

版本: V1.43.2

一、安装插件

1. 编程语言类
C/C++、**Python**、**Arduino**、markdownlint
2. 智能编程类(错误检测、提示、自动补全等)
C++ Intellisense、**Code Spell Checker**、**Include Autocomplete**、koroFileHeader、**Anaconda Extension Pack**、**Auto Import**、TODO Highlight、**Codelf**
3. 编译运行类
Code-runner、**Cmake**、**Cmake Tools**(还要再电脑端下载 **Cmake** 和 **mingw64**)
4. 美化类
Background-cover、**Bracket Pair Colorizer**、**One Dark Pro**、**vscode-icons**、**background**
5. 辅助插件
Chinese (Simplified) Language Pack for Visual Studio Code、**Vim**、**carbon-now-sh**、**Settings Sync**
6. git
Git Automator、**Git History**、**Git Project Manager**、Git Tree Compare、Run git difftool

二、网站

1. 在 VS Code 中编译调试 C、Cpp
<https://blog.csdn.net/bat67/article/details/76095813>
<https://www.zhihu.com/question/30315894>
<https://zhuanlan.zhihu.com/p/77645306>
<http://cauyxy.com/2019/04/20/C++/>
2. 网易云音乐插件:
<https://github.com/electron/electron/releases>
<https://npm.taobao.org/mirrors/electron/4.2.5/>
3. Python:
<https://www.bilibili.com/video/BV1H7411U7LF/?p=2>
<https://www.cnblogs.com/DesignerA/p/11604200.html>
<https://www.cnblogs.com/it-tsz/p/9346110.html>
4. VS Code 说明
<https://code.visualstudio.com/>
<https://jeasonstudio.gitbooks.io/vscode-cn-doc/content/>

三、插件安装地址转移

[mklink /D "C:\Users\brain" "E:\World\knowledge"](#)

四、注意事项

#注意事项

1. 脑子里要有源代码文件这个概念。写程序不是说就在框框里写一堆代码然后就能跑了！代码都要写成文件保存起来（到硬盘里），然后才有后续的工作；
2. VS Code 是支持多语言的编辑器，并不只为 C/C++ 服务，所以还要确保把代码文件的后缀名写对，C 语言的后缀名是.c，C++ 的是.cpp。这是 VS Code 区分你代码类型的标志；
3. 上面安装的"C/C++"插件提供了对这两门语言的基本支持。C 跟 C++ 是两门不同的语言，请向你的老师问清楚学的是哪一门。无论是在.c 文件里敲 C++ 代码，还是在.cpp 文件里敲 C 代码，都是 XX 的行为。
4. 编写 C/C++ 程序有两个基本步骤，一是把代码文件变成程序文件，这叫编译或生成，是编译器做的事；二才是把刚刚生成的程序跑起来，这叫执行，是操作系统做的事；
5. 代码的编译和一般程序的执行，都是在命令行的环境下进行的，输入输出也是在这里面。VS Code 的优势之一，就是把系统的命令行集成进了软件内部，名唤“终端”，从而不必再开新窗口。

#相关目录标识符

`${workspaceFolder}` - the path of the folder opened in VS Code

`${workspaceRootFolderName}` - the name of the folder opened in VS Code

without any slashes (/)

`${file}` - the current opened file

`${relativeFile}` - the current opened file relative to workspaceRoot

`${fileBasename}` - the current opened file's basename

`${fileBasenameNoExtension}` - the current opened file's basename with no file extension

`${fileDirname}` - the current opened file's dirname

`${fileExtname}` - the current opened file's extension

`${cwd}` - the task runner's current working directory on startup

`${lineNumber}` - the current selected line number in the active file

#调试运行说明

1. `simplefilecode-runner` 直接运行 (Terminal) ;
`Task->build+run(Terminal)<ctrl+shift+B>` ; 系统自带 `Debug(Terminal/externendTerminal)<F5>`
2. `project` : `Cmake->build<F7>`、`run(Terminal)<shift+F5>`、`Degug`(同系统自带 `Debug<shift+F6>`); `Task->build+run(Terminal)<ctrl+shift+B>`; 系统自带 `Debug`(同上)`<F5>`
3. 单个文件用 `cpde-runner/mingw`，工程用 `Cmake+mingw`
4. 注意`<ctrl+F5>`是系统自带的 `Runwithoutdebuging`，既直接运行，但是对 C/Cpp 语言程序没有用，还是会直接调试，对其他语言程序有效,所以要想直接运行，只能通过 `code-runner`、`Task` 和 `Cmake` 的 `Runwithoutdebuging<shift+F5>`

#编码问题

1. 文件乱码：更改编码格式即可
2. 输出乱码：VS Code 输出窗口内置为 UTF-8,不同于 setting.json 中的 encoding,

而对于输出的汉语，在某些情况下会出现乱码，所以遇到乱码再解决，目前没有办法一次性解决

#运行问题

所有要运行的文件必须放到含有`.vscode` 的文件夹中，然后在相应的工作区中打开 **Floder**，在此环境下进行编写、编译、运行、调试等工作