Mac 下配置 Vim 代码补全：YouCompleteMe (/a/1190000002793897)

冰雪殿 (/u/iceflying) **336** 5月25日 发布

引言

Vim 无疑是世界上最好用的编辑器之一（为了不引起战争 →_→）。在广大程序员用 Vim 敲代码的过程中，代码补全功能能够大大提高生产力，尤其是对于从各种 IDE 转到 Vim 的程序员来说更是福音般地存在。本文将介绍一种推荐的代码补全工具 YouCompleteMe，并且一步步介绍它的安装方式。

YouCompleteMe

YouCompleteMe (http://valloric.github.io/YouCompleteMe) 是一个比较完备，并且正在日渐完备的 Vim 代码补全插件。它的功能十分强大，支持自动补全的语言包括：

- C/C++/Objective-C/Objective-C++ (基于 Clang (http://clang.llvm.org/))
- Python (基于 Jedi (https://github.com/davidhalter/jedi))
- C# (基于 OmniSharp (https://github.com/nosami/OmniSharpServer))
- Go (基于 GoCode (https://github.com/nsf/gocode))
- 其他 Vim 的 omnicomplete system 支持的语言，比如 (Ruby, PHP 等)

先上一张作者给的 demo 效果图。

```
int LongestCommonSubsequenceLength( const std::string &first,
                                    const std::string &second ) {
  const std::string &longer  = first.size() > second.size() ? first  : second;
  const std::string &shorter = first.size() > second.size() ? second : first;

  int longer_len  = longer.size();
  int shorter_len = shorter.size();

  std::vector<int> previous( shorter_len + 1, 0 );
  std::vector<int> current(  shorter_len + 1, 0 );
  int foo = previous.size() + current.size()
    LongestCommonSubsequenceLength
    LongestCommonSubsequenceLength
    LongestCommonSubsequenceLength
    length
  f LongestCommonSubsequenceLength ++i ) {
    for ( int j = 0; j < shorter_len; ++j ) {
      if ( toupper( longer[ i ] ) == toupper( shorter[ j ] ) )
        current[ j + 1 ] = previous[ j ] + 1;
      else
        current[ j + 1 ] = std::max( current[ j ], previous[ j + 1 ] );
    }

    for ( int j = 0; j < shorter_len; ++j ) {
      previous[ j + 1 ] = current[ j + 1 ];
    }
  }
```

安装

接下来介绍安装过程。

确定 Vim 版本满足支持

如果你用的是 Mac OS X 自带的 vim 的话，那么肯定是不能满足需求的。首先，需要升级 vim。这里建议安装 macvim (https://github.com/macvim-dev/macvim)，当然 vim 的官网 (http://www.vim.org/download.php#mac) 也是这么推荐的。推荐使用 brew (http://brew.sh) 来安装。

1. MacVim 依赖 Xcode，首先需要在 App Store 中安装 Xcode。如果是全新安装的 Xcode，请打开一次，并且同意 license。
2. 使用 brew 安装 MacVim

```
brew install macvim
```

3. 使用 MacVim 替换系统自带的 Vim，在当前 shell 的配置文件中添加

```
alias vim='mvim -v'
```

安装 Vundle (https://github.com/gmarik/Vundle.vim)

Vundle (缩写自 `Vim bundle`) 是一个很方便的 Vim 插件管理器。它的使用方法很简单，安装一个插件只需要在 `.vimrc` 按照规则中添加 Plugin 的名称，某些需要添加路径，之后在 Vim 中使用 `:PluginInstall` 既可以自动化安装。具体的使用过方法详见官网 (https://github.com/gmarik/Vundle.vim)。

下面将介绍 Vundle 的安装及基本配置。

1. 使用 git 克隆 Vundle 工程到本地。

```
git clone https://github.com/gmarik/Vundle.vim.git ~/.vim/bundle/Vundle.vim
```

2. 修改 `.vimrc` 配置 Plugins。在 `.vimrc` 文件中添加如下内容。

```
set nocompatible
filetype off

set rtp+=~/.vim/bundle/Vundle.vim
call vundle#begin()

Plugin 'gmarik/Vundle.vim'

call vundle#end()
filetype plugin indent on
```

安装 YouCompleteMe

接下来将要安装的是我们的主角，YouCompleteMe。解决了上面的依赖软件之后，安装它将变得非常简单。

1. 在 `.vimrc` 中添加如下内容。位置在 `call vundle#begin()` 和 `call vundle#end()` 之间。

```
Bundle 'Valloric/YouCompleteMe'
```

2. 编译 YouCompleteMe

   首先说明，编译过程需要 `CMake`，可以使用 `brew` 来安装。

```
brew install CMake
```

   - 带 C-family languages 语义支持的版本

```
cd ~/.vim/bundle/YouCompleteMe
./install.sh --clang-completer
```

   - 不带 C-family languages 语义支持的版本

```
cd ~/.vim/bundle/YouCompleteMe
./install.sh --clang-completer
```

   - 带 C# 语义支持的版本

```
cd ~/.vim/bundle/YouCompleteMe
./install.sh --omnisharp-completer
```

   - 带 Go 语言语义支持的版本

```
cd ~/.vim/bundle/YouCompleteMe
./install.sh --gocode-completer
```

完成

至此，我们已经拥有了 YouCompleteMe 这款自动补全神器。请尽情享用。

Just enjoy it !

vim (/t/vim/blogs) macvim (/t/macvim/blogs)
代码补全 (/t/%E4%BB%A3%E7%A0%81%E8%A1%A5%E5%85%A8/blogs)
youcompleteme (/t/youcompleteme/blogs)

链接 (/a/1190000002793897) 更多

你可能感兴趣的文章

Xcode上的VIM (/a/1190000000667223) 875 浏览

vim (/a/1190000003851913) 80 浏览

使用vim-jedi插件自动补全django的蹩脚方法 (/a/1190000000405249) 2.7k 浏览

讨论区

很详细

#1 (/c/1050000003076584) **ShaoXin** (/u/shaoxin) · 8月11日 · 回复

请先 登录 后评论

本文隶属于专栏

冰雪之都 (/blog/iceflying)

关注专栏

分享扩散:

http://segmentfault.com/a/1190000002793897 缩短