Vim Cheat Sheet for Programmers

Esc Normal

Revision 2.0 Sept. 11, 2011

Vim 7.3+ :version

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>

Keyboard Layout

Key	Ctrl `	Ctrl 1	Ctrl @	Ctrl 3	Ctrl 4	Ctrl 5	Ctrl ^	Ctrl 7	Ctrl 8	Ctrl 9	Ctrl 0	Ctrl _	Ctrl =
top	~ 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

	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
	q• record macro	w word ↘	e end word ↘	r replace char	t• until char →	y copy	u undo	i ⊢ insert	o open ↓	p paste ↓	[• misc.	]• misc.

Tab | Caps

	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		goto col#
	a append ⌐	s subst char	d del	f• find char →	g• extra	h ←	j ↓	k ↑	l →	; "next" f/F/t/T	goto mark	\•

Ctrl ^ | Shift ⇧

	Z• quit	X ← del char	C change →	V select lines	B ↖ WORD	N "prev" find	M Middle screen	< undent	> indent	?• find ↖
	Z• extra	x ← del char	c change	v select chars	b ↖ word	n find "next"	m• set mark	, "prev" f/F/t/T	. repeat cmd	/• find ↘

16 The search direction is relative; **next** is the initial direction, **previous** is the opposite direction.   n , repeat same initial direction.   N , repeat opposite initial direction.   **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 arg 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 , dst , len);
Note: There is no whitespace in-between 'Foo(src,' but before/after 'dst,'
WORD | Foo (src dst , len);

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

Broken Keys Ctrl-L = 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	
:q quit	
:e <file>	edit file in new buffer
:source %	exec cmds in cur file
:exec '...'	do cmd
:gui	switch to GUI
:q!	quit w/o save

: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
[[begin this func {	
]] begin 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
'> end of select

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

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=	

Convert <eol>
c cpp sh make perl python
Note: choose only ONE type!
:set fileformat=
unix or dos or mac
then :w to convert

\ :map \ :Explore<CR> manually type <,C,R,>	
§0 "• before del/copy/paste to use register	
"+x cut to system clipboard reg. '+'	
"+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	

: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

Windows
: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	

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	

Cursor Bookmarks
:marks ` ' . ^
ma mark local 'a'	
'A goto global 'A'	
'' prev location	

File / Directory
:Explore or :e .	
:set browsedir=…	
one of buffer last	

Code
= < > < > < >	
: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
noremap - :s/^\/\//\///<CR>	uncomment
: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 numbers!	toggle line numbers
:set wrap!	toggle linewrap display
:set showmatch	hilite matching ()

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

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