

## 【文本编辑器】三、文本编辑功能

原创 Qt 学习 Qt 学习 2020-02-15

点击上方 蓝色 文字，快来 关注 我吧！

基本的文本操作包括『复制』『剪切』『粘贴』『撤销』与『重做』，这些功能都由 `QTextEdit` 类提供，可以直接使用。与这些操作对应的工具条按钮的可用（激活）状态设置也在实现这些文本操作后实现。

### 本篇目录

1. 文本操作基础
2. 工具按钮激活状态设置

#### 运行环境：

win 10 + Qt 5.12.5 + Qt Creator 4.10

### 1. 文本操作基础

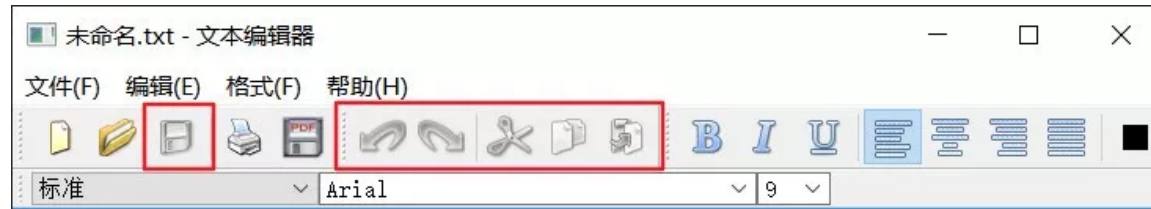
由于针对文本的『复制』『剪切』『粘贴』『撤销』与『重做』操作在 `QTextEdit` 类中都已经实现了，这里只需要调用相应的函数即可。在 `setupEditActions()` 函数中去掉这些操作对应动作后面的 `connect()` 语句前的注释符

```
1 void TextEdit::setupEditActions()
2 {
3     // 编辑主菜单动作集
4     ...
5     connect(actionUndo, &QAction::triggered, textEdit, &QTextEdit::undo);
6
7     ...
8     connect(actionRedo, &QAction::triggered, textEdit, &QTextEdit::redo);
9
10    ...
11    connect(actionCut, &QAction::triggered, textEdit, &QTextEdit::cut);
12
13    ...
14    connect(actionCopy, &QAction::triggered, textEdit, &QTextEdit::copy);
15
16    ...
17    connect(actionPaste, &QAction::triggered, textEdit, &QTextEdit::paste);
18
19    ...
20 }
```

运行程序，可在文本框中进行文本的『复制』『剪切』『粘贴』『撤销』与『重做』操作。

## 2. 工具按钮激活状态设置

可以发现，在文本框中对文本进行操作的时候，工具条中的按钮一直处于激活状态，即工具条按钮一直呈可点击的彩色图标状态。这里我们期望工具条按钮对应的功能在可用时呈激活状态，不可用时工具条按钮呈灰色的不可点击状态（非激活状态），如下图所示。



图中红色方框中的按钮处于非激活状态，其余按钮为激活状态

在 "textedit.cpp" 文件中添加头文件

```
1 #include <QClipboard>
2 #include <QMimeData>
```

在构造函数中添加如下代码：

```
1  TextEdit::TextEdit(QWidget *parent)
2      : QMainWindow(parent)
3  {
4      ...
5
6      connect(textEdit->document(), &QTextDocument::modificationChanged,
7              actionSave, &QAction::setEnabled);
8      connect(textEdit->document(), &QTextDocument::undoAvailable,
9              actionUndo, &QAction::setEnabled);
10     connect(textEdit->document(), &QTextDocument::redoAvailable,
11             actionRedo, &QAction::setEnabled);
12
13     actionSave->setEnabled(textEdit->document()->isModified());
14     actionUndo->setEnabled(textEdit->document()->isUndoAvailable());
15     actionRedo->setEnabled(textEdit->document()->isRedoAvailable());
16
17     actionCut->setEnabled(false);
18     connect(textEdit, &QTextEdit::copyAvailable, actionCut, &QAction::setEnabled);
19     actionCopy->setEnabled(false);
20     connect(textEdit, &QTextEdit::copyAvailable, actionCopy, &QAction::setEnabled);
```

```
21     connect(QApplication::clipboard(), &QClipboard::dataChanged, this, &TextEdit::clipboardDataChanged);
22
23     ...
24 }
```

