Vim: the why and how

Provo Linux User Group   Seth House <seth@eseth.com>

2013-07-16

1 Why Vim?

- A (very) little history
- A philosophy
- Getting started
- Getting help

2 Basics

3 An exercise left to the reader

A (very) little history

- First release — 1991

A (very) little history

- First release — 1991
- Latest release —

A (very) little history

- First release — 1991
- Latest release — last week

A (very) little history

Figure: Lear Siegler ADM-3A terminal

A (very) little history

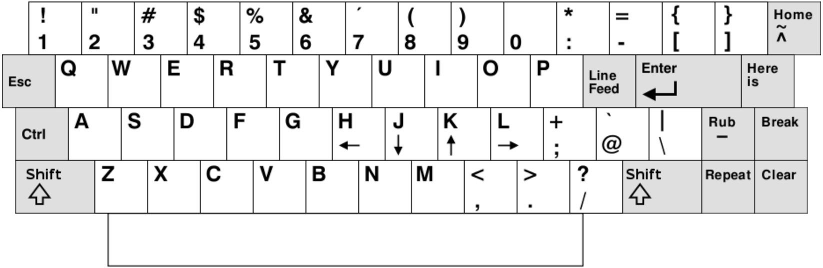


Figure: Lear Siegler ADM-3A keyboard layout

A philosophy

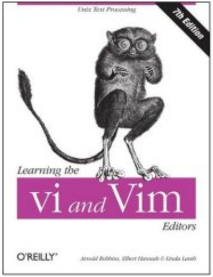
- Fingers on the home-row
- Terse, predictable commands
- Extreme speed
- Extreme flexibility
- UNIX as the IDE

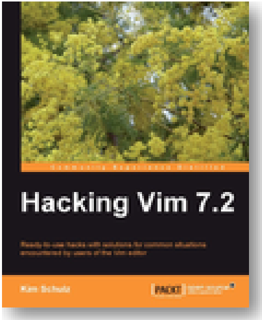
Getting started

Getting started

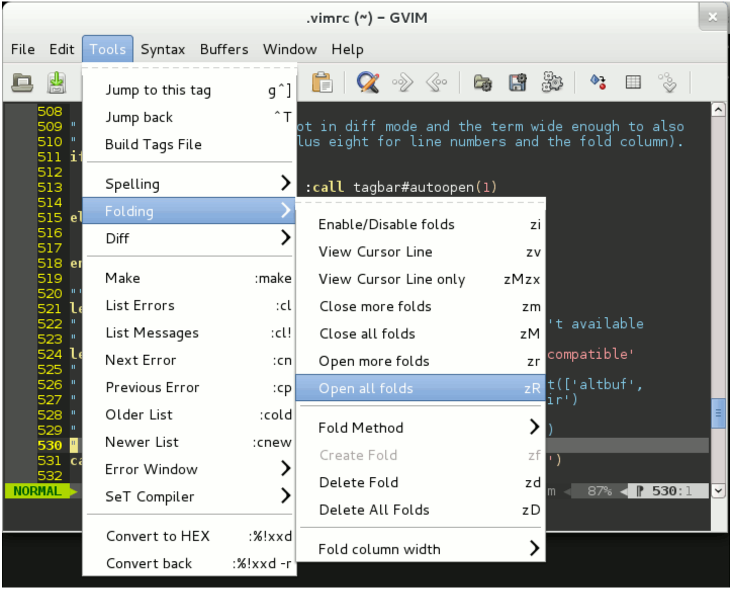
However you’re comfortable

Getting started





Getting started



```
508
509 "   ot in diff mode and the term wide enough to also
510 "   lus eight for line numbers and the fold column).
511 i
512
513     :call tagbar#autoopen(1)
514
515 el
516
517
518 en
519
520 "
521 le
522 "                                   't available
523 "
524 le                                  compatible'
525 "
526 "                                   t(['altbuf',
527 "                                   ir')
528 "
529 "                                   )
530 "
531 ca                                  ')
532
```

Getting started

1. Turn off vi-compatibility
   - Create a `~/.vimrc` or `~/.gvimrc`

Getting started

1. Turn off vi-compatibility
   - Create a `~/.vimrc` or `~/.gvimrc`
2. Turn on filetype detection (plugins and indentation)
   - `filetype plugin indent on`

Getting started

1. Turn off vi-compatibility
   - Create a `~/.vimrc` or `~/.gvimrc`
2. Turn on filetype detection (plugins and indentation)
   - `filetype plugin indent on`
3. Turn on the completion menu
   - `set wildmenu`

Getting help

- `:help`

Getting help

- `:help`
  - `:tab help`

Getting help

- `:help`
  - `:tab help`
- `:help <bs` <tab>

Getting help

- `:help`
  - `:tab help`
- `:help <bs` <tab>
- `:help i_` <tab>
- `:help modes` <tab>

