包含头文件#include

声明绘图事件

widget.h

```
void paintEvent(QPaintEvent *);
```

函数实现

widget.cpp

```
void Widget::paintEvent(QPaintEvent *)
{

}
```

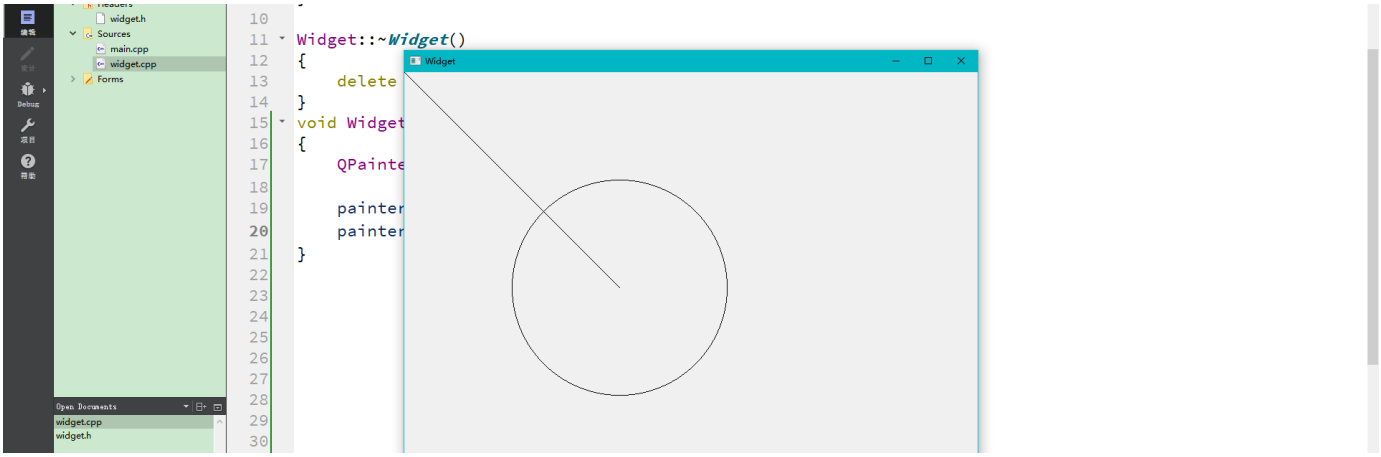
实例化对象

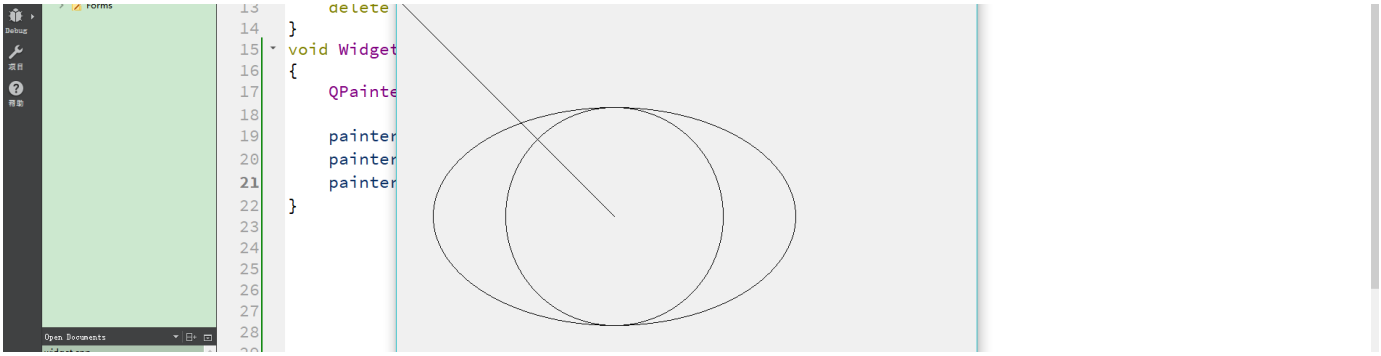
```
void Widget::paintEvent(QPaintEvent *)
{
    QPainter painter(this);//实例化一个画家    this指定的是绘图设备

    painter.drawLine(QPoint(0,0),QPoint(300,300));//画一条线
}
```

