

PyQt5

2021 年 6 月 18 日

目录

1	实例开启学习	3
1.1	转换文件格式	3
1.2	布局	4

1 实例开启学习

实例

```
1 import sys
2 from PyQt5.QtWidgets import QApplication, QWidget
3
4 if __name__ == '__main__':
5     # 创建类的实例
6     app = QApplication(sys.argv)
7     # 创建一个窗口
8     w = QWidget()
9     # 设置窗口的尺寸
10    w.resize(400, 200)
11    # 移动窗口
12    w.move(300, 300)
13
14    # 设置窗口的标题
15    w.setWindowTitle('第一个基于PyQt5的桌面应用')
16    # 显示窗口
17    w.show()
18
19    # 进入程序的主循环，并通过exit函数确保主循环安全结束
20    sys.exit(app.exec_())
```

以上的案例为自己打印的，为了生成更为复杂的界面，要调用 Anaconda 中专门生成界面的软件 QtDesigner 命名为 Start Designer，路径为 $D : \backslash \text{Anaconda3} \backslash \text{Library} \backslash \text{bin} \backslash \text{designer.exe}$

文件菜单中的设置，工具——外部工具。添加新的外部工具，名称和描述自己加，程序指定上述路径，参数可写：工作目录指定要运行 Qt designer 的工作程序目录，

在项目对话框中单击右键，选定 External tools 中所加的外部工具，就能调用起 Qt designer。

1.1 转换文件格式

将.ui 格式文件 (Qt designer 保存格式) 转换为.py 格式。

调用终端在项目所在的目录下运行 `pyuic5 demo.ui -o demo.py` 还有更简便的是将上述的程序路径改为 $D : \backslash \text{Anaconda3} \backslash \text{Library} \backslash \text{bin} \backslash \text{pyuic5.bat}$ 。参数改为 $\$FileName\$ -o \$FileNameWithoutExtension\$.py$ 。工作目录改

为 $\$FileDir\$$ 。之后直接选中要转换的.ui 文件之后右键调用 External tools 中的外部工具。

1.2 布局

1. 水平布局

选中要布局的对话框，单机右键，选定布局 (Horizontal Layout)，选定水平布局。也能选定布局后，直接将元素拖进趋于内，注意要沿着边线放置，出现蓝线后放置。完成水平设置布局后保存为 text.ui，转换文件格式生成 text.py 文件。

水平布局

```
1 import sys
2 import text
3 from PyQt5.QtWidgets import QApplication,QMainWindow
4
5 if __name__ == '__main__':
6     #创建类的实例
7     app = QApplication(sys.argv)
8     #创建一个窗口
9     mainWindow = QMainWindow()
10    #向主窗口上添加控件
11    ui = text.Ui_MainWindow()
12    ui.setupUi(mainWindow)
13    mainWindow.show()
14
15    #进入程序的主循环，并通过exit函数确保主循环安全结束
16    sys.exit(app.exec_())
```

2. 垂直布局

选中要布局的对话框，单机右键，选定布局，选定垂直布局 (Vertical Layout)。也能选定布局后，直接将元素拖进趋于内，注意要沿着边线放置，出现蓝线后放置。完成水平设置布局后保存为 text.ui，转换文件格式生成 text.py 文件。

垂直布局

```
1 import sys
2 import text
3 from PyQt5.QtWidgets import QApplication,QMainWindow
```

```

4
5 if __name__ == '__main__':
6     #创建类的实例
7     app = QApplication(sys.argv)
8     #创建一个窗口
9     mainWindow = QMainWindow()
10    #向主窗口上添加控件
11    ui = text.Ui_MainWindow()
12    ui.setupUi(mainWindow)
13    mainWindow.show()
14
15    #进入程序的主循环，并通过exit函数确保主循环安全结束
16    sys.exit(app.exec_())

```

3. 网格布局

选中要布局的对话框，单机右键，选定布局，选定网格布局 (Grid Layout)。也能选定布局后，直接将元素拖进趋于内，注意要沿着边线放置，出现蓝线后放置。完成水平设置布局后保存为 text.ui，转换文件格式生成 text.py 文件。

网格布局

```

1 import sys
2 import text
3 from PyQt5.QtWidgets import QApplication,QMainWindow
4
5 if __name__ == '__main__':
6     #创建类的实例
7     app = QApplication(sys.argv)
8     #创建一个窗口
9     mainWindow = QMainWindow()
10    #向主窗口上添加控件
11    ui = text.Ui_MainWindow()
12    ui.setupUi(mainWindow)
13    mainWindow.show()
14
15    #进入程序的主循环，并通过exit函数确保主循环安全结束
16    sys.exit(app.exec_())

```

4. 表单布局

选中要布局的对话框，单机右键，选定布局，选定表单布局 (Form Layout)。也能选定布局后，直接将元素拖进趋于内，注意要沿着边线放置，出现蓝线

后放置。完成水平设置布局后保存为 text.ui，转换文件格式生成 text.py 文件。

表单布局

```
1 import sys
2 import text
3 from PyQt5.QtWidgets import QApplication, QMainWindow
4
5 if __name__ == '__main__':
6     # 创建类的实例
7     app = QApplication(sys.argv)
8     # 创建一个窗口
9     mainWindow = QMainWindow()
10    # 向主窗口上添加控件
11    ui = text.Ui_MainWindow()
12    ui.setupUi(mainWindow)
13    mainWindow.show()
14
15    # 进入程序的主循环，并通过exit函数确保主循环安全结束
16    sys.exit(app.exec_())
```