# 下载安装

## 下载Vim

同时提供32bit和64bit下载地址：

<https://github.com/vim/vim-win32-installer/releases>

我的配置文件：



# 使用模式

## 编辑模式

编辑模式共有 6 种方式：crosia 克罗西亚：

|  |  |
| --- | --- |
| c | 修改 |
| r | 替换 |
| o | 打开 |
| s | 替换 |
| i | 插入 |
| a | 附加 |

参考资料：

<http://www.manongjc.com/article/51057.html>

<https://www.runoob.com/linux/linux-vim.html>

<https://zhuanlan.zhihu.com/vim-video>

Vim 从入门到精通：  
<https://github.com/wsdjeg/vim-galore-zh_cn/blob/master/README.md>

# 命令模式

## 重新载入配置文件

我们在编辑完vimrc配置文件的时候，希望不关闭vim并实现重新载入配置：

:source ~/\_vimrc

也可以通过快捷键映射的方式来实现：

# 配置文件

## 配置文件地址

在Windows中，配置文件地址为：C:\Users\Administrator\\_vimrc

在Linux中，配置文件地址为：

## 打开配置文件

启动Vim之后，用命令模式打开配置文件，如下命令所示：

在Windows中，:edit ~\\_vimrc

在Linux中，:edit ~/.vimrc

## 配置项说明

### 显示行号

set number

或

set nu

#### 取消显示行号

set nonu

或者

set nonumber

### 语法高亮

syntax on

### 自动缩进

set autoindent

### 设置tab宽度

set shiftwidth=4

set tabstop=4

### 把tab转换成空格

set expandtab

### 颜色主题

|  |
| --- |
| colorscheme murphy |

在命令模式下输入：

:colorscheme空格

然后按tab键，选择喜欢的颜色主题

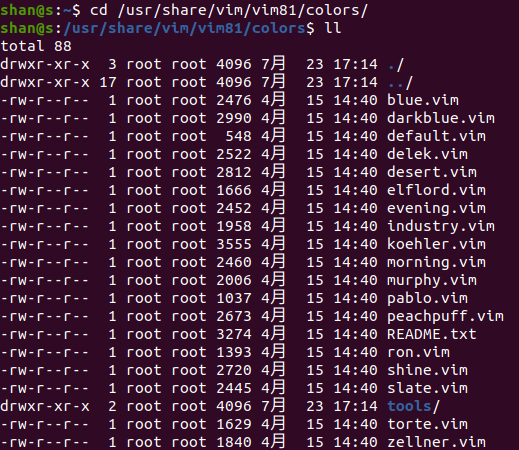
#### 颜色主题位置

在Windows系统中，颜色主题位置为：安装目录下\vim82\colors目录中。

比如：C:\Vim\vim82\colors

在Linux系统中，颜色主题位置为：/usr/share/vim/vim81/colors/

比如：



#### 安装 onedark 主题

打开 <https://github.com/joshdick/onedark.vim> ，下载，或者用 git 下载，然后把 **autoload** 目录下的 onedark.vim 复制到 Vim 软件的 autoload目录下，以及把 **colors** 目录下的 onedark.vim 复制到 Vim 软件的 colors目录下，经过这两个步骤才能完成安装。

### 设置编码

|  |
| --- |
| set encoding=utf-8  set fileencodings=utf-8,gb18030,gbk,gb2312  set termencoding=utf-8  set fileencodings=utf-8  source $VIMRUNTIME/delmenu.vim  source $VIMRUNTIME/menu.vim  language messages zh\_CN.utf-8 |

### 设置文本字体和大小

|  |
| --- |
| " 设置字体和大小  " set guifont=Courier\_new:h14:b:cDEFAULT  set guifont=Consolas:h14:cDEFAULT |

### 显示不可见字符

|  |
| --- |
| " 设置空白字符的视觉提示  set list listchars=tab:>-,trail:~,extends:>,precedes:< |

