

专栏首页 恩蓝脚本 python GUI库图形界面开发之PyQt5拖放控件实例详解

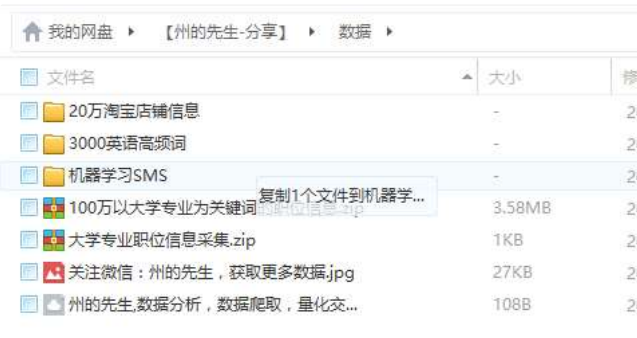
# python GUI库图形界面开发之PyQt5拖放控件实例详解

发布于2020-11-05 15:25:28 阅读 1.8K

本篇，我们学习PyQt5界面中拖放（Drag 和Drop）控件。

## 拖放动作

在GUI中，拖放指的是点击一个对象，并将其拖动到另一个对象上的动作。比如百度云PC客户端支持的拖放文件以快速移动文件：

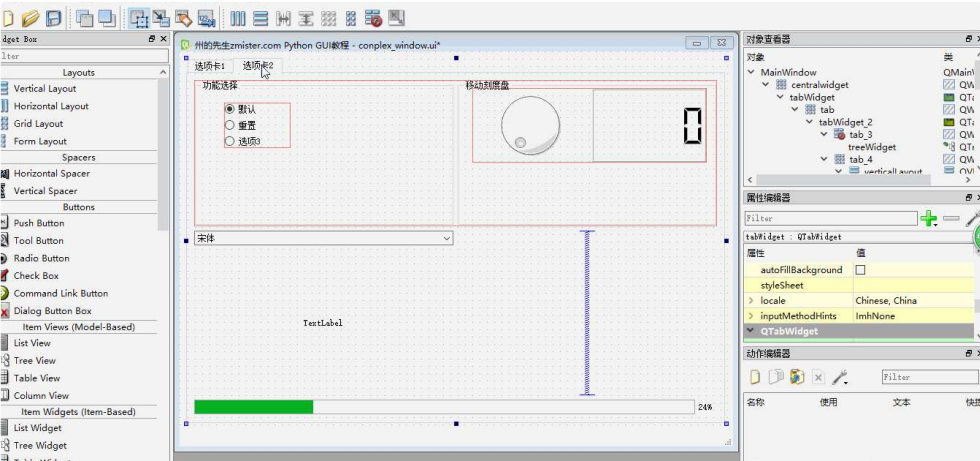


拖放动作能够很直观很方便的在GUI程序中完成一些很复杂或繁琐的操作。

## 在PyQt5中实现拖放

在PyQt5中，我们也可以很轻松地使用拖放功能。

使用Qt设计师或者使用API都可以实现。我们先使用Qt设计师将GUI的图形设计出来，在之前的GUI的基础上，我们新建一个选项卡。



我们新建了一个选项卡，然后在里面放置了一个LineEdit部件，一个PushButton部件，两个ListWidget部件。

对于简单的拖放效果，我们可以直接使用Qt设计师中的选项进行设置。例如，我们直接使用dragEnable属性、dragDropOverwriteMode属性、dragDropMode属性为ListWidget部件设置拖放功能：

