# Qt 之 QLabel

# 简述

QLabel提供了一个文本或图像的显示，没有提供用户交互功能。

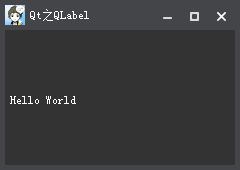
一个QLabel可以包含以下任意内容类型：

| **内容** | **设置** |
| --- | --- |
| 纯文本 | 使用setText()设置一个QString |
| 富文本 | 使用setText()设置一个富文本的QString |
| 图像 | 使用setPixmap()设置一个图像 |
| 动画 | 使用setMovie()设置一个动画 |
| 数字 | 使用setNum()设置int或double，并转换为纯文本。 |
| Nothing | 空的纯文本，默认的，使用clear()设置 |

**|**版权声明：一去、二三里，未经博主允许不得转载。

# 纯文本

## 显示



首先我们构造一个QLabel对象，其中this为其所在的父窗体。通过调用setText可以为标签设置文本（Hello World），这时标签就可以正常显示出来了。为了显示更佳的效果，我们可以通过调用setStyleSheet来设置样式。color: white-顾名思义，就是为标签设置一个文本色（白色）。

QLabel \*pLabel = new QLabel(this);

pLabel->setText("Hello World");

pLabel->setStyleSheet("color: white");

## 对齐方式

默认的标签文本对齐方式为：左对齐、垂直居中，我们可以通过setAlignment来设置，包括：左、上、右、下、居中对齐，一般情况，我们会进行两两组合（水平方向、垂直方向）。

比如：居中对齐

setAlignment(Qt::AlignCenter);

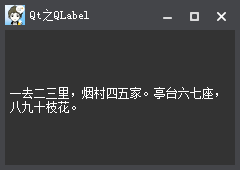
使用样式表来控制（水平居右、垂直居下）：

setStyleSheet("qproperty-alignment: 'AlignBottom | AlignRight';");

## 自动换行

如果文本过长，我们可以采用自动换行的方式来显示。

setWordWrap(true);



注意：当使用英文的时候，如果写为类似形式”abcdefghijklmnopqrstuvwxyz”则是不能换行的，why？因为中间没有空格，所以需要写为”abcde fghij klmno pqrst uvwxyz”。

## 设置行高

一般情况下，自动换行之后文本上下行会距离比较近，我们可以通过下面方式来设置行高。

pLabel->setWordWrap(true);

QString strText = QStringLiteral("一去二三里，烟村四五家。亭台六七座，八九十枝花。");

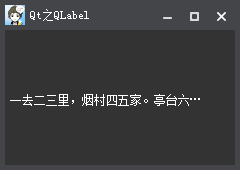
QString strHeightText = "<p style=\"line-height:%1%\">%2<p>";

strText = strHeightText.arg(150).arg(strText);

pLabel->setText(strText);

## 省略

如果过长，我们又不想换行，只想把其中一部分省略为…，那么我们可以通过QFontMetrics来实现，这里先不介绍QFontMetrics，感兴趣的童鞋可以先自行研究。



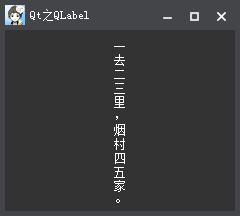
QString strText = QStringLiteral("一去二三里，烟村四五家。亭台六七座，八九十枝花。");

QString strElidedText = pLabel->fontMetrics().elidedText(strText, Qt::ElideRight, 200, Qt::TextShowMnemonic);

pLabel->setText(strElidedText);

## 垂直显示

默认情况下，文本显示方式为水平方向，如果我们需要在垂直方向上显示，需要用一些小技巧来处理。



QString strText = QStringLiteral("一去二三里，烟村四五家。");

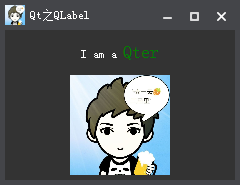
pLabel->setText(strText.split("", QString::SkipEmptyParts).join("\n"));

pLabel->setAlignment(Qt::AlignCenter);