1 Why Vim?

2 Basics

- Modes
- Normal mode: motions
- Normal mode: counts
- Normal mode: edits
- Normal mode: window movement
- Normal mode: cursor movement
- Insert mode: edits
- Visual mode: selections
- Command mode
- Command mode: edits
- Command mode: search (`/`, `?`)
- Command mode: filter (`!`)
- Stupid Vim tricks
- UNIX as an IDE

Modes

- Normal
- Insert
- Visual
- Command mode

Normal mode: motions

Cursor motions

- Right/up/down/left: `h`, `j`, `k`, `l`

Normal mode: motions

Cursor motions

- Right/up/down/left: `h`, `j`, `k`, `l`
- Beginning/end of line: `0`, `$`

Normal mode: motions

Cursor motions

- Right/up/down/left: `h`, `j`, `k`, `l`
- Beginning/end of line: `0`, `$`

Word motions

- Next/prev word beginning: `w`, `b`
- Next/prev word ending: `e`, `ge`

Normal mode: motions

Cursor motions

- Right/up/down/left: `h`, `j`, `k`, `l`
- Beginning/end of line: `0`, `$`

Word motions

- Next/prev word beginning: `w`, `b`
- Next/prev word ending: `e`, `ge`

Character jump

- Find char: `f{char}`, `F{char}`
- Find char: `t{char}`, `T{char}`

Normal mode: motions

Cursor motions

- Right/up/down/left: `h`, `j`, `k`, `l`
- Beginning/end of line: `0`, `$`

Word motions

- Next/prev word beginning: `w`, `b`
- Next/prev word ending: `e`, `ge`

Character jump

- Find char: `f{char}`, `F{char}`
- Find char: `t{char}`, `T{char}`
- Repeat find: `;`, `,`

Other movement

- Sentences: `)`, `(`
- Paragraphs: `}`, `{`

Normal mode: counts

Motions can be preceeded by counts

- `5j`
- `3w`
- `2}`

Normal mode: counts

Motions can be preceeded by counts

- `5j`
- `3w`
- `2}`

See line numbers as relative distance:

```
:set relativenumber
```

Normal mode: edits

Verbs and nouns (and counts)

- Change `c`, `C`, `cc`
- Delete `d`, `D`, `dd`
- Copy (yank) `y`, *Y*, `yy`

Normal mode: edits

Verbs and nouns (and counts)

- Change `c`, `C`, `cc`
- Delete `d`, `D`, `dd`
- Copy (yank) `y`, *Y*, `yy`
  - `map Y y$`

Modifiers (adjectives)

- A word/inner word: `aw`, `iw`
- A sentence/inner sentence: `as`, `is`
- A paragraph/inner paragraph: `ap`, `ip`
- A tag/inner tag: `at`, `it`
- Other text objects: `"`, `′`, `>`, `}`, `)`, `]`, `` ` ``

Normal mode: edits

Verbs and nouns (and counts)

- Change c, C, cc
- Delete d, D, dd
- Copy (yank) y, *Y*, yy
    - map Y y$

Modifiers (adjectives)

- A word/inner word: aw, iw
- A sentence/inner sentence: as, is
- A paragraph/inner paragraph: ap, ip
- A tag/inner tag: at, it
- Other text objects: ", ′, >, },),], `

Repeat

- .

Joins

- J

Normal mode: window movement

Scrolling window

- Page up/down: `ctrl-f`, `ctrl-b`
- Half-page up/down: `ctrl-d`, `ctrl-u`
- Single-line up/down: `ctrl-y`, `ctrl-e`

Normal mode: window movement

Scrolling window

- Page up/down: `ctrl-f`, `ctrl-b`
- Half-page up/down: `ctrl-d`, `ctrl-u`
- Single-line up/down: `ctrl-y`, `ctrl-e`

Window positioning relative to cursor

- Top/middle/bottom: `zt`, `zz`, `zb`

Normal mode: cursor movement

Cursor placement within window

- Head/middle/last: `H`, `M`, `L`

Normal mode: cursor movement

Cursor placement within window

- Head/middle/last: `H`, `M`, `L`

Moving the cursor line-wise

- Top of file, bottom of file: `gg`, `G`

Normal mode: cursor movement

Cursor placement within window

- Head/middle/last: `H`, `M`, `L`

Moving the cursor line-wise

- Top of file, bottom of file: `gg`, `G`
- Specific line: `<number>gg`, `<number>G`

Insert mode: edits

- Insert: `i`, `I`
- Append: `a`, `A`
- Delete word: `ctrl-w`
- Delete line: `ctrl-u`
- Dedent/indent: `ctrl-d`, `ctrl-t`

Visual mode: selections

- Character-wise selection: `v`
- Line-wise selection: `V`
- Block-wise selection: `ctrl-v`

