# 1.按钮pushbutton

|  |
| --- |
| **import** sys **from** PyQt5.QtWidgets **import** QApplication,QWidget,QPushButton  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app = QApplication(sys.argv)  w = QWidget()  w.setWindowTitle(**"btn demo"**)  *# 创建按钮* btn = QPushButton(**"hello"**)  *# 为按钮设置父窗口* btn.setParent(w)  btn.move(100,100)  w.show()   app.exec() |
| **import** sys **from** PyQt5.QtWidgets **import** QApplication,QWidget,QPushButton  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app = QApplication(sys.argv)  w = QWidget()  w.setWindowTitle(**"btn demo"**)  *# 创建按钮  # btn = QPushButton("hello")  # # 为按钮设置父窗口  # btn.setParent(w)* btn = QPushButton(**"hello"**,w) *# 可以把上面的两句话合并* btn.move(100,100)  w.show()   app.exec() |

# 2.标签控件Qlabel

|  |
| --- |
| **import** sys **from** PyQt5.QtWidgets **import** QApplication,QWidget,QLabel  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app = QApplication(sys.argv)  w = QWidget()  w.setWindowTitle(**"btn demo"**)  *# 创建标签* lbl = QLabel(**"这是一个标签"**)  *# lbl.setGeometry(20,20,30,30)  # 为按钮设置父窗口* lbl.setParent(w)  *# lbl.move(100,100)* w.show()   app.exec() |
| **import** sys **from** PyQt5.QtWidgets **import** QApplication,QWidget,QLabel  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app = QApplication(sys.argv)  w = QWidget()  w.setWindowTitle(**"btn demo"**)  *# 创建标签  # lbl = QLabel("这是一个标签")  # # 为按钮设置父窗口  # lbl.setParent(w)* lbl = QLabel(**"QLable标签"**,w) *#把上面两句话合在一起写，这样子创建的窗口比较小，需要调用下面的语句修改大小* lbl.setGeometry(20,20,300,300) *#设置窗口大小  # lbl.move(100,100)* w.show()   app.exec() |

# 3.单行编辑框控件LineEdit

|  |
| --- |
| **import** sys **from** PyQt5.QtWidgets **import** QWidget, QApplication, QLabel, QPushButton, QLineEdit, QMessageBox   **def** clicked():  QMessageBox.information(**None**, **"info"**,edit.text())  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app = QApplication(sys.argv)  w = QWidget()  w.setWindowTitle(**"demo"**)  label = QLabel(**"用户名："**, w)  label.setGeometry(20, 20, 50, 20)  edit = QLineEdit(**"请输入用户名"**, w)  edit.setGeometry(65, 20, 180, 20)  btn = QPushButton(**"ok"**, w)  btn.setGeometry(50, 80, 70, 30)  *# 绑定事件* btn.clicked.connect(clicked)  w.show()  app.exec() |

# 4.布局LayOut

## Qt里面有4大类布局

|  |
| --- |
|  |

## 1.QboxLayout，这个使用的比较多

### 1）垂直：QHBoxLayout

|  |
| --- |
| **import** sys **from** PyQt5.QtWidgets **import** QApplication, QVBoxLayout, QWidget, QPushButton **from** PyQt5.QtCore **import** Qt   **class** MyWidget(QWidget):  **def** \_\_init\_\_(self):  super().\_\_init\_\_()  *# 设置窗口大小* self.resize(500, 400)  self.setWindowTitle(**"vbox layout demo"**)  *# 创建垂直布局对象* layout = QVBoxLayout()  layout.addStretch(1)  btn1 = QPushButton(**"button1"**)  layout.addWidget(btn1)   layout.addStretch(1)  btn2 = QPushButton(**"button2"**)  layout.addWidget(btn2)   layout.addStretch(1) *# 设置间距* btn3 = QPushButton(**"button3"**)  layout.addWidget(btn3)   self.setLayout(layout) *# 布局需要调用QWidget的setLayout(layout)来设置,否则没有效果* **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app = QApplication(sys.argv)  w = MyWidget()  w.show()  app.exec() |

