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:

```
apt update
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:

```
mkdir -p /opt/vim/bundle
# a fancy file tree (envoked by CTRL+k):
git clone https://github.com/scrooloose/nerdtree /opt/vim/bundle/nertdtree


# this pimps the status line
git clone https://github.com/itchyny/lightline.vim /opt/vim/bundle/lightline


# this is for smart copy and paste (without automatic indent and messing around with :set
paste)
git clone https://github.com/ConradIrwin/vim-bracketed-paste /opt/vim/bundle/bracketed-paste


# make some often used shell commands available
git clone https://github.com/tpope/vim-eunuch.git /opt/vim/bundle/eunuch


# extended syntax checks
git clone https://github.com/vim-syntastic/syntastic.git /opt/vim/bundle/syntastic


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

Und hier noch die globale vimrc:

```
" File managed by salt.ovtec.it (packages/vim).
" Your changes will be overwritten.
"
```

```vim
runtime! debian.vim

if exists('g:loaded_sensible') || &compatible
  finish
else
  let g:loaded_sensible = 1
endif


if has('autocmd')
  filetype plugin indent on
endif


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


set autoindent
set autoread
set background=dark
set backspace=indent,eol,start
set cmdheight=2
set complete=i
set expandtab " enable this to use whitespaces instead of tabs
set formatoptions=qtj
set hidden
set incsearch
set iskeyword+=_,$,@,%,#,-
set laststatus=2
set lazyredraw
set modeline
set modelines=2
set nobackup
set nohlsearch
set nrformats=octal,hex,alpha
set ruler
set shiftwidth=2
set showcmd
set showmatch
set showmode
set smarttab
```

```vim
set softtabstop=2
set tabstop=2
set ttimeout
set ttimeoutlen=100
set visualbell
set wildmenu

" Use <C-L> to clear the highlighting of :set hlsearch.
if maparg('<C-L>', 'n') ==# ''
  nnoremap <silent> <C-L> :nohlsearch<C-R>=has('diff')?'<Bar>diffupdate':''<CR><CR><C-L>
endif

if !&scrolloff
  set scrolloff=1
endif

if !&sidescrolloff
  set sidescrolloff=5
endif

set display+=lastline

if &encoding ==# 'latin1' && has('gui_running')
  set encoding=utf-8
endif

if &listchars ==# 'eol:$'
  set listchars=tab:>-,trail:-,extends:>,precedes:<,nbsp:+
endif

if has('path_extra')
  setglobal tags-=./tags tags-=./tags; tags^=./tags;
endif

if &shell =~# 'fish$'
  set shell=/bin/bash
endif

if &history < 1000
  set history=1000
endif
```

```vim
if &tabpagemax < 50
  set tabpagemax=50
endif


if !empty(&viminfo)
  set viminfo^=!
endif


set sessionoptions-=options


" Allow color schemes to do bright colors without forcing bold.
if &t_Co == 8 && $TERM !~# '^linux\|^Eterm'
  set t_Co=16
endif


" I don't remember what that does
inoremap <C-U> <C-G>u<C-U>


" next and previous buffer / file
nnoremap <C-N> :next<Enter>
nnoremap <C-P> :prev<Enter>


" toggle displaying listchars
nnoremap <F3> :set list!<Enter>
au BufReadPost * if line("'\"") > 1 && line("'\"") <= line("$") | exe "normal! g'\"" | endif
au BufRead,BufNewFile *.sls setfiletype sls
au FileType yaml,yml,sls setlocal ts=2 sts=2 sw=2 ai expandtab fo-=c fo-=r fo-=o
"au FileType * setlocal ts=2 sts=2 sw=2 ai expandtab fo-=c fo-=r fo-=o


" Load plugins
execute pathogen#infect('bundle/{}', '/opt/vim/bundle/{}')


" colorscheme
let g:solarized_termcolors=256
colorscheme solarized


" syntastic syntax checker
set statusline+=%#warningmsg#
set statusline+=%{SyntasticStatuslineFlag()}
set statusline+=%*
```

```
let g:syntastic_always_populate_loc_list = 1

let g:syntastic_auto_loc_list = 1

let g:syntastic_check_on_open = 1

let g:syntastic_check_on_wq = 0


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