### 文件类型检测

filetype plugin indent on

### 标点符号自动闭合脚本

|  |
| --- |
| " 自动括号  inoremap ' ''<ESC>i  inoremap " ""<ESC>i  inoremap ( ()<ESC>i  inoremap [ []<ESC>i  inoremap { {<CR>}<ESC>O<tab> |

### 配置编程语言文件头

|  |
| --- |
| "创建文件头  autocmd BufNewFile \*.py,\*.tex exec ":call SetTitle()"  func! SetTitle()  if &filetype == 'python'  call setline(1,"#!/usr/bin/env python3")  call append(line("."),"# -\*- coding:UTF-8 -\*-")  call append(line(".")+1,"##########################################################################")  call append(line(".")+2, "# File Name: ".expand("%"))  call append(line(".")+3, "# Author: stubborn vegeta")  call append(line(".")+4, "# Created Time: ".strftime("%c"))  call append(line(".")+5, "##########################################################################")  endif  if &filetype == 'plaintex'  call setline(1,"% -\*- coding:UTF-8 -\*-")  call append(line("."),"%#########################################################################")  call append(line(".")+1, "% File Name: ".expand("%"))  call append(line(".")+2, "% Author: stubborn vegeta")  call append(line(".")+3, "% Created Time: ".strftime("%c"))  call append(line(".")+4, "%#########################################################################")  endif  normal Go  endfunc |

### 设置背景透明度

打开<https://github.com/mattn/vimtweak> 进行下载。

把 vimtweak64.dll复制到gvim.exe所在目录。

然后在配置文件中添加如下代码：

|  |
| --- |
| au GUIEnter \* call libcallnr("vimtweak64.dll","SetAlpha",180) |

这样就可以实现半透明效果了。

### 代码运行环境

<https://www.cnblogs.com/juneman/p/8592776.html>

# 插件管理

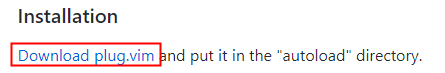
在这里我们将使用vim-plug这款插件管理器，它支持异步加载。

在安装任何插件之前，我们需要先安装插件管理器，这里是vim-plug插件管理器。

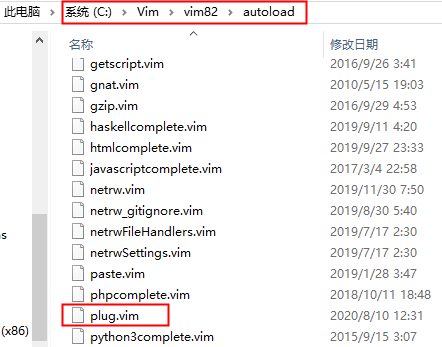
## 安装插件管理器

vim-plug官网地址为：<https://github.com/junegunn/vim-plug>

下载plug.vim文件



下载下来之后，保存到Vim安装目录下的vim82\autoload目录中，如下图所示：



这样就算是安装完成了。

然后重启Vim编辑器。

## 插件管理代码

创建插件保存的相关目录结构，如下所示：

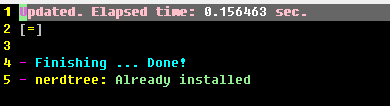
D:\ProgramConfigs\Vim\vim82\plugged

如果要使用Vim相关插件，则需要在配置文件中添加如下代码：

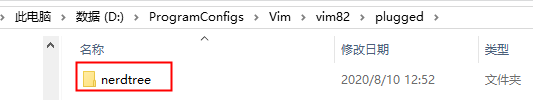
|  |
| --- |
| " 双引号是注释语句  " 插件管理器  call plug#begin('D:\ProgramConfigs\Vim\vim82\plugged') " 插件保存目录  Plug 'preservim/nerdtree'  call plug#end() |

