Archived:How to use QLabel

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Qt Quick should be used for all UI development on mobile devices. The approach described in this article (based on QWidget) is deprecated.

Overview

In this article we will see how to use QLabel to display text or an image. No user interaction functionality is provided. The visual appearance of the label can be configured in various ways, and it can be used for specifying a focus mnemonic key for another widget.

Screen Shot

```
QLabel *label = new QLabel("Hello Qt!");
label->show();
```

Various Function

- This function `setBuddy(QWidget)` is used to set the buddy of any particular widget.
- Without Function

(C) Copyright Nokia 2013. All rights reserved.

- With using `setBuddy(QWidget)`

```
QLineEdit *line=new QLineEdit()
label->setBuddy(line);
```

- To set the text Format.

```
label->setTextFormat(Qt::RichText)
```

- Specifies how the label should interact with user input if it displays text.

```
label->setTextInteractionFlags(Qt::LinksAccessibleByKeyboard);
```

- To set the frame style of label.

```
label->setFrameStyle(QFrame::Panel | QFrame::Sunken);
```

- For more Function See Here

(C) Copyright Nokia 2013. All rights reserved.

ScreenShot

```
QLabel *label = new QLabel("<h2><i>Hello</i> <font color=blue>Qt for<br>S60!</br> </font></h2>");
label->show();
```

Here is the complete code that will help the beginners.

```
#include <QApplication>
#include <QLabel>
int main(int argc, char *argv[])
{
QApplication app(argc, argv);
QLabel *label = new QLabel("Hello Qt!");
label->show();
return app.exec();
}
```

In the above example 1st and 2nd line include the definitions of the `QApplication` and `QLabel` classes. For every Qt class, there is a header file with the same name (and capitalization)as the class that contains the class's definition.

5th line creates a `QApplication` object to manage application-wide resources. The `QApplication` constructor requires argc and argv because Qt supports a few command-line arguments of its own. 6th line creates a `QLabel` widget that displays "Hello Qt!". Whatever widgets are created that all are invisible by defaults so next lline will set the widgets to visible. Last line passes control of the application to Qt. At this point, the program waits for user actions such as mouse clicks and key presses.

Related Article

- This is used to set the image on the label and also shows how scale the content of label.Scaling QPixmap image

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.