

## 目录

1、常见问题.....	2
01、图标图库素材.....	2
(1) 阿里巴巴矢量库.....	2
(2) 背景图库推荐网址.....	2
02、resource_rc.qrc 文件的生成配置.....	2
03、resource_rc.py import 引入报错.....	2
2、qtdesigner 使用问题.....	3
01、MainWindow 创建的时候，可以删除多余的对象.....	3
02、为当前的窗口创建资源库.....	3
3、技巧.....	4
01、Label 做背景.....	4
02、widget 做按钮母体.....	4
03、同一个地方，控件按逻辑先后展示.....	5
4、样式（控件）.....	6
01、添加自己想要的样式.....	6
02、QLabel 的样式.....	6
03、Push Button 的样式.....	7
5、自定义组件.....	9
01、右上角的关闭按钮.....	9

# 1、常见问题

## 01.、图标图库素材

- (1) [阿里巴巴矢量库](#)
- (2) [背景图库推荐网址](#)

## 02、resource\_rc.qrc 文件的生成配置

```
p\Scripts\pyrcc5.exe resource.qrc -o resource_rc.py
```

## 03、resource\_rc.py import 引入报错

在 qtdesigner 生成.ui 文件的时候，  
会将本 ui 文件对应的 qrc 文件生成的 resource\_rc.py 文件  
导入当前 ui 文件生成的 py 文件的最后一行。

