

可视化 UI 设计 (设计器 Qt Designer 实现)

原创 Qt 学习 Qt 学习 2019-11-07

在上一篇，通过一个简单的应用程序，分析了 Qt 创建的 GUI 应用程序中各个文件的作用，剖析了可视化设计的 UI 文件是如何被转换为 C++ 的类定义，并自动创建界面的。这些是使用 QtCreator 可视化设计用户界面，并使各个部分融合起来运行的基本原理。

Qt 开发应用程序既可以采用 Qt 设计器 (Qt Designer) 方式，也可以采用编写代码的方式。这一篇采用 Qt 设计器进行 GUI 应用程序开发，使读者对 Qt 开发程序的流程有一个初步的认识，下一篇将采用编写代码的形式详细介绍 Qt 程序开发步骤。

本篇目录：

1. 实例效果
2. 界面设计与布局管理
3. 界面组件的层次关系
4. 伙伴关系与 Tab 顺序

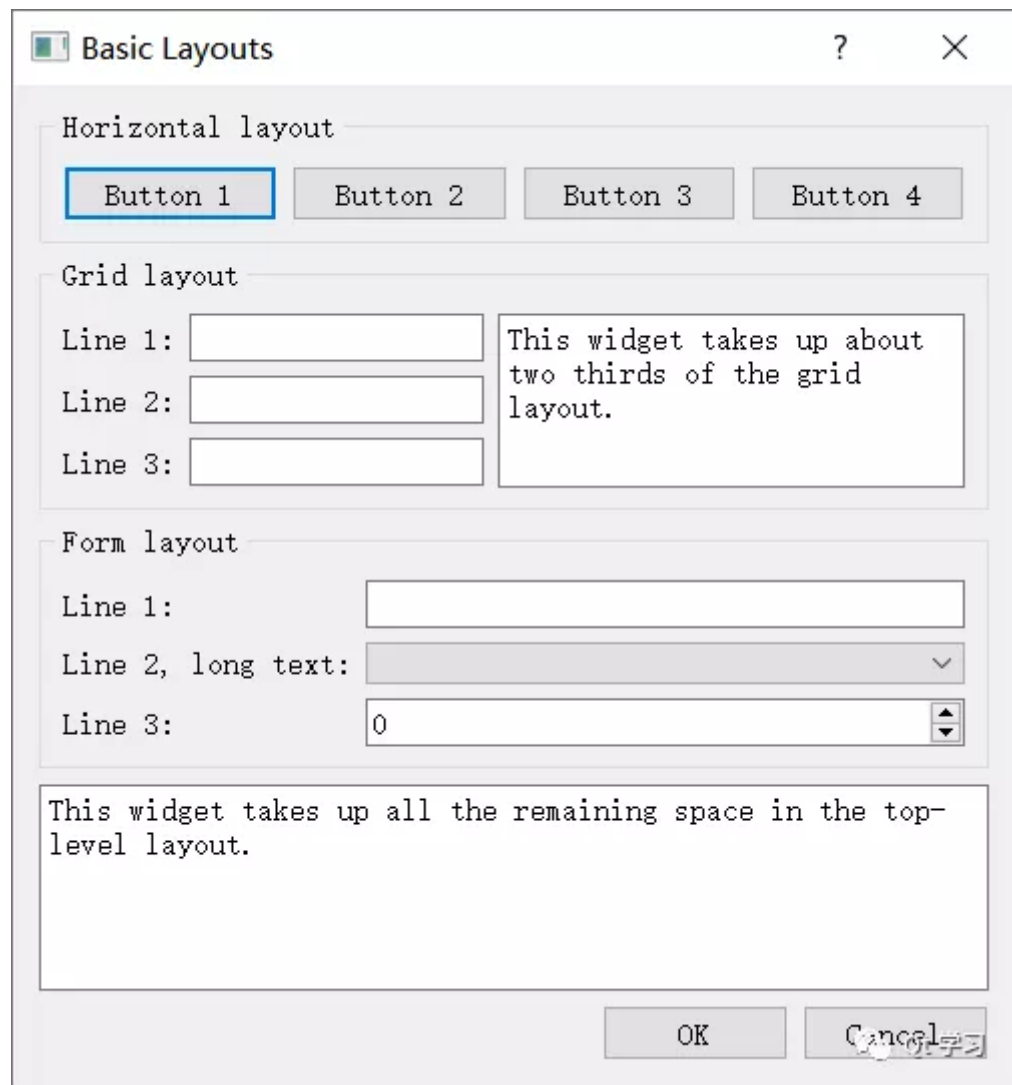
运行环境：

Win 10 + Qt 5.12.5 + Qt Creator 4.10

1. 实例效果

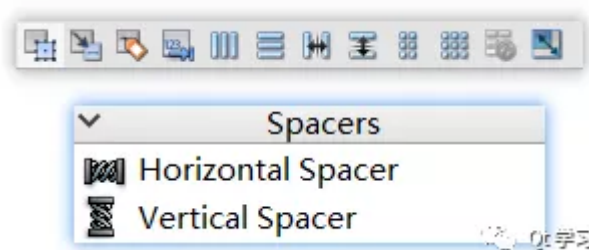
创建一个 Qt Widget Application 项目 BasicLayouts，在创建窗体时选择基类 QDialog，并选择生成窗体。

新建的项目 BasicLayouts 有一个界面文件 dialog.ui，一个头文件 dialog.h 和源程序文件 dialog.cpp。此外，还有项目文件 BasicLayouts.pro 和主程序文件 main.cpp。BasicLayouts 实例效果如下图所示



2. 界面设计与布局管理

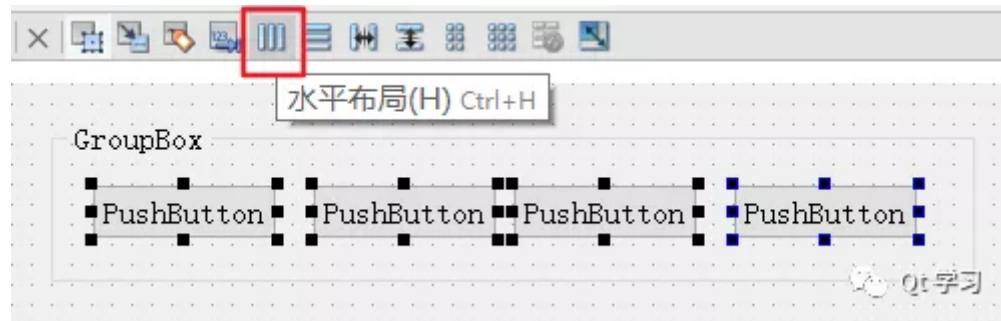
布局管理是 GUI 设计的必备技巧，Qt 的界面设计使用了布局 (Layout) 功能。**所谓布局，就是界面上组件的排列方式，使用布局可以使组件有规则地分布，并且随着窗体大小变化自动地调整大小和相对位置。**常用布局组件、按钮以及分隔空格如下图



