

ステータス:	新規	開始日:	2016/02/25
優先度:	通常	期日:	
担当者:		進捗 %:	0%
カテゴリ:		予定工数:	0.00時間
対象バージョン:		作業時間の記録:	0.00時間

説明

```
$ cat /.vimrc
```

```
[spyder1211@sv965 ]$ cat /.vimrc
```

```
"-----
```

```
" 文字コード
```

```
"-----
```

```
" 文字コードの自動認識
```

```
if &encoding !=# 'utf-8'
```

```
  set encoding=japan
```

```
  set fileencoding=japan
```

```
endif
```

```
if has('iconv')
```

```
  let s:enc_euc = 'euc-jp'
```

```
  let s:enc_jis = 'iso-2022-jp'
```

```
  " iconvがeucJP-msに対応しているかをチェック
```

```
  if iconv("¥x87¥x64¥x87¥x6a", 'cp932', 'eucjp-ms') ==# "¥xad¥xc5¥xad¥xcb"
```

```
    let s:enc_euc = 'eucjp-ms'
```

```
    let s:enc_jis = 'iso-2022-jp-3'
```

```
  " iconvがJISX0213に対応しているかをチェック
```

```
  elseif iconv("¥x87¥x64¥x87¥x6a", 'cp932', 'euc-jisx0213') ==# "¥xad¥xc5¥xad¥xcb"
```

```
    let s:enc_euc = 'euc-jisx0213'
```

```
    let s:enc_jis = 'iso-2022-jp-3'
```

```
  endif
```

```
  " fileencodingsを構築
```

```
  if &encoding ==# 'utf-8'
```

```
    let s:fileencodings_default = &fileencodings
```

```
    let &fileencodings = s:enc_jis .',' s:enc_euc .',cp932'
```

```
    let &fileencodings = &fileencodings .',' s:fileencodings_default
```

```
    unlet s:fileencodings_default
```

```
  else
```

```
    let &fileencodings = &fileencodings .',' s:enc_jis
```

```
    set fileencodings+=utf-8,ucs-2le,ucs-2
```

```
    if &encoding = # '^¥(euc-jp¥!euc-jisx0213¥!eucjp-ms¥)¥$'
```

```
      set fileencodings+=cp932
```

```
      set fileencodings=euc-jp
```

```
      set fileencodings=euc-jisx0213
```

```
      set fileencodings=eucjp-ms
```

```
      let &encoding = s:enc_euc
```

```
      let &fileencoding = s:enc_euc
```

```
    else
```

```
      let &fileencodings = &fileencodings .',' s:enc_euc
```

```

endif
endif
" 定数を処分
unlet s:enc_euc
unlet s:enc_jis
endif
" 日本語を含まない場合は fileencoding に encoding を使うようにする
if has('autocmd')
function! AU_ReCheck_FENC()
    if &fileencoding = # 'iso-2022-jp' && search("[^¥x01-¥x7e]", 'n') == 0
        let &fileencoding=&encoding
    endif
endfunction
autocmd BufReadPost * call AU_ReCheck_FENC()
endif
" 改行コードの自動認識
set fileformats=unix,dos,mac
"   とか   の文字があってもカーソル位置がずれないようにする
if exists('&ambiwidth')
    set ambiwidth=double
endif
"-----
" 基本的な設定
"-----
" viとの互換性をとらない(vimの独自拡張機能を使う為)
set nocompatible
" 改行コードの自動認識
set fileformats=unix,dos,mac
" ビープ音を鳴らさない
set vb t_vb=
" バックスペースキーで削除できるものを指定
" indent : 行頭の空白
" eol   : 改行
" start : 挿入モード開始位置より手前の文字
set backspace=indent,eol,start

"-----
" バックアップ関係
"-----
" バックアップをとらない
set nobackup
" ファイルの上書きの前にバックアップを作る
" (ただし、backup がオンでない限り、バックアップは上書きに成功した後削除される)
set writebackup
" バックアップをとる場合
"set backup
" バックアップファイルを作るディレクトリ
"set backupdir= /backup
" スワップファイルを作るディレクトリ
"set directory= /swap

"-----
" 検索関係