然后执行 :w 进行保存，然后执行 :PlugInstall

安装过程如下所示：



结果如下图所示：



插件安装到我们指定的目录中了。

## 插件操作命令

检查插件状态

:PlugStatus

安装插件

:PlugInstall

更新插件

:PlugUpdate

查看更改

:PlugDiff

删除插件

把插件代码注释掉，然后执行如下命令

:PlugClean

升级vim-plug插件管理器

:PlugUpgrade

## 常用插件说明

### nerdtree

官方地址：

<https://github.com/preservim/nerdtree>

Plug 'preservim/nerdtree'

### vim-airline

<https://github.com/vim-airline/vim-airline>

Plug 'vim-airline/vim-airline' " 状态条

Plug 'vim-airline/vim-airline-themes' " 状态条主题

效果如下图所示：



#### 状态栏乱码问题

安装完vim-airline后初次打开vim状态栏可能会出现乱码和部分符号不能正常显示（状态栏箭头不显示）等问题。这种情况一般是编码和字体设置引起的，我们所使用的字体没有airline配置的符号，因此airline状态栏上无法正常显示图标和三角形箭头符号。

可以在<https://github.com/powerline/fonts>下载使用powerline打过补丁的字体，里面也有安装方法。下载字体后，Linux系统上直接执行 ./install.sh 文件来安装字体（注意当前工作目录），Windows系统上选择所需要的字体双击打开，然后点击安装。

我安装的字体是 [Powerline Consolas.ttf](https://github.com/runsisi/consolas-font-for-powerline) （下载地址），安装之后打开vim输入命令 :set guifont=\* 回车即可跳出vim字体选择窗口，然后选择该字体设置。当然，如果要使用该字体，还是要写在.vimrc文件中。

我们直接在配置文件中添加如下代码即可：

|  |
| --- |
| set guifont=Powerline\_Consolas:h14:cDEFAULT |

### YouCompleteMe – 自动补全

<https://github.com/ycm-core/YouCompleteMe>

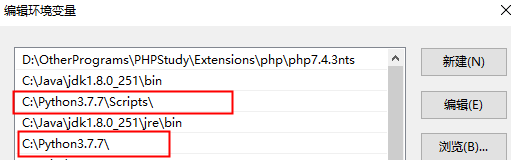
查看插件官网，可以看到如下文字：

|  |
| --- |
| The following additional language support options are available:  **C# support**: install Mono with Homebrew or by downloading the Mono macOS package and add --cs-completer when calling install.py.  **Go support**: install Go and add --go-completer when calling install.py.  **JavaScript and TypeScript support**: install Node.js and npm and add --ts-completer when calling install.py.  **Rust support**: add --rust-completer when calling install.py.  **Java support**: install JDK8 (version 8 required) and add --java-completer when calling install.py. |
| 各编程语言对应的安装命令。 |

