

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>

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

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.

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

- 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);

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
```

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>]

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.
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

- :w save
- :q quit
- :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
- 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 ↓
- 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=
- c cpp sh make perl python
- Note: chose only ONE type!

Convert <eol>

- :set fileformat=
- unix or dos or mac
- then :w to convert

:help recording

- q* start recording
- q playback
- q stop recording
- @@ repeat

set tabstop=#

- :set expandtab!
- :set listchars=
- :set list!
- :set colorcolumn=80
- noremap + :s/^\\//<CR>

set tab stop every #th col

- toggle hard/soft tabs
- tab:-:trail-:,nbsp:%,eol:\$
- toggle whitespace
- visible right margin indicator
- block comment

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

toggle line numbers

- :set numbers!
- toggle linewrap display
- highlite matching ()
- toggle line numbers
- toggle linewrap display
- highlite matching ()
- toggle line numbers
- toggle linewrap display
- highlite matching ()

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 # #

Windows

- :help windows
- ^w* or :wincmd *
- w :switch to next
- c :close!
- n :new
- s :split horz.
- v :split 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

Cursor Bookmarks

- :marks
- ma mark local 'a'
- 'A goto global 'A'
- ' prev location

File / Directory

- :Explore OR :e .
- :set browsefor=
- one of buffer last