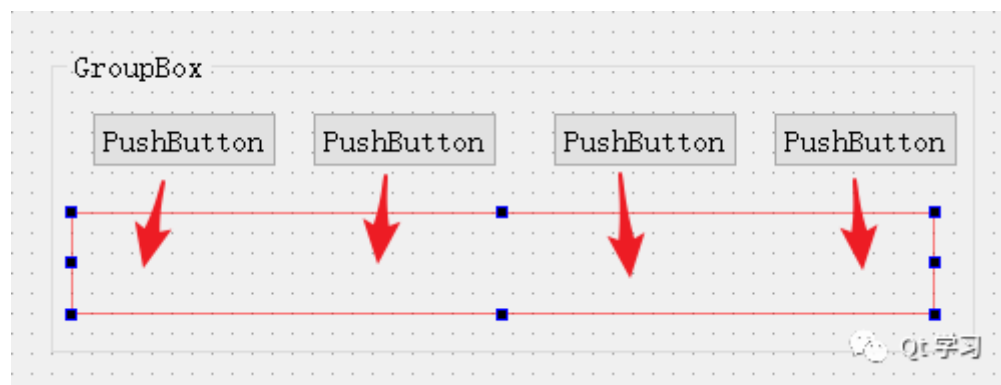
将鼠标放置在布局按钮上会出现按钮功能提示

双击 dialog.ui 进入 Qt 界面设计器，在组件面板的 Containers 中找到 Group Box 并拖拽至窗口文件中，适当调整其大小；同样，在组件面板的 Buttons 中找到 Push Button 并拖拽至窗口文件的 Group Box 中。