在安装该插件之前，我们需要先安装如下依赖：

1、安装Python3.7及以上版本，并把Python配置到环境变量。

C:\Python3.7.7



2、安装Git，并配置到环境变量。

C:\Git

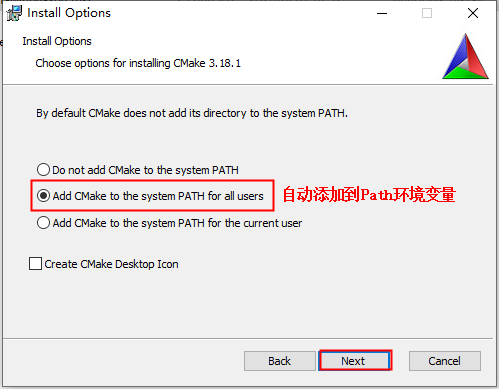


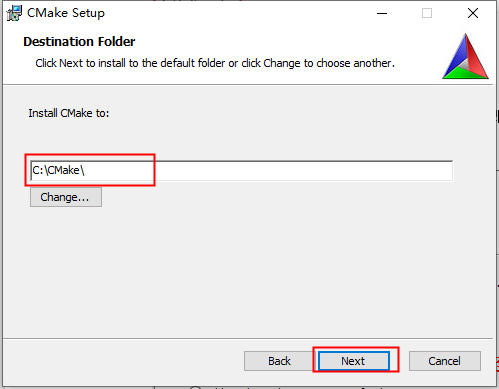
3、安装CMake

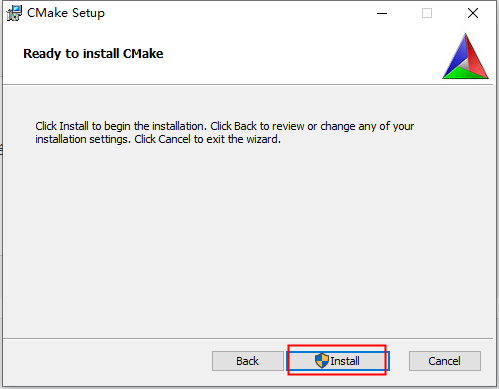
<https://cmake.org/>

下载“[Windows win64-x64 Installer](https://github.com/Kitware/CMake/releases/download/v3.18.1/cmake-3.18.1-win64-x64.msi)”版本，即 [cmake-3.18.1-win64-x64.msi](https://github.com/Kitware/CMake/releases/download/v3.18.1/cmake-3.18.1-win64-x64.msi)

下载下来之后进行安装即可，安装过程如下：







安装完成之后，自动加入Path环境变量，如下图所示：

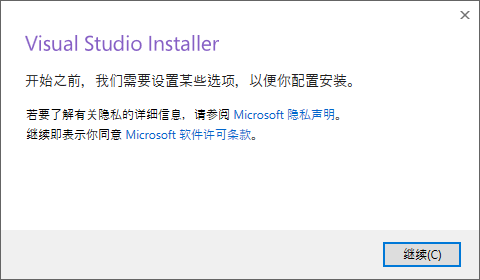


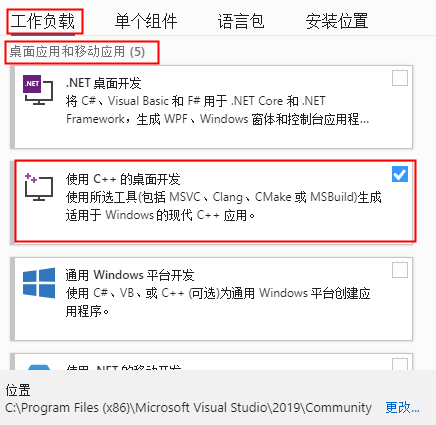
4、安装“Visual Studio 社区版本”

打开“[Download Visual Studio 2019 for Windows & Mac](https://visualstudio.microsoft.com/zh-hans/downloads/)”，选择社区进行下载：（跟进YouCompleteMe网址的说明下载相应版本的社区版。）

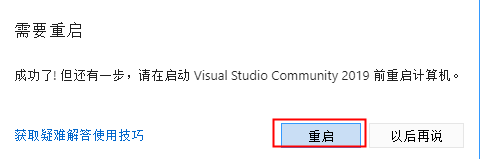


安装过程如下：





然后进行安装，安装过程比较漫长，需要慢慢等待。



安装完成之后，需要重启操作系统。

4、安装LLVM软件

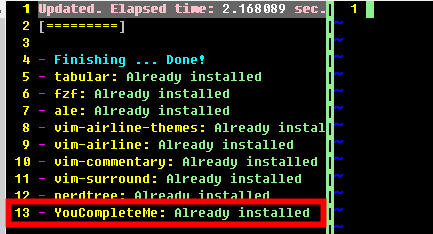
<https://github.com/llvm/llvm-project>

<https://github.com/llvm/llvm-project/releases/download/llvmorg-10.0.0/LLVM-10.0.0-win64.exe>

5、安装插件YouCompleteMe插件

Plug 'ycm-core/YouCompleteMe'

安装结果如下：



6、编译YouCompleteMe插件

在D:\ProgramConfigs\Vim\vim82\plugged\YouCompleteMe目录下，打开命令行工具，然后执行如下命令：

python install.py --clang-completer

提示如下错误：

|  |
| --- |
| File D:\ProgramConfigs\Vim\vim82\plugged\YouCompleteMe\third\_party\ycmd\build.py does not exist; you probably forgot to run  git submodule update --init --recursive |

