

Python 圖形界面框架PyQt5 使用指南！

錢魏Way 濤哥聊Python 2021-12-08 22:41



作者：錢魏Way

<https://www.biaodianfu.com/pyqt5.html>

使用Python開發圖形界面的軟件其實並不多，相對於GUI界面，可能Web方式的應用更受人歡迎。但對於像我一樣對其他編程語言比如C#或WPF並不熟悉的人來說，未必不是一個好的工具。

常見GUI框架

- **PyQt5**^[1]：Qt^[2]是一個跨平台的C++圖形用戶界面庫。QT一度被諾基亞擁，後出售給芬蘭的軟件公司Digia Oyj。PyQt5是基於Digia公司Qt5的Python接口，由一組Python模塊構成。PyQt5本身擁有超過620個類和6000函數及方法。在可以運行於多個平台，包括：Unix, Windows, and Mac OS。
- **PySide6**^[3]：PySide是QT公司官方提供的Python包，上一版本為PySide2，對應的是QT5，最新版命名規則進行了調整，更改為PySide6，對應的是QT6版本。由於官方出品的比較看好，缺點是發布比較晚，網上的資料沒有PyQt5多。
- **Tkinter**^[4]：Python內置的GUI框架，使用TCL實現，Python中內嵌了TCL解釋器，使用它的時候不用安裝額外的擴展包，直接import，跨平台。不足之處在於UI佈局全靠代碼實現，只有15種常用部件，顯示效果簡陋。
- **PySimpleGUI**^[5]：PySimpleGUI是Tkinter一層包裝。使用PySimpleGUI實現自定義GUI所需的代碼量要比使用Tkinter直接編寫相同的GUI要少得多。

^[6]

- **WxPython**^[10]：wxPython是Python語言對流行的wxWidgets跨平台GUI工具庫的綁定。用得比較廣泛，跨平台，C++編寫，文檔少，用戶可能就需要根據編程內容對不同平台中的GUI代碼做一些調整。遇到問題不好解決，代碼佈局控件，不直觀。
- **Wax**^[7]：基於wxPython，為克服wxPython的問題而製作的一個包。
- **Kivy**^[8]：主要針對多點觸控程序，智能手機平板等，也可以在沒有觸屏功能的系統上，全平台支持（Windows, Linux, Mac OS X, Android and iOS.）使用Python和cython編寫，中文支持差，需要自己下載中文庫並且制定路徑。
- **BeeWare**^[9]：Write once. Deploy everywhere.需要與Kivy配合使用。
- **Toga**^[10]：一個使用Python開發原生APP的GUI工具包。Toga由一個具有共享接口的基礎組件庫組成，以簡化與平台無關的GUI開發。Toga適用於Mac OS、Windows、Linux（GTK）以及Android和iOS等移動平台。
- **Eel**^[11]：一個輕量的Python庫，用於製作簡單的類似於Electron（但是比它更輕量）的離線HTML/JS GUI 應用程序，並具有對Python 功能（capabilities）和庫的完全訪問權限。
- **Flexx**^[12]：一個純Python 工具包，用來創建圖形化界面應用程序。其使用Web 技術進行界面的渲染。你可以用Flexx 來創建桌面應用，同時也可以導出一個應用到獨立的HTML 文檔。因為使用純Python 開發，所以Flexx 是跨平台的。只需要有Python 和瀏覽器就可以運行。
- **pywebview**^[13]是圍繞webview 組件的輕量型跨平台包裝器（wrapper），它允許在其自己的本機GUI 窗口中顯示HTML 內容。它使您可以在桌面應用程序中使用Web 技術，同時盡最大可能隱藏使用瀏覽器構建GUI的事實。
- **enaml**^[14]：一種能夠讓你用最小的努力就可以實現高質量GUI界面的Python框架，也是一種獨特的編程語言。enaml將聲明性語言與基於約束的佈局系統結合在一起，使用戶可以輕鬆地定義靈活佈局的UI。enaml應用程序可以在任何支持Python和Qt的平台上運行。

個人想法：太多學不完，先學PyQt5，原因是資料多，學有餘力再學pyside6，最後看下PySimpleGUI，看能否解決一些簡單問題。

PyQt5簡介

PyQt是Qt框架的Python語言實現，由Riverbank Computing開發，是最強大的GUI庫之一。PyQt提供了一個設計良好的窗口控件集合，每一個PyQt控件都對應一個Qt控件，因此PyQt的API接口與Qt的API接口很接近，但PyQt不再使用QMake系統和Q_OBJECT宏。

PyQt5提供GPL版和商業版證書，自由開發者可以使用免費的GPL許可，如果需要將PyQt用於商業應用，則必須購買商業許可。

PyQt5特性如下：

- 基於高性能的Qt的GUI控件集。
- 能夠跨平台運行在Linux、Window和Mac OS系統上。
- 使用信號槽機制進行通信。
- 對Qt庫進行完全封裝。
- 可以使用成熟的IDE進行界面設計，並自動生成可執行的Python代碼。
- 提供一整套種類齊全的窗口控件。