### 作者介绍



砸漏

工程师项目总监

关注

文章	阅读量	获
12.6K	9.7M	31.1

### 精选专题

腾讯云原生专题  
云原生技术干货地。

### 活动推荐

云安全最佳实践-创作...  
火热征文中，发布文章赢千元好礼！

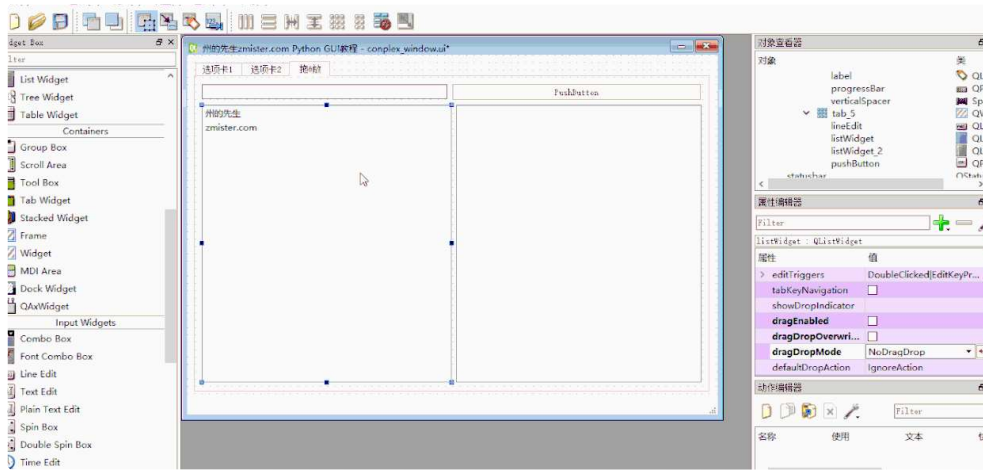
腾讯云自媒体分享计划  
入驻腾讯云开发者社区，共享百万资源包。

### 运营活动



### 目录

- 拖放动作
- 在PyQt5中实现拖放



而一些稍微复杂的拖放功能，就需要编写Python逻辑处理代码来完成了。

我们先将UI文件保存并转换为Python文件。

```
1 | pyuic5 -o complex_window_drag.py complex_window.ui
```