### 2）水平BoxLayout,注意,不能直接把图片添加到layout上面,它只能添加控件

|  |
| --- |
|  |
| **import** sys **from** PyQt5.QtWidgets **import** QApplication, QHBoxLayout, QWidget,QLabel **from** PyQt5.QtCore **import** Qt **from** PyQt5.QtGui **import** QImage,QPixmap   **class** MyWidget(QWidget):  **def** \_\_init\_\_(self):  super().\_\_init\_\_()  *# 设置窗口大小* self.resize(500, 400)  self.setWindowTitle(**"vbox layout demo"**)  *# 创建水平布局对象* layout = QHBoxLayout()  layout.addStretch(1)  *#写法1  # img1 = QImage("jerry.png")  # lbl1 = QLabel()  # lbl1.setPixmap(QPixmap.fromImage(img1))    #写法2* pix = QPixmap(**"jerry.png"**)  lbl1 = QLabel()  lbl1.setPixmap(pix)  layout.addWidget(lbl1) *# 不能把图片直接添加到layout上面* layout.addStretch(1)  img2 = QImage(**"rocket.jpg"**)  lbl2 = QLabel()  lbl2.setPixmap(QPixmap.fromImage(img2))  layout.addWidget(lbl2) *# 不能把图片直接添加到layout上面* layout.addStretch(1)  img3 = QImage(**"tom.jpg"**)   lbl3 = QLabel()  lbl3.setPixmap(QPixmap.fromImage(img3))  layout.addWidget(lbl3) *# 不能把图片直接添加到layout上面* self.setLayout(layout) *# 布局需要调用QWidget的setLayout(layout)来设置,否则没有效果* **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app = QApplication(sys.argv)  w = MyWidget()  w.show()  app.exec() |

#### 注意:一个窗口只能使用一种布局器风格,如果你写了2给布局器,后面写的那个会覆盖前面写的那个,除非你使用布局器嵌套.,例如可以中一个垂直布局器里面嵌套使用一个垂直布局器和一个水平布局器

## 学习了水平和垂直布局后我们做一个总案例,使用两个布局的嵌套:

### combolayout.py

|  |
| --- |
| *#一般来说,一个窗口只能使用一种布局,但是布局可以嵌套.我们来做一个嵌套的,用垂直布局器里面嵌套一个垂直布局器和一个水平布局器* **import** sys **from** PyQt5.QtWidgets **import** QApplication,QWidget,QVBoxLayout,QHBoxLayout,QGroupBox,QRadioButton  **class** MyLayoutWidget(QWidget):  **def** \_\_init\_\_(self):  super().\_\_init\_\_()  self.initUi()   **def** initUi(self):  *# 创建一个兴趣爱好组框* interestBox = QGroupBox(**"兴趣爱好"**)  vl = QVBoxLayout() *# 创建一个垂直布局器,注意先将控件添加到布局器,然后在组框上面设置布局  # 创建单选按钮* btn1 = QRadioButton(**"看电视"**)  btn2 = QRadioButton(**"吃饭"**)  btn3 = QRadioButton(**"泡妞"**)  *# 把一组的单选按钮添加到垂直布局器上面* vl.addWidget(btn1)  vl.addWidget(btn2)  vl.addWidget(btn3)  *#在组框上面设置布局* interestBox.setLayout(vl)    *# 创建一个性别组框* genderBox = QGroupBox(**"性别"**)  *# 创建一个水平布局器* hl = QHBoxLayout()  *# 创建两个性别对应的单选按钮* btn4 = QRadioButton(**"男"**)  btn5 = QRadioButton(**"女"**)  *# 把单选按钮添加到水平布局器上* hl.addWidget(btn4)  hl.addWidget(btn5)  *# 在性别组框上面设置水平布局器* genderBox.setLayout(hl)   *# 创建垂直总布局* vlayout = QVBoxLayout()  *#把这两个组框添加到总布局器上* vlayout.addWidget(interestBox)  vlayout.addWidget(genderBox)  *# 最后在窗口上设置布局器为我们的总布局器* self.setLayout(vlayout)  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app = QApplication(sys.argv)  w = MyLayoutWidget()  w.show()  app.exec() |