运行会报错：ImportError : no module named resource\_rc

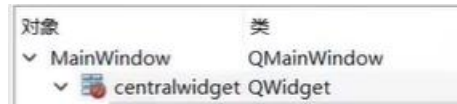
解决方法：

将 ui 文件生成的 py 文件中，引入的 import resource\_rc，去掉引入  
然后在 main.py 文件中引入,即可正常使用

## 2、qtdesigner 使用问题

### 01、MainWindow 创建的时候，可以删除多余的对象

例如只剩下面几个对象

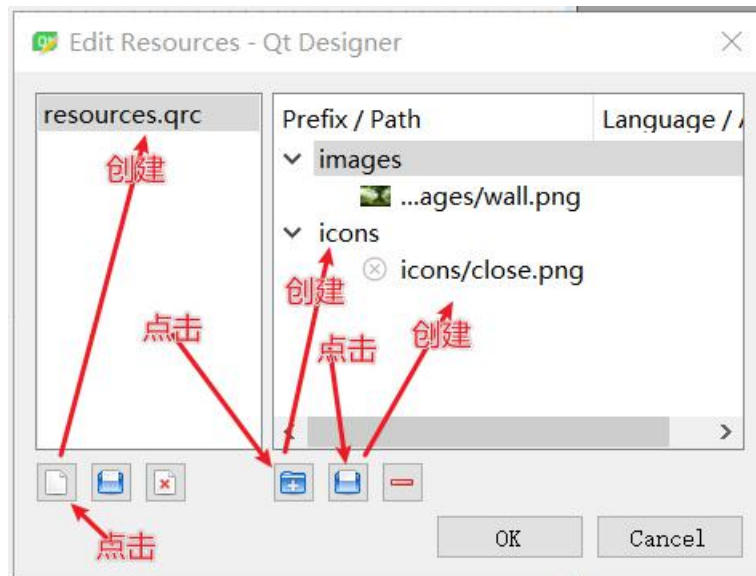


### 02、为当前的窗口创建资源库

在 qtdesigner 界面的右下角 Resource Browser 框中，点击小笔编辑资源



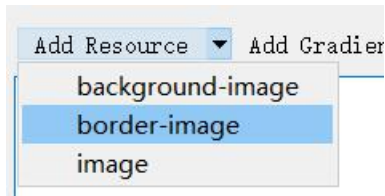
在弹出框界面，点击创建 qrc 文件。



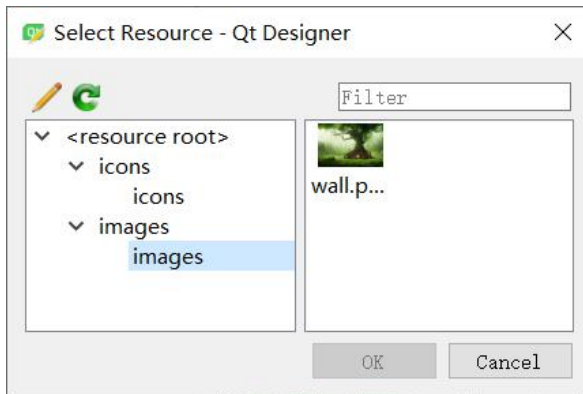
## 3、技巧

### 01、Label 做背景

- a、右键 Label 控件，点击修改样式表
- b、点击 Add Resource，选择 border-image



- c、在弹出框中，选择我们在本 pdf 第二章《2、qtdesigner 使用问题》中 02 问题中，创建的资源



- d、得到如下效果



### 02、widget 做按钮母体

好处，不会被打乱。

创建一个 widget 控件，然后在里面放置两个按钮，然后右键 widget 控件，选布局，将布局修改为水平布局。如下图



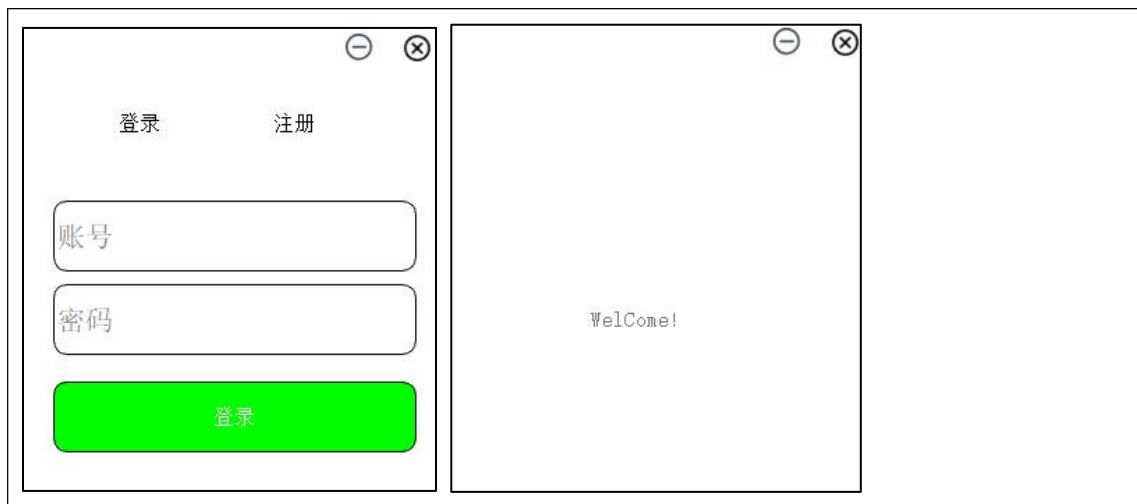
### 03、同一个地方，控件按逻辑先后展示

两者在 qtdesigner 的对象视图中，必须是同级关系，不是上下级关系。

通常：第一个控件 widget 正常放置、第二个 widget 要放在同一个地方，**第二块控件的大小一定要大于第一块控件的大小，并且包含第一块，注意他们不是继承关系。**

如下需求：点击登录后，隐藏登录控件，放置欢迎的文本控件

左图点击登录后，同一块地方显示另外一个控件



代码层面应该做的

```
class LoginComponent(QMainWindow, Ui_YCtools):
    def __init__(self):
        super(LoginComponent, self).__init__()
        self.setupUi(self)
        self.setWindowFlag(Qt.FramelessWindowHint)
        self.setAttribute(Qt.WA_TranslucentBackground)

        self.widget_4.hide()
        self.pushButton_3.clicked.connect(self.HideLogin)
    def HideLogin(self):
        self.widget_2.hide()
        self.widget.hide()
        self.widget_4.show()
```

## 4、样式（控件）

### 01、添加自己想要的样式

通过右键控件，选择修改样式表，添加我们想要的样式。

### 02、QLabel 的样式

- a、如下样式：第二个和第三个样式修改图片为圆角
- ```
border-image: url(/images/images/wall.png);  
border-top-left-radius:20px;  
border-bottom-left-radius:20px;
```
- 效果图



- b、或者如下样式：将背景置为白色修改背景为圆角
- ```
background-color: rgb(255, 255, 255);  
border-top-right-radius:20px;  
border-bottom-right-radius:20px;
```
- 效果图：如下图右侧 QLabel 控件



## 03、Push Button 的样式

### 样式 1:

右键创建的 button，选择“修改样式表”

修改前的样式：如下图



如下：第一个样式修改按钮的边框，为无边框，第二个样式为鼠标放上去后，按钮的颜色  
其中样式的格式为：# 加上 pushButton 的 objectName 名称

```
#pushButton{ border:none; }  
#pushButton:focus{ color: rgb(170, 170, 127); }
```

修改后样式：如下图



### 样式 2:

右键创建的 button，选择“修改样式表”

修改前的样子：如下图



添加如下三个样式：

第二个 hover 表示鼠标放置在按钮上方时候，按钮呈现的样式

第三个 pressed 表示鼠标按下按钮的时候，按钮呈现的样式

```
#pushButton_3{  
    border-radius:10px;  
    border:1px solid black;  
    background-color: rgb(170, 170, 127);  
}  
#pushButton_3:hover{  
    background-color: rgb(0, 255, 0);  
    color: rgb(255, 255, 255);  
}  
#pushButton_3:pressed{
```