PyQt5是由一系列Python模塊組成，有超過620個類，6000個函數和方法，主要模塊如下：

- QtCore：包含了核心的非GUI 的功能。主要和時間、文件與文件夾、各種數據、流、URLs、mime 類文件、進程與線程一起使用。
- QtGui：包含了窗口系統、事件處理、2D 圖像、基本繪畫、字體和文字類。
- QtWidgets：包含了一系列創建桌面應用的UI 元素。
- QtMultimedia：包含了處理多媒體的內容和調用攝像頭API 的類。
- QtBluetooth：包含了查找和連接藍牙的類。
- QtNetwork：包含了網絡編程的類，這些工具能讓TCP/IP 和UDP 開發變得更加方便和可靠。
- QtPositioning：包含了定位的類，可以使用衛星、WiFi 甚至文本。
- Enginio：包含了通過客戶端進入和管理Qt Cloud 的類。
- QtWebSockets：包含了WebSocket 協議的類。
- QtWebKit：包含了一個基WebKit2 的web 瀏覽器。
- QtWebKitWidgets：包含了基於QtWidgets 的WebKit1 的類。
- QtXml：包含了處理xml 的類，提供了SAX 和DOM API 的工具。
- QtSvg：提供了顯示SVG 內容的類，Scalable Vector Graphics (SVG) 是一種是一種基於可擴展標記語言(XML)，用於描述二維矢量圖形的圖形格式（這句話來自於維基百科）。
- QtSql：提供了處理數據庫的工具。
- QTest：提供了測試PyQt5 應用的工具。

PyQt5的安裝

由於後期要使用fbs進行打包，fbs對Python 3.7以後的版本可能存在兼容問題，所以我選擇了Python 3.6.8進行了整個環境的搭建。主要內容為：Python + PyCharm + PyQt5

安裝PyQt5



```
pip install pyqt5  
  
pip install pyqt5-tools
```

其中pyqt5-tools為Qt Designer拖拽式的界面設計工具。安裝過程中可能會報如下錯誤：

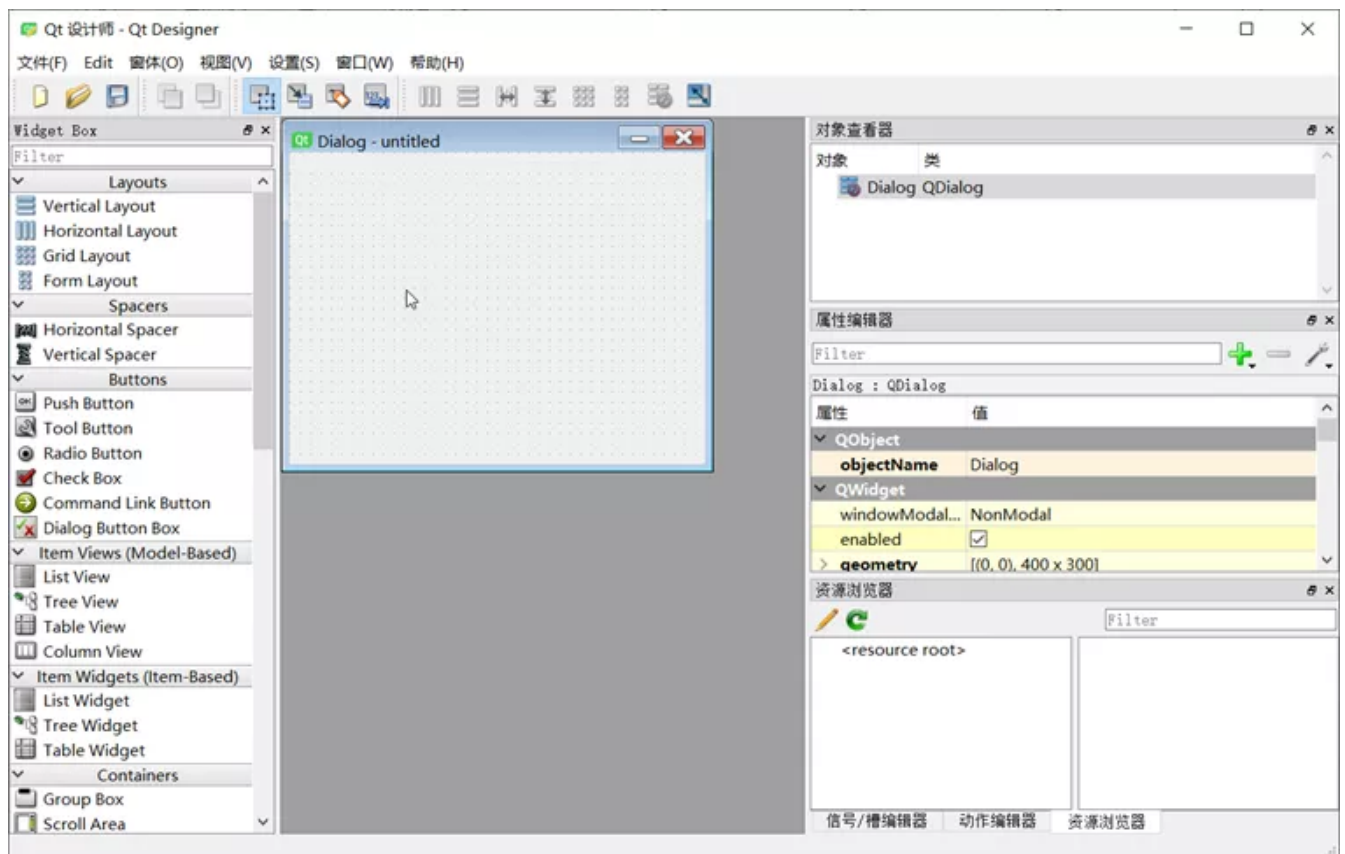
```
qt5-tools 5.15.2.1.2 has requirement click~=7.0, but you'll have click 8.0.1 which is incompatible
```