#### 注意控件的添加流程和添加顺序,把他们的关系理清楚

## 2.GridLayout

## 也叫做九宫格布局，需要使用双重循环来实现添加组件，而且注意将一个布局添加到另外一个布局使用addLayout方法，不能使用addWidget方法，因为改方法只添加普通组件

### gridlayout/gridlayout.py

|  |
| --- |
| **import** sys **from** PyQt5.QtWidgets **import** QGridLayout,QWidget,QApplication,QVBoxLayout,QLineEdit,QPushButton  **class** MyGridLayout(QWidget):  **def** \_\_init\_\_(self):  super().\_\_init\_\_()  self.setupUI()   **def** setupUI(self):  self.setWindowTitle(**"模拟计算器界面"**)  *# 网格时间* data = {  0:[**"7"**,**"8"**,**"9"**,**"+"**,**"("**], #注意 python中字典的key是不可变的  1:[**"4"**,**"5"**,**"6"**,**"-"**,**")"**],  2:[**"1"**,**"2"**,**"3"**,**"\*"**,**"←"**],  3:[**"0"**,**"."**,**"="**,**"/"**,**"c"**],  }   *#整体垂直布局* layout = QVBoxLayout()   *# 输入框* edit = QLineEdit()  edit.setPlaceholderText(**"请输入内容"**)  layout.addWidget(edit)  *#创建网格布局* grid = QGridLayout()  *#利用循环来给网格布局添加组件* **for** num,data **in** data.items():  **for** index,val **in** enumerate(data):  btn = QPushButton(val)  grid.addWidget(btn,int(num),index)  layout.addLayout(grid) *# 将一个布局添加到另外一个布局使用addLayout方法，不能使用addWidget方法，因为改方法只添加普通组件* self.setLayout(layout)  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app = QApplication(sys.argv)  w = MyGridLayout()  w.show()  app.exec() |
|  |

## 3布局器FormLayout，一般用来提交form表单数据的

|  |
| --- |
| **import** sys **from** PyQt5.QtWidgets **import** QGridLayout,QWidget,QApplication,QVBoxLayout,QLineEdit,QPushButton,QFormLayout **from** PyQt5.QtCore **import** Qt  **class** MyFormLayout(QWidget):  **def** \_\_init\_\_(self):  super().\_\_init\_\_()  self.setupUI()   **def** setupUI(self):  self.setWindowTitle(**"form layout"**)  self.resize(300,150)  *#整体表单布局* container = QVBoxLayout()  formlayout = QFormLayout()  *# 输入框* edit = QLineEdit()  edit.setPlaceholderText(**"请输入用户名"**)  formlayout.addRow(**"账号"**,edit)   edit2 = QLineEdit()  edit2.setPlaceholderText(**"请输入密码"**)  formlayout.addRow(**"密码"**,edit2)   container.addLayout(formlayout)  login = QPushButton(**"登录"**)  login.setFixedSize(100,30)  container.addWidget(login,alignment=Qt.AlignRight)  self.setLayout(container)  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app = QApplication(sys.argv)  w = MyFormLayout()  w.show()  app.exec() |
|  |