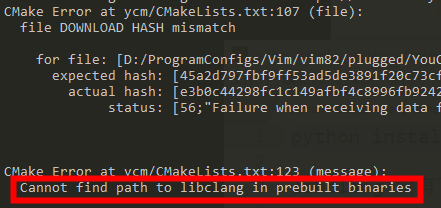
需要先执行如下命令：

|  |
| --- |
| git submodule update --init --recursive |

安装子模块，然后再执行安装命令：

|  |
| --- |
| python install.py --clangd-completer --java-completer --go-completer --cs-completer --ts-completer |
| 安装c、java、go的代码提示 |
| 会下载java、go相关库文件。 |
| 安装go的话，需要配置go的代理，否则下载非常慢。 |
|  |

出现如下错误:

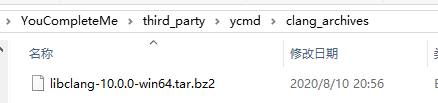


意思是libclang 没有下载成功。

下载地址为:

<https://dl.bintray.com/ycm-core/libclang/libclang-10.0.0-win64.tar.bz2>

然后保存到 YouCompleteMe\third\_party\ycmd\clang\_archives\ 的目录下，如下图所示：



再次执行：

python install.py --clang-completer

提示：“可能丢失数据”，这个可以忽略不计。没有其它错误则说明安装成功了。

如果发现“Downloading jdt.ls from <http://download.eclipse.org/jdtls/snapshots/jdt-language-server-0.57.0-202006172108.tar.gz>”下载很慢，则手工进行下载，然后保存到如下目录结构中：

\YouCompleteMe\third\_party\ycmd\third\_party\eclipse.jdt.ls\target\cache

然后中断命令，再次重新执行。

7、配置ycm\_extra\_conf.py路径

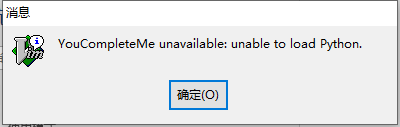
打开Vim配置文件，添加如下配置：

|  |
| --- |
| let g:ycm\_server\_python\_interpreter='C:\Python3.7.7\python.exe'  let g:ycm\_global\_ycm\_extra\_conf='D:\ProgramConfigs\Vim\vim82\plugged\YouCompleteMe\third\_party\ycmd\.ycm\_extra\_conf.py' |

完成YouCompleteMe的全部安装步骤。

8、YouCompleteMe unavailable: unable to load Python

启动时，提示如下错误信息：



测试该插件：

#### 各种语言代码提示

##### JavaScript 和 TypeScript

##### Java

##### C 和 C++

c 文件后缀为 .c

c++ 文件后缀为 .cpp

##### C#

根据 <https://github.com/ycm-core/YouCompleteMe> 文档说明：

|  |
| --- |
| **C# support**: install [Mono](https://www.mono-project.com/download/stable/#download-lin) and add --cs-completer when calling install.py. |

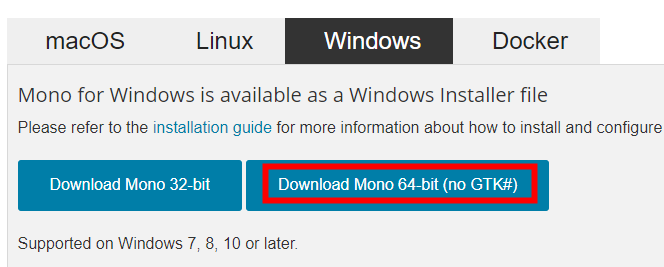
根据说明：如果要支持 C# 代码提示，需要先安装 Mono 软件，然后再通过 --cs-completer 完成安装。

