Vscode 配置指南

作者:向仔州

这里提前说明一下，vscode 创建的工程不允许有中文路径。

安装 windows 平台的 C/C++编译器

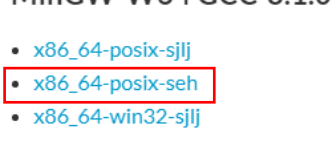
千万不要去微软官方下载

微软官方下载完成后就是这么一个软件，点击安装，你会发现在 bin 目录下面没有 gdb 调试工具，还要用这个 mingw 软件去安装其它 c/c++编译库。

我还是直接去网上找一个下载好的编译器实在

去这个网站下载 https://sourceforge.net/projects/mingw-w64/files/

根据你的 windows 平台来选择，我的电脑是 windows64 位的，所以我把网页拉到最下方

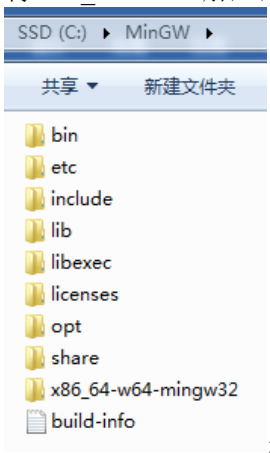


选择 x86_64-posix-she 来下载

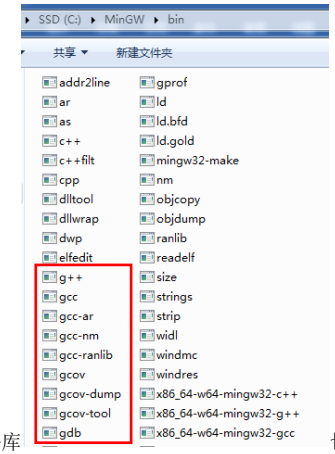
下载完成后就是这么一个压缩包

我在 C 盘创建一个 MinGW 的目录

将 x86_64-8.1….解压到 MinGW 目录下



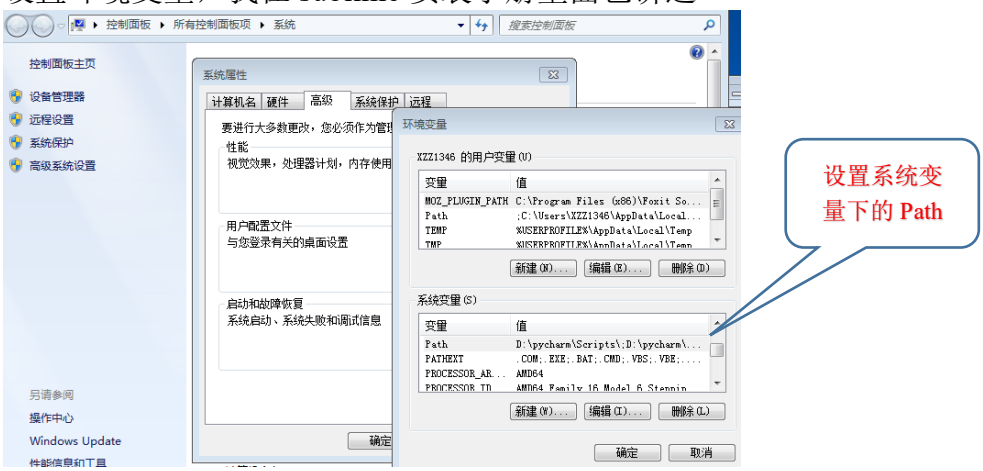
解压后有我要的 C/C++库

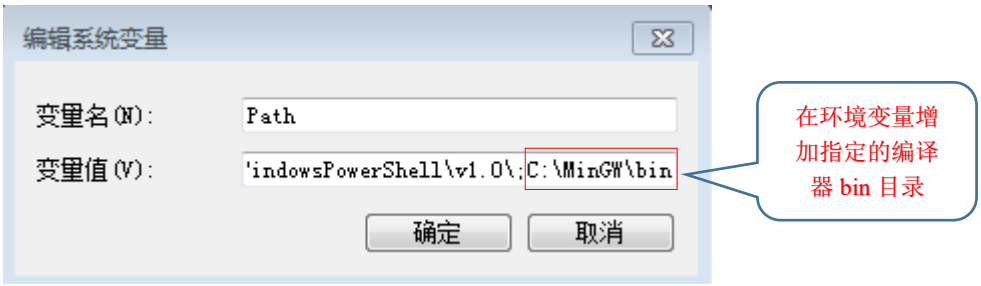


也有 gcc/g++/gdb 编译器

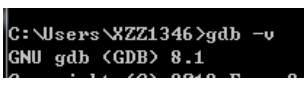
你看这些交叉编译器都有了，比如 gcc g++。

设置环境变量，我在 sublime 安装手册里面也讲过





执行 gcc –v 和 g++ -v，gdb –v



这三个工具都显示出版本号，就证明编译器环境变量设置成功。

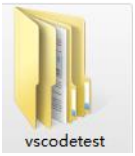
VScode 使用 gcc/g++编译器配置 C/C++环境

去微软官方下载 vscode，然后进行安装

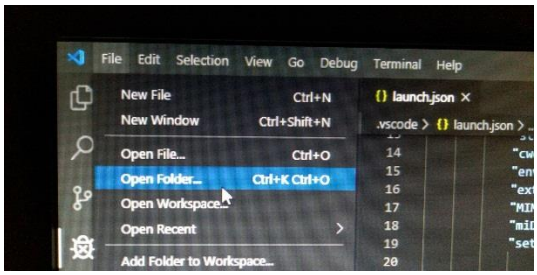
VScode 安装好后，打开 VScode 先安装 C/C++语法插件

然后建立 c++工程，

1.先创建 c++工程目录

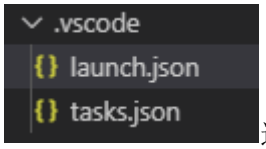


我这里取名为 vscodetest

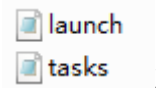


打开 Vscode 软件，用这个 openfolder 打开 vscodetest 工程目录，这样这个目录就是 vscode 的工程了，你在 vscode 下面创建的文件都会存放在这个 vscodetest 目录下，类似 sublime 工程创建的方式。