# 富文本

我们可以在助手中查找关于Using HTML Markup in Text Widgets的资料，查看Qt支持哪些HTML标记。

下面我们来写一段HTML代码，O(∩\_∩)O哈哈~。。。显示不同颜色的文本以及图片



QString strHTML = QString("<html> \

<head> \

<style> \

font{color:white;} #f{font-size:18px; color: green;} \

</style> \

</head> \

<body>\

<font>%1</font><font id=\"f\">%2</font> \

<br/><br/> \

<img src=\":/Images/logo\" width=\"100\" height=\"100\"> \

</body> \

</html>").arg("I am a ").arg("Qter");

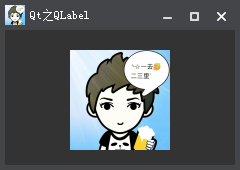
pLabel->setText(strHTML);

pLabel->setAlignment(Qt::AlignCenter);

是不是很神奇？经常我们要用好几个控件来组合才能实现的功能，就仅仅几行HTML代码就搞定了。。。何乐而不为！

# 图像

首先我们构建一个QPixmap来作为显示的图片，然后设置标签的大小，可以通过setScaledContents按比例缩放图片达到理想的效果。



QPixmap pixmap(":/Images/logo");

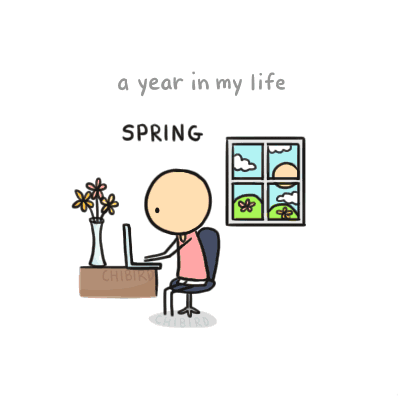
pLabel->setPixmap(pixmap);

pLabel->setFixedSize(100, 100);

pLabel->setScaledContents(true);

# 动画

这里我们需要使用另外一个类QMovie来控制动画，start()可以进行播放与stop()则可以停止，也可以通过调用setSpeed()来设置动画的播放速度。



QMovie \*pMovie = new QMovie(":/Images/movie");

pLabel->setMovie(pMovie);

pLabel->setFixedSize(135, 200);

pLabel->setScaledContents(true);

pMovie->start();

# 数字

如果我们需要显示一个数字，则可以调用setNum()，他可以将内容转换为纯文本。

pLabel->setNum(66.6);

* 1

很简单就一段代码，我们可以打开源码瞅瞅，究竟setNum是如何实现的。

void QLabel::setNum(int num)

{

QString str;

str.setNum(num);

setText(str);

}

就三行代码，呵呵哒。。。超简单吧，我们要有一颗随时看源码的心！

# 超链接

我们需要简单使用标签<a></a>写一段简单的HTML超链接代码

## 方法一

比较简单，直接调用setOpenExternalLinks(true)即可。

pLabel->setText(QString("<a href = \"%1\">%2</a>").arg("http://blog.csdn.net/liang19890820").arg(QStringLiteral("一去丶二三里")));

pLabel->setOpenExternalLinks(true);

声明一个槽openUrl，将其与linkActivated信号关联。

pLabel->setText(QString("<a href = \"%1\">%2</a>").arg("http://blog.csdn.net/liang19890820").arg(QStringLiteral("一去丶二三里")));

connect(pLabel, SIGNAL(linkActivated(QString)), this, SLOT(openUrl(QString)));

void MainWindow::openUrl(const QString &link)

{

QDesktopServices::openUrl(QUrl(link));

}

# 总结

通过上面的学习，我们基本将标签的大部分用法都分享了，可以看出HTML、CSS的重要性，所以无论你现在所从事的工作是什么，或者以后做什么，我都建议好好研究下Web，技多不压身。。。上面的内容虽多，但都很简单，也很重要，所以都需要掌握，我们不只是Qter，我们更是工匠-精益求精、严谨、耐心，专注，坚持。