```

```

"-----
" コマンド、検索パターンを100個まで履歴に残す
set history=100
" 検索の時に大文字小文字を区別しない
set ignorecase
" 検索の時に大文字が含まれている場合は区別して検索する
set smartcase
" 最後まで検索したら先頭に戻る
set wrapscan
" インクリメンタルサーチを使わない
set noincsearch

"-----
" 表示関係
"-----
" タイトルをウインドウ枠に表示する
" set title
" 行番号を表示しない
set number
" ルーラーを表示
set ruler
" タブ文字を CTRL-I で表示し、行末に $ で表示する
"set list
" 入力中のコマンドをステータスに表示する
set showcmd
" ステータスラインを常に表示
set laststatus=2
" 括弧入力時の対応する括弧を表示
set showmatch
" 対応する括弧の表示時間を2にする
" set matchtime=2
" シンタックスハイライトを有効にする
syntax on
" 検索結果文字列のハイライトを有効にする
set hlsearch
" コメント文の色を変更
highlight Comment ctermfg=DarkCyan
" コマンドライン補完を拡張モードにする
set wildmenu

" 入力されているテキストの最大幅
" (行がそれより長くなると、この幅を超えないように空白の後で改行される)を無効にする
set textwidth=0
" ウィンドウの幅より長い行は折り返して、次の行に続けて表示する
set wrap

" 全角スペースの表示
highlight ZenkakuSpace cterm=underline ctermfg=lightblue guibg=darkgray
match ZenkakuSpace / /

" ステータスラインに表示する情報の指定
set statusline=%n¥:%y%F¥ ¥!%{(&fenc!="?&fenc:&enc).¥!'.&ff.¥!}'%m%r%=<%l/%L:%p%%>
" ステータスラインの色

```

```
highlight StatusLine term=NONE cterm=NONE ctermfg=black ctermbg=white
```

```
"-----  
" インデント  
"-----  
" オートインデントを無効にする  
"set noautoindent  
" タブが対応する空白の数  
set tabstop=4  
" タブやバックスペースの使用等の編集操作をするときに、タブが対応する空白の数  
set softtabstop=4  
" インデントの各段階に使われる空白の数  
set shiftwidth=4  
" タブを挿入するとき、代わりに空白を使わない  
set noexpandtab  
  
"-----  
" オートコマンド  
"-----  
if has("autocmd")  
" ファイルタイプ別インデント、プラグインを有効にする  
filetype plugin indent on  
" カーソル位置を記憶する  
autocmd BufReadPost *  
  ¥ if line("¥") > 0 && line("¥") <= line("$) ;  
  ¥ exe "normal g`¥" ;  
  ¥ endif  
endif  
  
"-----  
" その他  
"-----  
" バッファを切替えてもundoの効力を失わない  
set hidden  
" 起動時のメッセージを表示しない  
set shortmess+=l  
" 横線の表示  
set cursorline  
" 縦線の表示  
"set cursorcolumn  
  
let g:neobundle_default_git_protocol='git'  
  
set nocompatible      " be iMproved  
filetype off  
  
if has('vim_starting')  
  set runtimepath+= /.vim/bundle/neobundle.vim  
  call neobundle#begin(expand(' /.vim/bundle/'))  
  NeoBundleFetch 'Shougo/neobundle.vim'  
  call neobundle#end()  
endif  
" originalrepos on github
```

```

NeoBundle 'Shougo/neobundle.vim'
"NeoBundle 'Shougo/vimproc.vim'
NeoBundle 'Shougo/neocomplcache'
NeoBundle 'Shougo/unite.vim'

filetype plugin indent on " required!
filetype indent on
syntax on

"" neocomplcache
NeoBundle 'Shougo/neocomplcache'
" Disable AutoComplPop.
let g:acp_enableAtStartup = 0
" " Use neocomplcache.
let g:neocomplcache_enable_at_startup = 1
" " Use smartcase.
let g:neocomplcache_enable_smart_case = 1
" " Set minimum syntax keyword length.
let g:neocomplcache_min_syntax_length = 3
let g:neocomplcache_lock_buffer_name_pattern = '¥*ku¥*'
"
" " Define dictionary.
let g:neocomplcache_dictionary_filetype_lists = {
    ¥ 'default' : "
        ¥ }
"
" Plugin key-mappings.
inoremap <expr><C-g> neocomplcache#undo_completion()
inoremap <expr><C-l> neocomplcache#complete_common_string()

" Recommended key-mappings.
" <CR>: close popup and save indent.
inoremap <silent> <CR> <C-r>=<SID>my_cr_function()<CR>
function! s:my_cr_function()
    return neocomplcache#smart_close_popup() . "¥<CR>"
endfunction
" <TAB>: completion.
inoremap <expr><TAB> pumvisible() ? "¥<C-n>" : "¥<TAB>"
" <C-h>, <BS>: close popup and delete backward char.
inoremap <expr><C-h> neocomplcache#smart_close_popup()."¥<C-h>"
inoremap <expr><BS> neocomplcache#smart_close_popup()."¥<C-h>"
inoremap <expr><C-y> neocomplcache#close_popup()
inoremap <expr><C-e> neocomplcache#cancel_popup()

" Unite
NeoBundle 'Shougo/unite.vim'
let g:unite_enable_start_insert=1
let g:unite_source_history_yank_enable =1
let g:unite_source_file_mru_limit = 200
nnoremap <silent> ,uy :<C-u>Unite history/yank<CR>
nnoremap <silent> ,ub :<C-u>Unite buffer<CR>
nnoremap <silent> ,uf :<C-u>UniteWithBufferDir -buffer-name=files file<CR>
nnoremap <silent> ,ur :<C-u>Unite -buffer-name=register register<CR>