解決方案：

```
pip install click~=7.0
```

Qt Designer的配置

Qt Designer 是通過拖拽的方式放置控件，並實時查看控件效果進行快速UI設計。



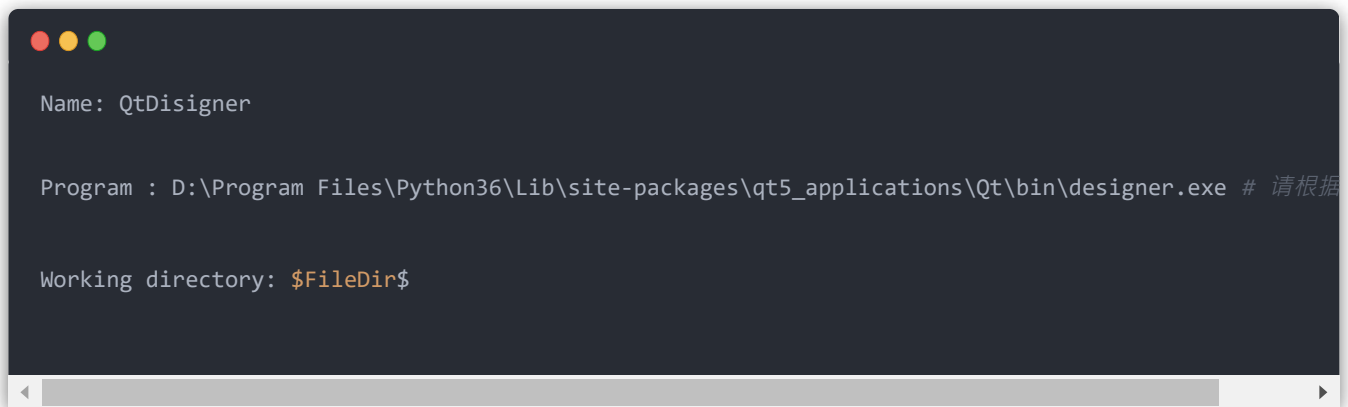
整個畫面的構成：

- 左側的“Widget Box”就是各種可以自由拖動的組件
- 中間的“MainWindow – untitled”窗體就是畫布
- 右上方的”Object Inspector”可以查看當前ui的結構
- 右側中部的”Property Editor”可以設置當前選中組件的屬性
- 右下方的”Resource Browser”可以添加各種素材，比如圖片，背景等等

最終生成.ui文件（實質上是XML格式的文件），可直接使用，也可以通過pyuic5工具轉換成.py文件。

QtDesigner配置

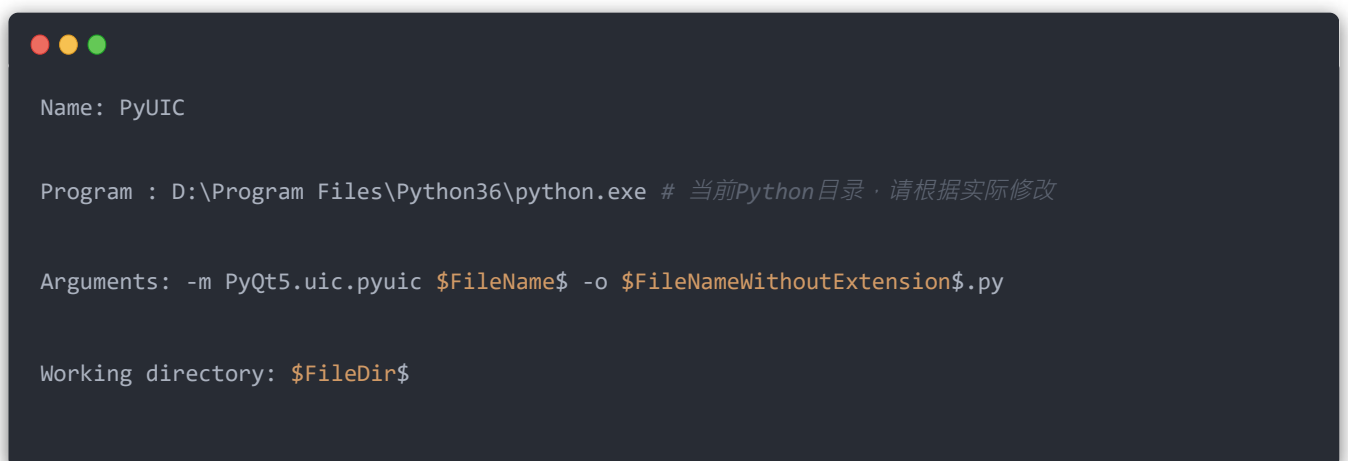
在Pycharm中，依次打開File – Settings – Tools – External Tools，點擊+ Create Tool，配置如下：



PyUIC配置

PyUIC主要是把Qt Designer生成的.ui文件換成.py文件。

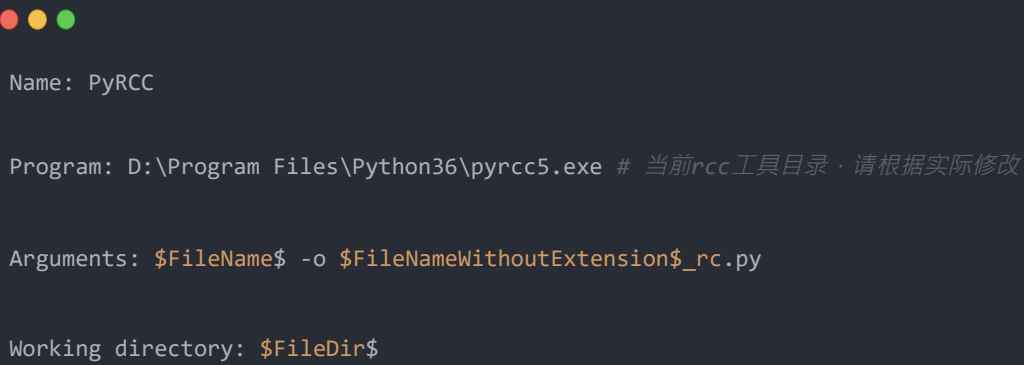
在Pycharm中，依次打開File – Settings – Tools – External Tools，點擊+ Create Tool，配置如下：