函数系统会自动调用

画一个圆

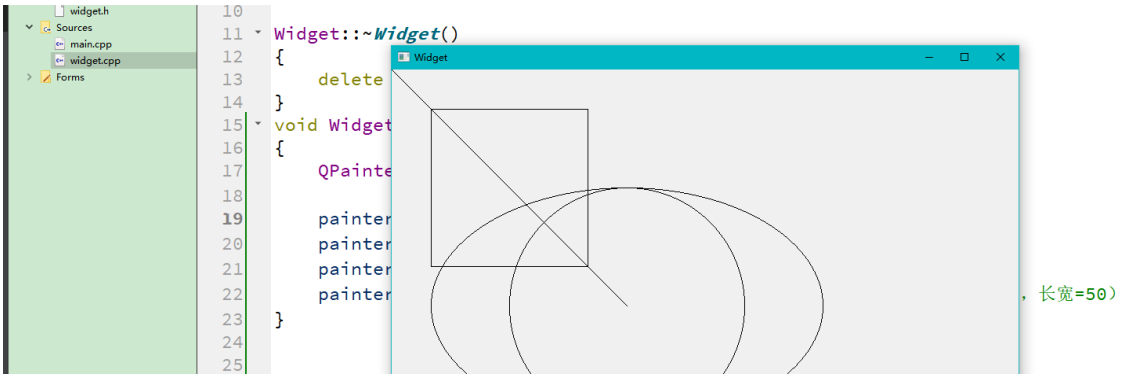
```
painter.drawEllipse(QPoint(300,300),150,150);//画一个圆
painter.drawEllipse(QPoint(300,300),250,150);//画一个椭圆
```





画一个矩形

```
painter.drawRect(QRect(50,50,200,200));//参数分别为（以左上角为坐标点，长宽=50）
```

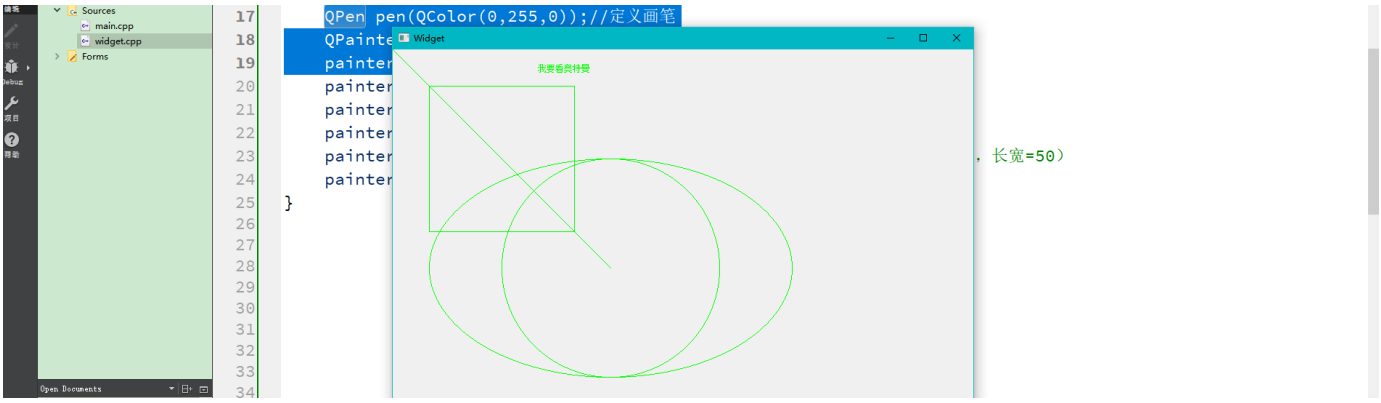


画文字

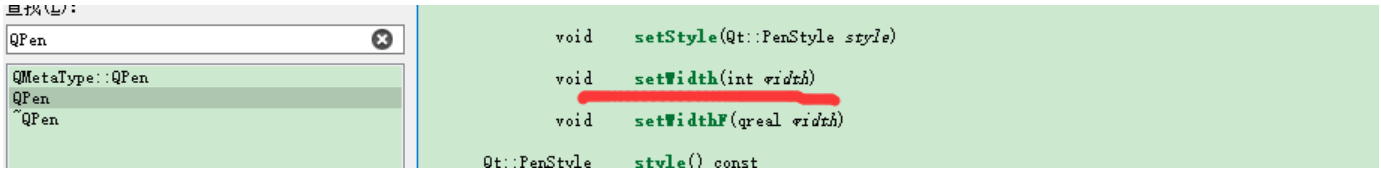
```
painter.drawText(QRect(200,20,100,50),"我要看奥特曼");
```

