

资料大全  
2019年最新!

html5全集

JavaScript全集

WEB前端全集

Bootstrap全集

VUE全集

angular全集

node.js全集

实战项目全集

PHP项目实战全集

java全集

java项目全集

Python实战全集

Linux实战项目

500本电子书集合

2019年面试题库

大小: 7,152.32 GB  
立即免费下载

# Qt QComboBox下拉框及用法（详解版）

< 上一页

下一页 >

**QComboBox** 是下拉列表框组件类，它提供一个下拉列表供用户选择，也可以直接当作一个 **QLineEdit** 用作输入。**QComboBox** 除了显示可见下拉列表外，每个项（item，或称列表项）还可以关联一个 **QVariant** 类型的变量，用于存储一些不可见数据。

实例 samp4\_6 演示 **QComboBox**（和 **QPlainTextEdit**，后续会讲）的使用，其运行时界面如图 1 所示。

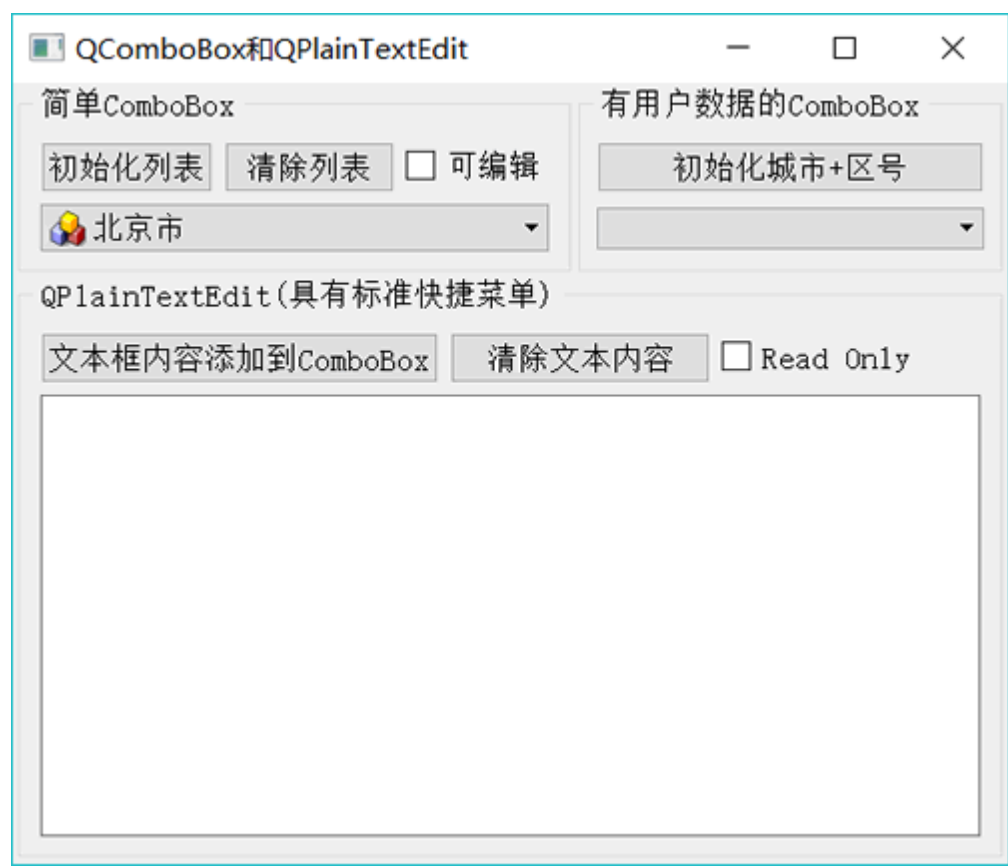


图 1 实例 samp4\_6 运行界面

QComboBox 主要的功能是提供一个下拉列表供选择输入。在界面上放置一个 QComboBox 组件后，双击此组件，可以出现如图 2 所示的对话框，对 QComboBox 组件的下拉列表的项进行编辑。在图 2 所示的对话框中，可以进行编辑，如添加、删除、上移、下移操作，还可以设置项的图标。

