Vim Cheat Sheet for Programmers

Esc Normal

Revision 2.3
May 23, 2013

Vim 7.3+
`:version`

Copyleft © 2013
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>

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

Unused & Duplicate keys

```
\ Ctrl-K Ctrl-S (free)
Ctrl-L (redraw)
13 ` near dup of '
14 Ctrl-Q = Ctrl-V
15 Ctrl-J = Ctrl-M = ^N
```

16 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` `N` searches buffer.

Legend:

Macro	Register name (0-9a-zA-Z) required
Op	Motion req.; act between cursor & dst
Cmd	Command
Ins	Command and enter insert mode
Move	Moves cursor or defines range for op
Find	Search (↖ = reverse, ↘ = forward)
tag	ctags / diffs / folding
Code	Code formatting, whitespace, etc.
Extra	Extended functionality; req. extra chars
•	Char param req. `g z Z ^w ' " ` …`

Modes

`:help modes`

n	Normal	Esc	^[	^c	
i	Insert	a	i	r	s
v	Visual	v	V	^v	^q
o	Op Pending	c	d	y	< >
c	Command Line	:	/	?	!

word `Foo ( src , ƀ dst , ƀ len );`

Note: There is no whitespace in-between 'Foo(src,' but before/after 'dst,'

WORD `Foo ( src , ƀ dst , ƀ len );`

Startup

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

bug **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*

:help cmdline

		`:r file`	insert file
`:w`	save	`:gui`	switch to GUI
`:q`	quit	`:q!`	quit w/o save
`:e <file>`	edit file in new buffer		
`:source %`	**exec cmds in cur file**		
`:exec '…'`	**do command(s)**		

:help movement

soft	^ ⇤ **Start of Line**	1st non-whitespace	
hard	0 ⇤ **Start of Line**	column 0	
	$ ⇥ **End of Line**		
	| move col 0	# |	**move col #**
^b page ↑		^f page ↓	
^u ½ page ↑		^d ½ page ↓	
^e scroll line ↑		^y scroll line ↓	
1g start of file		0g end of file	
#g **goto line #**		G **end of file**	
[[**prev func**		]] **next func**	

:set matchpairs=(:),{:},[:],<:>,?:\:

% **goto matching { } < > [ ]**

:help range

`:s/Foo/Bar`	find Foo replace w/ Bar
`:s/Foo/Bar/g`	…all instances on line
`:%s/Foo/Bar`	**apply to whole file**
`.,.+#`	**cur line, cur line + # lines**
$ **last line**	'< start of select
% whole file 1,$	'> end of select
\v **literal mode;**	**:help magic**

Code

`=` `<` `>` `<<` `>>`

`:set backspace=indent,eol,start`	**allow backspace join lines**
`:set shiftwidth=#`	**indent width for ai**
`:set autoindent!`	**toggle auto-indent**
`:set lisp`	**lisp indent mode**
`:set tabstop=#`	**set tab stop every #th col**
`:set expandtab!`	**toggle hard/soft tabs**
`:set listchars=…`	*tab:>-:trail:-,nbsp:%,eol:$*
`:set list!`	**toggle whitespace**
`:set colorcolumn=80`	**visible right margin indicator**
`noremap + :s/^/\/\//<CR>`	**block comment**

Tags

`:ts`	list active tags
`^]`	**jump to tag under cursor**
`^t`	**restore cursor before tag jump**
`^p`	**complete word**
`:ta Foo`	manual jump to tag 'Foo'

Diff

`[c` **prev diff**	`:hi DiffAdd`	`guifg=#rrggbb`	
`]c` **next diff**	`:hi DiffChange`	`guibg=#rrggbb`	
`:diffupdate`	`:hi DiffText gui=none`		
resync	`:hi DiffDelete`		
`:diffget`	**copy selected text →**		
`:diffput`	**copy selected text ←**		

Folding

zR	**fold remove**
zo	fold open
zc	fold close
zi	**invert all**
zr	fold reduce
zm	fold more

Changes

`:changes`	
g;	older change
g,	newer change

Syntax

`:syntax enable`	
`:set filetype=`	
c cpp sh make perl python	

Note: chose only ONE type!

:help recording

q•	start recording
@•	playback
q	stop recording
@@	repeat

Convert <eol>

`:set fileformat=`

unix or *dos* or *mac*

then :w to convert

`:set number!`	**toggle line numbers**
`:set wrap!`	**toggle linewrap display**
`:set showmatch`	**highlite matching ()**
`noremap - :s/^\/\///<CR>`	**uncomment**

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

Insert mode

9	^p	prev auto-complete	^n	next
10	^d	**undent**	^t	**indent**
§11	^x• ^f	filename completion		
	^s	spelling	`:set spell!`	
	^k	dictionary	]s	**next bad**
	^t	thesaurus	`:help spell`	
12	^r•	paste register 0-9a-zA-Z or …		
	+	**clipboard** (or '*')	`:help c_CTRL-R`	
	"	last del/copy	%	filename

:buffer

`:buffers`	list files
`:new`	blank file/buffer
`:bn`	**next file**
`:bp`	prev file
`:bd`	**close file**
`:bd!`	force close
`:set lines=#`	
`:set columns=#`	
`:winpos # #`	GUI

:help windows

^w•	or	`:wincmd •`
w	switch to **next**	
c	**:close!**	
n	:new	
s	**:split** horz.	
v	:vsplit vertical	
o	:only maximize	
=	**all same size**	
h	move to win ←	
j	move to win ↓	
k	move to win ↑	
l	move to win →	
`:sp [<filename>]`	*edit in split window*	

:help marks

`:marks`	list ` ' ^o
ma	mark local 'a'
'A	goto global 'A'
''	**prev location**

File / Directory

`:pwd`	`:cd …`
`:Explore`	or `:e .`
`:set browsedir=`	*buffer* or *last*