

Mein VIM-Setup (vimrc und Plugins)

Tastenkürzel	Aktion
STRG+N	nächste Datei öffnen (wenn Vim mit mehreren Dateien gestartet wurde)
STRG+P	vorherige Datei öffnen (wenn Vim mit mehreren Dateien gestartet wurde)
F3	Tabs als >- anzeigen umschalten

Folgende Pakete müssen installiert sein:

```
xxxxxxxxxx
```

1

```
apt update
```

2

```
apt install vim-common vim-nox vim-runtime vim-tiny vim-scripts vim-addon-manager vim-pathogen
```

Für Fancy-Aussehen, Syntax-Checks und so weiter habe ich diese Vim-Addons installiert:

```
xxxxxxxxxx
```

1

```
mkdir -p /opt/vim/bundle
```

2

```
# a fancy file tree (envoked by CTRL+k):
```

3

```
git clone https://github.com/scrooloose/nerdtree /opt/vim/bundle/nertdtree
```

4

```
?
```

5

```
# this pimps the status line
```

6

```
git clone https://github.com/itchyny/lightline.vim /opt/vim/bundle/lightline
```

7

```
?
```

8

```
# this is for smart copy and paste (without automatic indent and messing around with :set pa
```

9

```
git clone https://github.com/ConradIrwin/vim-bracketed-paste /opt/vim/bundle/bracketed-paste
```

10

```
?
```

11

```
# make some often used shell commands available
```

12

```
git clone https://github.com/tpope/vim-eunuch.git /opt/vim/bundle/eunuch
```

13

```
?
```

14

```
# extended syntax checks
```

15

```
git clone https://github.com/vim-syntastic/syntastic.git /opt/vim/bundle/syntastic
```

16

```
?
```

17

```
# optimzize colors
```

```
git clone https://github.com/altercation/vim-colors-solarized.git /opt/vim/bundle/colors-sol
```

Und hier noch die globale vimrc:

xxxxxxxxxx

1

"

2

" File managed by salt.ovtec.it (packages/vim).

3

" Your changes will be overwritten.

4

"

5

6

runtime! debian.vim

7

8

if exists('g:loaded_sensible') || &compatible

9

finish

10

else

11

let g:loaded_sensible = 1

12

endif

13

14

if has('autocmd')

15

filetype plugin indent on

16

endif

17

if has('syntax') && !exists('g:syntax_on')

18

`syntax enable`

19

`endif`

20

21

`set autoindent`

22

`set autoread`

23

`set background=dark`

24

`set backspace=indent,eol,start`

25

`set cmdheight=2`

26

`set complete=i`

27

`set expandtab " enable this to use whitespaces instead of tabs`

28

`set formatoptions=qtj`

29

`set hidden`

30

`set incsearch`

31

`set iskeyword+=_,$,@,%,#,-`

32

`set laststatus=2`

33

`set lazyredraw`

34

`set modeline`

35

`set modelines=2`

36

```
set nobackup
```

37

```
set nohlsearch
```

38

```
set nrformats=octal,hex,alpha
```

39

```
set ruler
```

40

```
set shiftwidth=2
```

41

```
set showcmd
```

42

```
set showmatch
```

43

```
set showmode
```

44

```
set smarttab
```

45

```
set softtabstop=2
```

46

```
set tabstop=2
```

47

```
set ttimeout
```

48

```
set ttimeoutlen=100
```

49

```
set visualbell
```

50

```
set wildmenu
```

51

52

```
" Use <C-L> to clear the highlighting of :set hlsearch.
```

53

```
if maparg('<C-L>', 'n') ==# ''
```

54

```
nnoremap <silent> <C-L> :nohlsearch<C-R>=has('diff')?'<Bar>diffupdate':''<CR><CR><C-L>
```

55

```
endif
```

56

57

```
if !&scrolloff
```

58

```
set scrolloff=1
```

59

```
endif
```

60

```
if !&sidescrolloff
```

61

```
set sidescrolloff=5
```

62

```
endif
```

63

```
set display+=lastline
```

64

65

```
if &encoding ==# 'latin1' && has('gui_running')
```

66

```
set encoding=utf-8
```

67

```
endif
```

68

69

```
if &listchars ==# 'eol:$'
```

70

```
set listchars=tab:>-,trail:-,extends:>,precedes:<,nbsp:+
```

71

```
endif
```

72

73


```
if has('path_extra')
```

74

```
setglobal tags-=./tags tags-=./tags; tags^=./tags;
```

75

```
endif
```

76

77

```
if &shell =~# 'fish$'
```

78

```
set shell=/bin/bash
```

79

```
endif
```

80

81

```
if &history < 1000
```

82

```
set history=1000
```

83

```
endif
```

84

```
if &tabpagemax < 50
```

85

```
set tabpagemax=50
```

86

```
endif
```

87

```
if !empty(&viminfo)
```

88

```
set viminfo^=!
```

89

```
endif
```

90

```
set sessionoptions-=options
```

91

92

```
" Allow color schemes to do bright colors without forcing bold.
```

93

```
if &t_Co == 8 && $TERM !~# '^linux\|^Eterm'
```

94

```
set t_Co=16
```

95

```
endif
```

96

97

```
" I don't remember what that does
```

98

```
inoremap <C-U> <C-G>u<C-U>
```

99

10

0

```
" next and previous buffer / file
```

10

1

```
nnoremap <C-N> :next<Enter>
```

10

2

```
nnoremap <C-P> :prev<Enter>
```

10

3

10

4

```
" toggle displaying listchars
```

10

5

```
nnoremap <F3> :set list!<Enter>
```

10

6

10

7

```
au BufReadPost * if line("'\"") > 1 && line("'\"") <= line("$") | exe "normal! g'\"" | endif
```

10

8

```
au BufRead,BufNewFile *.sls setfiletype sls
```

10

9

```
au FileType yaml,yml,sls setlocal ts=2 sts=2 sw=2 ai expandtab fo-=c fo-=r fo-=o
```

11

0

```
"au FileType * setlocal ts=2 sts=2 sw=2 ai expandtab fo-=c fo-=r fo-=o
```

11

1

11

2

```
" Load plugins
```

11

3

```
execute pathogen#infect('bundle/{}', '/opt/vim/bundle/{}')
```

11

4

11

5

```
" colorscheme
```

11

6

```
let g:solarized_termcolors=256
```

11

7

```
colorscheme solarized
```

11

8

11

9

```
" syntastic syntax checker
```

12

0

```
set statusline+=%#warningmsg#
```

12

1

```
set statusline+=%{SyntasticStatuslineFlag()}
```

12

2

```
set statusline+=%*
```

12

3

```
let g:syntastic_always_populate_loc_list = 1
```

12

4

```
let g:syntastic_auto_loc_list = 1
```

12

5

```
let g:syntastic_check_on_open = 1
```

12

6

```
let g:syntastic_check_on_wq = 0
```

12

7

12

8

```
" vim:set ft=vim et sw=2:
```

Revision #1

Created 29 April 2021 23:09:38 by magenbrot

Updated 29 April 2021 23:10:01 by magenbrot