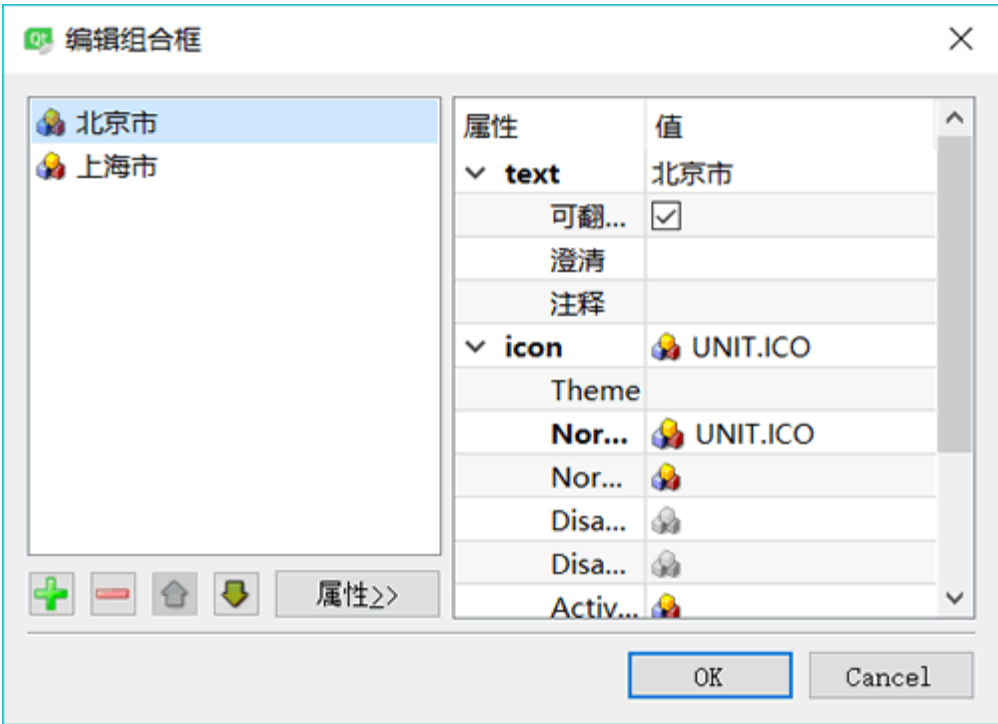


图 2 QComboBox 组件设计时的列表项编辑器

用代码添加简单项

窗口上的“初始化列表”按钮初始化下拉列表框的列表内容，其代码如下：

```
01. void Widget::on_btnIniItems_clicked()
02. { // "初始化列表" 按键
03.     QIcon icon;
04.     icon.addFile(":/images/icons/aim.ico");
05.
06.     ui->comboBox->clear(); // 清除列表
07.     for (int i=0; i<20; i++)
08.         ui->comboBox->addItem(icon, QString::asprintf("Item %d", i)); // 带图
// 标
09.         // ui->comboBox->addItem(QString::asprintf("Item %d", i)); // 不带图标
10. }
```

添加一个项时可以指定一个图标，图标来源于资源文件。

addItem() 用于添加一个列表项，如果只是添加字符串列表项，而且数据来源于一个 QStringList 变量，可以使用 addItems() 函数，示例代码如下：

```
01. ui->comboBox->clear();
02. QStringList strList;
03. strList<<"北京"<<"上海"<<"天津"<<"河北省"<<"山东省"<<"山西省";
```

```
04. ui->comboBox->addItem(strList);
```

添加具有用户数据的项 QComboBox::addItem() 函数的两种参数的原型定义如下：

```
void addItem (const QString &text, const QVariant &userData = QVariant())  
void addItem (const QIcon &icon, const QString &text, const QVariant &userData =  
QVariant())
```

不管是哪一个 addItem() 函数，后面都有一个可选的 QVariant 类型的参数 userData，可以利用这个变量存储用户定义数据。

界面上另一个 ComboBox 组件使用了用户数据，“初始化城市+区号”按钮的槽函数代码如下：

```
01. void Widget::on_btnIni2_clicked()  
02. { //初始化具有自定义数据的comboBox  
03.     //QMap自动根据 key排序  
04.     QMap<QString, int> City_Zone;  
05.     City_Zone.insert("北京",10);  
06.     City_Zone.insert("上海",21);  
07.     City_Zone.insert("天津",22);  
08.     City_Zone.insert("大连",411);  
09.     City_Zone.insert("锦州",416);  
10.     City_Zone.insert("徐州",516);  
11.     City_Zone.insert("福州",591);  
12.     City_Zone.insert("青岛",532);  
13.     ui->comboBox2->clear();  
14.     foreach(const QString &str, City_Zone.keys())  
15.         ui->comboBox2->addItem(str, City_Zone.value(str));  
16. }
```