然后你需要创建 2 个配置文件在这个工程里面



这两个目录最好不要自己创建，让工程运行 C++程序时候提示你创建



就是这两个配置文件，工程运行后你创建的 launch 和 tasks 会自动在.vscode 目录下

你运行 C++程序后就会出现这个，你点击 c++(GDB/LLDB)，就会自动帮你创建 launch 和 tasks。

然后我们只需要修改 launch 和 tasks 参数就行了

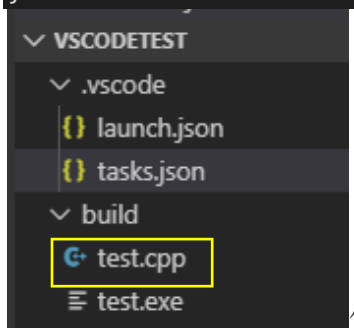
launch 和 tasks 参数修改

launch 文件修改

```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?li
nkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "name": "(gdb) Launch",
            "type": "cppdbg",
            "request": "launch",
            "program": "D:\\vscodetest\\build\\test.exe", //这里写入我编
译后生产的可执行程序文件路径，因为 gdb 会去自动寻找这个程序然后执行，你才可以看
到输出
            "args": [],
            "stopAtEntry": false,
            "cwd": "${workspaceFolder}", //这里就这样
            "environment": [],
            "externalConsole": true, //这里写 true
            "MIMode": "gdb",
            "miDebuggerPath": "C:\\MinGW\\bin\\gdb.exe", //这里填入你的
gdb.exe 程序位置
            "setupCommands": [
                {
                    "description": "Enable pretty-printing for gdb",//这
里一定要改成英文，不然会编译 g++ not found file
                    "text": "-enable-pretty-printing",
                    "ignoreFailures": true
                }
            ],
            "preLaunchTask": "compile" //这个名字随便去，但是 tasks 的
label 名字必须和这里一样
        }
    ]
}
```

tasks 文件修改

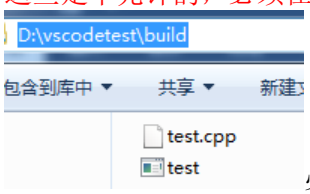
```
{
    // See https://go.microsoft.com/fwlink/?LinkId=733558
    // for the documentation about the tasks.json format
    "version": "2.0.0",
    "tasks": [
        {
            "type": "shell",
            "label": "compile", //你看 label 和 lunch 的 preLaunchTask 是一
样的
            "command": "g++", //这里写你的编译命令，g++或 gcc
            "args": [
                "-ggdb",
                "\"${file}\"",
                "--std=c++11",
                "-o",
                "\"${fileDirname}\\${fileBasenameNoExtension}.exe\""
            ],
            "options": {
                "cwd": "C:\\MinGW\\bin" //这里写你的编译器路径
            },
            "problemMatcher": [
                "$gcc"
            ],
            "group": "build"
        }
    ]
}
```



你自己建立 build 目录创建的你的 C++程序文件，然后调试就是。

如果运行出现 collect2.exe: error: ld returned 1 exit status ，这是程序被执行了，你关闭 vscode 然后看看任务管理有没有你的调试程序进程，有的话就直接关闭，重启 vscode 就好了

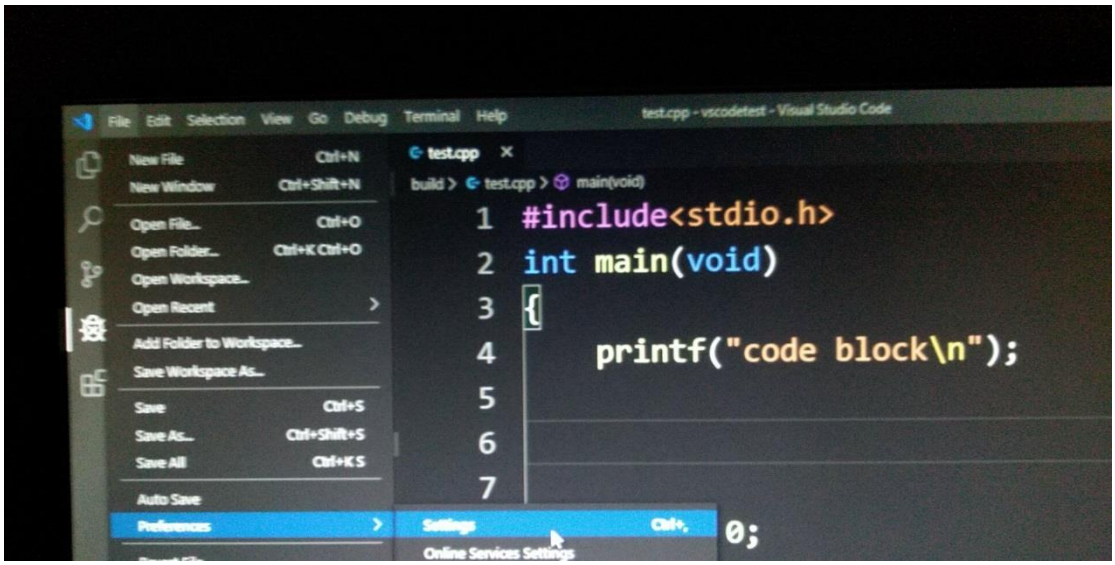
这里有个关键问题一定要记住，vscode 的所有工程不允许有中文路径，比如 D:\我的桌面\...，这些是不允许的，必须在英文路径下。不然会出现运行报错，说找不到文件什么的。



