**QDialog**

前言

为了更好的实现人机交互，比如window和linux等系统均会提供一系列的标准对话框来完成特定场景下的功能，比如选择字号大小。字体颜色等，在PyQt5中定义了一系列的标准对话框类，让使用者能够方便快捷地通过各个类完成字号大小，字体颜色以及文件的选择等

QDialog类的子类主要有QMessageBox，QFileDialog，QColorDialog，QFontDialog，QInputDialog等，这些内容随后就会介绍到

QDialog类中的常用方法

方法 描述

setWindowTitle() 设置对话框标题

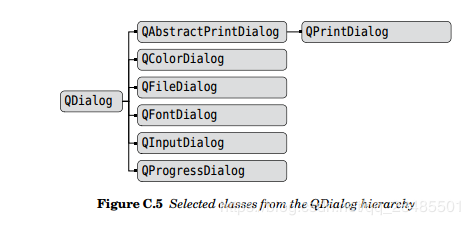
setWindowModality() 设置窗口模态，取值如下

Qt.NonModal：非模态，可以和程序的其他窗口进行交互

Qt.WindowModal:窗口模态，程序在未处理玩当前对话框时，将阻止和对话框的父窗口进行交互

Qt.ApplicationModal：应用程序模态，阻止和任何其他窗口进行交互

QDialog的继承的类



模态划分：

根据显示方式可以分为模态和非模态两种用法。

模态对话框：

一旦调用该对话框，它就会成为应用程序唯一能够与用户进行交互的部件，它会阻止其父窗口，父窗口的父窗口并直至顶层窗口等的交互，当然也会阻止与父窗口同层各兄弟窗口的交互

我们不需要“实时”或者交互式地更新画笔的属性，所以模态对话框足够。

用法：

d.exec\_()：模态化显示 d。 意味着，对话框的父窗口以及其兄弟窗口都会被阻塞，直到这个对话框关闭。

非模态对话框

如果用户可以重复更改数字格式并能马上看到修改效果，那么就无须不停的调用，接收对话框中数字格式修改所给出的提示，从而变得非常方便。使用非模态对话框。用户可以与该对话框以及应用程序的其他部分交互。

非模态的实时对话框

用法：

d.show()非模态显示 d。

