作者：聂飞
转载请注明出处

在终端下使用vim进行编辑时，默认情况下，编辑的界面上是没有显示行号、语法高亮度显示
工作，需要手动设置一个配置文件：.vimrc。
在启动vim时，当前用户根目录下的.vimrc文件会被自动读取，该文件可以包含一些设置甚至
前用户的根目录下比较方便，即创建的命令为：

```
$vi ~/.vimrc
```

设置完后

```
$:x 或者 $wq
```

进行保存退出即可。
下面给出一个例子，其中列出了经常用到的设置，详细的设置信息请参照参考资料：

```
“双引号开始的行为注释行，下同

“去掉讨厌的有关vi一致性模式，避免以前版本的一些bug和局限

set nocompatible

“显示行号

set nummber

“检测文件的类型

filetype on

“记录历史的行数

set history=1000

“背景使用黑色

set background=dark

“语法高亮度显示

syntax on

“下面两行在进行编写代码时，在格式对起上很有用；

“第一行，vim使用自动对起，也就是把当前行的对起格式应用到下一行；

“第二行，依据上面的对起格式，智能的选择对起方式，对于类似C语言编

“写上很有用

set autoindent
set smartindent

“第一行设置tab键为4个空格，第二行设置当行之间交错时使用4个空格
```

```
set tabstop=4
set shiftwidth=4

“设置匹配模式，类似当输入一个左括号时会匹配相应的那个右括号

set showmatch

“去除vim的GUI版本中的toolbar

set guioptions-=T

“当vim进行编辑时，如果命令错误，会发出一个响声，该设置去掉响声

set vb t_vb=

“在编辑过程中，在右下角显示光标位置的状态行

set ruler

“默认情况下，寻找匹配是高亮度显示的，该设置关闭高亮显示

set nohls

“查询时非常方便，如要查找book单词，当输入到/b时，会自动找到第一

“个b开头的单词，当输入到/bo时，会自动找到第一个bo开头的单词，依

“次类推，进行查找时，使用此设置会快速找到答案，当你找要匹配的单词

“时，别忘记回车

set incsearch

“修改一个文件后，自动进行备份，备份的文件名为原文件名加“~“后缀

if has(“vms”)

set nobackup

else

set backup

endif
```

如果去除注释后，一个完整的.vimrc配置信息如下所示：

```
set nocompatible

set nummber

filetype on
```

```
set history=1000

set background=dark

syntax on

set autoindent
set smartindent

set tabstop=4
set shiftwidth=4

set showmatch

set guioptions-=T

set vb t_vb=

set ruler

set nohls

set incsearch

if has(“vms”)

set nobackup

else

set backup

endif
```

如果设置完后，发现功能没有起作用，检查一下系统下是否安装了vim-enhanced包，查询命令

```
$rpm －q vim-enhanced
```

参考资料：

1. vim的完全翻译版在下面连接处可以找到

http://vimcdoc.sourceforge.net/

可以下载其中的一个PDF版本，里面介绍的很详细，强烈推荐：）
2. 更详细的vim信息可以访问：

http://www.vim.org/

3. 一个带有英文注释的.vimrc例子

http://www.vi-improved.org/vimrc.php

主要实现自动缩进，换行等功能。

一个很简单的.vimrc

主要实现自动缩进，换行等功能。

```
set nocompatible
set columns=80
set tabstop=4
set ruler
set noexpandtab
set number
set ignorecase
set wrapmargin=15
set cindent
set shiftwidth=4
syntax on

"关掉备份文件
set nobackup

"暂时没用，留着吧
hi CursorLine guibg=LightBlue

"高亮当前行
set cursorline

"设置颜色方案
colorscheme torte

"显示行号
set number

"关掉工具条
set go=e
```

Windows 和 Linux 下适用，无需更改，除了要重新设置用到的程序路径；
GUI 和 Console 或终端下适用，无需更改。

可在我的站点上下载/查看： _vimrc, _vimrc.txt, _vimrc.html
http://linxiao.li.googlepages.com/vim
http://www.leal.cn

使用前需要先下载一些 scripts， 见 _vimrc 或 plugin_list.txt, plugin_list.html 同上网址。

使用前先查看 vimrc 的使用说明。

Good Luck!