这里定义了一个关联容器类 QMap<QString, int> City\_Zone，用于存储<城市，区号>映射表。为 City\_Zone 填充数据后，给 comboBox2 添加项时，使用了 foreach 关键字遍历 City\_Zone.keys()。添加项的语句如下：

```
ui->comboBox2->addItem(str, City_Zone.value(str));
```

城市名称作为项显示的字符串，电话区号作为项关联的用户数据，但是在列表框里只能看到城市名称。

需要注意的是，将 City\_Zone 的内容添加到列表框之后，列表框里显示的列表项的顺序与源程序中设置 City\_Zone 的顺序不一致，因为 QMap<Key,T> 容器类会自动按照 Key 排序。

### QComboBox列表项的访问

QComboBox 存储的项是一个列表，但是 QComboBox 不提供整个列表用于访问，可以通过索引访问某个项。访问项的一些函数主要有以下几种：



- `int currentIndex()`: 返回当前项的序号，第一个项的序号为0。
- `QString currentText()`: 返回当前项的文字。
- `QVariant currentData(int role = Qt::UserRole)`: 返回当前项的关联数据，数据的缺省角色为 `role = Qt::UserRole`，角色的意义在后续章节会详细介绍。
- `QString itemText(int index)`: 返回指定索引号的项的文字。
- `QVariant itemData(int index, int role = Qt::UserRole)`: 返回指定索引号的项的关联数据。
- `int count()`: 返回项的个数。

在一个 QComboBox 组件上选择项发生变化时，会发射如下两个信号：

```
void currentIndexChanged(int index)
void currentIndexChanged(const QString &text)
```

这两个信号只是传递的参数不同，一个传递的是当前项的索引号，一个传递的当前项的文字。

为使用方便，选择为 `currentIndexChanged(const QString &text)` 信号编写槽函数。窗体上只存储字符串列表的 `comboBox` 的槽函数代码如下：

```
01. void Widget::on_comboBox_currentIndexChanged(const QString &arg1)
02. {
03.     ui->plainTextEdit->appendPlainText(arg1);
04. }
```

关联有城市区号的 `comboBox2` 的槽函数代码如下：

```
01. void Widget::on_comboBox2_currentIndexChanged(const QString &arg1)
02. {
03.     QString zone=ui->comboBox2->currentData().toString() ;//项关联的数据
04.     ui->plainTextEdit->appendPlainText(arg1 + ":区号=" + zone);
05. }
```

< [上一页](#)

[下一页](#) >



## 所有教程

[socket](#)[Python基础教程](#)[C#教程](#)[MySQL](#)[MySQL函数](#)[C语言入门](#)[C语言专题](#)[C语言编译器](#)[C语言编程实例](#)[GCC编译器](#)[数据结构](#)

[C语言项目案例](#)[C++教程](#)[OpenCV](#)[Qt教程](#)[Unity 3D教程](#)[UE4](#)[STL](#)[Hibernate](#)[Mybatis](#)[Spring MVC](#)[TCP/IP](#)[JavaScript](#)[Spring Cloud](#)[Maven](#)[vi命令](#)[Android教程](#)[PHP](#)[Redis](#)[Spring Boot](#)[Linux](#)[Linux命令](#)[Shell脚本](#)[Java教程](#)[设计模式](#)[Spring](#)[Servlet](#)[Struts2](#)[Java Swing](#)[JSP教程](#)[CSS教程](#)[TensorFlow](#)[区块链](#)[Go语言教程](#)[Docker](#)[编程笔记](#)[资源下载](#)[关于我们](#)[汇编语言](#)[大数据](#)[云计算](#)[VIP视频](#)

## 优秀文章

[Go语言continue（继续下一次循环）](#)[TensorFlow常用Python扩展包](#)[符号——链接的粘合剂](#)[Shell关联数组（下标是字符串的数组）](#)[C语言atan\(\)函数：求反正切](#)[MyBatis BlobTypeHandler读取Blob类型字段](#)[MyBatis关联查询（级联查询）](#)[Go语言网络爬虫调度器的实现](#)[JSP JSTL <c:forEach>标签：对对象进行迭代输出标签](#)[Python VS Code下载和安装教程](#)

精美而实用的网站，提供C语言、C++、STL、Linux、Shell、Java、Go语言等教程，以及socket、GCC、vi、Swing、设计模式、JSP等专题。

Copyright ©2011-2018 biancheng.net, 陕ICP备15000209号

*biancheng.net*

