GitHub Gist

Sign up for a GitHub account Sign in

Instantly share code, notes, and snippets.

Create a gist now

mikehaertl / **vimrc.local**

Last active 5 days ago

A global vim configuration on Ubuntu that uses Vundle. Just put it in /etc/vim/vimrc.local and follow the instructions

vimrc.local

```
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" /etc/vim/vimrc.local V1.1.5 2016-07-08 https://gist.github.com/mikehaertl/1612035
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"
" A Vundle based Vim configuration with globally shared plugins on Ubuntu.
"
" This is a Vundle based Vim setup that keeps all plugins in a global
" directory, namely /etc/vim/bundle. It's trimmed towards PHP development
" with Yii.
"
" Before you install, make sure you have a ruby enabled vim. On fresh Ubuntu you can do:
"
"   sudo apt-get install vim-gtk
"
" Installation:
"
"   mkdir /etc/vim/bundle.
"   git clone http://github.com/gmarik/vundle.git /etc/vim/bundle/vundle
"   vi
"  :PluginInstall
"
" Brief help
"  :PluginList          - list configured bundles
"  :PluginInstall(!)    - install(update) bundles
"  :PluginSearch(!) foo - search(or refresh cache first) for foo
"  :PluginClean(!)      - confirm(or auto-approve) removal of unused bundles
"
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" Shared plugin setup for Vundle
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
filetype off                        " required by Vundle!
set rtp+=/etc/vim/bundle/vundle
call vundle#rc('/etc/vim/bundle')   " Use a shared folder instead of ~/.vimrc/bundle
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""


""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" Vundle plugins
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
Plugin 'gmarik/vundle'

" Styling
Plugin 'guns/xterm-color-table.vim'
Plugin 'vim-airline/vim-airline'
Plugin 'vim-airline/vim-airline-themes'
Plugin 'CSApprox'
Plugin 'Colour-Sampler-Pack'

" Movement
Plugin 'Lokaltog/vim-easymotion'
Plugin 'vim-scripts/matchit.zip'

" Files & Search
Plugin 'LustyExplorer'
Plugin 'The-NERD-tree'
Plugin 'mileszs/ack.vim'

" Editing utilities
Plugin 'tpope/vim-surround'
Plugin 'tpope/vim-repeat'
```

```

" Help
Plugin 'powerman/vim-plugin-viewdoc'

" Syntax / Indenting
Plugin 'pangloss/vim-javascript'
Plugin 'StanAngeloff/php.vim'
Plugin '2072/PHP-Indenting-for-VIm'
Plugin 'rodjek/vim-puppet'
Plugin 'evanmiller/nginx-vim-syntax'
Plugin 'plasticboy/vim-markdown'
Plugin 'groenewege/vim-less'

" Git
Plugin 'tpope/vim-fugitive'

" Clojure
Plugin 'guns/vim-clojure-static'
Plugin 'tpope/vim-fireplace'
Plugin 'paredit.vim'

" PHP / Yii
Plugin 'mikehaertl/pdv-standalone'
" To activate add project specific .vimrc with
" set runtimepath+=/etc/vim/bundle/yii2-apidoc-vim
Plugin 'mikehaertl/yii-api-vim', {'rtp': 'none'}
Plugin 'mikehaertl/yii2-apidoc-vim', {'rtp': 'none'}


""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" General
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
set nocompatible                " get out of horrible vi-compatible mode
set history=50                  " How many lines of history to remember
set confirm                     " Ask for confirmation in some situations (:q)
set ignorecase smartcase        " case insensitive search, except when mixing cases
set modeline                    " we allow modelines in textfiles to set vim settings
set hidden                      " allows to change buffer without saving
set mouse=a                     " enable mouse in all modes
set noerrorbells                " don't make noise
set novisualbell                " don't blink
set t_Co=256                    " Enable 256 color mode
set exrc                        " Scan working dir for .vimrc
set secure                      " Make the above work safely