选中四个 Push Button 按钮，然后点击水平布局按钮

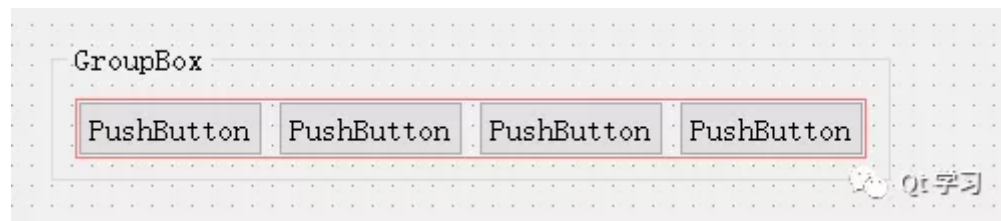


或者将水平布局一并放入 Group Box 组件，然后把四个 Push Button 按钮拖入水平布局组件之中

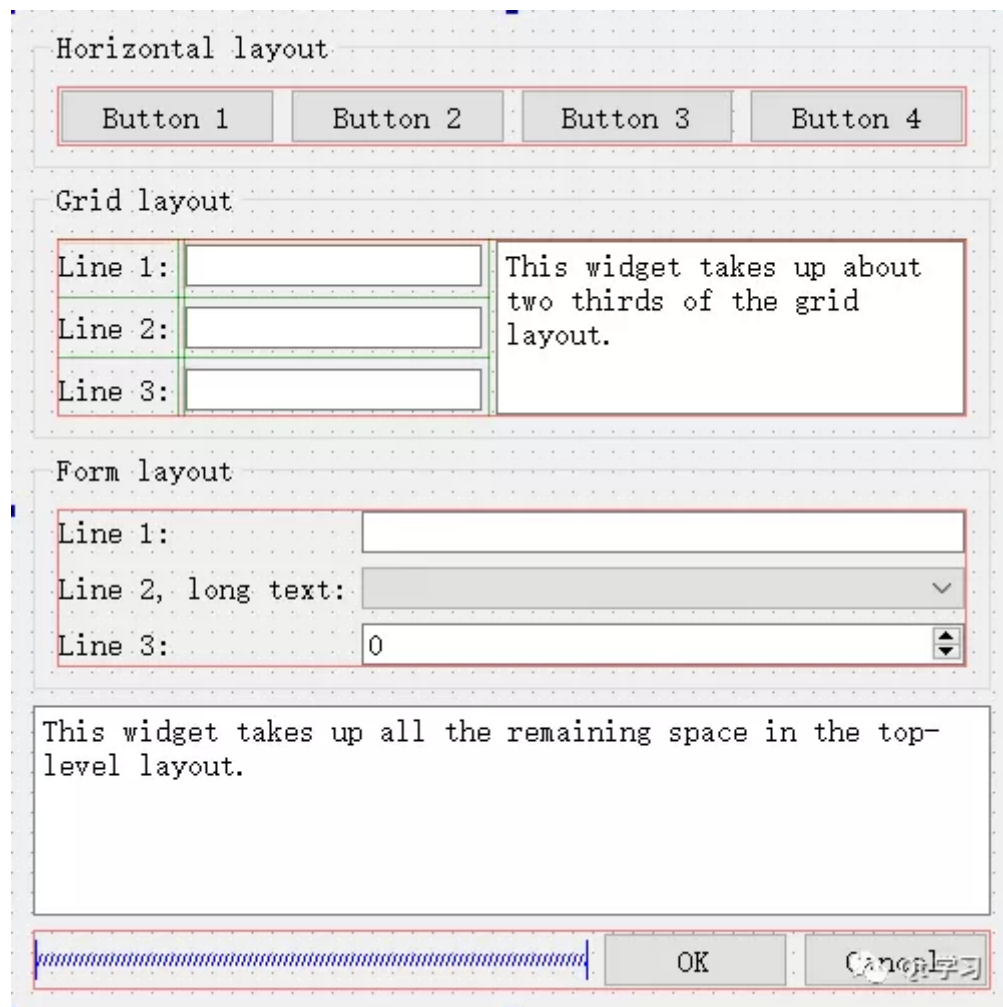


按钮拖入水平布局组件的左侧或者右侧时，会显示一条蓝色的竖直线，表示按钮插入的位置序列

最后选中 Group Box 组件，点击水平、垂直、窗体或栅格布局中的任意一个，这时，包含四个按钮的水平组件会填充整个 Group Box 组件。第一个 Group Box 组件布局效果如下图所示