附 _vimrc

```
"================================================================================
    " Description: My daily vimrc with dozens of plugins.
    " Originally derived from MetaCosm's http://vi-improved.org/vimrc.php
    " Last Change: 0 14/05/2006 20:54:07 Leal@RAINT:_vimrc
    "
    " Author: Leal <linxiao.li AT gmail DOT com>
    " http://www.leal.cn
    " Version: 1.0065
    "
    " Usage: 1. Prepare necessary dirs and files
    " $VIMDATA X:\Vim\vimdata on Win, ~/vimdata on Linux
    " |
    " |-- temp to put swap files <DIR>
    " |-- backup to put backup files <DIR>
    " |-- diary to save calendar.vim's diaries <DIR>
    " |-- GetLatest to save GetLatestVimScripts.vim's <DIR>
    " | |
    " | `-- GetLatestVimScripts.dat to store GLVS's items
    " |
    " |-- _vim_fav_files to store favmenu.vim's items
    " `-- _vim_mru_files to store mru.vim's items
    "
    " 2. Get all scripts you favor on www.vim.org better with GLVS
    "
    " 3. Get all needed utilities, especially on Windows, e.g.
    " wget -- WGET for Windows (win32) http://users.ugent.be/~bpuype/wget/
    " ctags -- Exuberant Ctags http://ctags.sf.net/
    "
    " 4. If you have no idea of some option, just press K (<S-k>) on it.
    "
    " 5. HTML file is produced with :TOhtml, with colo default.
    "
    "============================================================================
    "----------------------------------------------------------------------------
    " TODO List
    "----------------------------------------------------------------------------
    " 1. Install Chinese Vim help helplang -> cn

    "----------------------------------------------------------------------------
    " general
    "----------------------------------------------------------------------------
    set nocompatible " use vim as vim, should be put at the very start
    set history=1000 " lines of Ex-mode commands, search history
    set browsedir=buffer " use directory of the related buffer for file browser
    set clipboard+=unnamed " use clipboard register '*' for all y, d, c, p ops
    set viminfo+=! " make sure it can save viminfo
    set isk+=$,%,#,- " none of these should be word dividers
    set confirm " raise a dialog confirm whether to save changed buffer
    set ffs=unix,dos,mac " favor unix ff, which behaves good under bot Winz & Lin
    set fenc=utf-8 " default fileencoding
    set fencs=utf-8,ucs-bom,euc-jp,gb18030,gbk,gb2312,cp936
    map Q gq
    " don't use Ex-mode, use Q for formatting
```

```
filetype on " enable file type detection
filetype plugin on " enable loading the plugin for appropriate file type

"----------------------------------------------------------------------------
" platform dependent
"----------------------------------------------------------------------------
if has("win32")
let $VIMDATA = $VIM.'/vimdata'
let $VIMFILES = $VIM.'/vimfiles'
let PYTHON_BIN_PATH = 'd:/foo/python/python.exe' " ensure the path right
else
let $VIMDATA = $HOME.'/vimdata'
let $VIMFILES = $HOME.'/.vim'
let PYTHON_BIN_PATH = '/usr/bin/python'
set guifont=courier\ 10
endif

"----------------------------------------------------------------------------
" path/backup
"----------------------------------------------------------------------------
set backup " make backup file and leave it around
set backupdir=$VIMDATA/backup " where to put backup file
set directory=$VIMDATA/temp " where to put swap file
set runtimepath+=$VIMDATA " add this path to rtp, support GetLatestVimScripts
set path=.,/usr/include/*,, " where gf, ^Wf, :find will search
set tags=./tags,tags,$VIMRUNTIME/doc/tags,$VIMFILES/doc/tags " tags files CTR
set makeef=error.err " the errorfile for :make and :grep

"----------------------------------------------------------------------------
" colors
"----------------------------------------------------------------------------
set background=dark " use a dark background
syntax on " syntax highlighting

"----------------------------------------------------------------------------