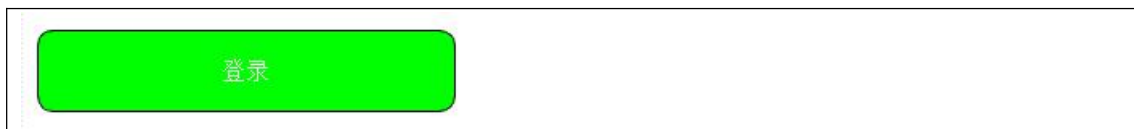
```
padding-top:7px;  
padding-left:5px;  
}
```

最终效果如下： 共有三种样式

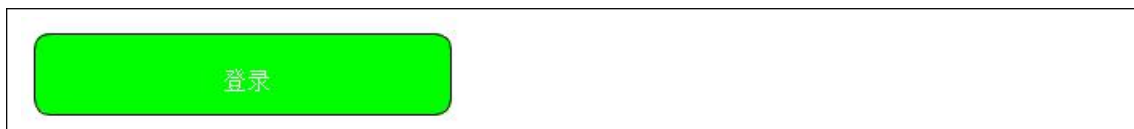
一下两种情况以外：背景是灰色的，字是黑色的



鼠标放置在按钮上：背景颜色，文字颜色变了



按下后：文字往下移了





## 5、自定义组件

### 01、右上角的关闭按钮

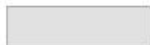
结果图：如下



1、首先在图标库内下载合适大小的矢量图，选择合适的大小，一般都是 32 位

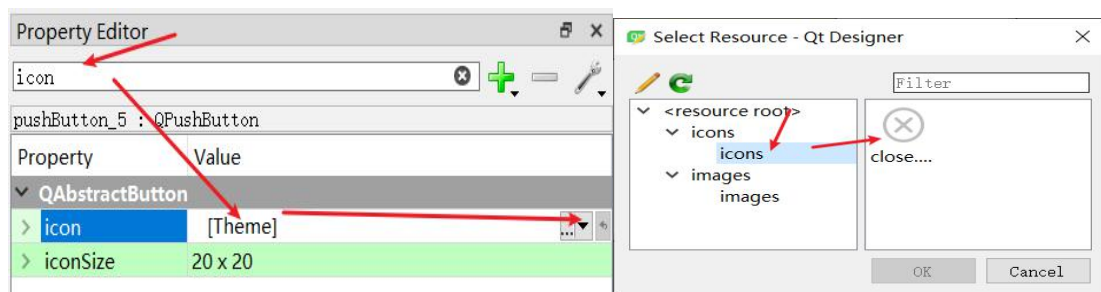


2、放置一个 Push Button 控件，并去除其中的文字，如下图



3、点击我们创建的 Push Button，然后在 qtdesigner 的右边“属性编辑器”中搜索“icon”

在下图中点击“倒三角符号▼”，选择“choose resources”，然后选择我们添加进去的图标



4、然后右键我们创建的 Push Button，选择“修改样式表”

将按钮的 border 去掉。

```
border:none;
```

走完 4 步后，得到的结果图

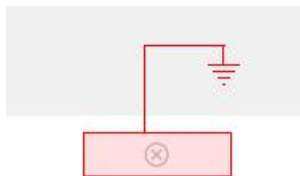


##### 5、给关闭按钮添加“关闭事件”

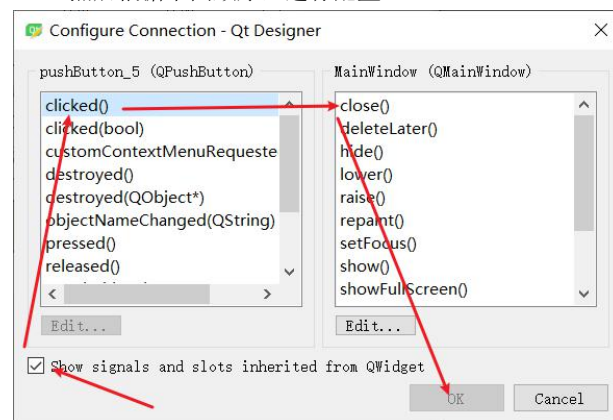
在 qtdesigner 工具栏中点击“编辑槽和信号”



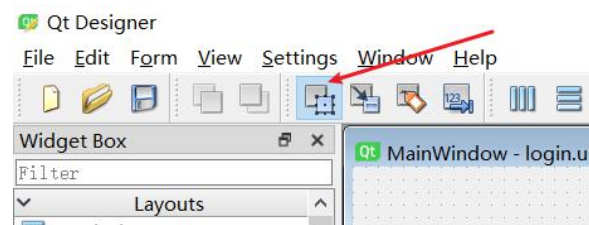
鼠标左键，点击刚刚创建的按钮，将红线拖动到外面，松开左键弹出槽函数编辑框



然后根据下图顺序，进行配置



配置完成后，请点击 qtdesigner 工具栏中的“编辑 widget”功能，结束编辑



到此已完成定制按钮的制作。