采用同样的方法可完成剩余的窗口布局 (包含 dialog 布局) , 最终的总体布局效果如下图所示

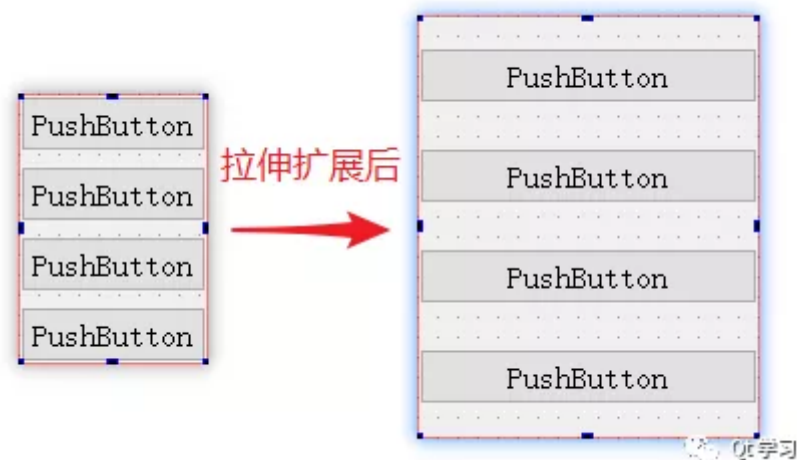


选中或者右击 dialog 窗体, 点击水平、垂直、窗体或栅格布局中的任意一个, 使放入的所有组件填充整个 dialog

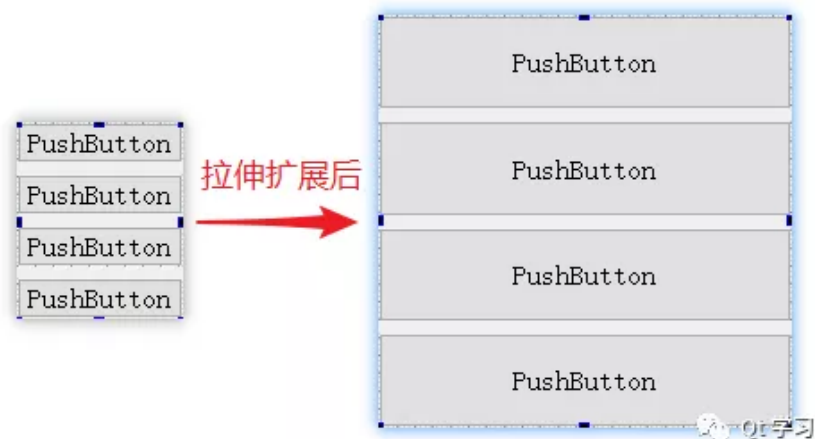
标签、按钮等上显示的文本可直接双击修改, 也可在对象属性框中修改; 文本编辑器 Text Edit 中的文本可通过双击文本编辑器后弹出的对话框中加入。