PyRCC配置

PyRCC主要是把編寫的.qrc資源文件換成.py文件。

在Pycharm中，依次打開File – Settings – Tools – External Tools，點擊+ Create Tool，配置如下：

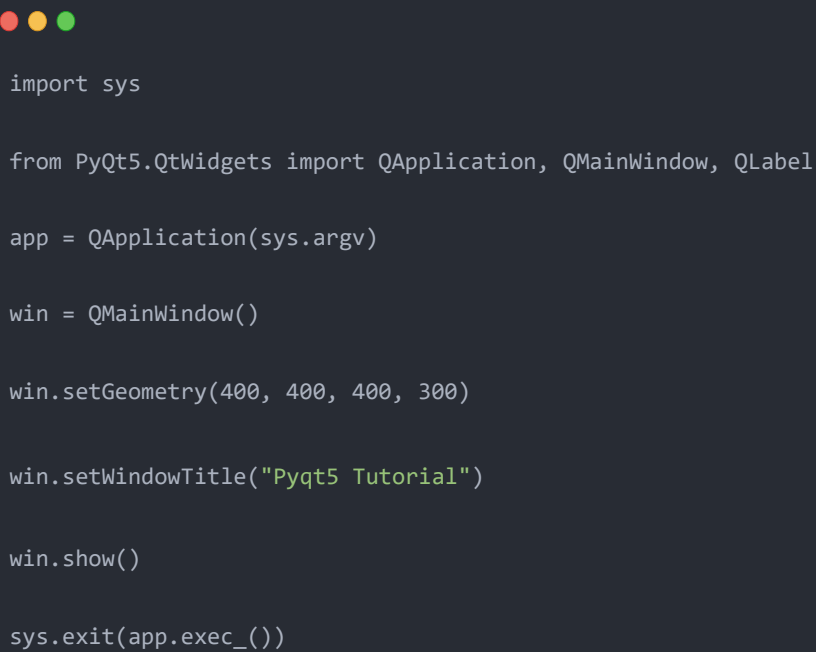


A screenshot of the PyCharm 'External Tools' configuration window. It shows the following fields:

- Name: PyRCC
- Program: D:\Program Files\Python36\pyrcc5.exe # 当前rcc工具目录，请根据实际修改
- Arguments: \$FileName\$ -o \$FileNameWithoutExtension\$_rc.py
- Working directory: \$FileDir\$

PyQt5使用示例

創建一個空白的界面：



```
import sys

from PyQt5.QtWidgets import QApplication, QMainWindow, QLabel

app = QApplication(sys.argv)

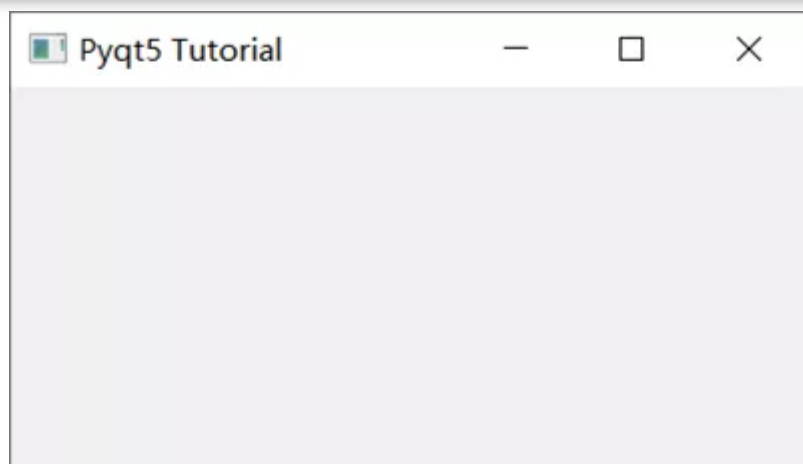
win = QMainWindow()

win.setGeometry(400, 400, 400, 300)

win.setWindowTitle("Pyqt5 Tutorial")

win.show()

sys.exit(app.exec_())
```

