

Example and Tutorial for Vim Editor (Linux)

Go back to [fan's Tex4Econ and Miscellaneous Repository](#).

1 Vim Installations

Vim ships with Debian. Also install neo-vim:

```
vim
sudo apt-get install neovim
```

Just installing onedark, neovim seems to work better than vim. Vim has delete lag, and extra white space rows at the bottom of the screen that could take up very large screen segment. Neovim with onedark does not have this issue, could be due to some basic setting differences.

1.1 Install Vim-Plug Plug-ins manager for Vim

Follow the [instructions here](#) for installations. Several steps:

1. generate ~/.vim and ~/.vimrc folders
2. curl plug.vim to ~/.vim/autoload
3. update ~/.vimrc file with instructions regarding which files to install.
4. open vim, type: `:PlugInstall` or `:PlugUpdate`, should start seeing updating packages.
5. type: `:scriptnames` to see loaded plugins
6. just comment out or delete plug lines for packages no longer needed.
7. how to install a particular package? See [vimawesome](#)

```
# curl plug.vim
curl -fLo ~/.vim/autoload/plug.vim --create-dirs \
    https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim

# create vimrc file
cd $HOME
vim .vimrc

# paste into file the text below.
```

Example .vimrc file with only one package one dark atom visual for vim:

```
if empty(glob('~/.vim/autoload/plug.vim'))
    silent !curl -fLo ~/.vim/autoload/plug.vim --create-dirs
        \ https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim
    autocmd VimEnter * PlugInstall --sync | source $MYVIMRC
endif

" Plugins will be downloaded under the specified directory.
call plug#begin('~/.vim/plugged')

" Declare the list of plugins.
Plug 'joshdick/onedark.vim'
```

```

" List ends here. Plugins become visible to Vim after this call.
call plug#end()

#####

" Settings
#####

syntax on
colorscheme onedark

```

1.2 Install Vim-Plug Plug-ins manager for Neovim

Follow the [instructions here](#) for installations. Very similar steps as above for Vim:

1. curl *plug.vim* to *.local/share/nvim*. Directory different than vim
2. create nvim folder under *.config*, vimrc contents should be in this file: *~/.config/nvim/init.vim*
3. enter nvim and *:PlugInstall*

```

# Neovim (~/.local/share/nvim/site/autoload)
curl -fLo ~/.local/share/nvim/site/autoload/plug.vim --create-dirs \
  https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim

# create vimrc file
mkdir ~/.config/nvim
nvim ~/.config/nvim/init.vim

# paste into file the text below.

```

Vimrc file with atom styling:

```

" Plugins will be downloaded under the specified directory.
call plug#begin('~/.local/share/nvim/site/plugged')

" Declare the list of plugins.
Plug 'vim-airline/vim-airline'

" Visual Settings
Plug 'junegunn/goyo.vim'
Plug 'junegunn/limelight.vim'

" Color theme plugins
Plug 'joshdick/onedark.vim'

" List ends here. Plugins become visible to Vim after this call.
call plug#end()

#####

" Color Settings
#####

syntax on

colorscheme onedark

#####

" Visual Settings
#####

```

```

" For Goyo
let g:goyo_width=85

" For LimeLight
" Color name (:help cterm-colors) or ANSI code
let g:limelight_conceal_ctermfg = 'gray'
let g:limelight_conceal_ctermfg = 240
" Color name (:help gui-colors) or RGB color
let g:limelight_conceal_guifg = 'DarkGray'
let g:limelight_conceal_guifg = '#777777'
" highlight line
let g:limelight_bop = '^.*$'
" let g:limelight_eop = '\n'
let g:limelight_paragraph_span = 0

#####
" UI Settings
#####
" Keep cursor in the middle of the page, useful for editing text
set so=999
" Turn limelight on by default
" autocmd VimEnter * Limelight
" Turn Goyo on by default
autocmd VimEnter * Goyo
" autocmd VimEnter * AirlineToggle
" In Goyo, if airline is turned on, do nto show scratch area
" autocmd! User GoyoEnter nested set eventignore=FocusGained
" autocmd! User GoyoLeave nested set eventignore=

```

1.3 Install Neovim in Windows

Install and access neovim all from windows powershell. Type in *powershell* in regular command prompt for example.

1.3.1 Install Neovim and Vim-Plug in Windows

```

# install scoop and use scoop
scoop install neovim

# follow vim-plug instructions
md ~\AppData\Local\nvim\autoload
$uri = 'https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim'
(New-Object Net.WebClient).DownloadFile(
    $uri,
    $ExecutionContext.SessionState.Path.GetUnresolvedProviderPathFromPSPath(
        "~\AppData\Local\nvim\autoload\plug.vim"
    )
)

```

1.3.2 Set up init.vim File in Windows

Inside powershell, create init.vim file:

```

cd ~\AppData\Local\nvim\
nvim init.vim