" gui-only settings
"----------------------------------------------------------------------------
if has("gui_running")
colo inkpot " colorscheme, inkpot.vim
set lines=40 " window tall and wide, only if gui_running,
set columns=120 " or vim under console behaves badly
endif

"----------------------------------------------------------------------------
" Vim UI
"----------------------------------------------------------------------------
set linespace=1 " space it out a little more (easier to read)
set wildmenu " type :h and press <Tab> to look what happens
set ruler " always show current position along the bottom
set cmdheight=2 " use 2 screen lines for command-line
set lazyredraw " do not redraw while executing macros (much faster)
set nonumber " don't print line number
set hid " allow to change buffer without saving
set backspace=2 " make backspace work normal
set whichwrap+=<,>,h,l " allow backspace and cursor keys to wrap
set mouse=a " use mouse in all modes
set shortmess=atI " shorten messages to avoid 'press a key' prompt
set report=0 " tell us when anything is changed via :...
set fillchars=vert:\ ,stl:\ ,stlnc:\
" make the splitters between windows be blank

"----------------------------------------------------------------------------
" visual cues
"----------------------------------------------------------------------------
set showmatch " show matching paren
set matchtime=5 " 1/10 second to show the matching paren
set nohlsearch " do not highlight searched for phrases
set incsearch " BUT do highlight where the so far typed pattern matches
set scrolloff=10 " minimal number of screen lines to keep above/below the cur
set novisualbell " use visual bell instead of beeping
set noerrorbells " do not make noise
set laststatus=2 " always show the status line
set listchars=tab:\|\ ,trail:.,extends:>,precedes:<,eol:$ " what to show whil
set statusline=%{VimBuddy()}\ %F%m%r%h%w\ [%{&ff}]\ [%Y]\ [\%03.3b\ \%02.2B]\

"----------------------------------------------------------------------------
" text formatting/layout
"----------------------------------------------------------------------------
set ai " autoindent
set si " smartindent
set cindent " do C-style indenting
set fo=tcrqn " see help (complex)
set tabstop=4 " tab spacing (settings below are just to unify it)
set softtabstop=4 " unify
set shiftwidth=4 " unify
set noexpandtab " real tabs please!
set smarttab " use tabs at the start of a line, spaces elsewhere
set nowrap " do not wrap lines
set formatoptions+=mM " so that vim can reformat multibyte text (e.g. Chinese

"----------------------------------------------------------------------------
" folding
"----------------------------------------------------------------------------
set foldenable " turn on folding
set foldmethod=indent " make folding indent sensitive
set foldlevel=100 " don't autofold anything (but I can still fold manually)
set foldopen -=search " don't open folds when you search into them
set foldopen -=undo " don't open folds when you undo stuff

"----------------------------------------------------------------------------
" plugin - a.vim
"----------------------------------------------------------------------------
" alternate files quickly (.c --> .h etc)

"----------------------------------------------------------------------------
" plugin - c.vim
```

```
″-----------------------------------------------------------------------------
″set makeprg=g++\ %
let g:C_AuthorName = 'Leal’
let g:C_Email = 'linxiao.li NOSPAM gmail DOT com’

″-----------------------------------------------------------------------------
″ plugin - runscript.vim (for Python)
″-----------------------------------------------------------------------------
″let PYTHON_BIN_PATH = ...

″-----------------------------------------------------------------------------
″ plugin - calendar.vim
″-----------------------------------------------------------------------------
let g:calendar_diary = $VIMDATA.'/diary’ ″ where to store your diary