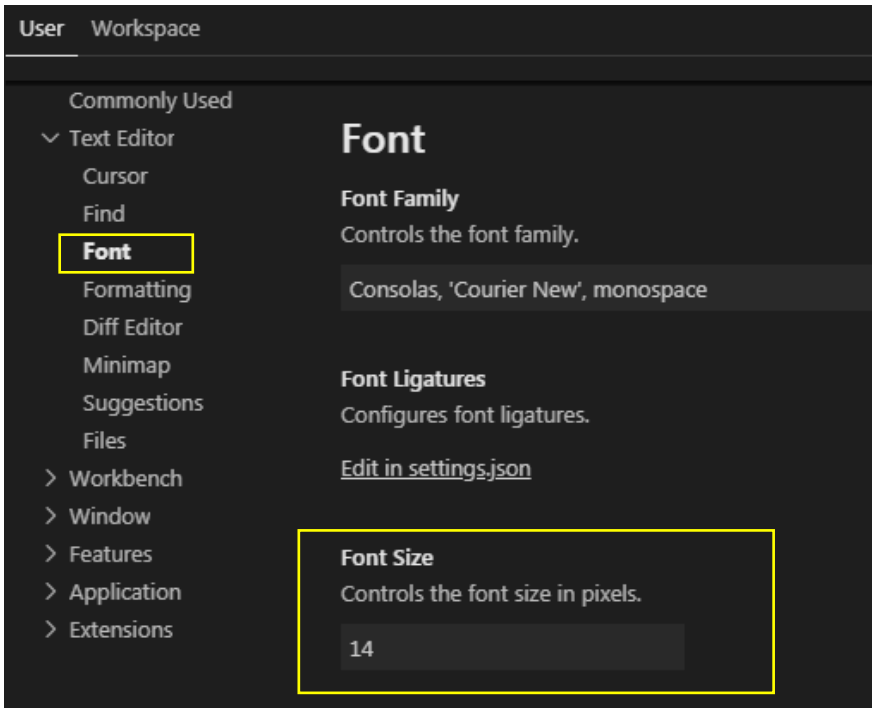
必须是因为路径，这样才正确。

你配置好的 launch 和 tasks 在其它新创建的工程里面可以直接复制过去用。放在.vscode 目录下就是

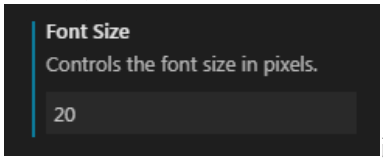
VScode 代码字体大小修改



选择 preferences->settings



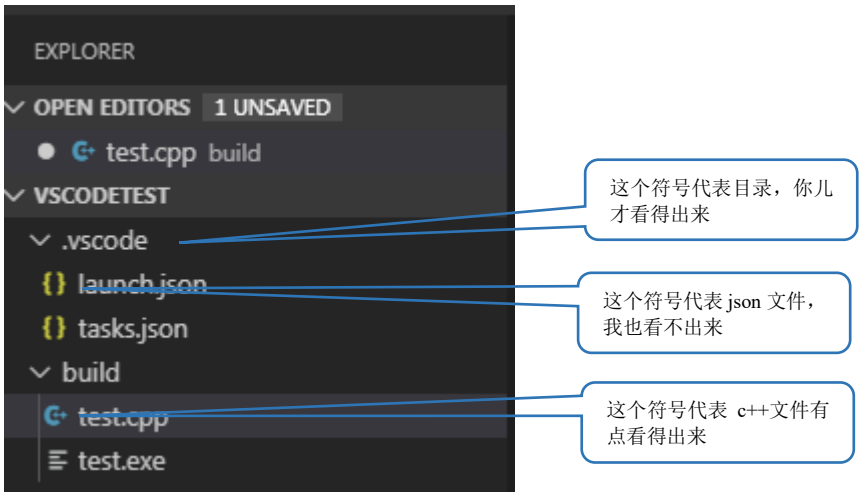
默认代码字体大小是 14，我改成 20 试试。



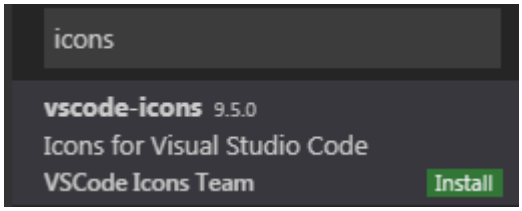
改完之后一定要执行 file->Save All 保存

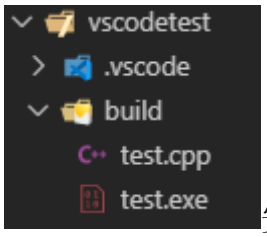
你看字体明显变大了，下次打开 VScode 也是这样大的字体

图标插件，让你的目录图标像目录，代码文件图标像代码，看起直观



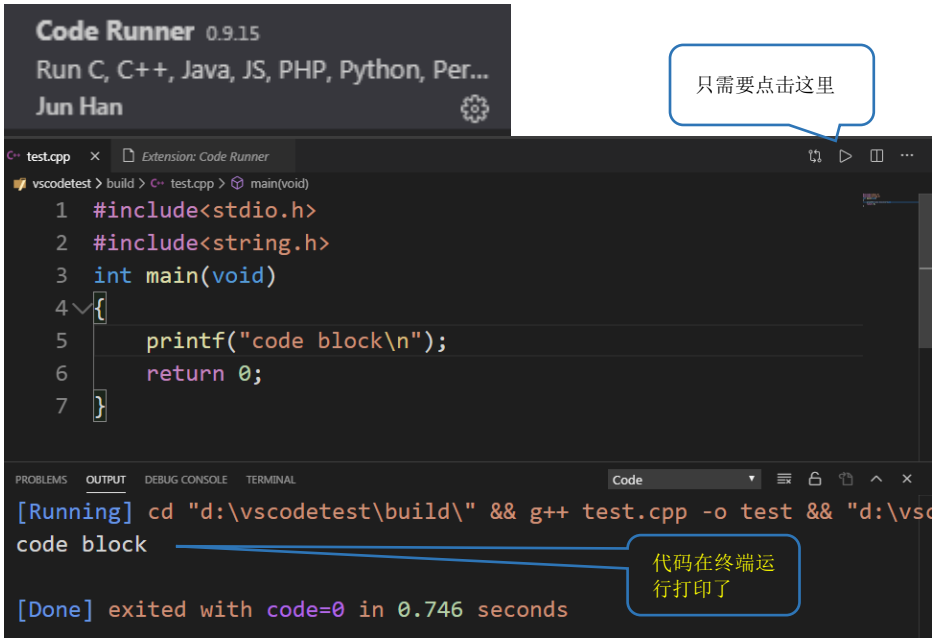
下面安装 vscode-icons 插件，解决图标难看问题



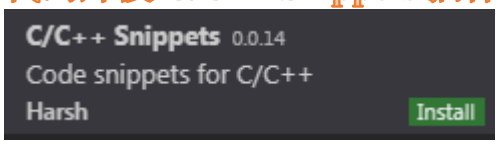


安装后，你看图标一目了然了目录是目录图标代码是代码图标

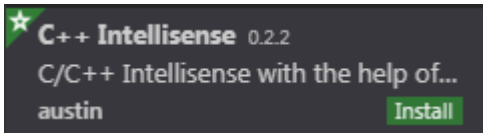
代码快速运行插件 Code Runner，方便点击运行按钮