然后，新建一个Python文件drag.py，在文件中引入刚刚转换好的Python文件：

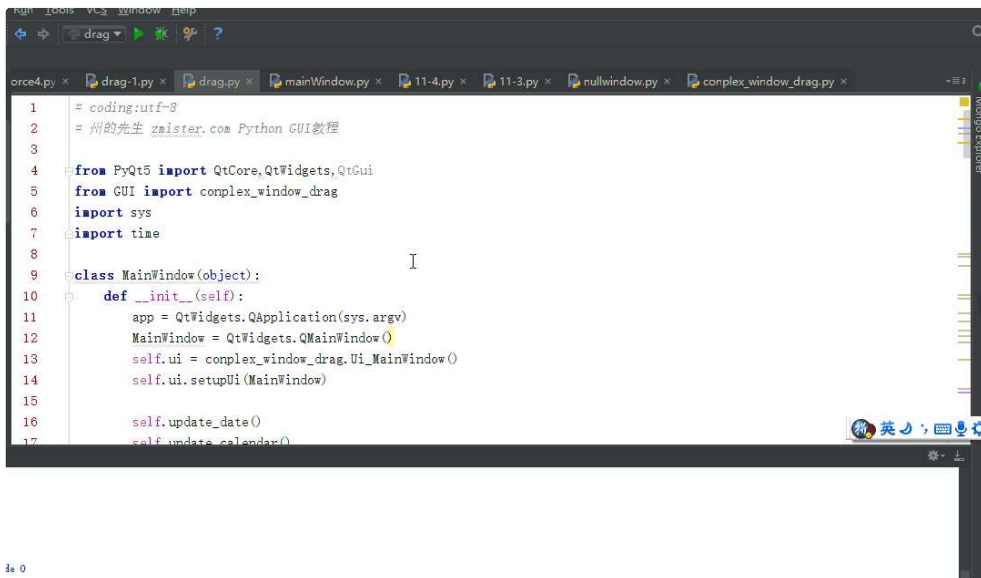
```
1 | # coding:utf-8
2 | # 州的先生 zmister.com Python GUI教程
3 | from PyQt5 import QtCore,QtWidgets,QtGui
4 | from GUI import complex_window_drag
5 | import sys
6 | import time
7 | class MainWindow(object):
8 |     def __init__(self):
9 |         app = QtWidgets.QApplication(sys.argv)
10 |         MainWindow = QtWidgets.QMainWindow()
11 |         self.ui = complex_window_drag.Ui_MainWindow()
12 |         self.ui.setupUi(MainWindow)
13 |         self.update_date()
14 |         self.update_calendar()
15 |         self.set_lcd()
16 |         self.set_dial()
17 |         self.update_progressbar()
18 |         self.set_font()
19 |         MainWindow.show()
20 |         sys.exit(app.exec_())
21 | # 修改日期修改器数值
22 | def update_date(self):
23 |     self.ui.dateEdit.setDate(self.ui.calendarWidget.selectedDate())
24 |     # 日历信号槽
25 | def update_calendar(self):
26 |     self.ui.calendarWidget.selectionChanged.connect(self.update_date)
27 |     # 设置LCD数字
28 |     def set_lcd(self):
29 |         self.ui.lcdNumber.display(self.ui.dial.value())
30 |     # 刻度盘信号槽
31 |     def set_dial(self):
32 |         self.ui.dial.valueChanged['int'].connect(self.set_lcd)
33 |     # 州的先生 zmister.com
34 |     # 按钮信号槽
35 |     def update_progressbar(self):
36 |         self.ui.radioButton.clicked.connect(self.start_progressbar)
37 |         self.ui.radioButton_2.clicked.connect(self.stop_progressbar)
38 |         self.ui.radioButton_3.clicked.connect(self.reset_progressbar)
39 |         self.progress_value = 0
40 |         self.stop_progress = False
41 |     def progressbar_counter(self, start_value=0):
42 |         self.run_thread = RunThread(parent=None, counter_start=start_value)
43 |         self.run_thread.start()
44 |         self.run_thread.counter_value.connect(self.set_progressbar)
45 |     def set_progressbar(self, counter):
46 |         if not self.stop_progress:
47 |             self.ui.progressBar.setValue(counter)
```

```

48 # 启动进度栏
49 def start_progressbar(self):
50     self.stop_progress = False
51     self.progress_value = self.ui.progressBar.value()
52     self.progressbar_counter(self.progress_value)
53 # 停止进度栏
54 def stop_progressbar(self):
55     self.stop_progress = True
56     try:
57         self.run_thread.stop()
58     except:
59         pass
60 # 重置进度栏
61 def reset_progressbar(self):
62     self.stop_progressbar()
63     self.progress_value = 0
64     self.ui.progressBar.reset()
65     self.stop_progress = False
66 # 字体选择
67 def set_font(self):
68     self.ui.fontComboBox.activated['QString'].connect(self.ui.label.setText)
69 class RunThread(QThread):
70     # 定义一个新的信号
71     counter_value = QtCore.pyqtSignal(int)
72     def __init__(self, parent=None, counter_start=0):
73         super(RunThread, self).__init__(parent)
74         self.counter = counter_start
75         self.is_running = True
76     def run(self):
77         while self.counter < 100 and self.is_running == True:
78             time.sleep(0.1)
79             self.counter += 1
80             print(self.counter)
81             # 发出一个新值的信号
82             self.counter_value.emit(self.counter)
83     def stop(self):
84         self.is_running = False
85         print('线程停止中...')
86         self.terminate()
87 if __name__ == "__main__":
88     MainWindow()

```

运行代码正常:



接着，我们创建一个DragDropButton()类，用来处理按钮的拖放：

```

1 class DragDropButton(QPushButton):
2     def __init__(self, text, parent):
3         super().__init__(text, parent)
4         self.setAcceptDrops(True)
5     def dragEnterEvent(self, event):