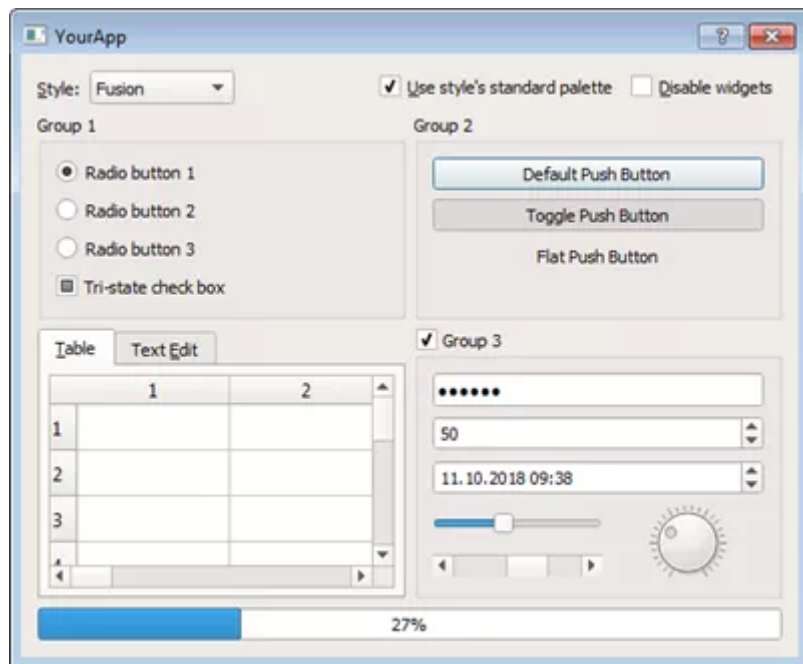




其中：

- **Qapplication()**：每個GUI都必須包含一個**Qapplication**，**argv**表示獲取命令行參數，如果不用獲取，則可以使用[]代替。
- **QMainWindow()**：類似一個容器（窗口）用來包含按鈕、文本、輸入框等**widgets**。arg標識可以獲取命令行執行時的參數。
- **SetGeometry**是用來定義**QMainWindow()** 窗口的尺寸，語法：**setGeometry(x, y, width, height)**，其中x,y為屏幕上的坐標點。
- **show()**：用來顯示窗口
- **exit(app.exec_())**：設置窗口一直運行指導使用關閉按鈕進行關閉

PyQt5支持的常見Widgets有：



從上到下，從左到右依次為：**QLabel**、**QcomboBox**、**QcheckBox**、**QradioButton**、**QpushButton**、**QtableWidget**、**QlineEdit**、**Qslider**、**QProgressBar**

對於使用Pyqt5設置文本內容，我們使用**QLabel**：

```
import sys
```

```
from PyQt5.QtWidgets import QApplication, QMainWindow, QLabel

app = QApplication(sys.argv)

win = QMainWindow()

win.setGeometry(400, 400, 400, 300)

win.setWindowTitle("Pyqt5 Tutorial")

\# Label Text

label = QLabel(win)

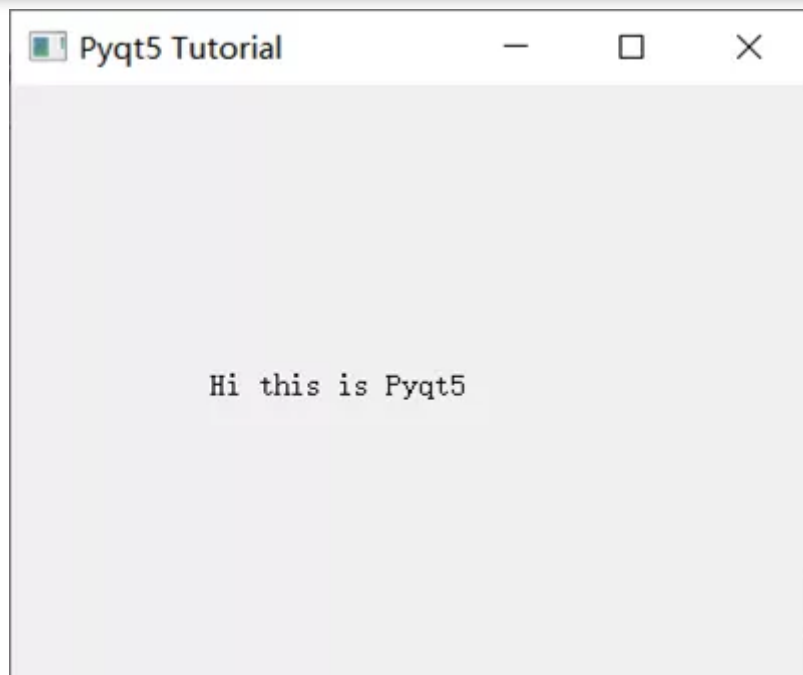
label.resize(200, 100)

label.setText("Hi this is PyQt5")

label.move(100, 100)

win.show()

sys.exit(app.exec_())
```



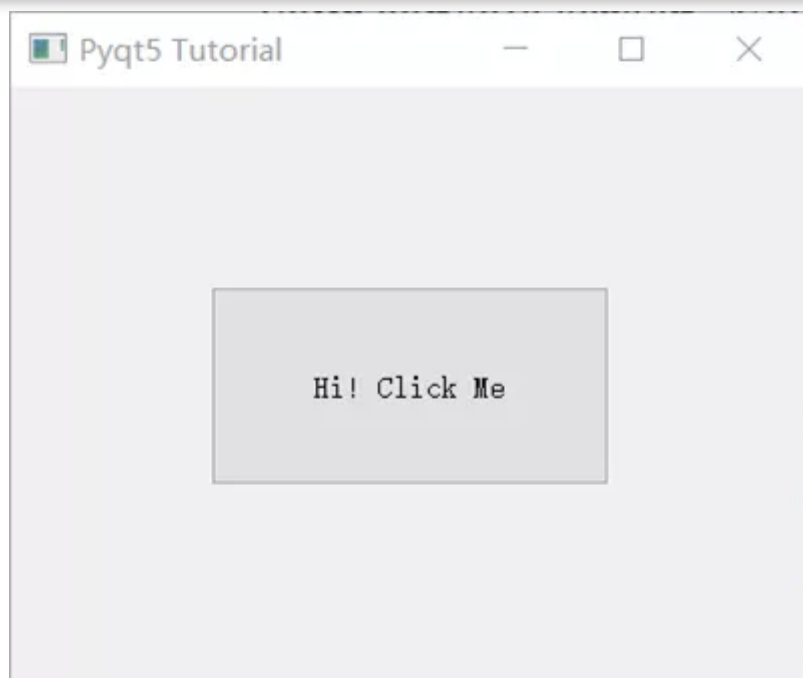
按鈕與事件：

```
import sys

from PyQt5.QtWidgets import QApplication, QMainWindow, QPushButton
```



```
def click():  
  
    print("Hy Button is clicked!")  
  
app = QApplication(sys.argv)  
  
win = QMainWindow()  
  
win.setGeometry(400, 400, 400, 300)  
  
win.setWindowTitle("Pyqt5 Tutorial")  
  
\# Button  
  
button = QPushButton(win)  
  
button.resize(200, 100)  
  
button.setText("Hi! Click Me")  
  
button.move(100, 100)  
  
button.clicked.connect(click)  
  
win.show()  
  
sys.exit(app.exec_())
```

