

python配置窗口大小和标题

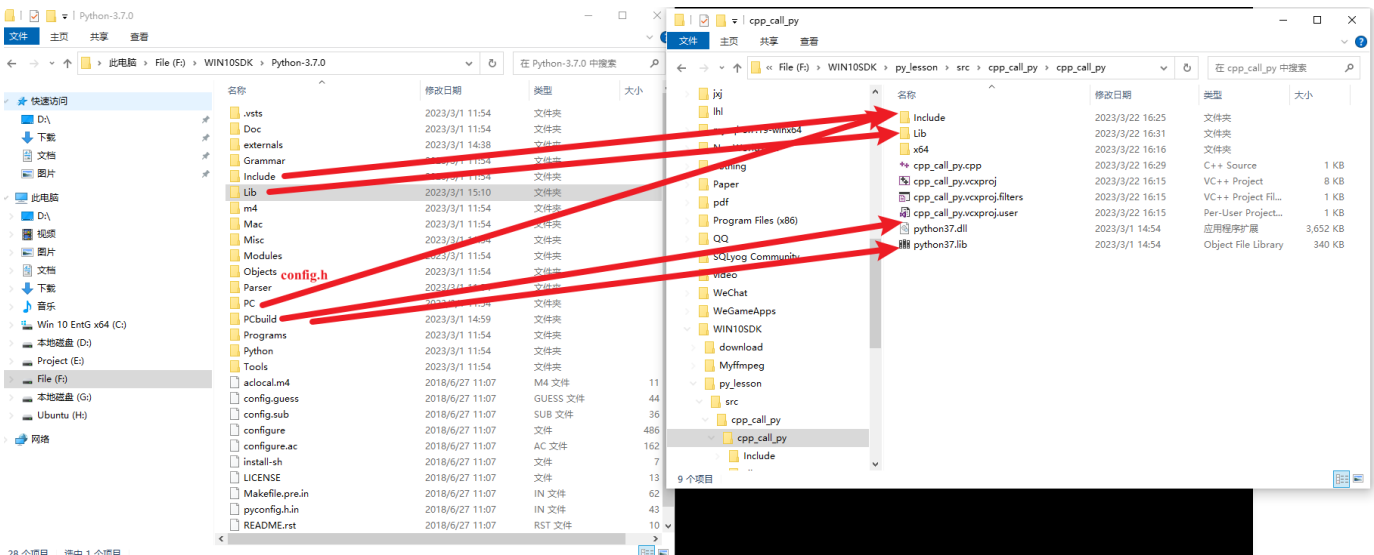
一、配置python库环境

一、配置python库环境

在C++调用python中 学习过

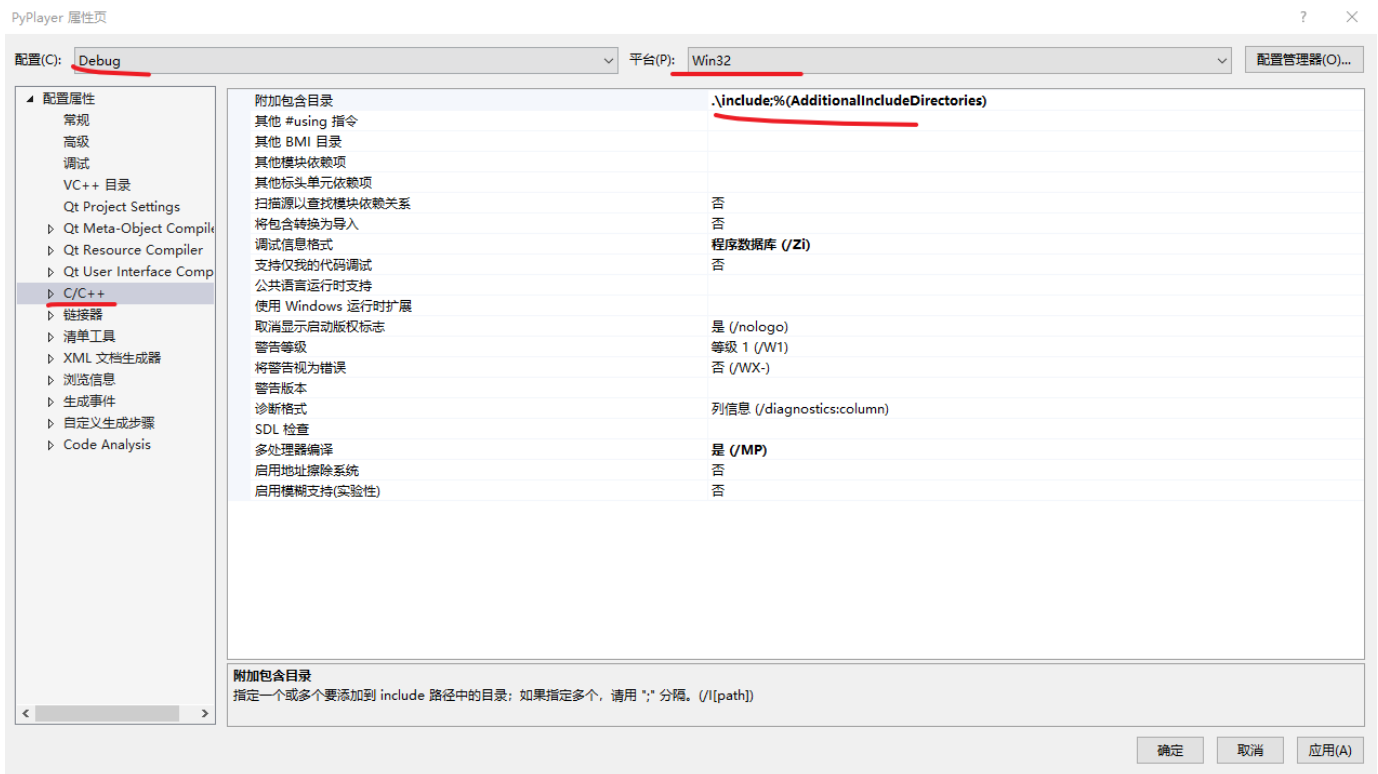
- 1. include 同时把PC\pyconfig.h 放到include中
- 2. PCbuild\win32\python37.dll
- 3. PCbuild\win32\python37.lib
- 4. Lib

这四个文件放到执行项目目录下



名称	修改日期	类型	大小
Debug	2023/3/28 10:42	文件夹	
Include	2023/3/28 10:45	文件夹	
Lib	2023/3/28 10:45	文件夹	
++ main.cpp	2023/3/28 10:42	C++ Source	1 KB
++ PyPlayer.cpp	2023/3/28 10:42	C++ Source	1 KB
PyPlayer.h	2023/3/28 10:42	C/C++ Header	1 KB
PyPlayer.qrc	2023/3/28 10:42	QRC 文件	1 KB
PyPlayer.ui	2023/3/28 10:42	Qt UI file	1 KB
PyPlayer.vcxproj	2023/3/28 10:42	VC++ Project	6 KB
PyPlayer.vcxproj.filters	2023/3/28 10:42	VC++ Project Fil...	2 KB
PyPlayer.vcxproj.user	2023/3/28 10:42	Per-User Project...	1 KB
python37.dll	2023/3/1 14:54	应用程序扩展	3,652 KB