单击左下角的运行按钮，可得到本篇第一节中的效果。

注： 方向布局与分裂器布局之差别



垂直方向布局



分裂器垂直布局

在垂直方向布局中，四个按钮自动垂直排列，并且进行水平拉伸，按钮总是水平方向变化，如上上图所示；而使用分裂器布局的按钮纵向是可以变化的，这就是分裂器和布局管理器的重要区别，如上图所示。

3. 界面组件的层次关系

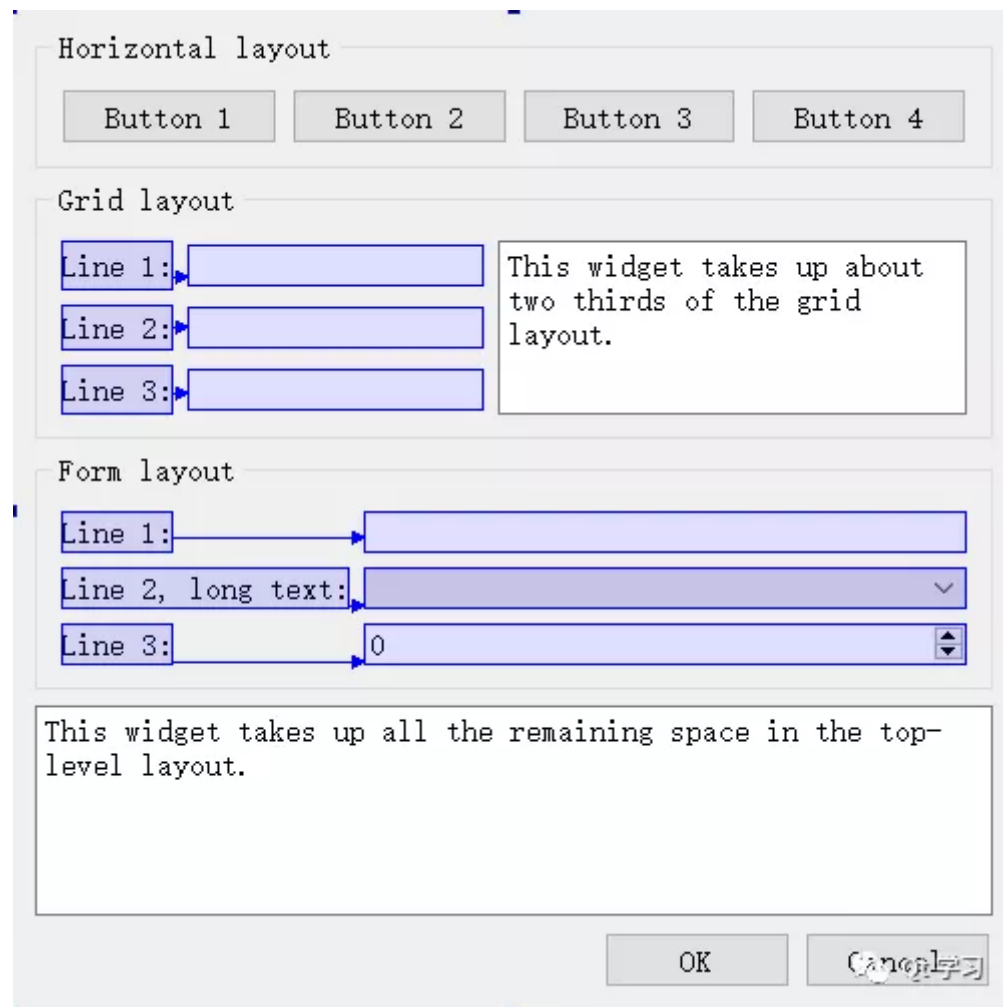
为了将界面上的各个组件的分布设计得更加美观，经常使用一些容器类，如 QGroupBox、QTabWidget、QFrame 等。例如，将 4 个 Push Button 组件放置在一个 Group Box 组件里，该 Group Box 组件就是这 4 个 Push Button 的容器，移动这个 Group Box 就会同时移动其中的 4 个 Push Button。