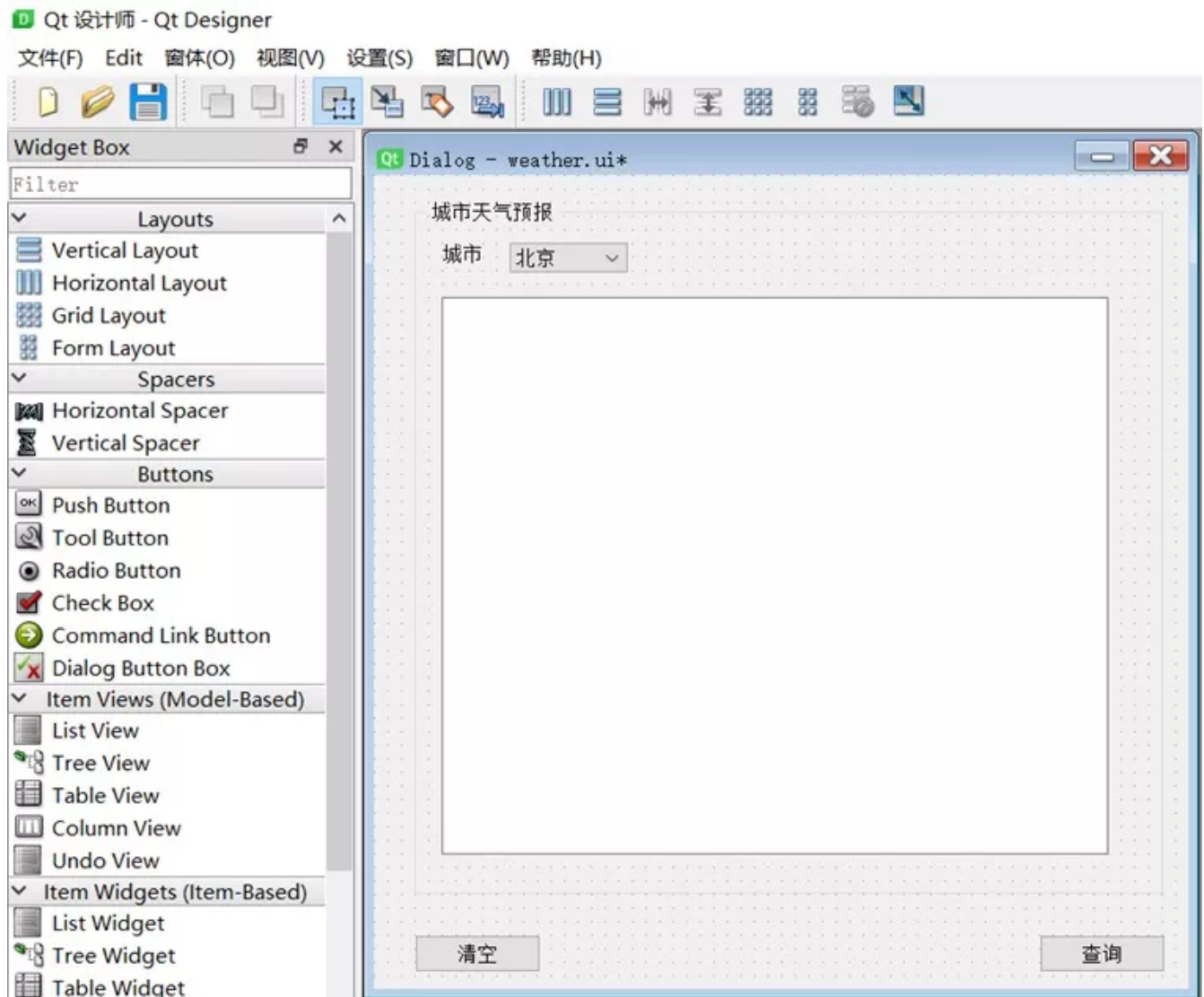


`button.clicked.connect()` 在按鈕點擊後執行特定的事件。

PyQt5實戰

實戰項目：簡易的天氣查詢軟件

1、使用Qt Designer設計一個界面



用到的控件有Button, GroupBox, Label, ComboBox, TextEdit，同時定義了兩個按鈕queryBtn及clearBtn，分別用來查詢及清空天氣數據。我們需要綁定槽函數，方法如下：

- 在Qt Designer右下角選擇信號/槽編輯器，點擊+號新增
- 分別選擇queryBtn及clearBtn，選擇信號clicked()，接收者Dialog 及槽accept()，（槽函數這裡不知道如何定義，後期在代碼裡再進行修改）

以上完成後保存為Weather.ui文件。

2、轉換.ui文件為.py文件

PyQt5支持直接使用.ui文件：

```
import sys

from PyQt5 import QtWidgets, uic

app = QtWidgets.QApplication(sys.argv)

window = uic.loadUi("mainwindow.ui")

window.show()

app.exec()
```

但是為了更好的自定義及修改上面的槽函數，可以使用External Tools – PyUIC，即可生成Weather.py，實際運行命令如下：

```
D:\Program Files\Python36\python.exe -m PyQt5.uic.pyuic Weather.ui -o Weather.py
```