```

```

paste these in :set paste:

" Plugins will be downloaded under the specified directory.
call plug#begin('~\AppData\Local\nvim\site\plugged')

" Declare the list of plugins.
Plug 'vim-airline/vim-airline'

" Markdown
Plug 'godlygeek/tabular'
Plug 'plasticboy/vim-markdown'

" Visual Settings
Plug 'junegunn/goyo.vim'
Plug 'junegunn/limelight.vim'
Plug 'reedes/vim-pencil'

" Color theme plugins
Plug 'joshdick/onedark.vim'
Plug 'morhetz/gruvbox'
Plug 'junegunn/seoul256.vim'
Plug 'jaredgorski/spacecamp'
Plug 'lifepillar/vim-solarized8'
Plug 'reedes/vim-colors-pencil'

" List ends here. Plugins become visible to Vim after this call.
call plug#end()

#####
" Color Settings
#####
syntax on

" onedark:
colorscheme onedark

" seoul256:
" seoul256 (dark), 236 is darkest
" let g:seoul256_background = 236
" colorscheme seoul256

" spacecamp:
" colorscheme spacecamp

" vim-solarized8:
" set background=dark
" colorscheme solarized8

" vim-colors-pencil:
" let g:pencil_higher_contrast_ui = 0
" colorscheme pencil
" set background=dark

#####

```

```

" Visual Settings
#####
" For Goyo
let g:goyo_width=85

" For LimeLight
" Color name (:help cterm-colors) or ANSI code
let g:limelight_conceal_ctermfg = 'gray'
let g:limelight_conceal_ctermfg = 240
" Color name (:help gui-colors) or RGB color
let g:limelight_conceal_guifg = 'DarkGray'
let g:limelight_conceal_guifg = '#777777'
" if Limelight should highlight line rather than paragraph
let g:limelight_bop = '^.*$'
" let g:limelight_eop = '\n'
let g:limelight_paragraph_span = 0

#####
" UI Settings
#####
" Keep cursor in the middle of the page, useful for editing text
set so=999
" Turn limelight on by default
" autocmd VimEnter * Limelight
" Turn Goyo on by default
autocmd VimEnter * Goyo

#####
" Markdown **, * etc settings
#####
" set concealcursor=n
" set conceallevel=3
" hi Asterisks NONE
" hi AsteriskBold cterm=bold gui=bold
" syn match Asterisks contained "***" conceal
" syn match AsteriskBold "\\@<!\*\*[^"*\]]+\*\*" contains=Asterisks
" autocmd VimEnter * highlight Normal ctermbg=black
" ctermfg=grey

#####
" Other Settings
#####
set paste " to enable pasting in windows
nvim "C:/Users/d674a/Tex4Econ/_other/equation/multilines.tex"

```

1.3.3 Full Screen Vim Window in Windows

It is nice to have full screen, no distraction work experience. To set command prompt to full screen (to get rid of scrolling bar on the right in full screen), use windows command prompt which allows for full screen, right click on window top border, choose property, then choose layout, then set height and width of window size and buffer size to be the same. This will allow for full screen experience.

2 Vim Basics

```
# to open vim
vim -v
# to exit vim
:q!
# open up file from google drive
vim "/mnt/chromeos/GoogleDrive/MyDrive/Reading Papers/Reading/Referee/pa_2019_oct_journal/jor_12345_com
# start editing, type i to enter insert mode
i
# when editing is done, to exit and save, write and quit
# esc first
:wq
```

3 Vim Cursor Movements

Move cursor to the beginning of a line? This should not be done in edit mode, esc, use movement commands, then back to insert mode. There are six most basic movements, *wbjk()0\$*:

```
# forward one word
w
# back one word to begin
b
# back to word to end
e

# forward one line, achieved by also pressing downward arrow
j
# backward one line, achieved by pressing upward arrow
k
# left arrow
h
# right arrow
l

# Jump forward one Sentence
)
# Jump backward one Sentence
(

# Forward one page and back one page
# ctrl+f, ctrl+b

# Begin of line
0
# End of Line
$

# Begin of File
gg
# End of File
G
```

In combination, the basic commands can be used in combination

```
# move to the beginning of the next line
j0
# move to the end of the next line
j$
```

4 Very Convenient Functions

Some functionalities take much more efforts to complete in many other editors, but are very easy in vim:

```
# combine lines: bunch of lines, with carriage break, one block of text, reflow/join
# 1. select block
Vjjjj
# 2. reflow/join
J
```

4.1 Vim go to Spot in Text Quickly

```
# ESC
# Search for word
/abc
# esc
# next occurrence of word
n
# last occurrence
N
# first occurrence overall
ggn
# last occurrence overall
G
# clear search highlighting
:noh
back to previous point
~.
```

4.2 Combo Operations

1. Delete:
 - delete current word and go into insert mode: *ciw*
 - Delete current word: *dw*
 - Delete from cursor until end of sentence: *dis*
 - Delete from cursor to end of line: *d\$* or *D*
 - Delete until end of paragraph: *dip*
 - Delete all text in file: *ggdG*

4.3 Vim Statistics

- word count: *g*, *ctrl+g*

5 Vim Errors

Vim seems frozen (hard for vim to freeze, it is not really frozen). This is likely due to pressing *ctrl + s*, then need to press *ctrl + q*.

vim error: "E297: Write error in swap file"

```
# Trying to save
```

```
:w
```

```
# E297 can not save, override changes elsewhere, and force save what is on screen.
```

```
:e!
```

6 Vim Set-Up Others

6.1 Chromebook

pressing esc key is an issue, too far away. On chromebook, remap the search key which is in the spot of the caps lock key to esc. and remap the esc key to search key.

1. search for keyboard under settings
2. launcher to escape, and escape to launcher