## 4.抽屉布局器StackedLayout

|  |
| --- |
|  |

### 代码1：stackedlayout.py

|  |
| --- |
| **import** sys **from** PyQt5.QtWidgets **import** QWidget,QApplication,QStackedLayout,QLabel,QPushButton,QVBoxLayout **from** PyQt5.QtGui **import** QPixmap  **class** Window1(QWidget):  **def** \_\_init\_\_(self):  super().\_\_init\_\_()  lbl= QLabel(**"抽屉1的内容"**,self)  lbl.setPixmap(QPixmap(**"xg5.jpg"**))  self.label = lbl  self.setStyleSheet(**"background-color:green;"**)  \_,\_,width,height = self.frameGeometry().getRect()  self.label.setGeometry(30,10,width//2,height//2)  **class** Window2(QWidget):  **def** \_\_init\_\_(self):  super().\_\_init\_\_()  lbl = QLabel(**"抽屉2的内容"**,self)  lbl.setPixmap(QPixmap(**"kdk.jpg"**))  self.label = lbl  self.setStyleSheet(**"background-color:orange;"**)  \_, \_, width, height = self.frameGeometry().getRect()  self.label.setGeometry(30, 10, width // 2, height // 2)  **class** MyStackedLayout(QWidget):  **def** \_\_init\_\_(self,parent=**None**):  super().\_\_init\_\_(parent)  self.create\_stack\_layout()  self.setupUI()   **def** create\_stack\_layout(self):  self.stackLayout = QStackedLayout()  *#创建抽屉widget* win1 = Window1()  win2 = Window2()  *#将widget添加到抽屉布局中* self.stackLayout.addWidget(win1)  self.stackLayout.addWidget(win2)   **def** setupUI(self):  self.setFixedSize(400,300)  self.setWindowTitle(**"抽屉布局demo"**)  container = QVBoxLayout()  container.addLayout(self.stackLayout)  btn1 = QPushButton(**"抽屉1"**)  btn2 = QPushButton(**"抽屉2"**)  btn1.clicked.connect(self.show1)  btn2.clicked.connect(self.show2)  container.addWidget(btn1)  *# container.addStretch(1)* container.addWidget(btn2)  *# container.addStretch(1)* self.setLayout(container)  **def** show1(self):  self.stackLayout.setCurrentIndex(0) *#stackedlayout显示指定组件的方法* **def** show2(self):  self.stackLayout.setCurrentIndex(1)   **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app = QApplication(sys.argv)  w = MyStackedLayout()  w.show()  app.exec() |
|  |

### 代码2：stackedlayout2.py

|  |
| --- |
| **import** sys **from** PyQt5.QtWidgets **import** QWidget, QApplication, QStackedLayout, QLabel, QPushButton, QVBoxLayout **from** PyQt5.QtGui **import** QPixmap   **class** Window1(QWidget):  **def** \_\_init\_\_(self):  super().\_\_init\_\_()  lbl = QLabel(**"抽屉1的内容"**, self)  lbl.setPixmap(QPixmap(**"xg5.jpg"**))  self.label = lbl  self.setStyleSheet(**"background-color:green;"**)  \_, \_, width, height = self.frameGeometry().getRect()  self.label.setGeometry(30, 10, width // 2, height // 2)   **class** Window2(QWidget):  **def** \_\_init\_\_(self):  super().\_\_init\_\_()  lbl = QLabel(**"抽屉2的内容"**, self)  lbl.setPixmap(QPixmap(**"kdk.jpg"**))  self.label = lbl  self.setStyleSheet(**"background-color:orange;"**)  \_, \_, width, height = self.frameGeometry().getRect()  self.label.setGeometry(30, 10, width // 2, height // 2)   **class** MyStackedLayout(QWidget):  **def** \_\_init\_\_(self, parent=**None**):  super().\_\_init\_\_(parent)  self.create\_stack\_layout()  self.setupUI()   **def** create\_stack\_layout(self):  self.stackLayout = QStackedLayout()  *# 创建抽屉widget* win1 = Window1()  win2 = Window2()  *# 将widget添加到抽屉布局中* self.stackLayout.addWidget(win1)  self.stackLayout.addWidget(win2)   **def** setupUI(self):  self.setFixedSize(400, 300)  self.setWindowTitle(**"抽屉布局demo"**)  container = QVBoxLayout()  widget = QWidget()  widget.setLayout(self.stackLayout)  widget.setStyleSheet(**"background-color:gray;"**)  container.addWidget(widget)  btn1 = QPushButton(**"抽屉1"**)  btn2 = QPushButton(**"抽屉2"**)  btn1.clicked.connect(self.show1)  btn2.clicked.connect(self.show2)  container.addWidget(btn1)  *# container.addStretch(1)* container.addWidget(btn2)  *# container.addStretch(1)* self.setLayout(container)   **def** show1(self):  self.stackLayout.setCurrentIndex(0) *# stackedlayout显示指定组件的方法* **def** show2(self):  self.stackLayout.setCurrentIndex(1)   **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app = QApplication(sys.argv)  w = MyStackedLayout()  w.show()  app.exec() |
|  |