

## 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。

## 基本使用

---

- 子部件必须使用**setWidget()**指定

```
1 QLabel *imageLabel =new QLabel(this);
2 imageLabel->setPixmap(QPixmap(":/Images/head"));
3 QScrollArea *scrollArea =new QScrollArea(this);
4 scrollArea->setBackgroundRole(QPalette::Dark); // 背景色
5 scrollArea->setWidget(imageLabel);
```

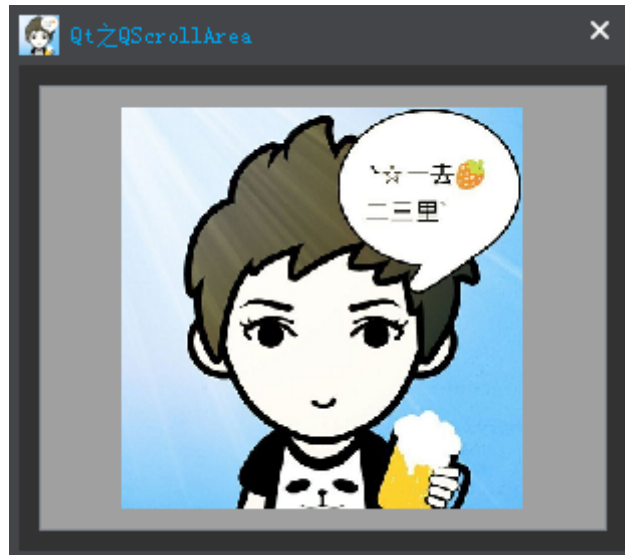


- 可以使用**widget()**函数来获取部件，视图可以使用**setWidgetResizable()**函数来调整大小。
- 有两个便利的函数：**ensureVisible()**和**ensureWidgetVisible()**，确保视口中的一定区域内容是可见的，必要时通过滚动内容。

## 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。
- **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。



```
1 QLabel *imageLabel =new QLabel(this);
2 QPixmap pixmap(":/Images/head");
3 pixmap = pixmap.scaled(200, 200);
4 imageLabel->setPixmap(pixmap);
5 // 用于“部件调整大小”处效果对比
6 imageLabel->setStyleSheet("background: white;");
7 QScrollArea *scrollArea =new QScrollArea(this);
8 scrollArea->setBackgroundRole(QPalette::Dark); // 背景色
9 scrollArea->setWidget(imageLabel);
10 scrollArea->setAlignment(Qt::AlignCenter); // 居中对齐
```

## 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**。滚动不会改变 **widget** 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。一般用自动调整  
**setWidgetResizable(true)**

## 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。

```
1 QLabel *imageLabel =new QLabel(this);
2 QPixmap pixmap(":/Images/head");
3 pixmap = pixmap.scaled(200, 200, Qt::KeepAspectRatio); //图片缩放
  为200*200
4 imageLabel->setPixmap(pixmap);
5 imageLabel->setStyleSheet("background: white;"); //标签白色背景
6 imageLabel->setAlignment(Qt::AlignCenter); //将pixmap图片居中
7
8 QScrollArea *scrollArea =new QScrollArea(this);
9 scrollArea->setBackgroundRole(QPalette::Dark);
10 scrollArea->setWidget(imageLabel);
11 scrollArea->setAlignment(Qt::AlignCenter); //居中对齐
12
13 scrollArea->widget()->resize(260, 260); //将imageLabel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。



```
1 QLabel *imageLabel =new QLabel(this);
2 QPixmap pixmap(":/Images/head");
3 pixmap = pixmap.scaled(200, 200, Qt::KeepAspectRatio); // 图片缩
  放
4 imageLabel->setPixmap(pixmap);
5 imageLabel->setStyleSheet("background: white;"); // 标签白色背景
6 imageLabel->setAlignment(Qt::AlignCenter); // 图片居中
7
8 QScrollArea *scrollArea =new QScrollArea(this);
9 scrollArea->setBackgroundRole(QPalette::Dark); // 背景色
10 scrollArea->setWidget(imageLabel);
11 scrollArea->setAlignment(Qt::AlignCenter); // 居中对齐
12
13 scrollArea->setWidgetResizable(true); // label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**。

```
1 QWidget *widget = scrollArea->widget();
```

## 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。

```
1 QWidget *widget = scrollArea->takeWidget();
```