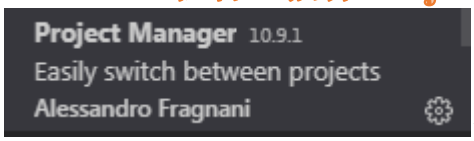
代码片段 C/C++ Snippets 插件安装



C++插件 intellisense 安装

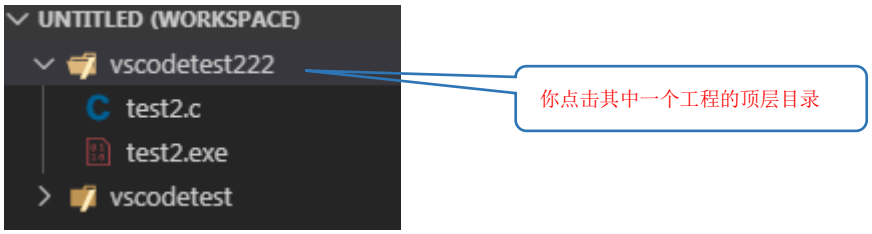


VScode 工程管理插件 Project Manager

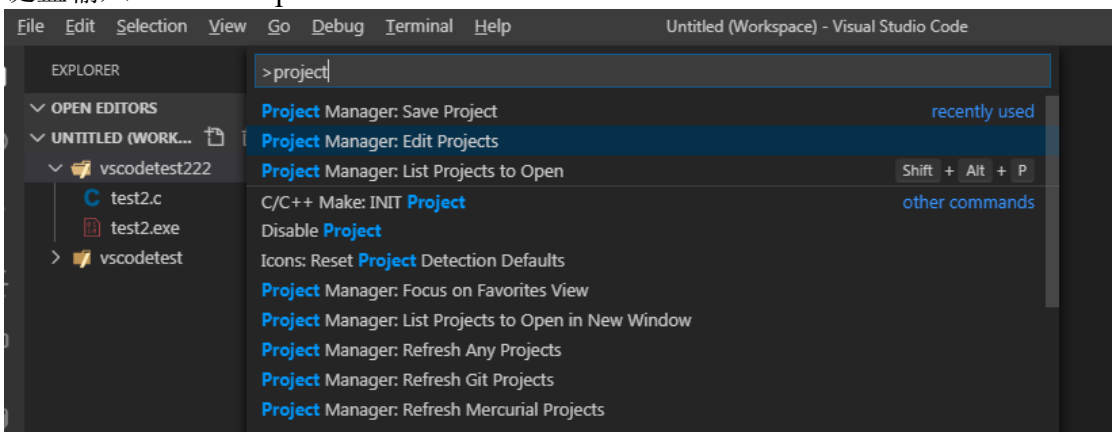


安装插件

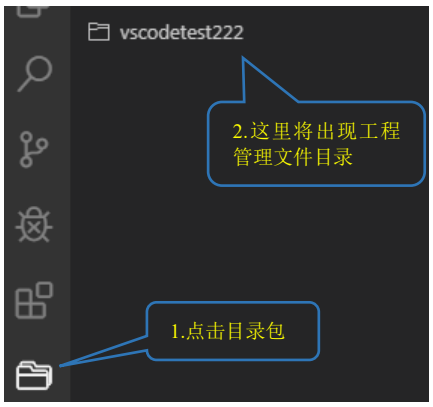
这两个目录就等于两个工程



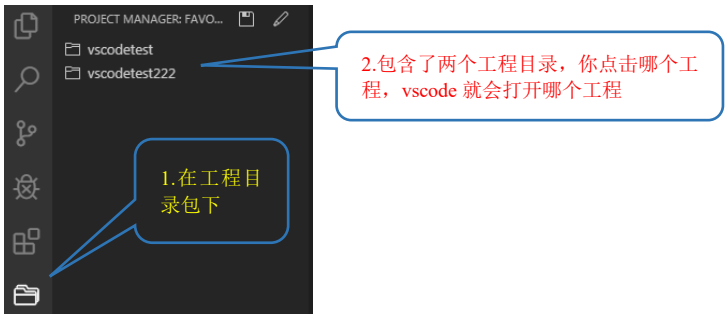
键盘输入 ctrl+shift+p

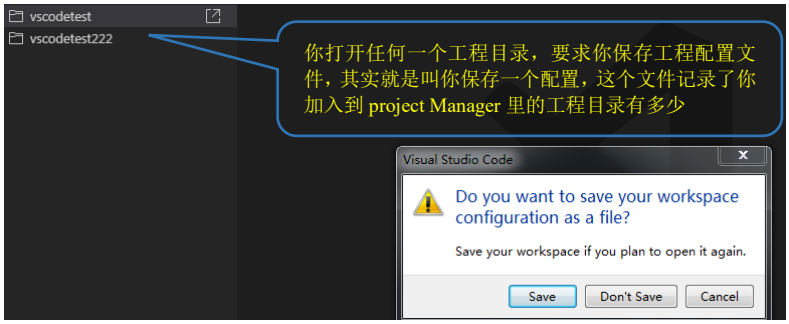


然后输入 Project Manager 选择 Edit Projects
将产生一个新的 json 文件

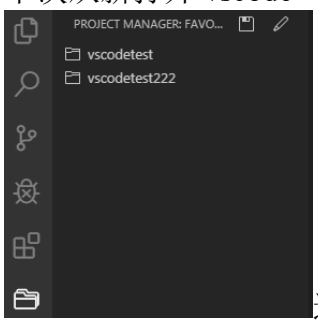


如果你想加入新的工程目录，就只需要修改 json 文件就行



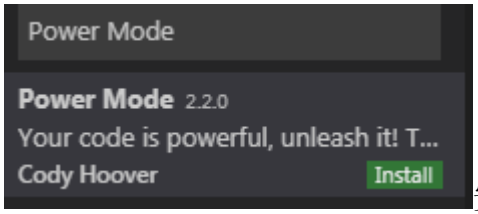


下次从新打开 vscode



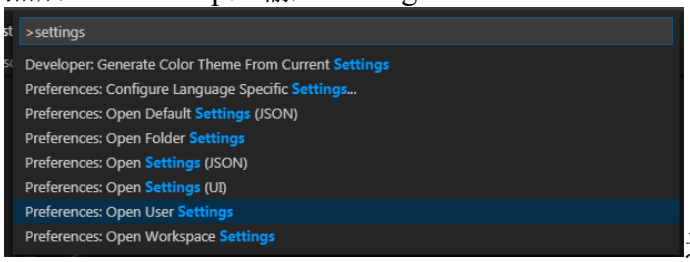
这两个工程都还在工程目录包下，你随便打开就是。

编辑代码花哨显示插件 Power Mode

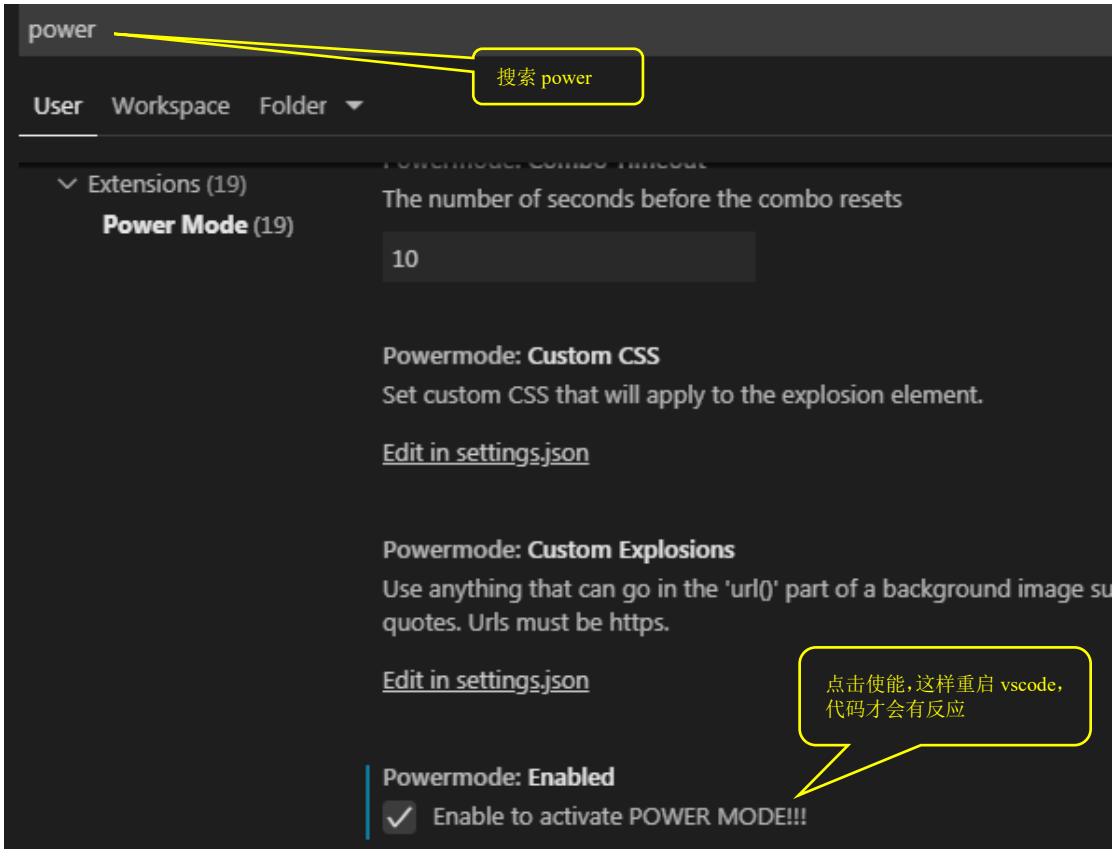