""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" UI/Colors
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
set wildmenu                    " Show suggestions on TAB for some commands
set ruler                       " Always show current positions along the bottom
set cmdheight=1                 " the command bar is 1 high
set number                      " turn on line numbers
"set nonumber                    " turn off line numbers (problematic with cut&paste)
set lazyredraw                  " do not redraw while running macros (much faster)
set backspace=indent,eol,start  " make backspace work normal
set whichwrap+=<,>,h,l          " make cursor keys and h,l wrap over line endings
set report=0                    " always report how many lines where changed
set fillchars=vert:\ ,stl:\ ,stlnc:\    " make the splitters between windows be blank
set laststatus=2                " always show the status line
set scrolloff=10                " Start scrolling this number of lines from top/bottom

syntax on                       " syntax highlighting on (set before colorscheme)
colorscheme darkspectrum
let g:airline_theme='sol'
let g:airline_theme_patch_func = 'AirlineThemePatch'
function! AirlineThemePatch(palette)
  if g:airline_theme == 'sol'
    " Current tab should always have hightes optical priority, so don't use
    " redish colors for modified tabs

    " Color for current tab if modified (darkened blue)
    let a:palette.tabline.airline_tabmod[3] = 30

    " Color for modified (grey
    let a:palette.normal_modified.airline_c[3] = 244
  endif
endfunction

""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" Text Formatting/Tab settings
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
set formatoptions=tcrqn         " autowrap and comments (see :h 'fo-table)
set autoindent                  " keep indent on next line and make BS work with indenting
set wrap                        " wrap lines that exceed screen
set smarttab                    " Make Tab work fine with spaces
set showmatch                   " show matching brackets
set matchtime=5                 " tenths of a second to blink matching brackets
set hlsearch                    " highlight search phrase matches (reset with :noh)
set incsearch                   " do highlight as you type you search phrase
set list                        " show tabs, trailings spaces, ...
set listchars=tab:\|\ ,trail:.,extends:>,precedes:<


""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" File settings
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
set encoding=utf-8              " Let Vim use utf-8 internally
set fileencoding=utf-8          " Default for new files
set termencoding=utf-8          " Terminal encoding
set fileformats=unix,dos,mac    " support all three, in this order
set fileformat=unix             " default file format

filetype plugin on              " load filetype plugins
filetype indent on              " load indent settings

" When editing a file, always jump to the last cursor position.
autocmd BufReadPost * if line("'\"") && line("'\"") <= line("$") | exe "normal `\"" | endif

" Set some file types by extension
autocmd BufNewFile,BufRead *.xt,*.xd            setf xml
autocmd BufNewFile,BufRead *.tpl,*.page         setf html

" Global indent settings
set tabstop=4 softtabstop=4 shiftwidth=4 expandtab

" Indenting per file type
" tabstop: number of spaces inserted for each tab
" softtabstop: make spaces feel like real tabs (e.g. for backspace)
" shiftwidth: number of spaces for indentation (e.g. > and < )
" expandtab: use spaces instaed of Tab. <c-v><TAB> gives real Tab
autocmd FileType php setlocal tabstop=4 softtabstop=4 shiftwidth=4 expandtab autoindent