打开Mono 下载地址页面（使用Chrome打开页面才能正常显示）：

<https://www.mono-project.com/download/stable/#download-lin>

查找：Mono for Windows is available as a Windows Installer file

然后找到[Download Mono 64-bit (no GTK#)](https://download.mono-project.com/archive/6.10.0/windows-installer/mono-6.10.0.104-x64-0.msi)下载链接进行下载，如下图所示：



下载下来之后，直接进行安装即可。

安装完成之后，在开始菜单中，找到Mono命令行，如下图所示：



在上图中，提示把Mono的bin目录添加到Path变量，所以，我们在Path环境变量中进行配置，如下图所示：



配置完成之后，我们进行下一步操作：

在YouCompleteMe目录下执行 python install.py --cs-completer命令。

##### Go

##### Python

默认支持。

#### 代码补全操作

|  |
| --- |
| let g:ycm\_min\_num\_identifier\_candidate\_chars=2 |

输入两个字母时，自动弹出代码提示。

|  |
| --- |
| let g:ycm\_key\_invoke\_completion='<c-z>' |

按住 Ctrl + Z 即可弹出代码提示。

#### 支持 PHP 代码提示

### coc – 新一代自动补全

coc项目地址为：

<https://github.com/neoclide/coc.nvim>

在安装coc之前，我们需要先安装node.js>=10.12，然后使用vim-plug做为插件管理器，则安装coc的方式如下：

|  |
| --- |
| Plug 'neoclide/coc.nvim', {'branch': 'release'} |

然后在Vim中执行命令模式: :PlugInstall

这样coc插件就安装完成了。

本地模块安装地址：

C:\Users\Administrator\AppData\Local\coc\extensions\node\_modules

安装完成之后，我们需要进行配置，对需要代码提示的编程语言进行配置。

|  |
| --- |
| :CocInstall coc-clangd # C++环境插件  :CocInstall coc-cmake # Cmake 支持  :CocInstall coc-emmet  :CocInstall coc-git # git 支持  :CocInstall coc-highlight # 高亮支持  :CocInstall coc-jedi # jedi  :CocInstall coc-json # json 文件支持  :CocInstall coc-python # python 环境支持  :CocInstall coc-sh # bash 环境支持  :CocInstall coc-snippets # python提供 snippets  :CocInstall coc-vimlsp # lsp  :CocInstall coc-yaml # yaml  :CocInstall coc-syntax  :CocInstall coc-pairs |

查看所有支持列表：<https://github.com/neoclide/coc.nvim/wiki/Using-coc-extensions> 找到 “Implemented coc extensions”即可看到。

添加语言支持：

<https://github.com/neoclide/coc.nvim/wiki/Language-servers>

如何配置 coc 模块安装路径？？？？

#### php 代码提示

:CocInstall coc-phpls

即可。

#### python 代码提示

:CocInstall coc-python

python -m pip install --upgrade pip -i https://pypi.doubanio.com/simple/

pip install -i https://pypi.doubanio.com/simple/ pylint

pip install -i https://pypi.doubanio.com/simple/ jedi

#### go 代码提示

gopls安装说明：

<https://github.com/golang/tools/blob/master/gopls/doc/user.md>

go get golang.org/x/tools/gopls@latest

#### java 代码提示

#### c 和 c++ 代码提示

#### c# 代码提示

#### javascript 和 typescript 代码提示

#### json 代码提示

coc-json

#### css 代码提示

coc-css

#### html 代码提示

coc-html

### fzf – 文件搜索器

<https://github.com/junegunn/fzf>

Plug 'junegunn/fzf'

### tabular – 代码格式化

<https://github.com/godlygeek/tabular>

Plug 'godlygeek/tabular'

### vim-commentary - 代码注释

<https://github.com/tpope/vim-commentary>

Plug 'tpope/vim-commentary'

### ale – 语法检查

<https://github.com/dense-analysis/ale>

Plug 'dense-analysis/ale'

### vim-surround – 删除引号内的内容

<https://github.com/tpope/vim-surround>

Plug 'tpope/vim-surround'