```

```

nnooremap <silent> ,uu :<C-u>Unite file_mru buffer<CR>
nnooremap <silent>bp :bprevious<CR>
nnooremap <silent>bn :bnext<CR>
nnooremap <silent>bb :b#<CR>
NeoBundle 'scrooloose/syntastic'
let g:syntastic_enable_signs=1
let g:syntastic_auto_loc_list=2
let g:syntastic_check_on_open = 1
let g:syntastic_enable_signs = 1

let g:syntastic_auto_loc_list = 2
let g:syntastic_enable_highlighting = 1
set statusline+=%#warningmsg#
set statusline+=%{SyntasticStatuslineFlag()}
set statusline+=%*
autocmd BufRead *.php%!*.ctp%!*.tpl :Errors

NeoBundle 'scrooloose/syntastic'
let g:syntastic_enable_signs=1
let g:syntastic_auto_loc_list=2
let g:syntastic_check_on_open = 1
let g:syntastic_enable_signs = 1

let g:syntastic_auto_loc_list = 2
let g:syntastic_enable_highlighting = 1
set statusline+=%#warningmsg#
set statusline+=%{SyntasticStatuslineFlag()}
set statusline+=%*
autocmd BufRead *.php%!*.ctp%!*.tpl :Errors

NeoBundle 'tpope/vim-fugitive'
NeoBundle 'Shougo/unite-outline'
nnooremap <silent> ,uo :Unite -vertical -no-quit outline<CR>

```

履歴

#1 - 2016/02/25 18:17 - 長谷川 結

Neobundle ver3.0以上

```

" Note: Skip initialization for vim-tiny or vim-small.

if 0 | endif

if &compatible
  set nocompatible          " Be iMproved
endif

" Required:
set runtimepath^= /.vim/bundle/neobundle.vim/

" Required: 変更前 : call neobundle#rc(expand(' /.vim/bundle/'))

```

```
call neobundle#begin(expand(' .vim/bundle/'))
```

```
" Let NeoBundle manage NeoBundle
" Required:
NeoBundleFetch 'Shougo/neobundle.vim'
```

```
" My Bundles here:
" Refer to !:NeoBundle-examples!.
" Note: You don't set neobundle setting in .gvimrc!
```

```
call neobundle#end()
```

```
" Required:
filetype plugin indent on
```

```
" If there are uninstalled bundles found on startup,
" this will conveniently prompt you to install them.
NeoBundleCheck
```

