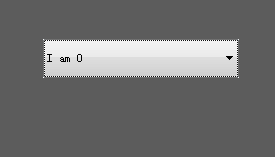
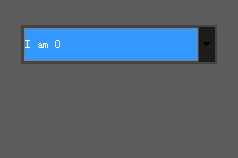
什么也不说，就拿QcomboBox来开刀吧。

1. 替换QcomboBox背景

**1）.什么也没设置效果：**



**2）.设置背景透明效果**



设置代码：

m\_Comb->setStyleSheet(

"QComboBox"

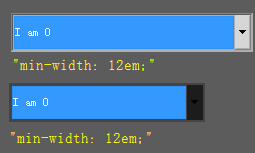
"{"

"background-color:rgba(0 , 0 , 0 , 50%);"

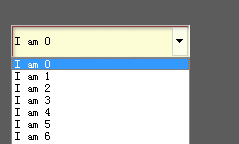
"min-width: 12em;"//最小宽度①

"}"

1. ：最小宽度指的是整个QCombobox的宽度比较一下两图



**3).设置背景图片效果**



m\_Comb->setStyleSheet(

"QComboBox"

"{"

"background-color:rgba(255 , 255 , 255 , 50%);"

"background-image: url(:/in.png);"

"}"

需要注意的是，设置背景图片和设置背景颜色必须要同时使用，否则背景图片显示不出来。

**4）.设置边框颜色**



m\_Comb->setStyleSheet(

"QComboBox"

"{"

"background-color:rgba(255 , 255 , 255 , 50%);"

"background-image: url(:/in.png);"

"border: 5px solid red;"

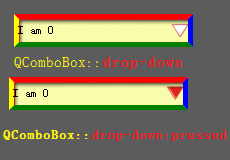
" border-color: red blue green yellow ;"

"}"

注释：border有三个参数，1.画笔像素2.画笔样式3.画笔颜色

Border-color后面可以跟一个、两个、三个、四个参数，分别控制4个边框的颜色。

**5）.更改右边的按下按钮**



m\_Comb->setStyleSheet(

"QComboBox"

"{"

"background-color:rgba(255 , 255 , 255 , 50%);"

"background-image: url(:/in.png);"

"border: 5px solid red;"

" border-color: red blue green yellow ;"

"QComboBox::drop-down"//默认向下按钮

"{"

"image: url(:/1.png);"

" max-width: 12px;"

" max-height: 12px;"

"}"

"QComboBox::drop-down:pressed"//鼠标按下改变向下按钮

"{"

"image: url(:/0.png);"

" max-width: 12px;"

" max-height: 12px;"

"}"

"QComboBox::drop-down:hover"//鼠标悬浮，改变向下按钮

"{"

"image: url(:/0.png);"

" max-width: 12px;"

" max-height: 12px;"

"}"

"}"

需要注意的是，drop-down和down-arrow的区别

drop-down:指的是向下的按钮，



down-arrow：是指向下的箭头，

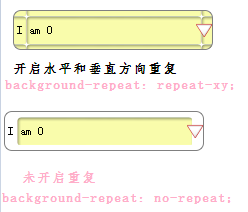


注意这两者的区别，drop-down有hover属性，down-arrow没有。

| **Background** | **{ [Brush](http://doc.qt.digia.com/4.6/stylesheet-reference.html" \l "brush)  | [Url](http://doc.qt.digia.com/4.6/stylesheet-reference.html" \l "url)  | [Repeat](http://doc.qt.digia.com/4.6/stylesheet-reference.html" \l "repeat)  | [Alignment](http://doc.qt.digia.com/4.6/stylesheet-reference.html" \l "alignment) }\*** |  |
| --- | --- | --- |
| **Type** | **Syntax** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |

Background分别有4个参数。

1. 画刷，可以是颜色值，可以是渐变
2. 图片路径，可以为当前可视控件加载一个背景图
3. 重复值，可以水平重复，可以垂直重复。效果如下：



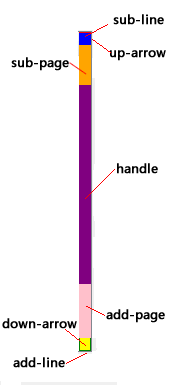
4.图片的位置。

QScrollBar样式

      滚动条在Qt里叫做QScrollBar，使用样式设置控件的外观很方便，这里分享一下滚动条的样式设置。例如下面的树形控件的滚动条的样式。

const QString VSCROLLBAR\_STYLE =

            "QScrollBar:vertical {margin: 13px 0px 13px 0px;background-color: rgb(255, 255, 255, 100); border: 0px; width: 12px;}\  
             .QScrollBar::handle:vertical {background-color: purple; width: 12px; }\  
             .QScrollBar::add-page:vertical {background-color: pink; }\  
             .QScrollBar::sub-page:vertical {background-color: orange; }\  
             .QScrollBar::sub-line:vertical {subcontrol-origin: margin; border: 1px solid red; min-height:13px}\  
             .QScrollBar::up-arrow:vertical { subcontrol-origin: margin; background-color: blue; height:13px}\  
             .QScrollBar::down-arrow:vertical { background-color: yellow; }\  
             .QScrollBar::add-line:vertical {subcontrol-origin: margin; border: 1px solid green; height:13px}";



QScrollBar分为sub-line、add-line、add-page、sub-page、up-arrow、down-arrow和handle几个部分，这几个相应的部位都己经在上面画出来了。sub-line包含了up-arrow，而add-line则包含了down-arrow区域。

需要注意的是，如果不对sub-line、add-line设置subcontrol的话，会出现下面的问题。如果不设置整个QScrollBar的marg-left和margin-bottom话，则不能定制add-line和sub-line区域。



没有定义sub-control为margin

没有定义margin-top和margin-bottom

