

# Overview zu Vim

Matthias Beyer

Furtwangen University

*matthias.beyer@hs-furtwangen.de*

6. Juni 2013

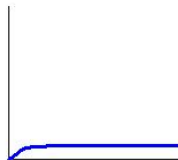
- 1 Vi vs. Vim
- 2 Steuerung von Vim
  - Modi
  - Navigation
  - Kopieren/Einfügen, Suchen/Ersetzen
  - Advanced stuff
- 3 Konfigurationsdatei (vimrc) schreiben
  - Basics
  - Weitergehendes
- 4 Sonstiges

- Navigation
- Einfache Texteditierung

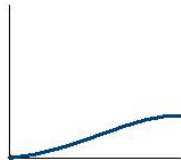
Classical learning curves for some common editors



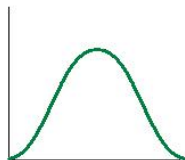
**Notepad**



**Pico**



**Visual Studio**



**vi**



**emacs**



- Besseres Userinterface
- Scripting
- Macros
- Tabbing
- Screen splitting
- Plugins

- Command Mode
- Insert Mode
- Visual Mode

- Kommandos eingeben
- Navigieren
- Text bearbeiten, aber nicht einfügen

- Betreten mit einer Vielzahl an Möglichkeiten:
  - a bzw A
  - i bzw I
  - o bzw O
  - C
  - r bzw R
- Text schreiben

Text markieren um damit zu arbeiten, zum Beispiel:

- Löschen
- Verändern
- Verschieben, wie z.B. einrücken
- Sortieren
- Durchsuchen
- Im Terminal ausführen (advanced)



## Im Command-Mode

- h j k l
- gg
- {
- (
- 0
- ^
- w
- b
- \$
- )
- }
- G

# Navigation - Komplet

vim - movement commands

Quelle: <http://beyermatthias.de/blog/2012/12/18/vim-shortcuts/>

- yy - Zeile kopieren
- 5yy - 5 Zeilen kopieren
- yt<character> - bis <character> kopieren
- y<number>G - bis Zeile <number> kopieren

Löschen == Kopieren und entfernen!

- p - kopiertes nach Cursor einfügen
- P - vor Cursor einfügen

- “<register><number>yy - Kopiere <number> Zeilen in <register>
- “<register><number>y) - Kopiere <number> Sätze in <register>

## Command-Mode:

- /<regex>
- ?<regex>
- n
- N
  
- :s/foo/bar
- :%s/foo/bar/g

## Vom Visual-Mode aus:

- : '<,'>s/foo/bar/

Macros aufnehmen und beenden mit "q<register>", bzw. "q", abspielen mit "@<macro register>"  
(Beispiel)

Variablen setzen mit “set <variable>“, zurücksetzen mit “set no<variable>“.

Kommentare mit “

# Umgebungsvariablen

- nocompatible
- restorescreen
- number
- showcmd
- numberwidth=3
- runtimepath
- history
- textwidth
- backupdir
- novisualbell
- noerrorbells
- title
- titlestring=%F %m



- cursorline
- smarttab
- autoindent
- shiftwidth=4
- softtabstop=4
- tabstop=4
- enc
- fileformats=unix,dos,mac

# Nützliche Zusatzvariablen

- hidden
- backspace=indent,eol,start
- scrolloff=5
- laststatus=2
- showmode
- cmdheight=2
- incsearch
- hlsearch
- smartcase
- ignorecase
- gdefault

## Listing 1: Statusline variable setzen

```
1 " Format string for statusline
  set statusline=[%l,%v\ %P%M]
3
  " some info about git
5 set statusline+=%{fugitive#statusline()}
```

## Listing 2: Statusline variable setzen

```
1 " and about the current file
  set statusline+=\ %f\ %r%h%w\|\| \ %L\ lines%{
    StatuslineFilesize()}
3
  set statusline+=\ \|\| Syntastic \:\  
5
  set statusline+=%{SyntasticStatuslineFlag()}
```

38 verschiedene Formatter.