画笔：

```
QPen pen(QColor(0,255,0));//定义画笔
    QPainter painter(this);//实例化一个画家    this指定的是绘图设备
    painter.setPen(pen);//让画家使用这个笔
```

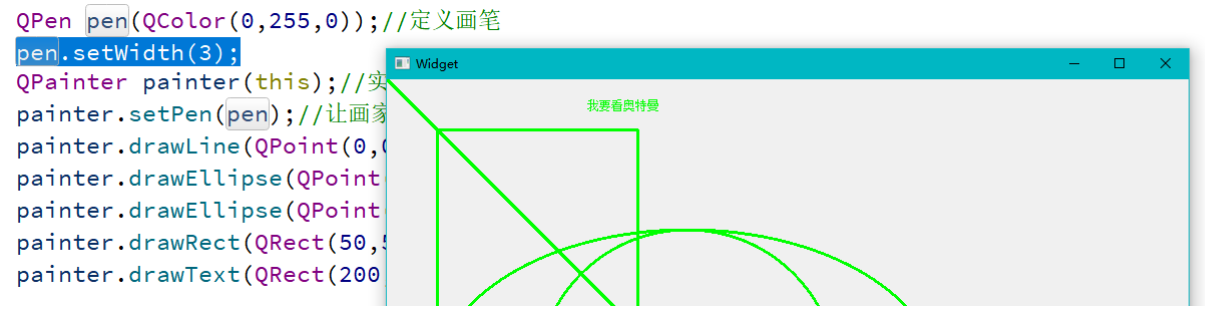


QPen查询文档

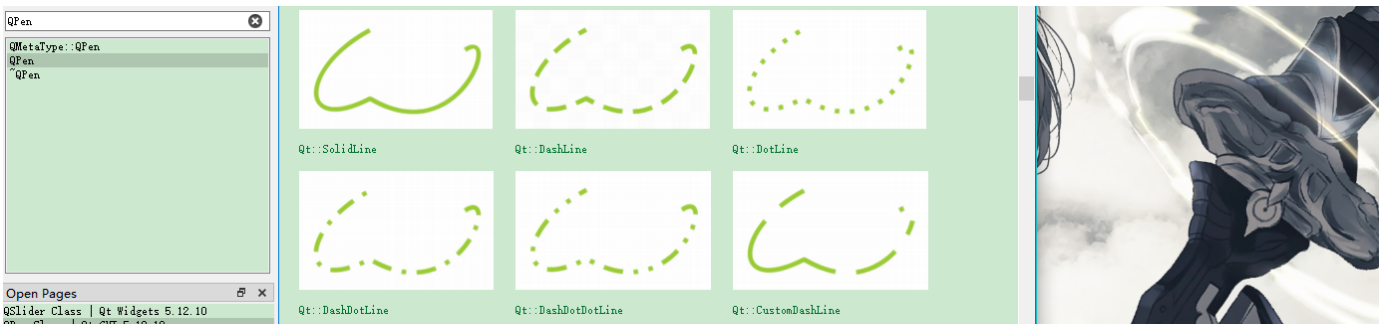


设置画笔粗细

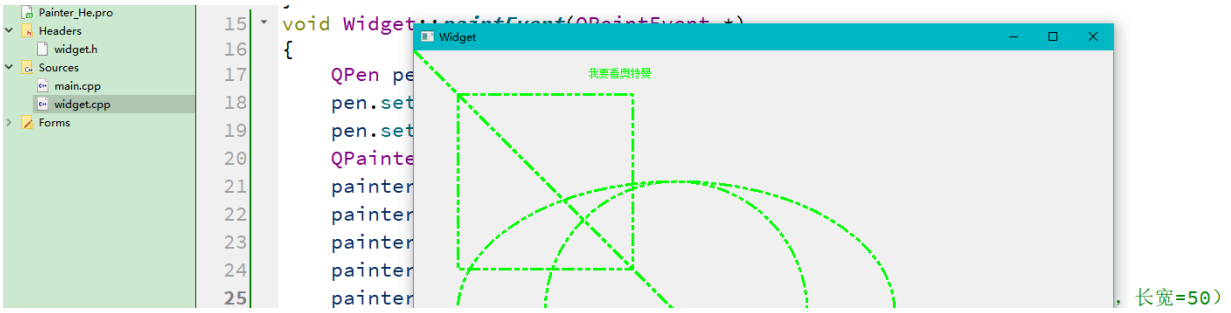
```
pen.setWidth(3);
```



为了设置风格，我们查询文档

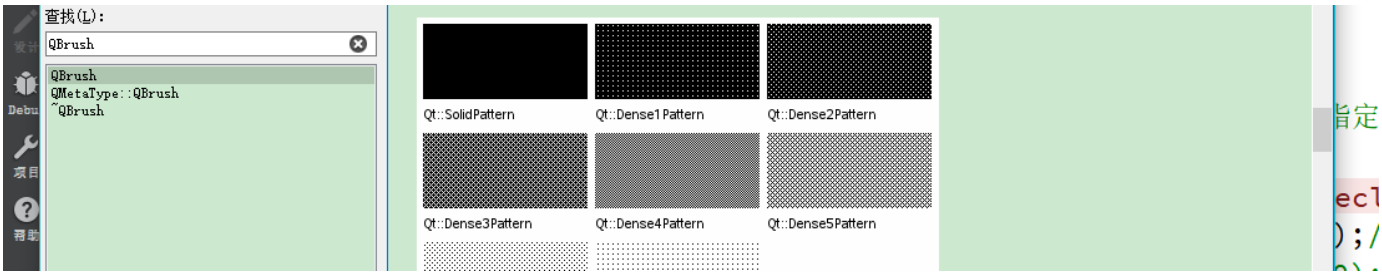