```

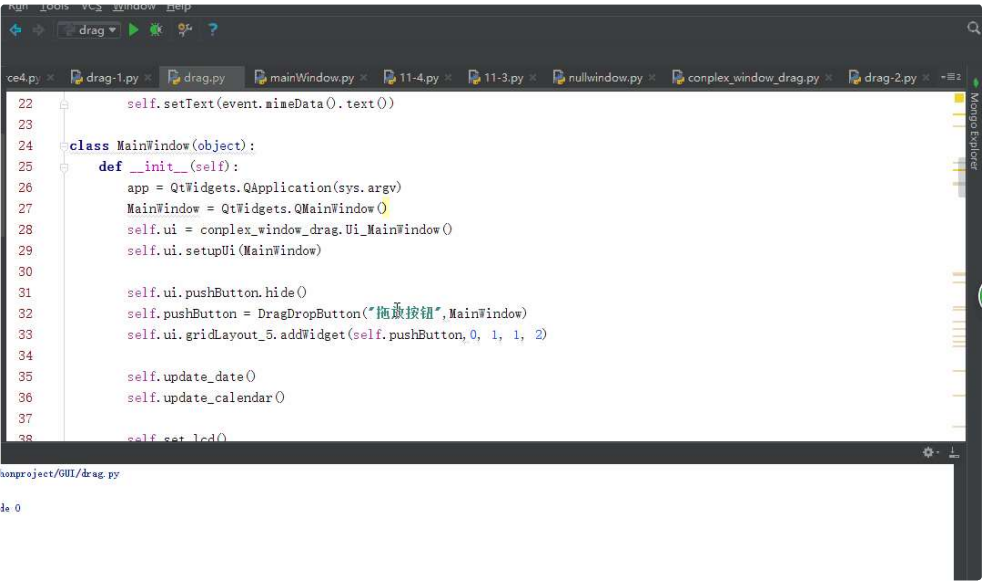
```
6 | if event.mimeData().hasFormat('text/plain'):
7 |     event.accept()
8 | else:
9 |     event.ignore()
10 | def dropEvent(self, event):
11 |     self.setText(event.mimeData().text())
```

我们使用setAcceptDrops属性设置按钮接收拖放事件，创建一个dragEnterEvent()方法用来设置拖的事件响应，创建一个dropEvent()方法用来设置放的事件响应。

接着我们在MainWindow()主类中，调用它：

```
1 | class MainWindow(object):
2 |     def __init__(self):
3 |         .....
4 |         self.ui.pushButton.hide()
5 |         self.pushButton = DragDropButton("拖放按钮",MainWindow)
6 |         self.ui.gridLayout_5.addWidget(self.pushButton,0, 1, 1, 2)
7 |         .....
```

最后，运行一下看看：



在上面的程序中，我们能够将文本拖放到按钮上。

好了python GUI库图形界面开发中PyQt5拖放控件的实例就是这些，更多关于python PyQt5 GUI库图形界面开发请查看下面的相关链接

本文参与 [腾讯云自媒体分享计划](#)，欢迎热爱写作的你一起参与！

本文分享自作者个人站点/博客：<https://www.zalou.cn> 复制

如有侵权，请联系 [cloudcommunity@tencent.com](mailto:cloudcommunity@tencent.com) 删除。

Qt

GUI

编程算法

Python

举报

点赞 2

分享

登录后参与评论

0 条评论



## 相关文章

### python GUI库图形界面开发之PyQt5布局控件QHBoxLayout详细使用...

采用QBOXLayout类可以在水平和垂直方向上排列控件，QHBoxLayout和QVBoxLayout类继承自QBoxLayout

🔍 砸漏

### python GUI库图形界面开发之PyQt5布局控件QGridLayout详细使用方法与实例

QGridLayout（网格布局）是将窗口分割成行和列的网格来进行排列，通常可以使用函数addWidget（）将被管理的控件（Widget）添加到窗口中，或者使...

🔍 砸漏

### python GUI库图形界面开发之PyQt5表格控件QTableView详细使用方法与实例

在通常情况下，一个应用需要和一批数据进行交互，然后以表格的形式输出这些信息，这时就需要用到QTableView类了，在QTableView中可以使用自定义的数据...

🔍 砸漏

### python GUI库图形界面开发之PyQt5布局控件QVBoxLayout详细使用方法与实例

在布局中要用到addStretch()函数，设置stretch伸缩量后，按比例分配剩余的空间

🔍 砸漏

### python GUI库图形界面开发之PyQt5单选按钮控件QRadioButton详细...

QRadioButton 继承自 QAbstractButton，其主要作用提供用户一些互斥的按钮。

🔍 砸漏

### python GUI库图形界面开发之PyQt5树形结构控件QTreeWidget详细使用方法与实例

QTreeWidget 使用类似于 QListView 类的方式提供一种典型的基于 item 的树形交互方法类，该类基于QT的“模型/视图”结构，提供了默认的模...