″-----------------------------------------------------------------------------
″ plugin - mru.vim (most recently used files)
″-----------------------------------------------------------------------------
let MRU_File = $VIMDATA.'/_vim_mru_files’ ″ which file to save mru entries
let MRU_Max_Entries = 20 ″ max mru entries in _vim_mru_files

″-----------------------------------------------------------------------------
″ plugin - favmenu.vim
″-----------------------------------------------------------------------------
let FAV_File = $VIMDATA.'/_vim_fav_files’ ″ which file to save favorite entri

″-----------------------------------------------------------------------------
″ plugin - minibufexpl.vim
″-----------------------------------------------------------------------------
let g:miniBufExplTabWrap = 1 ″ make tabs show complete (no broken on two line
let g:miniBufExplModSelTarget = 1

″-----------------------------------------------------------------------------
″ plugin - taglist.vim
″-----------------------------------------------------------------------------
if has(″win32″)
let Tlist_Ctags_Cmd = $VIMFILES.'/ctags.exe’ ″ location of ctags tool
else
let Tlist_Ctags_Cmd = '/usr/local/bin/ctags’
endif

let Tlist_Sort_Type = ″name″ ″ order by
let Tlist_Use_Right_Window = 1 ″ split to the right side of the screen
let Tlist_Compart_Format = 1 ″ show small meny
let Tlist_Exist_OnlyWindow = 1 ″ if you are the last, kill yourself
let Tlist_File_Fold_Auto_Close = 0 ″ do not close tags for other files
let Tlist_Enable_Fold_Column = 0 ″ do not show folding tree

″-----------------------------------------------------------------------------
″ plugin - matchit.vim
″-----------------------------------------------------------------------------
let b:match_ignorecase = 1
```

```
"---------------------------------------------------------------------------
" plugin - supertab.vim
"---------------------------------------------------------------------------
" <Tab> has been mapped to SuperTab() func in that plugin

"---------------------------------------------------------------------------
" plugin - timestamp.vim
"---------------------------------------------------------------------------
let g:timestamp_regexp = '\v\C%(<Last %([cC]hanged?|[Mm]odified):\s+)@<=.*$’
let g:timestamp_rep = '%w %d/%m/%Y %H:%M:%S #u@#h:#f’
augroup TimeStampHtml
au filetype html let b:timestamp_regexp = '\v\C%(\<!-- Timestamp:\s{-1,})@<=.
au filetype html let b:timestamp_rep = '%a %d/%m/%Y %r #u@#h:#f’
augroup END

"---------------------------------------------------------------------------
" plugin - perl-support.vim
"---------------------------------------------------------------------------
let g:Perl_AuthorName = 'Leal’
let g:Perl_Email = 'linxiao.li NOSPAM gmail DOT com’
let tlist_perl_settings = 'perl;c:constants;l:labels;s:subroutines;d:POD’

"---------------------------------------------------------------------------
" utilities
"---------------------------------------------------------------------------
" select range, then hit :SuperRetab($width) - by p0g and FallingCow
fu! SuperRetab(width) range
sil! exe a:firstline . ',' . a:lastline . 's/\v%(^ *)@<= {'. a:width .'}/\t/g
endf

" bind :CD to :cd %:h, then change cwd to the dir that includes current file
sil! com -nargs=0 CD exe 'cd %:h’

"TODO just do it
fu! AddLineNo(isVM)
if(a:isVM == 1)
sil! exe 's/^/\=' . strpart((line('.')-line("'<")+1)." ", 0, 4)
else
sil! exe '%s/^/\=' . strpart(line('.')." ", 0, 4)
endif
endf
" add line number befor each line of the text
" :g/^/exec "s/^/".strpart(line(".")." ", 0, 4)
" :%s/^/\=strpart(line('.')." ", 0, 4)
" visual mode
" :s/^/\=strpart((line('.')-line("'<")+1)." ", 0, 4)

" XML support, e.g. element,,,<CR> -> <element> </element>
" Bart van Deenen , www.vandeenensupport.com
fu! MakeElement()
if match(getline('.'),'^\s*>\s*$') == -1
"the deleted word was not alone on the line
let @w = "i< pla</ pa> F<i"
```

```
else
"the deleted word was on it's own on the line