```
pen.setStyle(Qt::DashDotDotLine);//风格
```



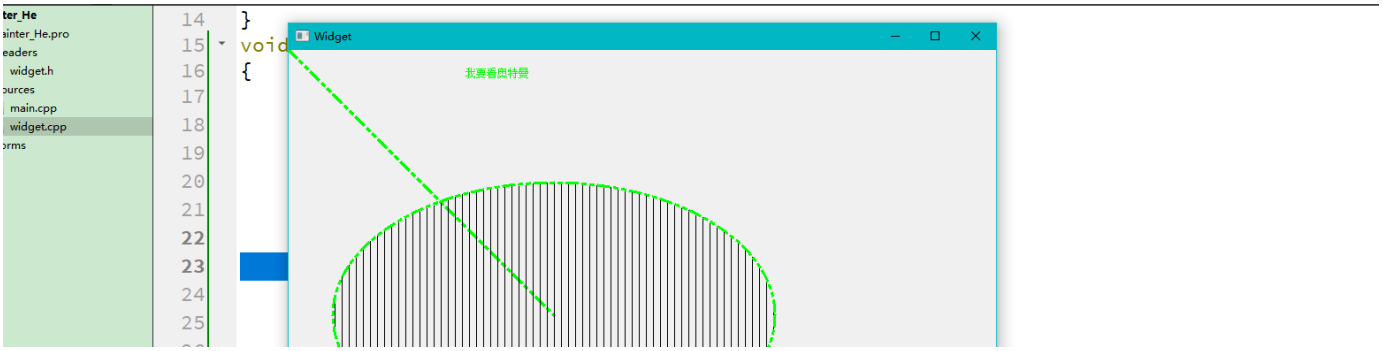
画刷

查询帮助文档，了解画刷的枚举值



使用画刷

```
QBrush brush(Qt::VerPattern);//定义画刷
painter.setBrush(brush);//让画家使用这个画刷
```



总结:

1.1 绘图事件 void paintEvent()
1.2 声明一个画家对象 QPainter painter(this) this指定绘图设备
1.3 画线、画圆、画矩形、画文字
1.4 设置画笔 QPen 设置画笔宽度 、风格
1.5 设置画刷 QBrush 设置画刷 风格