完成『保存』『复制』『剪切』『撤销』与『重做』按钮的初始激活状态设置，并添加『保存』『复制』『剪切』『粘贴』『撤销』与『重做』按钮激活状态的关联函数。

其中『粘贴』按钮的初始激活状态设置在 `ditsetupEditActions()` 函数中完成

```
1 void TextEdit::setupEditActions()
2 {
3     ...
4
5     // 新建剪贴板 mimeType 对象，并根据剪贴板中是否存在文本设置粘贴按钮的激活状态
6     if (const QMimeData *md = QApplication::clipboard()->mimeType())
7         actionPaste->setEnabled(md->hasText());
8 }
```

这是因为『粘贴』按钮的激活状态是根据剪贴板中的文本状态来设置的。

为获取剪贴板中文本状态，在 "textedit.h" 文件中添加槽函数

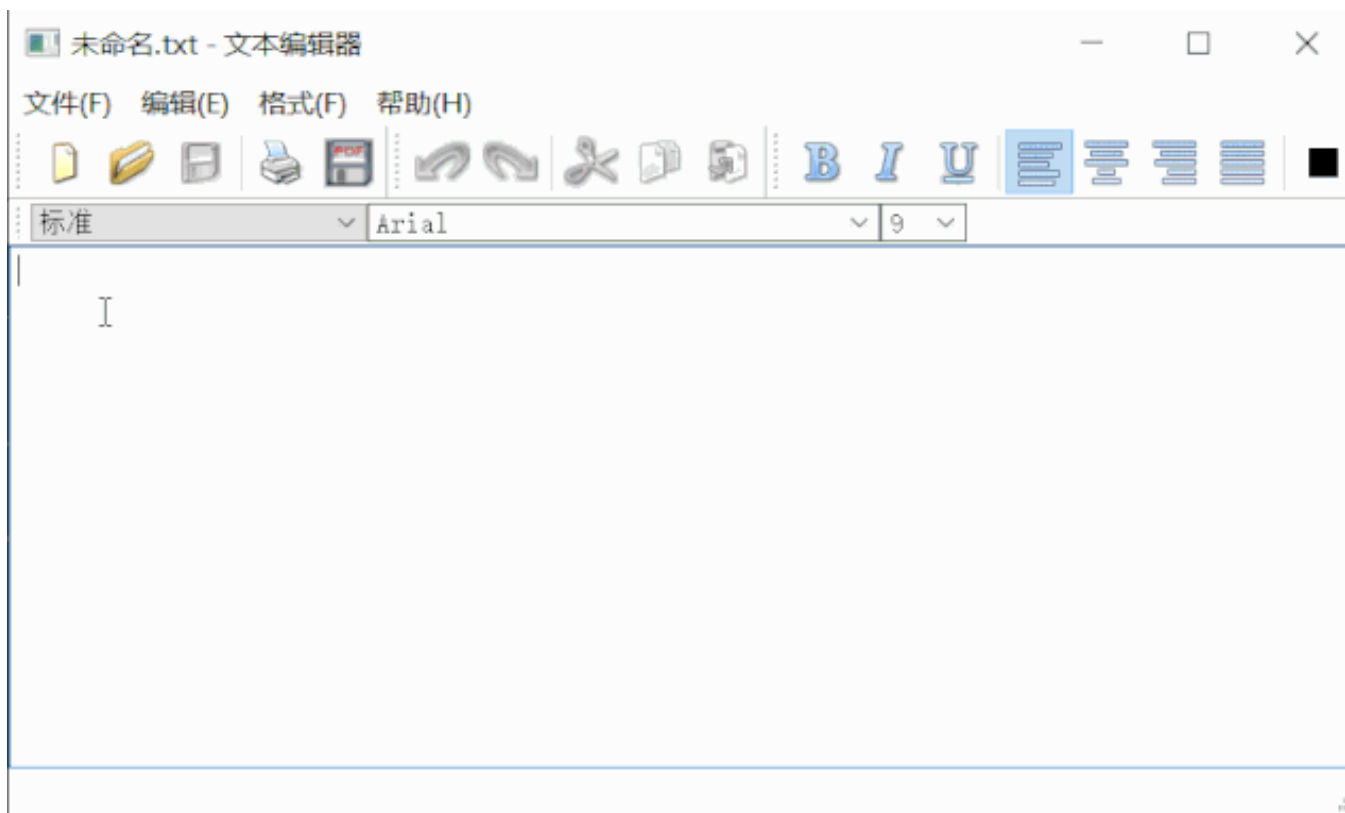
```
1 private slots:
2     clipboardDataChanged();
```