| 对象 | 类 |
|--|---|
| ▼  Dialog | QDialog |
| ▼  horizontalLayout |  QHBoxLayout |
| horizontalSpacer | Spacer |
| pushButton_5 | QPushButton |
| pushButton_6 | QPushButton |
| ▼  groupBox | QGroupBox |
| ▼  horizontalLayout_2 |  QHBoxLayout |
| pushButton_1 | QPushButton |
| pushButton_2 | QPushButton |
| pushButton_3 | QPushButton |
| pushButton_4 | QPushButton |
| ▼  groupBox_2 | QGroupBox |
| ▼  formLayout |  QFormLayout |
| comboBox | QComboBox |
| label_4 | QLabel |
| label_5 | QLabel |
| label_6 | QLabel |
| lineEdit_4 | QLineEdit |
| spinBox | QSpinBox |
| ▼  groupBox_3 | QGroupBox |
| ▼  gridLayout |  QGridLayout |
| label | QLabel |
| label_2 | QLabel |
| label_3 | QLabel |
| lineEdit | QLineEdit |
| lineEdit_2 | QLineEdit |
| lineEdit_3 | QLineEdit |
| textEdit | QTextEdit |
| textEdit_2 | QTextEdit |

上图左侧显示的是 BasicLayouts 界面上各组件之间的层次关系，右侧是各组件对应的类。在窗体上放置了 3 个 Group Box 组和 1 个 Text Edit 文本编辑器，在各 Group Box 里分别放置相应数量的 Push Button 按钮、Label 标签、Line Edit 行编辑器、Text Edit 文本编辑器、Combo Box 组合框和 Spin Box 选值框。