python37.lib	2023/3/1 14:54	Object File Library	340 KB

编辑器项目属性：



从python脚本里面读取配置

Qt按钮事件的Open槽函数调用python的open函数，

1. 在PyPlayer::PyPlayer构造函数中完成窗口大小和标题的初始化 和 开放接口函数的声明
 - a. //初始化 根目录

- b. //载入模块 pyqt.py
- c. //获取python pyqt.py配置项conf
- d. 开放选择文件的接口给python PyModule_AddFunctions

2. 定义信号槽函数

- a. public slots:
 - i. void Open();

3. 实现槽函数的功能

- a. void PyPlayer::Open()

4. 槽函数调用python的Open函数

- a. PyObject_CallObject(open, 0); //调用python的Open函数

5. python的Open函数调用c++的接口函数OpenDialog获取filename

```
def open():
    print(" PyPlayer Open ")
    filename = OpenDialog()
    print(filename)
```

6. 定义OpenDialog接口函数

- a. PyObject* OpenDialog(PyObject* self)

7. 传递接口函数给python脚本

```
static PyMethodDef cfuncs[] = {
    {"OpenDialog", (PyCFunction)OpenDialog, METH_NOARGS, 0}
    ,{NULL} };

int re = PyModule_AddFunctions(pMoudle, cfuncs);
```