其实代码如下：

```
1 void TextEdit::clipboardDataChanged()
2 {
```

```
3     if (const QMimeData *md = QApplication::clipboard()->mimeTypeData())  
4         actionPaste->setEnabled(md->hasText());  
5 }
```

运行程序，效果如下

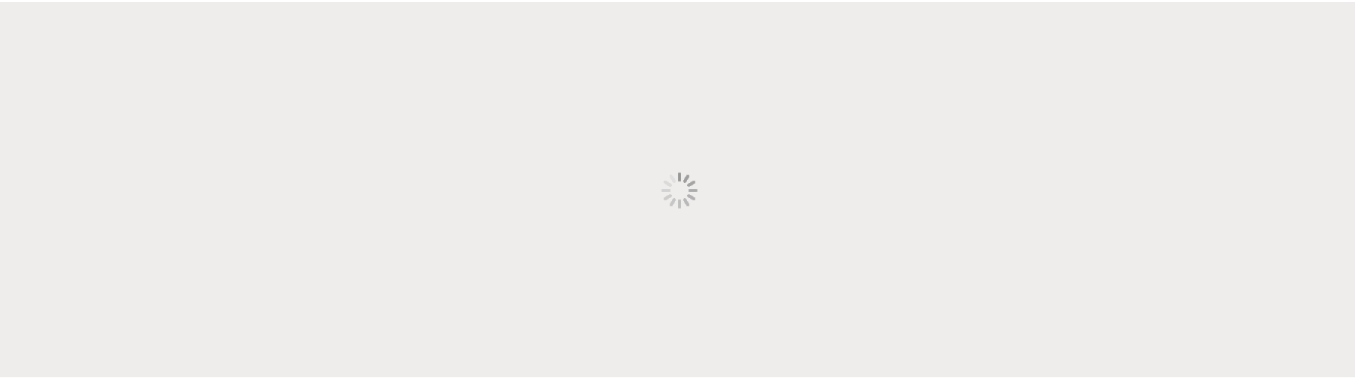


## 小结

基于 `QTextEdit` 类，本篇很轻松地就实现了文本的『复制』『剪切』『粘贴』『撤销』与『重做』操作，以及与这些操作对应的按钮可用（激活）状态设置。

相关阅读：

- 《【文本编辑器】二、文件操作功能（下）》
- 《【文本编辑器】二、文件操作功能（上）》
- 《【文本编辑器】一、界面设计》
- 《主要的窗体类和主窗体构成》
- 《Qt 模块简介》
- 《信号与槽》



喜欢此内容的人还喜欢

美国何时给国际社会一个交代？

央视军事



住大平层到底有多爽？

知乎日报



