QtWidgets

# QWidget

## paintEvent()

当产生一个绘制事件并且调用paintEvent()函数的时候，会出现如下几种情况：

* 在窗口部件第一次显示时，系统会自动产生一个绘制事件，从而强制绘制这个窗口部件本身。
* 当重新调整窗口部件大小的时候，系统也会产生一个绘制事件。
* 当窗口部件被其他窗口部件遮挡，然后又再次显示出来的时候，就会对那些隐藏的区域产生一个绘制事件（除非这个窗口系统存储了整个区域）。

也可以通过调用QWidget::update()或者QWidget::repaint()来强制产生一个绘制事件。这个两个函数的区别是：repaint()函数会强制产生一个即时的重绘事件，而update()函数则只是在Qt下一次处理事件时才简单地调用一个绘制事件。如果窗口部件在屏幕上是不可见的，那么这两个函数什么也不做。如果多次调用update()，Qt就会把连续多次的绘制事件压缩成一个单一的绘制事件，这样就可以避免闪烁现象。

## palette()

palette()函数可以返回窗口部件的调色板，它是一个QPalette类型的对象。

每个窗口部件都会配备一个调色板，由它来确定做什么事应该使用什么颜色。一个窗口部件的调色板由三个颜色组构成：激活组(Active)、非激活组(Inactive)和不可用组(Disabled)。应该使用哪一个颜色组取决于该窗口部件的当前状态：

* Active颜色组可用于当前激活窗口中的那些窗口部件。
* Inactive颜色组可用于其他窗口中的那些窗口部件。
* Disabled颜色组可用于任意窗口中的那些不可用窗口部件。

## setAttribute()

Qt::WA\_StaticContents属性

这个属性告诉Qt，当重新改变窗口部件的大小时，这个窗口部件的内容并没有发生变化，而且它的内容仍旧保留从窗口部件左上角开始的特性。当重新定义窗口部件的大小时，通过使用这个信息，Qt就可以避免对已经显示区域的重新绘制。

通常情况下，当重新定义一个窗口部件的大小时，Qt会为窗口部件的整个可见区域生成一个绘制事件。但是如果该窗口部件在创建时使用了Qt::WA\_StaticContents属性，那么绘制事件的区域就会严格限制在之前没有被显示的像素部分上。这也就意味着，如果重新把窗口部件改变为比原来要小的尺寸，那么就根本不会产生任何绘制事件。

## setSizePolicy()

设置窗口部件的大小策略。它会告诉布局系统是否可以对这个窗口部件进行拉长或者缩短。

## sizeHint()

返回一个窗口部件的理想大小。在和布局联合使用时，窗口部件的大小提示非常有用。当Qt的布局管理器摆放一个窗体的子窗口部件时，它会尽可能多的考虑这些窗口部件的大小提示。

## update()

强制重绘窗口部件。

## updateGeometry()

告诉包含这个窗口部件的任意布局，这个窗口部件的大小提示已经发生改变了。于是，该布局将会自动适应这个新的大小提示。