autocmd FileType javascript setlocal tabstop=2 softtabstop=2 shiftwidth=2 expandtab

" PHP settings
" Let the surround plugin use `-` for <?php ?>
autocmd FileType php let b:surround_45 = "<?php \r ?>"
" Let the surround plugin use `=` for <?= ?>
autocmd FileType php let b:surround_61 = "<?= \r ?>"
" Fix javascript word boundaries (erratically activated for PHP files): exclude $
autocmd FileType php setlocal iskeyword-=$

" Function for autodetecting tab settings
function Kees_settabs()
    if len(filter(getbufline(winbufnr(0), 1, "$"), 'v:val =~ "^\\t"')) > len(filter(getbufline(winbufnr(0), 1, "$"), 'v:val =~ '
        set noet
    else
        set et
    endif
endfunction
autocmd BufReadPost * call Kees_settabs()

" No folding for markdown files
let g:vim_markdown_folding_disabled=1

""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" Plugin Settings
```

```
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" surround
let g:surround_indent = 0           " Make indenting on block selection + S work

" matchit
let b:match_ignorecase = 1

" PHP syntax
let php_sql_query = 1
let php_htmlInStrings = 1
let php_baselib = 1
let php_asp_tags = 0
let php_parent_error_close = 1
let php_parent_error_open = 1
let php_no_shorttags = 0
let php_folding = 0
let php_sync_method = 0

" PHP indent
let g:PHP_outdentphpescape = 0
let g:PHP_vintage_case_default_indent = 1

" viewdoc
let g:viewdoc_open = 'new'
let g:viewdoc_openempty = 0
" Custom help handler that tries Yii help and PHP help, in this order.
" Note: PHP help requires sudo pear install doc.php.net/pman
function ViewDoc_pman_custom(topic, filetype, synid, ctx)
  let l:tagpath = '/etc/vim/bundle/yii2-apidoc-vim/doc/tags'
  let l:shell = printf('grep %s %s', shellescape(a:topic,1), l:tagpath)
  let l:output = system(l:shell)

  if !v:shell_error
    return g:ViewDoc_help(a:topic, a:filetype, a:synid, a:ctx)
  else
    return g:ViewDoc_pman_old(a:topic, a:filetype, a:synid, a:ctx)
  endif
endfunction
autocmd VimEnter * let g:ViewDoc_pman_old = g:ViewDoc_pman
autocmd VimEnter * let g:ViewDoc_pman = function('ViewDoc_pman_custom')
autocmd VimEnter * let g:ViewDoc_php = function('ViewDoc_pman_custom')
" make :h an alias for :viewDoc
cnoreabbrev <expr> h ((getcmdtype() is# ':' && getcmdline() is# 'h')?('ViewDocHelp!'):('h'))

" airline
let g:airline#extensions#tabline#enabled = 1
let g:airline#extensions#tabline#buffer_nr_show = 1
let g:airline_powerline_fonts = 1

""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" Key Mappings
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
let mapleader = ","             " Map <leader> to , instead of default \

" ESC is 'jj', typed quickly (:help 'timeout')
imap jj <Esc>

" Don't use Ex mode, use Q for formatting
map Q gq

" Use cp{motion} to replace {motion} with unnamed register content
nmap <silent> cp :set opfunc=ChangePaste<CR>g@
function! ChangePaste(type, ...)
    silent exe "normal! `[v`]\"_c"
    silent exe "normal! p"
endfunction

" Use <tab> instead of % to match brackets
nnoremap <tab> %
vnoremap <tab> %

" Make p in Visual mode replace the selected text with the "" register.
vnoremap p <Esc>:let current_reg = @"<CR>gvdi<C-R>=current_reg<CR><Esc>

```

```
" Toggle paste mode (no autoindenting) with F11
set pastetoggle=<F11>

" use <C-[jl]> to switch to prev/next buffer
noremap <C-j> :bp<CR>
noremap <C-l> :bn<CR>

" PDV (PHP Documentor)
nnoremap <C-K> :call PhpDocSingle()<CR>
vnoremap <C-K> :call PhpDocRange()<CR>

" Show NERDtree on F8
noremap <silent> <F8> :NERDTreeToggle<CR>

" Show LustyExplorer on ",f", ",r", ",g"
noremap <leader>f :LustyFilesystemExplorer<CR>
noremap <leader>r :LustyFilesystemExplorerFromHere<CR>
noremap <leader>g :LustyBufferGrep<CR>

" Tagbar on _
nmap <silent> _ :TagbarToggle<CR>
```