#2 - 2016/02/25 18:34 - 長谷川 結

??Neobundle ver3.0以上の場合??

```
"-----
" 文字コード
"-----
" 文字コードの自動認識
if &encoding !=# 'utf-8'
  set encoding=japan
  set fileencoding=japan
endif
if has('iconv')
  let s:enc_euc = 'euc-jp'
  let s:enc_jis = 'iso-2022-jp'
  " iconvがeucJP-msに対応しているかをチェック
  if iconv("\x87\x64\x87\x6a", 'cp932', 'eucjp-ms') ==# "\xad\x5\xad\xcb"
    let s:enc_euc = 'eucjp-ms'
    let s:enc_jis = 'iso-2022-jp-3'
  " iconvがJISX0213に対応しているかをチェック
  elseif iconv("\x87\x64\x87\x6a", 'cp932', 'euc-jisx0213') ==# "\xad\x5\xad\xcb"
    let s:enc_euc = 'euc-jisx0213'
    let s:enc_jis = 'iso-2022-jp-3'
  endif
  " fileencodingsを構築
  if &encoding ==# 'utf-8'
    let s:fileencodings_default = &fileencodings
    let &fileencodings = s:enc_jis .',' s:enc_euc .','cp932'
    let &fileencodings = &fileencodings .',' s:fileencodings_default
    unlet s:fileencodings_default
  else
```

```

let &fileencodings = &fileencodings .',' s:enc_jis
set fileencodings+=utf-8,ucs-2le,ucs-2
if &encoding = # '^%(euc-jp%|euc-jisx0213%|eucjp-ms%)$'
  set fileencodings+=cp932
  set fileencodings=euc-jp
  set fileencodings=euc-jisx0213
  set fileencodings=eucjp-ms
  let &encoding = s:enc_euc
  let &fileencoding = s:enc_euc
else
  let &fileencodings = &fileencodings .',' s:enc_euc
endif
endif
" 定数を処分
unlet s:enc_euc
unlet s:enc_jis
endif
" 日本語を含まない場合は fileencoding に encoding を使うようにする
if has('autocmd')
function! AU_ReCheck_FENC()
  if &fileencoding = # 'iso-2022-jp' && search("[^%x01-%x7e]", 'n') == 0
    let &fileencoding=&encoding
  endif
endfunction
autocmd BufReadPost * call AU_ReCheck_FENC()
endif
" 改行コードの自動認識
set fileformats=unix,dos,mac
" とか の文字があってもカーソル位置がずれないようにする
if exists('&ambiwidth')
  set ambiwidth=double
endif
"-----
" 基本的な設定
"-----
" viとの互換性をとらない(vimの独自拡張機能を使う為)
set nocompatible
" 改行コードの自動認識
set fileformats=unix,dos,mac
" ビープ音を鳴らさない
set vb t_vb=
" バックスペースキーで削除できるものを指定
" indent : 行頭の空白
" eol : 改行
" start : 挿入モード開始位置より手前の文字
set backspace=indent,eol,start
"-----
" バックアップ関係
"-----
" バックアップをとらない
set nobackup
" ファイルの上書きの前にバックアップを作る

```

```
" (ただし、backup がオンでない限り、バックアップは上書きに成功した後削除される)
set writebackup
" バックアップをとる場合
"set backup
" バックアップファイルを作るディレクトリ
"set backupdir= /backup
" スワップファイルを作るディレクトリ
"set directory= /swap

"-----
" 検索関係
"-----
" コマンド、検索パターンを100個まで履歴に残す
set history=100
" 検索の時に大文字小文字を区別しない
set ignorecase
" 検索の時に大文字が含まれている場合は区別して検索する
set smartcase
" 最後まで検索したら先頭に戻る
set wrapscan
" インクリメンタルサーチを使わない
set noincsearch

"-----
" 表示関係
"-----
" タイトルをウインドウ枠に表示する
" set title
" 行番号を表示しない
set number
" ルーラーを表示
set ruler
" タブ文字を CTRL-I で表示し、行末に $ で表示する
"set list
" 入力中のコマンドをステータスに表示する
set showcmd
" ステータスラインを常に表示
set laststatus=2
" 括弧入力時の対応する括弧を表示
set showmatch
" 対応する括弧の表示時間を2にする
" set matchtime=2
" シンタックスハイライトを有効にする
syntax on
" 検索結果文字列のハイライトを有効にする
set hlsearch
" コメント文の色を変更
highlight Comment ctermfg=DarkCyan
" コマンドライン補完を拡張モードにする
set wildmenu

" 入力されているテキストの最大幅
" (行がそれより長くなると、この幅を超えないように空白の後に改行される)を無効にする
```

```

set textwidth=0
" ウィンドウの幅より長い行は折り返して、次の行に続けて表示する
set wrap

" 全角スペースの表示
highlight ZenkakuSpace cterm=underline ctermfg=lightblue guibg=darkgray
match ZenkakuSpace / /

" ステータスラインに表示する情報の指定
set statusline=%n%:~%y%F% ¥!%{(&fenc!="?&fenc:&enc).¥!'&ff.¥!'}%m%r%=<%l/%L:%p%>
" ステータスラインの色
highlight StatusLine term=NONE cterm=NONE ctermfg=black ctermbg=white

"-----
" インデント
"-----
" オートインデントを無効にする
"set noautoindent
" タブが対応する空白の数
set tabstop=4
" タブやバックスペースの使用等の編集操作をするときに、タブが対応する空白の数
set softtabstop=4
" インデントの各段階に使われる空白の数
set shiftwidth=4
" タブを挿入するとき、代わりに空白を使わない
set noexpandtab

"-----
" オートコマンド
"-----
if has("autocmd")
" ファイルタイプ別インデント、プラグインを有効にする
filetype plugin indent on
" カーソル位置を記憶する
autocmd BufReadPost *
  ¥ if line("¥") > 0 && line("¥") <= line("$) ;
  ¥ exe "normal g`¥" ;
  ¥ endif
endif

"-----
" その他
"-----
" バッファを切替えてもundoの効力を失わない
set hidden
" 起動時のメッセージを表示しない
set shortmess+=l
" 横線の表示
set cursorline
" 縦線の表示
"set cursorcolumn
" netrwは常にtree view
let g:netrw_liststyle = 3