安装插件

然后 ctrl+shift+p，输入 settings



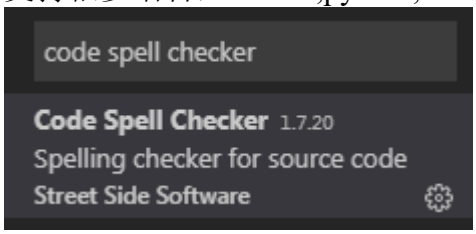
选择 open User Settings

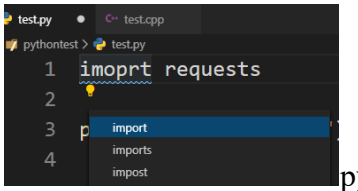


写代码就会有冒泡的效果

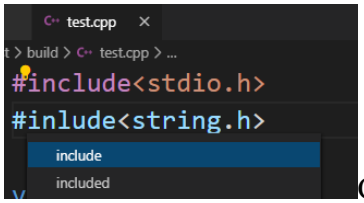
代码关键字写错，提示正确关键字插件 code spell Checker

支持很多语言，C/C++,python,Java….等等



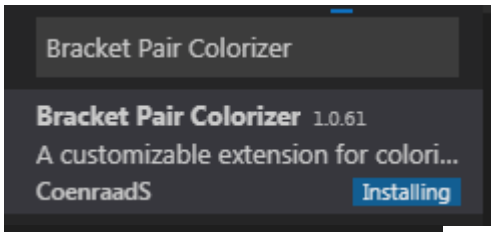


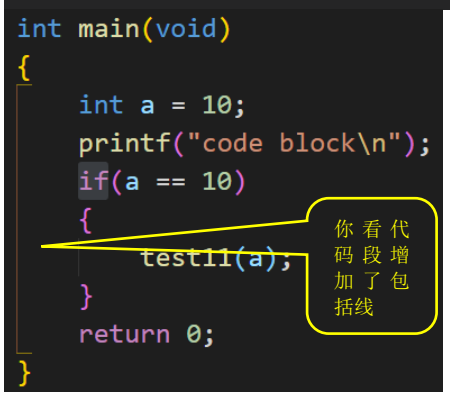
python import 写错提示选择



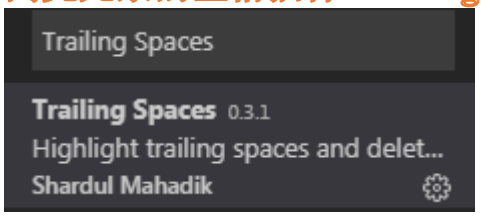
C++ include 写错提示选择

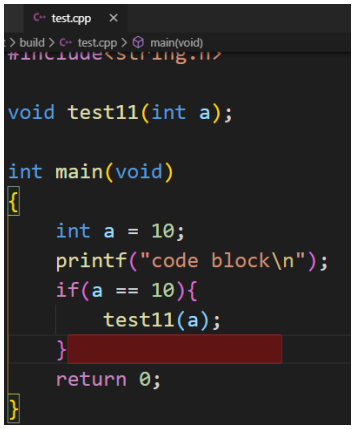
函数代码段颜色更加明显 Bracket Pair Colorizer 插件



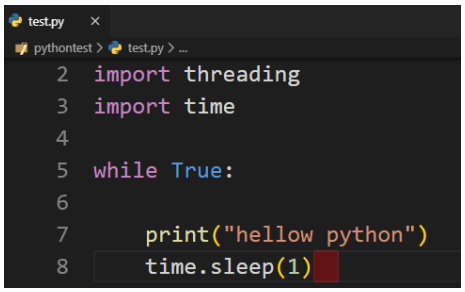


高亮冗余的空格插件 Trailing Spaces，我觉得对 python 有点用



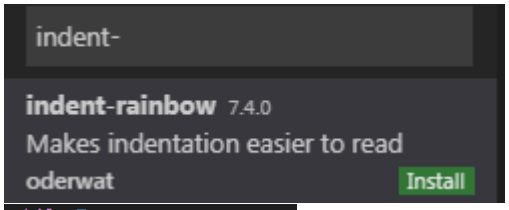


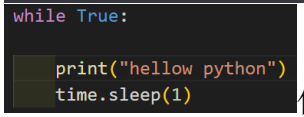
