 'a'
<code>'A</code>	goto global 'A'
<code>''</code>	prev location
<b>File / Directory</b>	
<code>:Explore</code> or <code>:e .</code>	
<code>:set browsedir=...</code>	
one of <code>buffer last</code>	