let @w = "i< po</ pa> kA"
endif
endf

" include colon(5Cool for namespaces in xsl for instance
"setlocal iskeyword=@,48-57,_,192-255,58
inoremap <buffer> ,,, ><Esc>db:call MakeElement()<enter>@w

"----------------------------------------------------------------------------
" mappings
"----------------------------------------------------------------------------
map <right> <ESC>:MBEbn<RETURN>
" -> switches buffers
map <left> <ESC>:MBEbp<RETURN>
" <- switches buffers
map <up> <ESC>:Sex<RETURN><ESC><C-W><C-W>
" up arrow to bring up a file explorer
map <down> <ESC>:Tlist<RETURN>
" down arrow to bring up the taglist
map <A-i> i <ESC>r
" Alt-i inserts a single char, and back to normal
map <F3> <ESC>ggVG:call SuperRetab()<left>
map <F4> ggVGg?
" Rot13 encode the current file

"noremap <silent> <C-F11> :cal VimCommanderToggle()<CR>

" plugin - php_console.vim
"map <F5> :call ParsePhpFile()<cr> " call function in normal mode
"imap <F5> <ESC>:call ParsePhpFile()<cr> " call function in insert mode

"----------------------------------------------------------------------------
" autocommands
"----------------------------------------------------------------------------
au BufEnter * :syntax sync fromstart " ensure every file does syntax highligh
au BufNewFile,BufRead *.asp :set ft=jscript " all my .asp files ARE jscript
au BufNewFile,BufRead *.tpl :set ft=html " all my .tpl files ARE html

" ftplugin - python_fold - $VIMFILES/ftplugin/python_fold.vim
" add names in pydiction to autocomplete class or object's class, attribute c
au FileType python set complete+=k$VIMFILES/dict/pydiction isk+=.,(
"au FileType python pyfile $VIMFILES/plugin/pyCallTips.py
au FileType python source $VIMFILES/plugin/python.vim
"au FileType java source $VIMFILES/ftplugin/JavaRun.vim
"au FileType php set complete+=k$VIMFILES/dict/php.dict isk+=.,(
au FileType text setlocal textwidth=78

"----------------------------------------------------------------------------
" highlight active line in normal mode, Vim7 don't need this
"----------------------------------------------------------------------------
"highlight CurrentLine guibg=darkgrey guifg=white ctermbg=darkgrey ctermfg=wh
```

```
"au! Cursorhold * exe 'match CurrentLine /\%' . line('.') . 'l.*/'
"set ut=19

"----------------------------------------------------------------------------
" stuffs I don't like
"----------------------------------------------------------------------------
"set ignorecase -- turns out, I like case sensitivity
"set list -- turns out, do not display unprintable characters such as Tab
"autocmd GUIEnter * :simalt ~x -- having it auto maximize the screen is annoy
"autocmd BufEnter * :lcd %:p:h -- switch to current dir (breaks some scripts)

"----------------------------------------------------------------------------
" useful abbrevs
"----------------------------------------------------------------------------
iab xasp <%@language=jscript%><CR><%<CR><TAB><CR><BS>%><ESC><<O<TAB>
iab xdate <c-r>=strftime("%m/%d/%y %H:%M:%S")<cr>

"----------------------------------------------------------------------------
" customize cursor color to indicate IM is on
"----------------------------------------------------------------------------
if has('multi_byte_ime')
hi Cursor guifg=NONE guibg=Green
hi CursorIM guifg=NONE guibg=Blue
endif

"----------------------------------------------------------------------------
" TVO defaults - otl.vim
"----------------------------------------------------------------------------
let otl_install_menu =1
let no_otl_maps =0
let no_otl_insert_maps =0

let otl_bold_headers =0
let otl_use_thlnk =0

let g:otl_use_viki =0
let maplocalleader =","

map <c-w><c-f> :FirstExplorerWindow<cr>
map <c-w><c-b> :BottomExplorerWindow<cr>
map <c-w><c-t> :WMToggle<cr>

let g:winManagerWidth = 35
let g:winManagerWindowLayout = 'TodoList'

let g:tskelDir = $VIMFILES."/skeletons"

"----------------------------------------------------------------------------
" vim: set et ft=vim tw=98 path+=$VIMFILES/*:
```