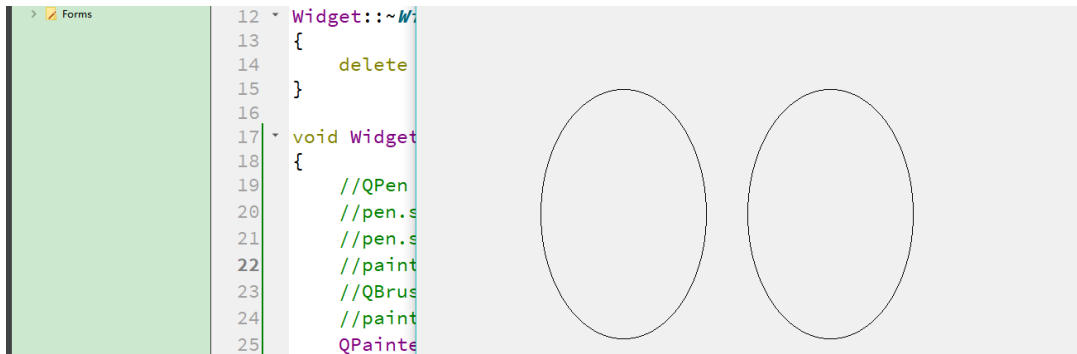


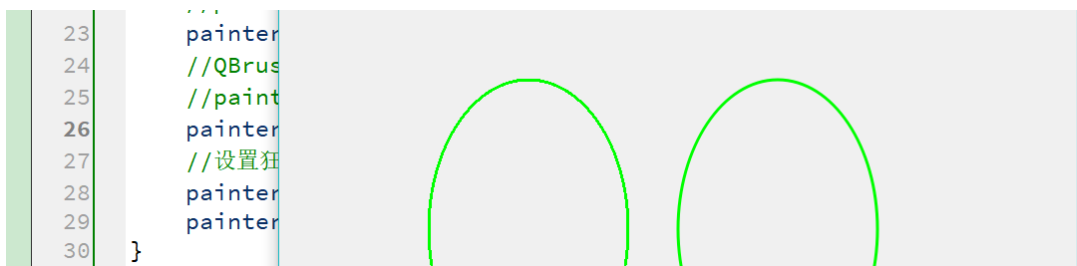
先用默认画笔画刷画两个个椭圆



设置抗锯齿能力

```
void Widget::paintEvent(QPaintEvent *)
{
    QPen pen(QColor(0,255,0)); //定义画笔
    pen.setWidth(3); //粗细
    QPainter painter(this); //实例化一个画家 this指定的是绘图设备
    //pen.setStyle(Qt::DashDotDotLine); //风格
    painter.setPen(pen); //让画家使用这个笔
    //QBrush brush(Qt::VerPattern); //定义画刷
    //painter.setBrush(brush); //让画家使用这个画刷
    painter.drawEllipse(QPoint(250,250),100,150); //画一个椭圆
    //设置抗锯齿能力
    painter.setRenderHint(QPainter::Antialiasing);
    painter.drawEllipse(QPoint(500,250),100,150);
}
```

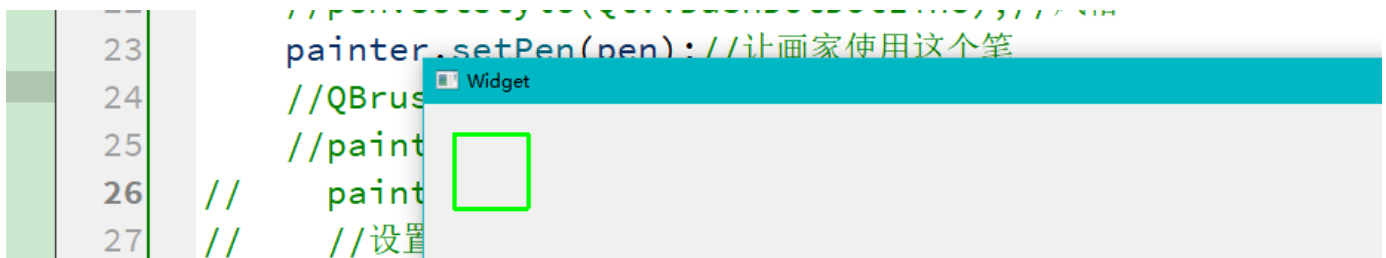
如图



当画重叠的图形

```
painter.drawRect(QRect(20,20,50,50));
painter.drawRect(QRect(20,20,50,50));
```