🔍 砸漏

### python GUI库图形界面开发之PyQt5图片显示控件QPixmap详细使用...

QPixmap类用于绘图设备的图像显示，它可以作为一个QPainterDevice对象，也可以加载到一个控件中，通常是标签或者按钮，用于在标签或按钮上显示图像

🔍 砸漏

### python GUI库图形界面开发之PyQt5状态栏控件QStatusBar详细使用...

MainWindow对象在底部保留有一个水平条，作为状态栏（QStatusBar），用于显示永久或临时的状态信息

🔍 砸漏

### python GUI库图形界面开发之PyQt5切换按钮控件QPushButton详细使...

QAbstractButton类为抽象类，不能实例化，必须由其他的按钮类继承QAbstractButton类，来实现不同的功能和表现形式，常见的按钮QPushButton...

砸漏

### python GUI库图形界面开发之PyQt5滑块条控件QSlider详细使用方法...

QSlider控件提供一个垂直或者水平的滑动条，滑动条是一个用于控制有界值典型的控件，它允许用户沿水平或者垂直方向在某一范围内移动滑块，并将滑块所在的位置转换为...

砸漏

### python GUI库图形界面开发之PyQt5打印控件QPrinter详细使用方法与...

打印图像是图像处理软件中的一个常用功能，打印图像实际上是在QPaintDevice中画图，与平时在QWidget、QPixmap和QImage中画图是一样的，都...

砸漏

### python GUI库图形界面开发之PyQt5窗口控件QWidget详细使用方法

基础窗口控件QWidget类是所有用户界面对象的基类，所有的窗口或者控件都直接或者间接的继承自QWidget类。

砸漏

### python GUI库图形界面开发之PyQt5计数器控件QSpinBox详细使用方...

QSPINBox是一个计数器控件，允许用户选择一个整数值通过单击向上向下或者键盘上的上下键来增加减少当前显示的值，当然用户也可以输入值

砸漏

### python GUI库图形界面开发之PyQt5滚动条控件QScrollBar详细使用方法与实例

可以看到，前面介绍的几个窗口控件的共同点是新建一些窗口来装载更多的控件，而QScrollBar提供了另一种思路：这个控件提供水平的或垂直的滚动条，这样可以扩大当...

砸漏

### python GUI库图形界面开发之PyQt5选项卡控件QTabWidget详细使用方法与实例

QTabWidget控件提供了一个选项卡和一个页面区域，默认显示第一个选项卡的页面，通过单击各选项卡可以查看对应的界面，如果在一个窗口中显示的输入字段很多，则可...

砸漏

### python GUI库图形界面开发之PyQt5复选框控件QCheckBox详细使用...

在这个例子中，将三个复选框添加到一个水平布局管理器中，并添加到一个QGroupBox组中

砸漏

### python GUI库图形界面开发之PyQt5动态(可拖动控件大小)布局控件Q...

PyQt还提供了特殊的布局管理器QSplitter。它可以动态地拖动子控件之间的边界，算是一个动态的布局管理器，QSplitter允许用户拖动子控件的边界控制子...

砸漏

### python GUI库图形界面开发之PyQt5时间控件QTimer详细使用方法与...

如果在应用程序中周期性地地进行某项操作，比如周期性的检测主机的cpu值，则需要用到QTimer定时器，QTimer类提供了重复和单次的定时器，要使用定时器，需要先...

砸漏

[更多文章](#)

社区

文章

问答

技术沙龙

团队主页

平台

活动

自媒体分享计划

邀请作者入驻

自荐上首页

技术竞赛

资源

技术周刊

社区标签

开发者手册

开发者实验室

关于

视频介绍

社区规范

免责声明

联系我们

友情链接

热门产品

域名注册

云服务器

区块链服务

消息队列

网络加速

云数据库

域名解析

云存储

热门推荐

人脸识别

腾讯会议

企业云

CDN 加速

视频通话

图像分析

MySQL 数据库

SSL 证书

更多推荐

数据安全

负载均衡

短信

文字识别

云点播

商标注册

小程序开发

网站监控

Copyright © 2013 - 2022 Tencent Cloud. All Rights Reserved. 腾讯云 版权所有 京公网安备 11010802017518 粤B2-20090059-1