```

```

" " CVSと.で始まるファイルは表示しない
let g:netrw_list_hide = 'CVS,*(^*!%s*s%)*z%*.%$*+'
" " 'v'でファイルを開くときは右側に開く。(デフォルトが左側なので入れ替え)
let g:netrw_altv = 1
" " 'o'でファイルを開くときは下側に開く。(デフォルトが上側なので入れ替え)
let g:netrw_alto = 1
if has('vim_starting')
    set nocompatible          " Be iMproved
    set runtimepath+= /.vim/bundle/neobundle.vim/
endif

" -----
" neobundle
" -----

" Note: Skip initialization for vim-tiny or vim-small.
if 0 | endif

filetype off

if has('vim_starting')
    if &compatible
        set nocompatible      " Be iMproved
    endif

    set runtimepath+= /.vim/bundle/neobundle.vim
endif

call neobundle#begin(expand(' /.vim/bundle/'))

" originalrepos on github
NeoBundle 'Shougo/neobundle.vim'
NeoBundle 'Shougo/vimproc', {
    ¥ 'build' : {
        ¥ 'windows' : 'make -f make_mingw32.mak',
        ¥ 'cygwin' : 'make -f make_cygwin.mak',
        ¥ 'mac' : 'make -f make_mac.mak',
        ¥ 'unix' : 'make -f make_unix.mak',
    ¥ },
    ¥ }
NeoBundle 'VimClojure'
NeoBundle 'Shougo/vimshell'
NeoBundle 'Shougo/unite.vim'
NeoBundle 'Shougo/neosnippet'
NeoBundle 'jpalardy/vim-slime'
NeoBundle 'scroolose/syntastic'
NeoBundle 'Shougo/vimfiler.vim'
NeoBundle 'itchyny/lightline.vim'
NeoBundle 't9md/vim-textmanip'

NeoBundle 'ujihisa/unite-colorscheme'
NeoBundle 'tomasr/molokai'
""NeoBundle 'https://bitbucket.org/kovisoft/slimv'

```

NeoBundle 'scrooloose/nerdtree'

NeoBundle 'kien/ctrlp.vim'

"" neocomplcache

NeoBundle 'Shougo/neocomplcache'

" Disable AutoComplPop.

let g:acp_enableAtStartup = 0

" " Use neocomplcache.

let g:neocomplcache_enable_at_startup = 1

" " Use smartcase.

let g:neocomplcache_enable_smart_case = 1

" " Set minimum syntax keyword length.

let g:neocomplcache_min_syntax_length = 3

let g:neocomplcache_lock_buffer_name_pattern = '¥*ku¥*'

"

" " Define dictionary.

let g:neocomplcache_dictionary_filetype_lists = {

¥ 'default' : "

¥ }

"

" Plugin key-mappings.

inoremap <expr><C-g> neocomplcache#undo_completion()

inoremap <expr><C-l> neocomplcache#complete_common_string()

" Recommended key-mappings.

" <CR>: close popup and save indent.

inoremap <silent> <CR> <C-r>=<SID>my_cr_function()<CR>

function! s:my_cr_function()

return neocomplcache#smart_close_popup() . "¥<CR>"

endfunction

" <TAB>: completion.

inoremap <expr><TAB> pumvisible() ? "¥<C-n>" : "¥<TAB>"

" <C-h>, <BS>: close popup and delete backward char.

inoremap <expr><C-h> neocomplcache#smart_close_popup()."¥<C-h>"

inoremap <expr><BS> neocomplcache#smart_close_popup()."¥<C-h>"

inoremap <expr><C-y> neocomplcache#close_popup()

inoremap <expr><C-e> neocomplcache#cancel_popup()

NeoBundle 'tpope/vim-fugitive'

call neobundle#end()

" Required

filetype plugin indent on

" If there are uninstalled bundles found on startup,

" " this will conveniently prompt you to install them.

NeoBundleCheck

- 説明を更新