# Qt 之 QScrollArea

# 简述

QScrollArea提供了一个滚动视图到另一个部件。

滚动区域用于显示一个画面中的子部件的内容。如果部件超过画面的大小，视图可以提供滚动条，这样就都可以看到部件的整个区域。

**|**版权声明：一去、二三里，未经博主允许不得转载。

# 基本使用

子部件必须使用setWidget()指定，例如：

QLabel \*imageLabel = new QLabel(this);

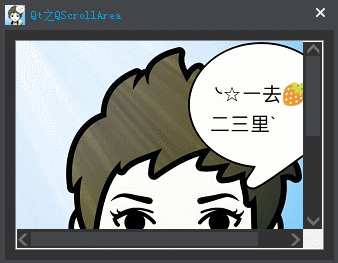
imageLabel->setPixmap(QPixmap(":/Images/head"));

QScrollArea \*scrollArea = new QScrollArea(this);

scrollArea->setBackgroundRole(QPalette::Dark); *// 背景色*

scrollArea->setWidget(imageLabel);

上面的代码创建了一个滚动区域（如下所示）包含一个图像。当缩放图片时，滚动区域可以提供必要的滚动条：

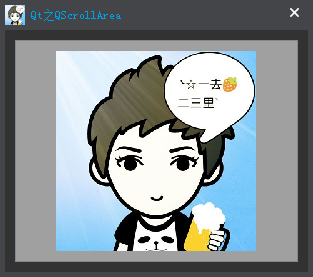


可以使用widget()函数来获取部件，视图可以使用setWidgetResizable()函数来调整大小。

有两个便利的函数：ensureVisible()和ensureWidgetVisible()，确保视口中的一定区域内容是可见的，必要时通过滚动内容。

# 对齐方式

默认的对齐方式为：水平居左，垂直居上。可以通过setAlignment(Qt::Alignment)来改变。



QLabel \*imageLabel = new QLabel(this);

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

pixmap = pixmap.scaled(200, 200);

imageLabel->setPixmap(pixmap);

imageLabel->setStyleSheet("background: white;"); *// 用于“部件调整大小”处效果对比*

QScrollArea \*scrollArea = new QScrollArea(this);

scrollArea->setBackgroundRole(QPalette::Dark); *// 背景色*

scrollArea->setWidget(imageLabel);

scrollArea->setAlignment(Qt::AlignCenter); *// 居中对齐*

Qt::Alignment有很多可选项，分别表示水平和垂直方向的对齐方式，例如：左上右下。

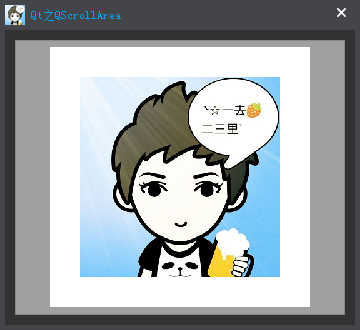
# 调整部件大小

上面，通过setStyleSheet()为QLabel部件设置了白色背景色，但是，我们并没有发现背景有任何变化。

因为默认情况下属性widgetResizable为false。滚动区域会保持部件的大小，也就是说QLabel自适应QPixmap的大小。

如果要调整部件大小，有两种可选方式。

## 手动调整



忽略这个属性，我们可以通过使用widget()->resize()来调整部件的大小，这时滚动区域会自动调整自身大小为新的尺寸。

QLabel \*imageLabel = new QLabel(this);

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

pixmap = pixmap.scaled(200, 200, Qt::KeepAspectRatio); *// 图片缩放*

imageLabel->setPixmap(pixmap);

imageLabel->setStyleSheet("background: white;"); *// 标签白色背景*

imageLabel->setAlignment(Qt::AlignCenter); *// 图片居中*

QScrollArea \*scrollArea = new QScrollArea(this);

scrollArea->setBackgroundRole(QPalette::Dark); *// 背景色*

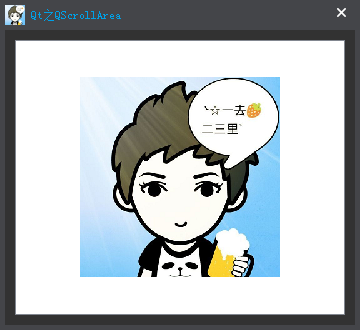
scrollArea->setWidget(imageLabel);

scrollArea->setAlignment(Qt::AlignCenter); *// 居中对齐*

scrollArea->widget()->resize(260, 260); *// 手动调整大小*

## 自动调整

如果widgetResizable被设置为true，则滚动区域部件将自动调整，以避免可以不显示的滚动条，或者利用额外的空间。



QLabel \*imageLabel = new QLabel(this);

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

pixmap = pixmap.scaled(200, 200, Qt::KeepAspectRatio); *// 图片缩放*

imageLabel->setPixmap(pixmap);

imageLabel->setStyleSheet("background: white;"); *// 标签白色背景*

imageLabel->setAlignment(Qt::AlignCenter); *// 图片居中*

QScrollArea \*scrollArea = new QScrollArea(this);

scrollArea->setBackgroundRole(QPalette::Dark); *// 背景色*

scrollArea->setWidget(imageLabel);

scrollArea->setAlignment(Qt::AlignCenter); *// 居中对齐*

scrollArea->setWidgetResizable(true); *// 自动调整大小*

# 获取与移除部件

## 获取

可以通过widget()函数来获取滚动区域的部件，如果没有，则返回0。

QWidget \*widget = scrollArea->widget();

QLabel \*pLabel = qobject\_cast<QLabel \*>(widget);

* 1
* 2

如果部件是一个QWidget的子类，可以用qobject\_cast进行类型转换。

## 移除

如果要将部件从滚动区域中移除（即：在上例中，滚动区域不再显示图片），使用takeWidget()函数。

QWidget \*widget = scrollArea->takeWidget();

QLabel \*pLabel = qobject\_cast<QLabel \*>(widget);

移除的同时，会将部件的所有权传递到调用者，可以用QWidget来接收。