## Listing 3: Vim für Quellcode optimieren

```
syntax on
2 filetype on
  filetype plugin on
4 filetype indent on

6 autocmd FileType ruby setlocal expandtab shiftwidth
  =2 tabstop=2 softtabstop=2

8 autocmd FileType ruby inoremap , ,<space>
  autocmd FileType ruby inoremap ,<space> ,<space>
```

Use vundle, or die!

# Plugins - why vundle?

- Ein Kommando zum Updaten aller Plugins
- Ein Kommando zum Installieren eines Plugins
- Ein Kommando zum Abschalten eines Plugins
- Ein Kommando zum Deinstallieren eines Plugins

# Plugins - Welche sollte man nutzen?

- 'Valloric/YouCompleteMe'
- 'majutsushi/tagbar'
- 'mbbill/undotree'
- 'mikewest/vimroom'
- 'mileszs/ack.vim'
- 'scrooloose/syntastic'
- 'tpope/vim-fugitive'



Die Steuerung von vim ist eine Sprache!

- `dt}` = delete till "`}`"
- `"oy5)` = yank 5 sentences to register "`o`"
- `l5w` = right 5 words
- `"oP` = paste from register "`o`" before cursor

Wenn man die Sprache kann, kann man vim *benutzen!*

# My vim

```
den web irc chat ssh home [2] ~/vimrc [5] [41,26,131,48 | -99,055 | (85 H) | H | 15, 05, 2013 10:26:21]
* Press ? for help.
476 * Global mappings
477 * -----
478
479 map <UP> <NOP>
480 map <DOWN> <NOP>
481 map <LEFT> <NOP>
482 map <RIGHT> <NOP>
483
484 * How between screen lines instead of real lines
485 noremap k gk
486 noremap j gj
487
488 let mapleader = ""
489
490 map <Leader>so :SyntaxCheckCR<CR>
491 map <Leader>nt :NERDTreeToggle!<CR>
492 map <Leader>tb :iTagbarToggle!<CR>
493 map <Leader>ut :undotreeToggle!<CR>
494 map <Leader>v :VimrcOpenToggle!<CR>
495
496 * -----
497 * own Scripts
498 * -----
499 :augroup HighlightTOD
500 * autocmd!
501 * autocmd VimEnter, VimEnter * silent! call matchadd('TOD', '\[TOD\]', -1, 24 )
502 :augroup END
503 *
504 autocmd FileType gitcommit call sgitcommit(%, f, B, i, r, g)
505
490 494 495 496 497 498 499 500 501 502 503
~/vimrc 434 lines | 12 KB | Syntax:V
14 " Display line numbers in front of each line and use space for 3 charact
15 set number
16 set numberwidth=3
17
18 * set runtimepath
19 set runtimepath+=~/config/vim/colors
20 set runtimepath+=~/config/vim/bundle/vundle/
21
22 * vundle setup
23 filetype off
24 call vundle#rc()
25 Bundle 'garik/vundle'
26 filetype plugin indent on
27
28 " Display commands in the bottom right corner as they are typed
29 set showcmd
30
31
32 * Size of the history
33 set history=250
34
35 * textwidth to 80
36 set textwidth=80
37
38 * customize the wildmenu
39 set wildmenu
40 set wildmode=list:full
41
42 * How Backup File to ~/config/vim/sessions
~/vimrc 434 lines | 12 KB | Syntax:V
Press ? for help
.. (up a dir)
~/archive/
blog/ -> ~/archive/blog/
config/
dev/ -> ~/archive/code/
dlisp/ -> ~/archive/download
doc/ -> ~/archive/docs/studius/
gitlab/
sir/
thesso_orga/
unfug/
via-talk/
laages/
sro/
@group
presentation_1.sux
presentation_1.log
presentation_1.nav
presentation_1.out
presentation_1.pdf
presentation_1.smx
presentation_1.tex
presentation_1.toc
presentation_1.wb
vim_Config_Files/
Hilfswiese_informatik__RIB_p
vorlagen/
tmp/ -> /tmp/
virt/ -> ~/archive/virt/
wpsdrabble.log
todo.txt
~/vimrc 434 lines | 12 KB | Syntax:V
```

Quelle: Selbst erstellt

# How to learn vim

- Practice!
- Practice!
- Practice!
- Don't map h-j-k-l to Left-Down-Up-Right!



Fragen?