

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>											
~ toggle case		! extern filter		@ play macro		# prev identifier		\$ →		% goto match		^ soft ↵		& repeat :s		*		(begin sentence) end sentence		_ cur line		+ ↓																							
• goto mark		1		2		3		4		5		6		7		8		9		0 hard ↵		- ↑		= auto-format																							
14 block select												^w window...		scroll line ↑		12 :redo		10 ctags return		scroll line ↓		half page		Ctrl+ ↵		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 ↵																							
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																																															

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

Macro

Op

Cmd

Ins

Move

Find

tag

Code

Extra

•

Modes

n

i

v

o

c

word

WORD

Startup

vim

vim

vim

GUI

GUI

GUI

diff

bug

Register name (0-9a-zA-Z) required

Motion req.; act between cursor & dst

Command

Command and enter insert mode

Moves cursor or defines range for op

Search (↖ = reverse, ↗ = forward)

ctags / diffs / folding

Code formatting, whitespace, etc.

Extended functionality; req. extra chars

Char arg req. g z Z ^w ' " < ...

:help modes

Normal Esc ^[^c

Insert a i r s

Visual v V ^v ^q

Op pending c d y < >

Command Line : / ? !

Foo (src , dst , len);

There is no whitespace in-between 'Foo(src)' but before/after 'dst.'

Foo (src , dst , len);

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

gvimdiff <file1> <file2> [<file3>]

Broken Keys Ctrl-I = Tab, Ctrl-[] = ESC

Vim is still unable to map certain keys for your own use...

Caps, Ctrl-I, Ctrl-Shift-I, Ctrl-I, Ctrl-I, etc.

See: src/ops.c -c "/valid_yank_reg" for ^ reg. names

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

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 cmd

:help movement

soft ^ ← Start of Line 1st non-whitespace

hard 0 ← Start of Line column 0

↵ End of Line

| move col 0 # | move col #

↵ page ↑ ↵ page ↓

↵ ½ page ↑ ↵ ½ page ↓

↵ scroll line ↑ ↵ scroll line ↓

lg start of file og 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

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

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=

:help recording

q start recording

q playback

q stop recording

@@ repeat

Convert <eol>

:set fileformat=

unix or dos or mac

then :w to convert

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

:set numbers! toggle line numbers

:set wrap! toggle linewidth display

:set showmatch highlight matching ()

noremap - :s/^\\//<CR> uncomment

: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 :split vertical

o :only maximize

o 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

Cursor Bookmarks

:marks

ma mark local 'a'

'A goto global 'A'

' prev location

File / Directory

:Explore OR :e .

:set browsefor=...

one of buffer last