其中，我們需要把兩個按鈕綁定的槽函數：

```
\# self.queryBtn.clicked.connect(Dialog.accept)

\# self.clearBtn.clicked.connect(Dialog.accept)

\# 修改为：

self.queryBtn.clicked.connect(Dialog.queryWeather)

self.clearBtn.clicked.connect(Dialog.clearText)
```

最終的Weather.py內容如下：

```
\# -*- coding: utf-8 -*-

\# Form implementation generated from reading ui file 'Weather.ui'
```

```
\#

\# Created by: PyQt5 UI code generator 5.15.4

\#

\# WARNING: Any manual changes made to this file will be lost when pyuic5 is

\# run again. Do not edit this file unless you know what you are doing.

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_Dialog(object):

    def setupUi(self, Dialog):

        Dialog.setObjectName("Dialog")

        Dialog.resize(600, 600)

        self.groupBox = QtWidgets.QGroupBox(Dialog)

        self.groupBox.setGeometry(QtCore.QRect(30, 20, 551, 511))

        self.groupBox.setObjectName("groupBox")

        self.label_2 = QtWidgets.QLabel(self.groupBox)

        self.label_2.setGeometry(QtCore.QRect(20, 30, 31, 16))

        self.label_2.setObjectName("label_2")

        self.comboBox = QtWidgets.QComboBox(self.groupBox)

        self.comboBox.setGeometry(QtCore.QRect(70, 30, 87, 22))

        self.comboBox.setObjectName("comboBox")

        self.comboBox.addItem("")

        self.comboBox.addItem("")

        self.comboBox.addItem("")

        self.textEdit = QtWidgets.QTextEdit(self.groupBox)

        self.textEdit.setGeometry(QtCore.QRect(20, 70, 491, 411))

        self.textEdit.setObjectName("textEdit")
```

```
self.queryBtn = QtWidgets.QPushButton(Dialog)

self.queryBtn.setGeometry(QtCore.QRect(490, 560, 93, 28))

self.queryBtn.setObjectName("queryBtn")

self.clearBtn = QtWidgets.QPushButton(Dialog)

self.clearBtn.setGeometry(QtCore.QRect(30, 560, 93, 28))

self.clearBtn.setObjectName("clearBtn")

self.retranslateUi(Dialog)

self.clearBtn.clicked.connect(Dialog.clearText)

self.queryBtn.clicked.connect(Dialog.queryWeather)

QtCore.QMetaObject.connectSlotsByName(Dialog)

def retranslateUi(self, Dialog):

    _translate = QtCore.QCoreApplication.translate

    Dialog.setWindowTitle(_translate("Dialog", "Dialog"))

    self.groupBox.setTitle(_translate("Dialog", "城市天气预报"))

    self.label_2.setText(_translate("Dialog", "城市"))

    self.comboBox.setItemText(0, _translate("Dialog", "北京"))

    self.comboBox.setItemText(1, _translate("Dialog", "苏州"))

    self.comboBox.setItemText(2, _translate("Dialog", "上海"))

    self.queryBtn.setText(_translate("Dialog", "查询"))

    self.clearBtn.setText(_translate("Dialog", "清空"))
```

3、調用MainDialog

在MainDialog中調用界面類Ui_Dialog，然後在其中中添加查詢天氣的業務邏輯代碼，這樣就做到了界面顯示和業務邏輯的分離。新增demo.py文件，在MainDialog類中定義了兩個槽函數

queryWeather()和clearText(),以便在界面文件Weather.ui中定義的兩個按鈕(queryBtn 和 clearBtn) 觸發clicked 信號與這兩個槽函數進行綁定。

完整代碼如下：

```
import sys

import Weather

from PyQt5.QtWidgets import QApplication, QDialog

import requests

class MainDialog(QDialog):

    def __init__(self, parent=None):

        super(QDialog, self).__init__(parent)

        self.ui = Weather.Ui_Dialog()

        self.ui.setupUi(self)

    def queryWeather(self):

        cityName = self.ui.comboBox.currentText()

        cityCode = self.getCode(cityName)

        r = requests.get(

            "https://restapi.amap.com/v3/weather/weatherInfo?key=f4fd5b287b6d7d51a3c60fee24e42002&cityCode)")

        if r.status_code == 200:

            data = r.json()['lives'][0]

            weatherMsg = '城市：{}\n天气：{}\n温度：{}\n风向：{}\n风力：{}\n湿度：{}\n发布时间：{}\n'.format(

                data['city'],

                data['weather'],

                data['temperature'],

                data['winddirection'],
```

```
        data['windpower'],

        data['humidity'],

        data['reporttime'],

    )

    else:

        weatherMsg = '天气查询失败，请稍后再试！'

        self.ui.textEdit.setText(weatherMsg)

def getCode(self, cityName):

    cityDict = {"北京": "110000",

                "苏州": "320500",

                "上海": "310000"}

    **return** cityDict.get(cityName, '101010100')

def clearText(self):

    self.ui.textEdit.clear()

if __name__ == '__main__':

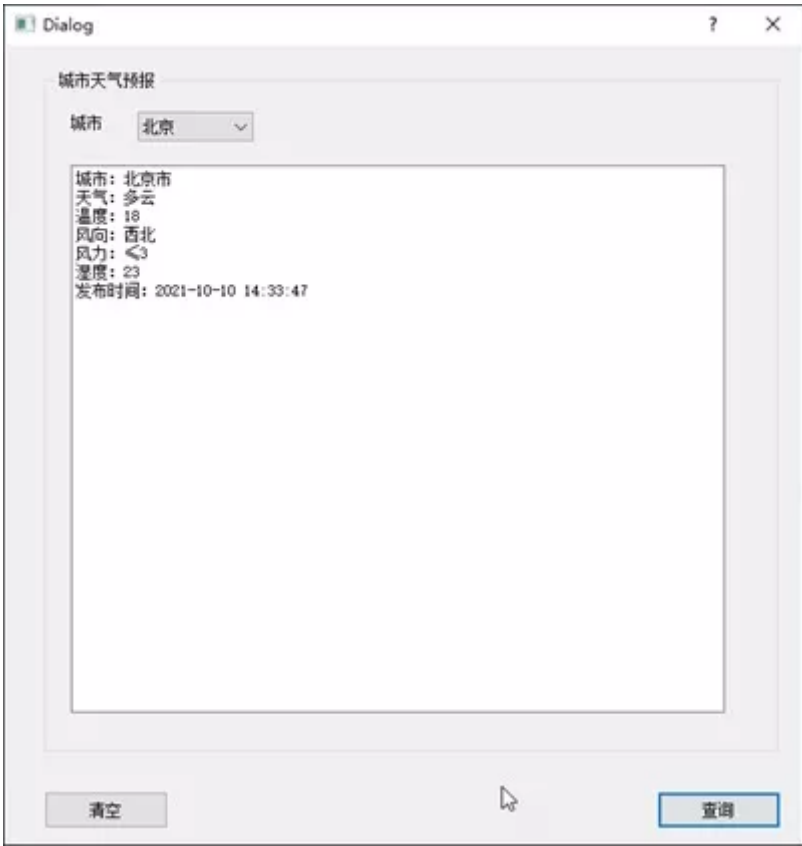
    myapp = QApplication(sys.argv)

    myDlg = MainDialog()

    myDlg.show()

    sys.exit(myapp.exec_())
```