这是 C/C++空格检查



这是 python 空格检查

彩虹缩进插件 indent-rainbow，比较适合 python

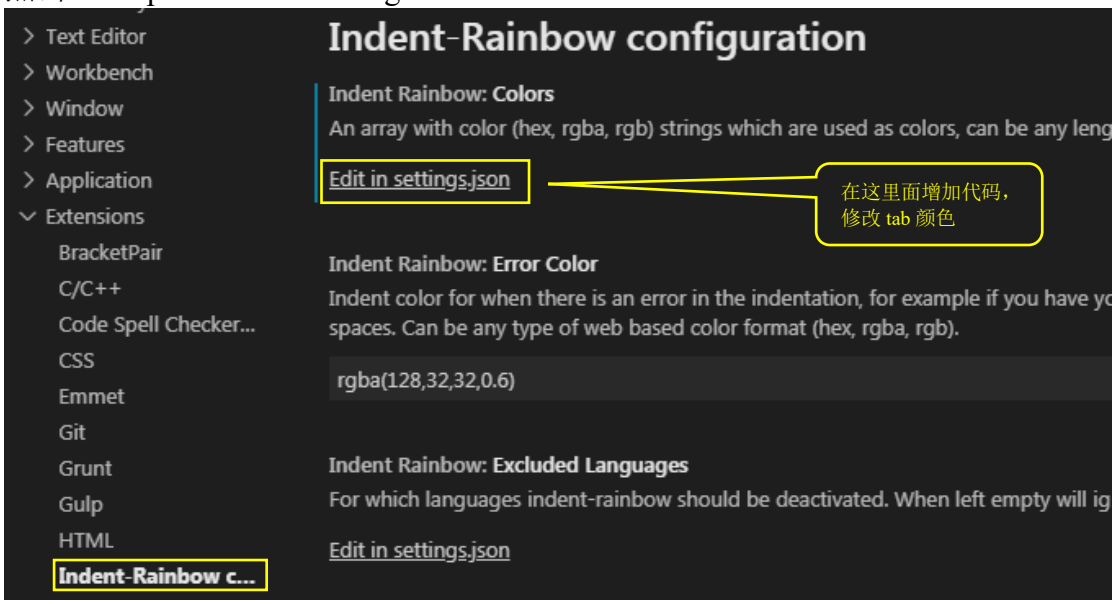




你看如果是缩进就是这种黄颜色，如果是空格就是红色。

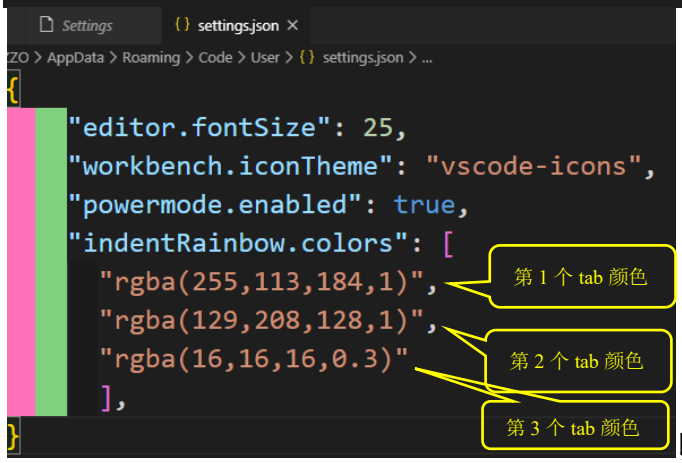
彩虹缩进语法规则

点击 file->prefencens->setting



输入下面这段语句

```
"indentRainbow.colors": [
        "rgba(255,113,184,1)",
        "rgba(129,208,128,1)",
        "rgba(16,16,16,0.3)"
        ],
```



以此类推

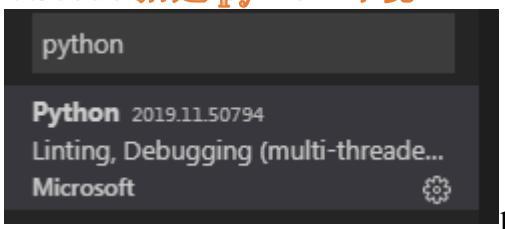
格式就是 rgba(红，绿，蓝，透明度) 透明度是 0~1,1 是不透明，0 是完全透明，0.5 是半透明，也可以是 0.1,0.2,0.3…决定了透明强度。用 windows 画板决定调色值。

设置完成后重启 Vscode 就能看见了。

```
"indentRainbow.colors": [
      "rgba(122,118,105,0.1)",
      "rgba(71,67,65,0.1)",
      "rgba(192,192,192,0.1)"
      ],
```

