

4.3.1 QMimeData

QMimeData 用于描述可以存储在剪贴板中并且可以通过拖放机制传输的信息。**QMimeData** 不是一个 **QWidget** 控件，因此不能直接看到。**QMimeData** 对象将其拥有的数据与相应的 **MIME** 类型相关联，以确保信息可以在应用程序之间安全地传输，并且可以在同一应用程序内进行复制。**QMimeData** 类的继承结构如图 4-38 所示。

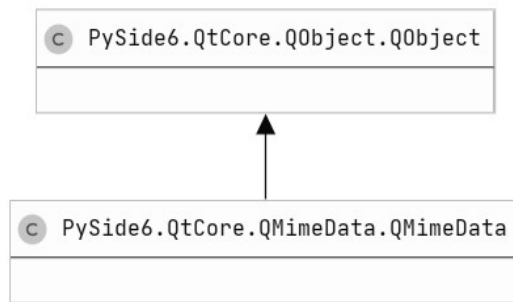


图 4-38

QMimeData 对象通常由 **QDrag** 对象或 **QClipboard** 对象创建，并作为数据传输通道提供给 **QDrag** 对象或 **QClipboard** 对象。这样，**Qt** 能够更好地管理它们使用的内存。

QMimeData 支持一些常见的格式，以便于调用，如表 4-21 所示。

表 4-21

判断函数	获取函数	设置函数	mimeType
hasText()	text()	setText()	text/plain
hasHtml()	html()	setHtml()	text/html
hasUrls()	urls()	setUrls()	text/uri-list
hasImage()	imageData()	setImageData()	image/*
hasColor()	colorData()	setColorData()	application/x-color

常见的 **QMimeData** 的使用方式如下：

```

class Button(QPushButton):
    def __init__(self, title, parent):
        super().__init__(title, parent)
        self.setAcceptDrops(True)

    def dragEnterEvent(self, e):
        if e.mimeData().hasFormat("text/plain"):
            e.accept()
        else:
            e.ignore()

    def dropEvent(self, e):
        self.setText(e.mimeData().text())
  
```

这是一个对按钮的拖曳事件。在上述代码中，**dragEnterEvent()** 函数中的 **e** 是

QDragEnterEvent 类型，拖曳动作进入该按钮时会触发该事件；**dropEvent()**函数中的 **e** 是 **QDropEvent** 类型，拖曳动作在按钮上被释放时会触发该事件。两个 **e.mimeData()**函数都是 **QMimeData** 类型的，存储拖曳的数据信息。该代码表示在一个拖曳操作中，**QMimeData** 如果判断拖曳的是文本就会接受这个拖曳操作，并且在拖曳释放的时候设置按钮的 **text** 为拖曳的文本。

QMimeData 中的常见格式如果无法满足需求，则可以使用自定义的格式 (**mimeType**)。一个 **QMimeData** 对象可以同时使用几种不同的 **mimeType** 存储相同的数据；**formats()**函数返回可用的 **mimeType** 列表；**data(mimeType:str)**函数返回 **mimeType** 对应的原始数据；如果使用其他 **mimeType**，则可以通过 **setData(mimeType,QByteArray)**函数新增或修改一个 **mimeType**。新增一个 **mimeType** 的代码如下：

```
class ButtonMyQMime(QPushButton):
    def __init__(self, title, parent):
        super().__init__(title, parent)
        self.setAcceptDrops(True)
        self.mime = QMimeData()
        qb = QByteArray(bytes('abcd1234', encoding='utf8'))
        self.mime.setData('my_mimetype', qb)

    def dragEnterEvent(self, e):
        if self.mime.hasFormat('my_mimetype'):
            e.accept()
        else:
            e.ignore()

    def dropEvent(self, e):
        self.setText('自定义 format 结果为: '+self.mime.data('my_mimetype').
data().decode('utf8'))
```

上述代码新增了一个 **mimeType**，并把按钮的 **text** 修改为 **mimeType** 对应的数据。实现的效果是只要拖曳到按钮，按钮的 **text** 自动修改为 **abcd1234**。需要注意的是，**setData** 接收的参数类型为 **QByteArray**，这是 Qt 支持的二进制格式，所以必须把数据转换成这种格式才可以通过 **QMimeData** 传输，这就增加了额外的工作量。实际上，对于上面的案例，也可以通过 **self.mime.setText('abcd1234')**传递字符串，同时使用 **self.mime.text()**函数获取字符串。

在 Windows 中，**QMimeData** 经常使用自定义 **mimeType** 存储数据，使用 **x-qt-windows-mime** 子类型指示它们表示非标准格式的数据。举例如下：

```
application/x-qt-windows-mime;value="FileGroupDescriptor"
application/x-qt-windows-mime;value="FileContents"
```

本节的案例给出了 Windows 系统中的 **txt** 文件完整的 **formats** 格式，由图 4-39 可以看到，除了 **text/uri-list** 是标准的 **QMimeData**，其他的都是自定义的 **mimeType**。