每条QSS样式都由两部分组成：

1. 选择器，该部分指定要美化的控件

2. 声明，该部分指定要在控件上使用的属性和值

如：QPushButton {color: red}

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

demo = Demo()

qss = 'QPushButton {color: red}'

demo.setStyleSheet(qss) # 调用setStyleSheet()方法设置整个程序的样式

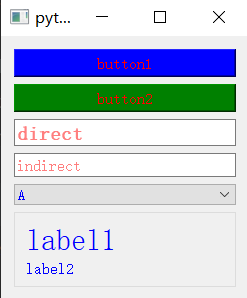
demo.show()

sys.exit(app.exec\_())

**可以在Qt Assistant中输入Qt Style Sheets Reference查询所有的属性和值，子控件以及伪状态等**

|  |  |  |
| --- | --- | --- |
| 选择器例子 | 例子 | 解释 |
| 通用选择器 | \* | 匹配所有控件 |
| 类型选择器 | QPushButton | 匹配所有QPushButton实例及其子类 |
| 属性选择器 | QPushButton[name='btn']  QPushButton[name~='btn'] | 匹配所有name属性为btn的QPushButton实例，~=代表匹配所有name属性中包含btn的QPushButton实例 |
| 类选择器 | QPushButton | 匹配所有objectName为okButton的QPushButton的实例 |
| ID选择器 | QPushButton#okButton | 匹配所有objectName为okButton的QPushButton实例 |
| 后代选择器 | QDialog QPushButton | 匹配所有QDialog控件中包含的QPushButton实例(无论是直接包含还是间接包含) |
| 子选择器 | QDialog > QPushButton | 匹配所有QDialog控件中直接包含的QPushButton实例 |

\* {color: red}



# 所有控件的文本颜色都设为红色

#

# 为什么不是所有控件文本颜色都是红色？

# 当选择器写的越具体，选择器的优先程度越高。通配符\*这一选择器写

# 法非常笼统，而之后几条样式的选择器都是指定了控件名称的，比通

# 配符更加具体，所以优先度更高

QPushButton {background-color: blue}

# 把所有QPushButton实例及其子类的背景颜色设为蓝色

QPushButton[name='btn'] {background-color: green}

# 把所有name属性为btn的QPushButton实例的背景色设为绿色

QLineEdit {font: bold 20px}

# 把所有QLineEdit实例(不包括子类)的字体变粗，大小设为15px

QComboBox#cb {color: blue}

# 把所有objectName为cb的下拉框文本颜色设为蓝色

QGroupBox QLabel {color: blue}

# 把所有包含在QStackedWidget中的QLabel控件的文本颜色设为蓝色(无论直接包含还是

间接包含)

QGroupBox > QLabel {font: 30px}

# 把所有QStackedWidget直接包含的QLabel文本字体大小设为30px