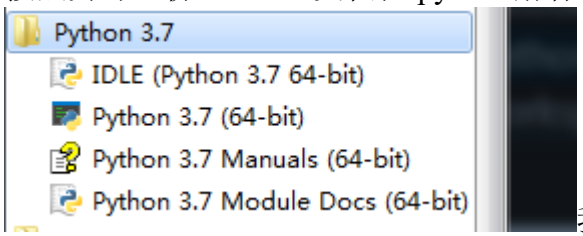
这是我的配色方案

VScode 搭建 python 环境



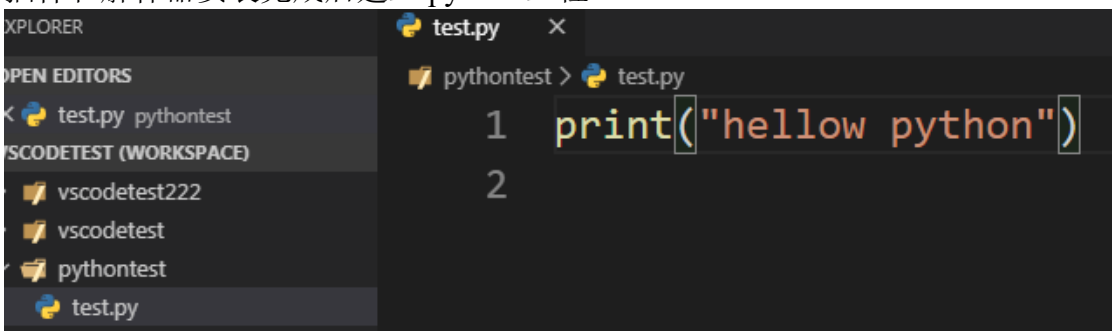
python 语法调试器安装

按照另外一份 sublime 文档把 python 解释器安装好

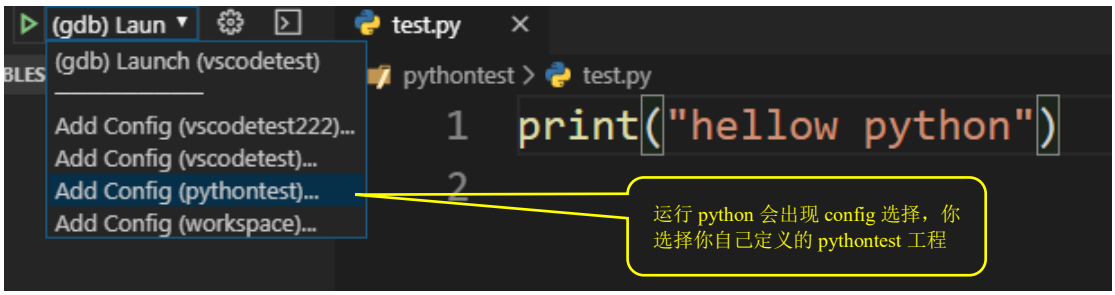


我这里是 python3.7

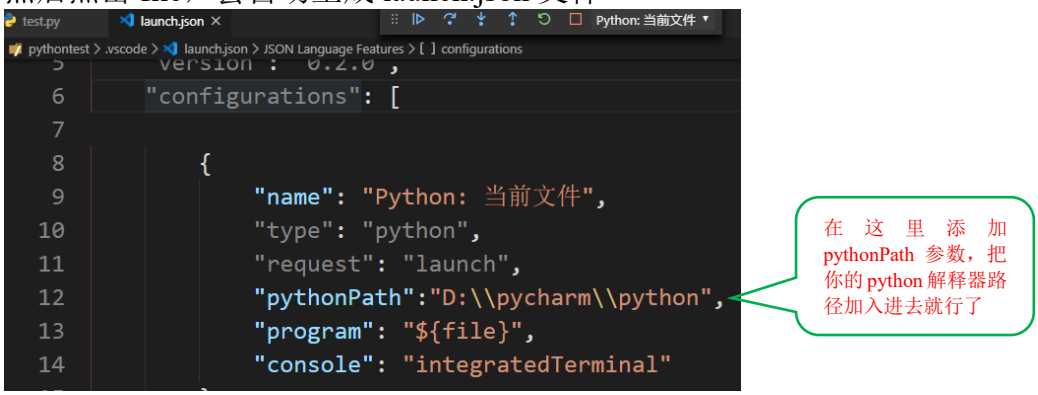
插件和解释器安装完成后建立 python 工程



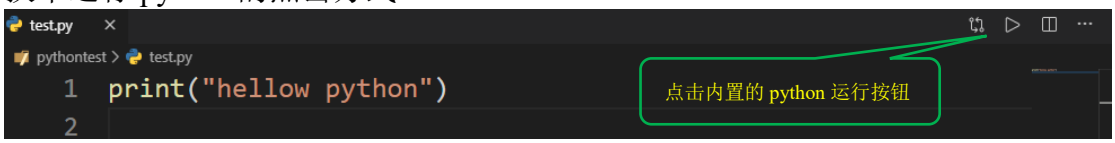
创建 python 目录和加载目录创建 python 工程

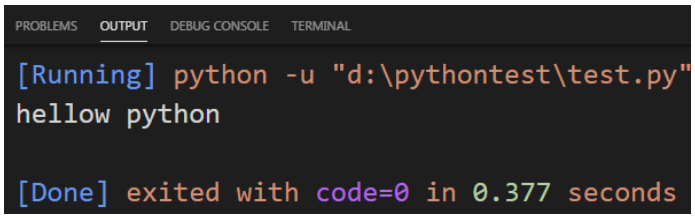


然后点击 file，会自动生成 launch.json 文件



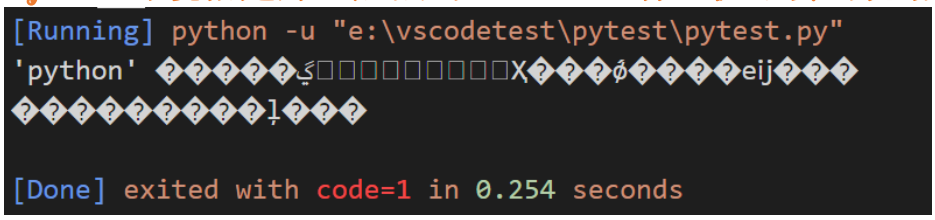
换个运行 python 的点击方式



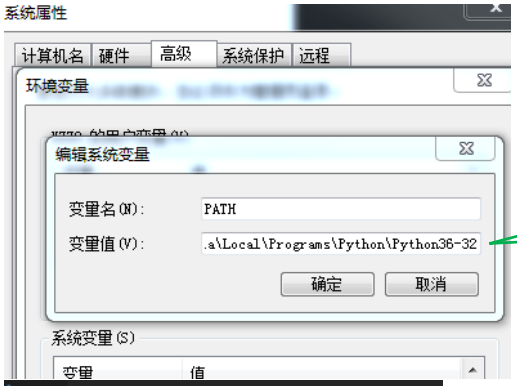


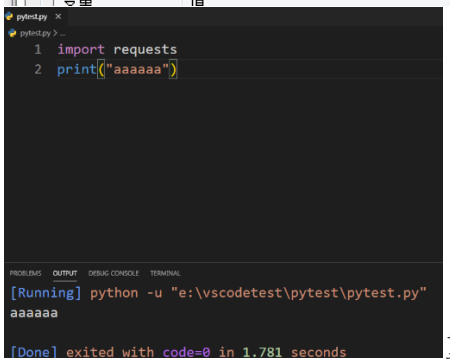
你看这种打印出来就很好看

Python 环境搭建好之后点击 Rum code 有些机器会出现错误



这是因为你忘记在 windows 电脑加入 python 的环境变量了





重启 VScode，问题得到解决

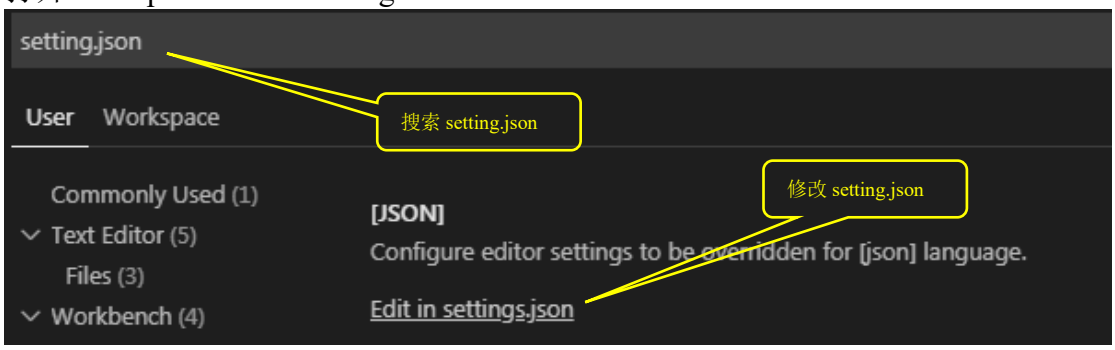
Run code 运行 python 程序，遇到中文输出乱码

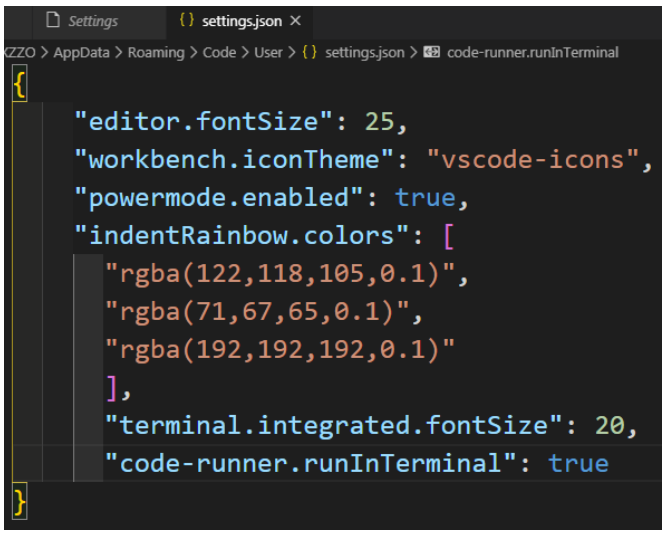
```
[Running] python -u "e:\vscodetest\pytest\pytest.py"
??????

[Done] exited with code=0 in 0.464 seconds
```

