
PyQt5 实现界面和逻辑代码分离

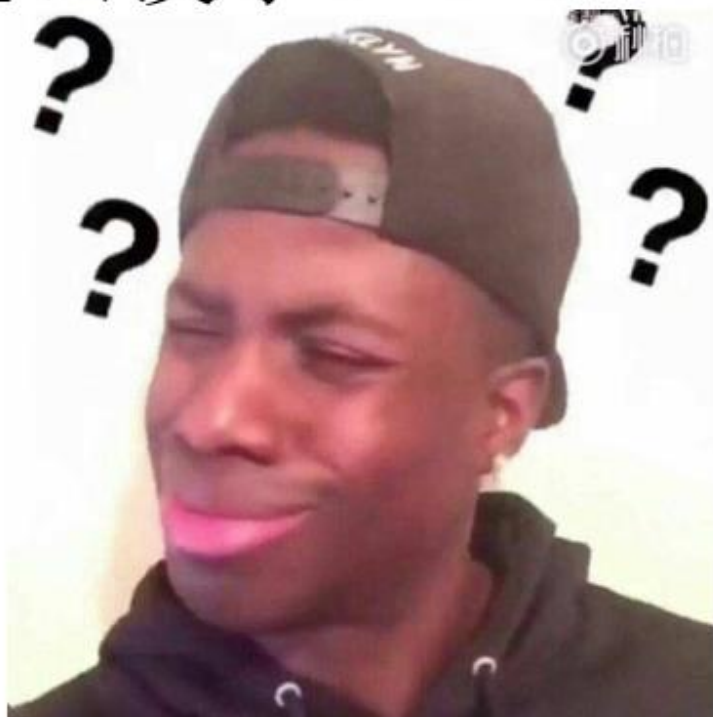
PyQt5 作为最强大的 Python GUI 设计模块，小编在这里告诉大家：PyQt5 可以实现界面与逻辑代码分离，实现非常简单，用处特别大！

初学者在学习 PyQt5 时，使用 Qt Designer 设计器设计完 GUI 窗体之后，将其转换为.py 代码文件，通常都会将逻辑代码直接写入到转换后的.py 文件中，类似这样：

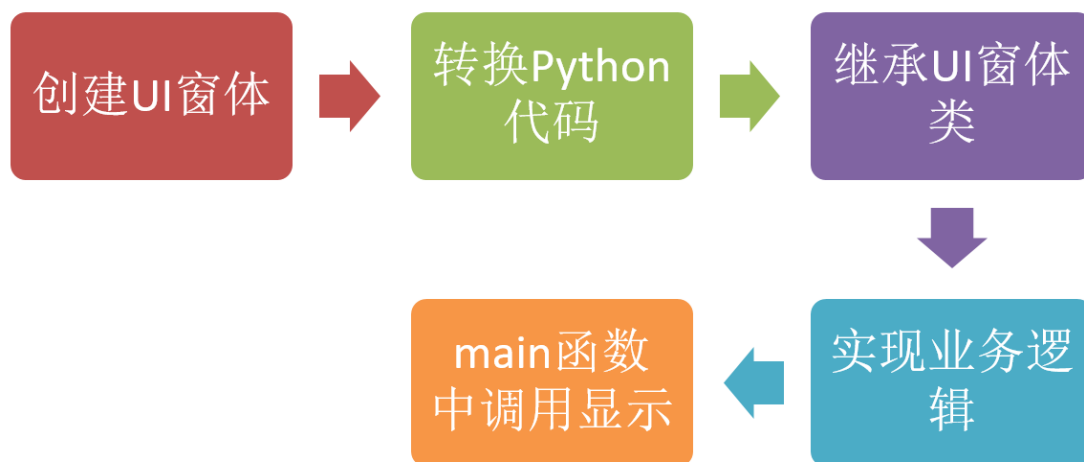
```
from PyQt5 import QtCore, QtGui, QtWidgets
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(290, 117)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QtWidgets.QMenuBar(MainWindow)
        self.menubar.setGeometry(QtCore.QRect(0, 0, 290, 23))
        self.menubar.setObjectName("menubar")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
        self.statusbar.setObjectName("statusbar")
        MainWindow.setStatusBar(self.statusbar)
        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)
    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow",
"MainWindow"))
import sys
# 主方法，程序从此处启动 PyQt 设计的窗体
if __name__ == '__main__':
    app = QtWidgets.QApplication(sys.argv)
    app.setStyle("Fusion") # 设置窗口风格
    MainWindow = QtWidgets.QMainWindow() # 创建窗体对象
    ui = Ui_MainWindow() # 创建 PyQt 设计的窗体对象
    ui.setupUi(MainWindow) # 对窗体对象进行初始化设置
    MainWindow.show() # 显示窗体
    sys.exit(app.exec_()) # 程序关闭时退出进程
```

代码运行并没有什么问题，但是，如果需要修改 UI 文件，并重新转换.py 文件，那么你之前写过的逻辑代码将会如滔滔江水，一去不复返……

辛辛苦苦写了两周的代码 就这么没了!!!



为了避免这种尴尬，我们在开发 PyQt5 程序时，可以使用 UI 与逻辑代码分离的方式进行开发，具体步骤如下图。



上图中的前两步就是正常的设计窗体和转换.py 代码操作，最后 3 步需要我们新创建一个用于编写逻辑代码的 Python 代码文件，然后在其中继承自动生成的 UI 窗体类，并编写相应的业务逻辑代码和 main 函数，参考代码如下：

```
from PyQt5 import QtWidgets, QtGui, QtCore
from py 文件名 import 类名
class MainWindow(QtWidgets.QMainWindow, 类名):
```

```
def __init__(self, parent=None):
    super(MainWindow, self).__init__(parent)
    self.setupUi(self)
    # 此处编写业务逻辑代码
if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    mainWindow = MainWindow()
    mainWindow.show()
    sys.exit(app.exec_())
```

通过这种方法，以后再也不用怕修改 UI 窗体了……