只能看到一个

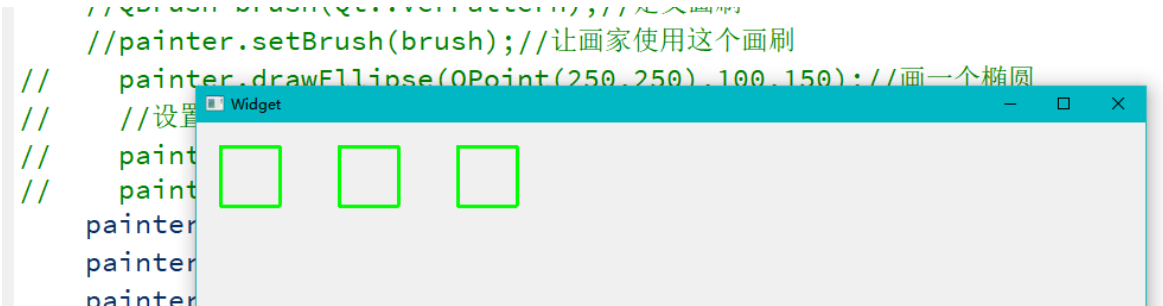


## 抱走画家 translate

```

painter.drawRect(QRect(20,20,50,50)); //画一个矩形
painter.translate(100,0); //移动画家到100, 0的位置
painter.drawRect(QRect(20,20,50,50)); //画一个完全相同的矩形
painter.translate(100,0); //移动画家到100, 0的位置
painter.drawRect(QRect(20,20,50,50)); //再画一个完全相同的矩形

```



相当于画完第一个之后，画家移动到坐标（100,0），由此可见，画家的位置是相对的

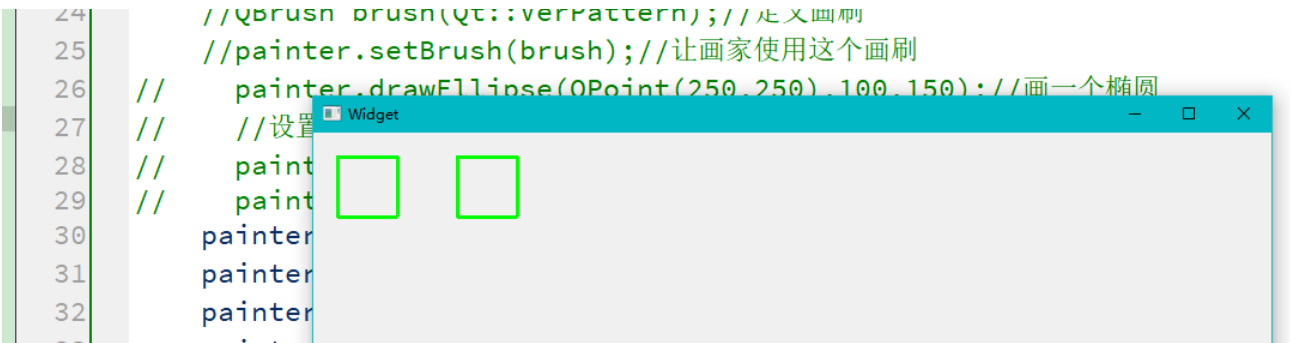
## 保存画家状态

### 抱走之后还原状态

```

painter.drawRect(QRect(20,20,50,50)); //画一个矩形
painter.translate(100,0); //移动画家到100, 0的位置
painter.save(); //保存画家状态
painter.drawRect(QRect(20,20,50,50)); //画一个完全相同的矩形
painter.restore(); //还原画家所保存的状态
painter.drawRect(QRect(20,20,50,50)); //再画一个完全相同的矩形

```



第三次画家状态被还原，所以画出的矩形和还原的位置相同

## 总结:

- 8.1 抗锯齿 效率低
- 8.1.1 `painter.setRenderHint(QPainter::Antialiasing);`
- 8.2 对画家进行移动
- 8.2.1 `painter.translate(100,0);`
- 8.2.2 保存状态 `save`
- 8.2.3 还原状态 `restore`