Use repeat operator with a visual selection:

```
vnoremap . :normal .<cr>
```

Command mode

- `:wq`, `:x`, `:wqa`, `:xa`
- `:qa!`
- `:r <filename>`

Command mode: edits

Search/replace

- `:%s/foo/bar/g`
- `:%s/foo/bar/gc`

Command mode: edits

Search/replace

- `:%s/foo/bar/g`
- `:%s/foo/bar/gc`

Ranges

- Line 23: `23`

Command mode: edits

Search/replace

- `:%s/foo/bar/g`
- `:%s/foo/bar/gc`

Ranges

- Line 23: `23`
- Entire file: `%` (same as `1,$`)

Command mode: edits

Search/replace

- `:%s/foo/bar/g`
- `:%s/foo/bar/gc`

Ranges

- Line 23: `23`
- Entire file: `%` (same as `1,$`)
- Line 4 to “some text”: `4,/some text/`

Command mode: edits

Search/replace

- `:%s/foo/bar/g`
- `:%s/foo/bar/gc`

Ranges

- Line 23: `23`
- Entire file: `%` (same as `1,$`)
- Line 4 to “some text”: `4,/some text/`
- Current line to last line: `.,$`

Command mode: edits

Search/replace

- `:%s/foo/bar/g`
- `:%s/foo/bar/gc`

Ranges

- Line 23: `23`
- Entire file: `%` (same as `1,$`)
- Line 4 to “some text”: `4,/some text/`
- Current line to last line: `.,$`
- Line below line containing “that”: `/that/+1`

Command mode: search (/, ?)

- /{pattern}

Search settings:

```
set hlsearch
set ignorecase
set smartcase
set incsearch
```

Command mode: filter (!)

- `:%!json_reformat`

Command mode: filter (!)

- `:%!json_reformat`
- `:.!ls /path/to/file`
- `:'<,'>!sort -rnu`

Stupid Vim tricks

Transient shell scripts

1. `ls *.JPEG | vi -`
2. `:%s/.*/mv -i & &/g`
3. `:%s/JPEG$/jpg/g`
4. `:w !sh`

UNIX as an IDE

- Filter
- Scripts
- `makeprg` (Syntastic)
- `ctags` (Tagbar)

Buffers, windows, and tabs

Buffer A file

Buffers, windows, and tabs

Buffer A file

Window A view of a file

Buffers, windows, and tabs

Buffer	A file
Window	A view of a file
Tabs	An arrangement of windows

Registers

- `:reg`

Registers

- `:reg`
- Numbered registers
  - `"0` last yank
  - `"1` to `"9` last 9 deletes

Registers

- `:reg`
- Numbered registers
  - `"0` last yank
  - `"1` to `"9` last 9 deletes
- Small delete (less than one line)

Registers

- `:reg`
- Numbered registers
  - `"0` last yank
  - `"1` to `"9` last 9 deletes
- Small delete (less than one line)
- 26 named registers

Registers

- `:reg`
- Numbered registers
  - `"0` last yank
  - `"1` to `"9` last 9 deletes
- Small delete (less than one line)
- 26 named registers
- Last search register

Registers

- `:reg`
- Numbered registers
  - `"0` last yank
  - `"1` to `"9` last 9 deletes
- Small delete (less than one line)
- 26 named registers
- Last search register
- Selection registers `*`, `+`
  - `:version`

1 Why Vim?

2 Basics

3 An exercise left to the reader

- What haven’t we talked about?
- File type settings
- Plugins

What haven't we talked about?

- Macros
- Completion
- Diffing / merging
- Encrypt files
- Scripting languages
- Folding
- Editing compressed files
- Editing network files
- Sessions
- Spell checking
- Unicode
- Syntax highlighting
- Cursor position histories
- Undo/redo tree

File type settings

- `:help ft-` <tab>

File type settings

- `:help ft-` <tab>
- `~/.vim/ftplugin/<filetype>.vim`

Plugins

General

- Pathogen / Vundle

Plugins

General

- Pathogen / Vundle
- Gundo

Plugins

General

- Pathogen / Vundle
- Gundo
- NERDtree

Plugins

General

- Pathogen / Vundle
- Gundo
- NERDtree
- Powerline

Plugins

General

- Pathogen / Vundle
- Gundo
- NERDtree
- Powerline

Scripting / programming

- Syntastic

Plugins

General

- Pathogen / Vundle
- Gundo
- NERDtree
- Powerline

Scripting / programming

- Syntastic
- Tagbar

Plugins

General

- Pathogen / Vundle
- Gundo
- NERDtree
- Powerline

Scripting / programming

- Syntastic
- Tagbar

Version control

- Fugitive

Plugins

General

- Pathogen / Vundle
- Gundo
- NERDtree
- Powerline

Scripting / programming

- Syntastic
- Tagbar

Version control

- Fugitive
- git-gutter