这种情况做好把 Run code 程序映射到 TERMINAL 终端运行

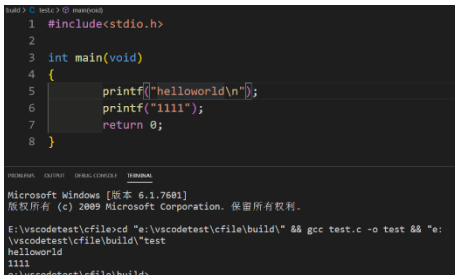
打开 file->prefercens->settings





加入`"code-runner.runInTerminal": true`

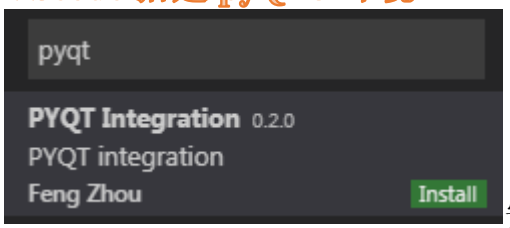
这样 python 和 C++的代码 Run code 运行都映射到 TERMINAL 终端了。



C++运行在 TERMINAL 输出　　Python 运行在 TERMINAL 输出

这样主要解决输出显示中文乱码问题

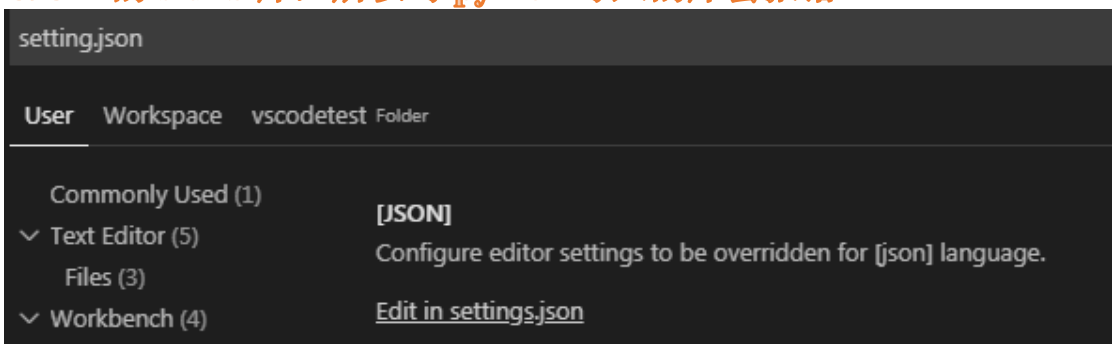
VScode 搭建 pyQT5 环境



安装 pyqt 插件

然后按照 pyQT5 使用手册在 windows 下用 pip 安装 pyQT5 库到 python 解释器目录下。

编写 pyQT5 代码出现红色波浪线问题解决，主要是 pylint 是检查 C/C++的 stdlib 库，所以对 python 导入的库会报错。



打开 setting.json 文件

关闭 pylint 报错

```
"python.linting.pylintArgs": [
        "--disable=W,C,E"
    ], //忽略指定报错
    "python.linting.enabled": false //关闭 pylint 报错 （不包括语法错误）
```

问题得到解决。如果你不想这样做也可以在 settings.json 后面加这句

```
"python.linting.pylintArgs": ["--generate-members"]
```

这样也是可以的

Python 智能自动补全

在 setting.json 中加入你 pip 安装在 python 解释器目录下的第三方库路径

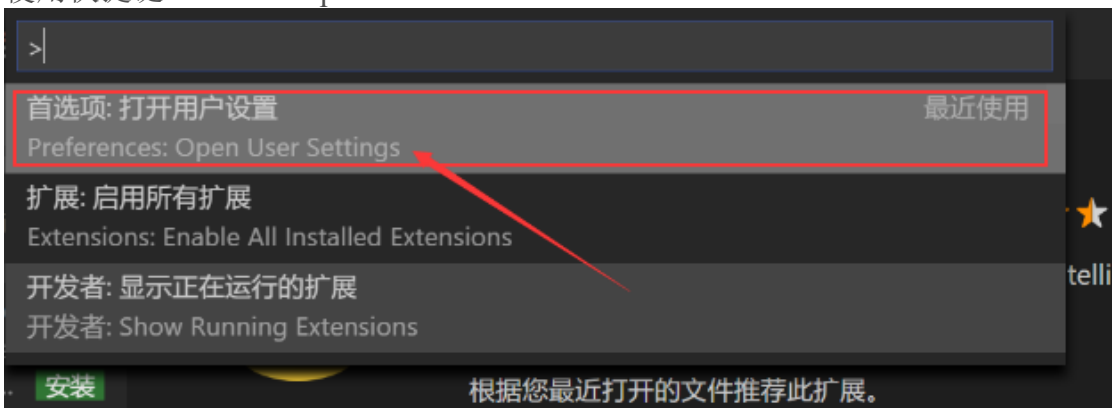
```
"python.autoComplete.extraPaths": [
            "C:/Users/XZZO/AppData/Local/Programs/Python/Python36-32/Lib/site-packages",
            "C:/Users/XZZO/AppData/Local/Programs/Python/Python36-32/Scripts",
    ],
```

支持一些 python 语法格式，这里要安装 flake8

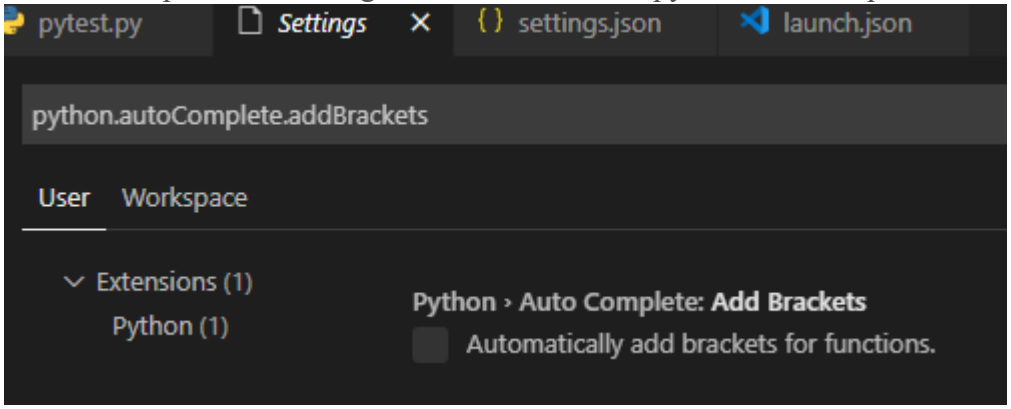
```
"python.formatting.provider": "yapf",
    "python.linting.flake8Args": ["--max-line-length=10"],
    "python.linting.pylintEnabled": false,
```

VScode 的 python 没有自动补全功能，导致安装的 PyQT5 也没有自动补全，如何增加自动补全

使用快捷键 Ctrl+Shift+p



在打开的 Open User Settings 文件界面，搜索：python.autoComplete.addBrackets



勾选上