4. 伙伴关系与 Tab 顺序

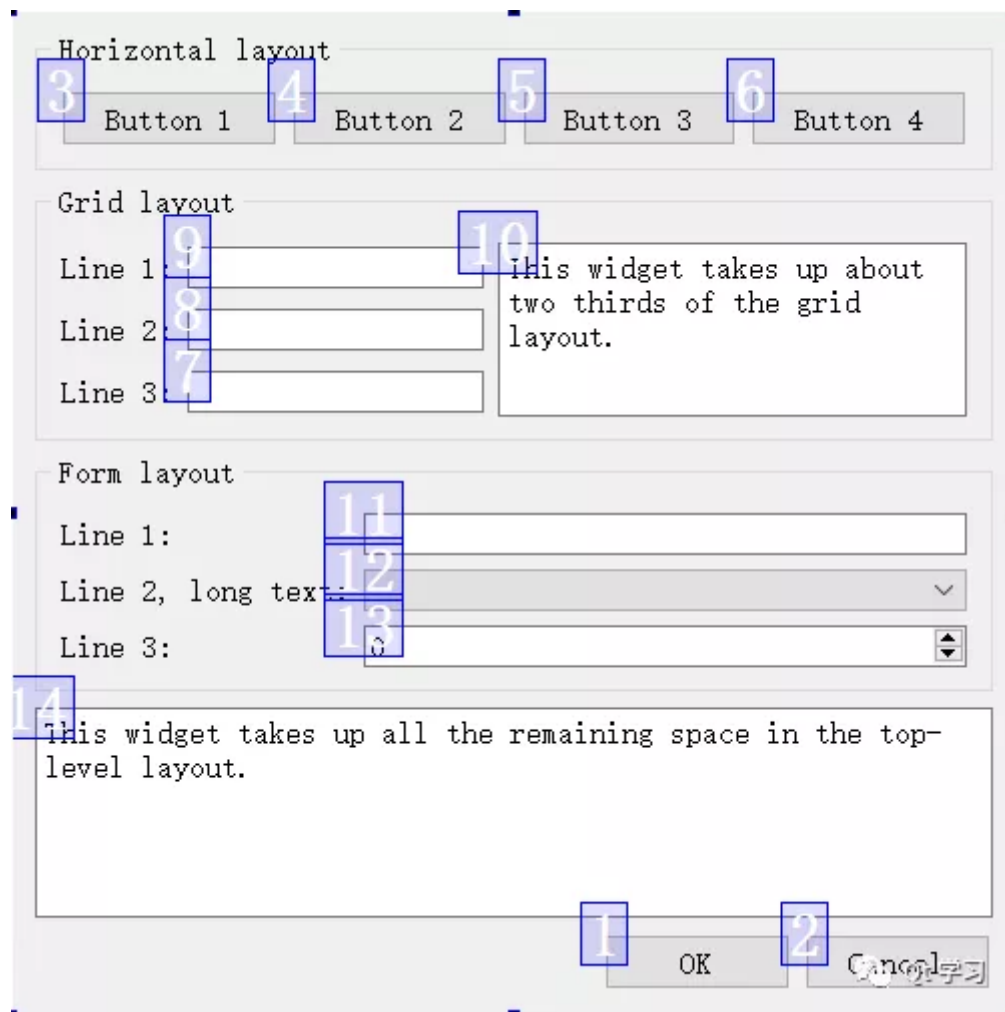
在设计工具栏上单击 "Edit Buddies" 按钮可以进入伙伴关系编辑状态，进入伙伴编辑状态之后的界面如下图所示



伙伴关系 (Buddy) 是指界面上一个 **Label** 和一个组件相关联，上图中的伙伴关系编辑状态，单击一个 Label，按住鼠标左键，然后拖向一个组件，就建立了 Label 和组件之间的伙伴关系。伙伴关系是为了在程序运行时，在窗体上用快捷键快速将输入焦点切换到某个组件上。

在 Qt 设计器工具栏上单击 "Edit Tab Order" 按钮进入 Tab 顺序编辑状态，如下图所示，Tab 顺序是指在程序运行时，按下键盘

上的 Tab 键时输入焦点的移动顺序。一个好的用户界面，在按 Tab 键时，焦点应该以合理的顺序在界面上移动，而不是随意地移动。



如上图，进入 Tab 顺序编辑状态后，在界面上会显示具有 Tab 顺序组件的编号，依次按希望的顺序单击组件，就可以重排 Tab 顺序了。没有输入焦点的组件是没有 Tab 顺序的，如 Label 标签组件。

总结:

介绍了基本的布局编排技巧，如何查看各组件之间的层次关系，以及设置组件的之间的伙伴关系和编号顺序。

下一篇《代码化 UI 设计》

全文结束，感谢大家！

相关阅读:

《UI 文件设计与运行机制》

《编写一个 Hello World 程序》

《Qt Creator 初步使用》

《Qt 的许可类型、主要版本以及安装步骤》

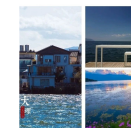
《Qt 的开源版本与商业版区别及 LGPL 与闭源程序》

《Qt 简介》

喜欢此内容的人还喜欢

走进图文并茂的攻略世界

Stata and Python数据分析



三星10多名员工确诊新冠；英国政府或考虑阻止英伟达收购Arm；代工厂首现成熟制程晶圆长单

与非网eefocus





20年的嵌入式经验：如何从零开始开发一款嵌入式产品(值得收藏的高质量文章!)

芯片之家

