

Qt中窗口背景的设置，下面介绍三种方法：

1. 使用QPalette
2. 使用Style Sheet
3. 绘图事件

一般不用Style Sheet设置窗口背景，也不建议使用。（这里是对于窗口而，如果是子部件当然可以）。因为窗口使用QSS设置背景之后，若子部件不使用同样的方式来设置，默认则会继承父窗口的样式。

使用QPalette

- Qt中提供的调色板QPalette类就是专门用于管理控件的外观显示。QPalette类相当于对话框或控件的调色板，管理着控件和窗体的所有颜色。
- 每个窗体和控件都包含一个QPalette对象，在显示时，对其做相应的设置即可。
- 设置控件颜色的方法是先调用QWidget::palette()获取当前面板，修改它为自定义的值后再通过方法QWidget::setPalette设置为新修改的面板
- 必须添加setAutoFillBackground(true)，设置窗体运行自动填充。

```
1 在widget的构造函数中：
2  QPalette palette=this->palette();
3  palette.setColor(QPalette::Window, Qt::black);
4  this->setPalette(palette);
5  this->setAutoFillBackground(true);
```

绘图事件

- 重写paintEvent，使用QPainter来绘制背景。

```
1 void Widget::paintEvent(QPaintEvent *event)
2 {
3     Q_UNUSED(event);
4
5     QPainter p(this);
6     p.setPen(Qt::NoPen);
7     p.setBrush(Qt::black);
```

```
8 p.drawRect(rect());  
9
```

使用Style Sheet

- 例如设置mainWindow界面的菜单栏menuBar,工具栏toolBar, 状态栏statusBar本景色。
- 菜单栏不光要设置背景的颜色, 还要设置里面的item背景颜色
- 使用#D5D5D5十六进制颜色设置, 如果用调色板设置背景颜色使用(RGB(74,75,75)十进制)颜色

```
1 setStyleSheet(  
2 "QMenuBar:item{background-color:#D5D5D5;} QMenuBar{background-color:#D5D5D5;}"  
3 "QToolBar:item{background-color:#D5D5D5;} QToolBar{background-color:#D5D5D5;}"  
4 "QStatusBar:item{background-color:#D5D5D5;} QStatusBar{background-color:#D5D5D5;}");
```