運行demo.py並執行查詢後的效果：



4、將代碼打包成exe文件

將.py文件打包成可執行的exe在Python中稱為freezing，常用的工具有：PyInstaller, py2exe, cx_Freeze, bbfreeze, py2app等。功能對比：

		fbs	py2app	py2exe	pyinstaller	appimage
windows	win10					
	win8					
	win7				should	
	xp					
mac						
linux						全linux发行版
交叉编译						

- py2exe：軟件更新已經不活躍，因此也就略過。
- pyinstaller：明確支持win8、win10、理論上支持win7，支持apple MacOS, linux。
pyinsaller可以打包成文件夾形式內含exe入口執行文件的形式，也可以是一個單獨的exe文件。
- fbs^[15]：基於PyInstaller，使用起來更加方便

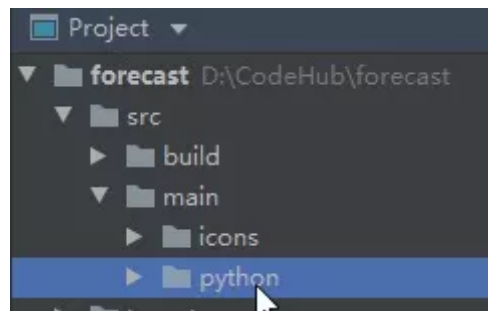
這裡選擇了fbs來打包。fbs的安裝方法：


```
pip install fbs
```

使用方法，在命令行中輸入：

```
fbs startproject
```

執行完成後需要輸入一些APP的名稱等。完成後會生成如下目錄：



將剛才編寫的PyQt5的代碼（demo.py和Weather.py）拖到src/main/python文件夾下，刪除原有的main.py，並將demo.py修改為main.py。然後打開main.py，在文件頭部添加如下代碼：

```
from fbs_runtime.application_context.PyQt5 import ApplicationContext
```

完成後執行：

```
fbs freeze
```

即可實現打包。生成的exe可執行文件在\target\MyApp文件下。

參考資料

- [1] PyQt5: <https://riverbankcomputing.com/software/pyqt/intro>
- [2] Qt: <https://www.qt.io/>
- [3] Pyside6: <https://doc.qt.io/qtforpython/index.html>

- [4] Tkinter: <https://wiki.python.org/moin/TkInter>
- [5] PySimpleGUI: <https://github.com/PySimpleGUI/PySimpleGUI>
- [6] WxPython: <https://wxpython.org/>
- [7] Wax: <https://pypi.org/project/wax/>
- [8] Kivy: <https://kivy.org/#home>
- [9] BeeWare: <https://beeware.org/>
- [10] Toga: <https://github.com/pybee/toga>
- [11] Eel: <https://github.com/ChrisKnott/Eel>
- [12] Flexx: <https://github.com/zooflO/flexx>
- [13] pywebview: <https://github.com/r0x0r/pywebview/>
- [14] enaml: <https://github.com/nucleic/enaml>
- [15] fbs: <https://build-system.fman.io/pyqt5-tutorial>

往期內容：

再見，Teamviewer！ 這款國產輕量級遠程桌面軟件超牛逼！

我在美團的八年

厲害了！ 推薦一個Web 端自動化神器- Automa

深耕Python快10年了，積累了不少好書，準備了一份Python經典電子書給廣大粉絲。大家有需要的添加下方微信，暗號「**p y資料**」，我私發給大家哈



暗號「**py資料**」獲取哦

喜歡此內容的人還喜歡

從Java 9 到Java 17之Java 13

碼農小胖哥



頂級Javaer，常用的14 個類庫

Java中文社群



帶薪搞個人項目：在vscode.dev 中直接運行Python ！ 純瀏覽器環境，無後端！

HJ說

