

QMessageBox::information常用

- 第一个参数**parent**，说明它的父组件
- 第二个参数**title**，也就是对话框的标题
- 第三个参数**text**，是对话框显示的内容
- 第四个参数**buttons**，声明对话框放置的按钮，默认是只放置一个**OK**按钮，这个参数可以使用或运算，例如我们希望有一个**Yes**和一个**No**的按钮，可以使用**QMessageBox::Yes | QMessageBox::No**，所有的按钮类型可以在**QMessageBox**声明的**StandardButton**枚举中找到
- 第五个参数**defaultButton**就是默认选中的按钮，默认值是**NoButton**，也就是哪个按钮都不选中。

```
1 QMessageBox::information(NULL, "Title", "Content",
2   QMessageBox::Yes | QMessageBox::No, QMessageBox::Yes);
```

Qt提供了五个类似的标准对话框

- 最后一个about()函数是没有button设置的按钮

```
1 QMessageBox::critical(NULL, "critical", "Content",
2   QMessageBox::Yes | QMessageBox::No, QMessageBox::Yes);
3
4 QMessageBox::warning(NULL, "warning", "Content",
5   QMessageBox::Yes | QMessageBox::No, QMessageBox::Yes);
6
7 QMessageBox::question(NULL, "question", "Content",
8   QMessageBox::Yes | QMessageBox::No, QMessageBox::Yes);
9
10 QMessageBox::about(NULL, "About", "About this application");
```

自定义图标的MessageBox

- 这里我们使用的是exec()函数，而不是show()，因为这是一个模态对话框，需要有它自己的事件循环，否则的话，我们的对话框会一闪而过

```
1 QMessageBox message(QMessageBox::NoIcon, "Title", "Content with
   icon.");
```

```
2 message.setIconPixmap(QPixmap(":/image/icon.png"));
3 message.exec();
```

处理对话框的交互

- static函数都是要返回一个StandardButton，我们就可以通过判断这个返回值来对用户的操作做出相应

```
1 QMessageBox::StandardButton button=
  QMessageBox::question(NULL, "Show Qt",
2   "Do you want to show Qt dialog?",
3   QMessageBox::Yes | QMessageBox::No, QMessageBox::Yes);
4 if(button == QMessageBox::Yes) {
5     QMessageBox::aboutQt(NULL, "About Qt");
6 }
7
8 //如果是自定义的对话框
9 QMessageBox message(QMessageBox::NoIcon, "Show Qt",
10  "Do you want to show Qt dialog?",
11  QMessageBox::Yes | QMessageBox::No, NULL);
12 if(message.exec() == QMessageBox::Yes) {
13     QMessageBox::aboutQt(NULL, "About Qt");
14 }
```