

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

