

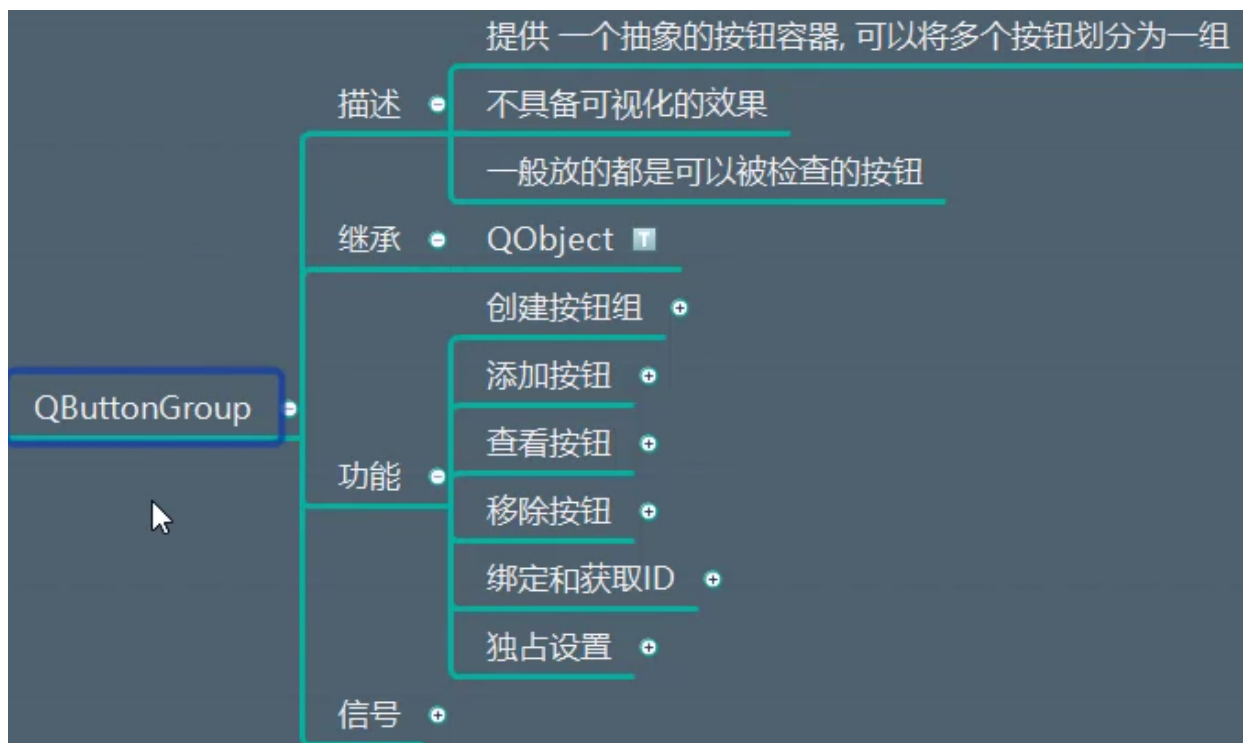
[笔记][LIKE-Python-GUI编程-PyQt5][14]

PyQt5

[笔记][LIKE-Python-GUI编程-PyQt5][14]

- 104. Python-GUI编程-PyQt5-QButtonGroup-创建和添加按钮
- 105. Python-GUI编程-PyQt5-QButtonGroup-查看按钮
- 106. Python-GUI编程-PyQt5-QButtonGroup-移除按钮
- 107. Python-GUI编程-PyQt5-QButtonGroup-ID的操作
- 108. Python-GUI编程-PyQt5-QButtonGroup-独占操作
- 109. Python-GUI编程-PyQt5-QButtonGroup-信号使用

104. Python-GUI编程-PyQt5-QButtonGroup-创建和添加按钮



可以解决多组互斥问题

继承自 `QObject` , 不具备可视化效果 , 一般要继承自 `QWidget` 才有可视化效果

```
# 0. 导入需要的包和模块
import sys
from PyQt5.Qt import *

# 1. 创建一个应用程序对象
app = QApplication(sys.argv)

# 2. 控件的操作
# 2.1 创建控件
window = QWidget()
# 2.2 设置控件
window.setWindowTitle('按钮组的使用')
window.resize(500, 500)

# 创建4个单选按钮
# 分别是男、女和是、否
r_male = QRadioButton('男', window)
r_male.move(100, 100)
r_female = QRadioButton('女', window)
r_female.move(100, 150)
r_yes = QRadioButton('是', window)
r_yes.move(300, 100)
r_no = QRadioButton('否', window)
r_no.move(300, 150)

# 创建按钮组 QButtonGroup
gender_group = QButtonGroup(window) # 父对象 window 销毁的时候，子对象也会被销毁
# 添加按钮
gender_group.addButton(r_male)
gender_group.addButton(r_female)

# 如果不写下面的代码，也能完成多组互斥
# 但是一般还是再创建一个按钮组
ans_group = QButtonGroup(window)
ans_group.addButton(r_yes)
ans_group.addButton(r_no)

# 2.3 展示控件
window.show()

# 3. 应用程序的执行，进入到消息循环
sys.exit(app.exec_())
```

105. Python-GUI编程-PyQt5-QButtonGroup-查看按钮



```
# 0. 导入需要的包和模块
import sys
from PyQt5.Qt import *

# 1. 创建一个应用程序对象
app = QApplication(sys.argv)

# 2. 控件的操作
# 2.1 创建控件
window = QWidget()
# 2.2 设置控件
window.setWindowTitle('按钮组的使用')
window.resize(500, 500)

# 创建4个单选按钮
# 分别是男、女和是、否
r_male = QRadioButton('男', window)
r_male.move(100, 100)
# 默认选中男
r_male.setChecked(True)
r_female = QRadioButton('女', window)
r_female.move(100, 150)
r_yes = QRadioButton('是', window)
r_yes.move(300, 100)
r_no = QRadioButton('否', window)
r_no.move(300, 150)

# 创建按钮组 QButtonGroup
gender_group = QButtonGroup(window) # 父对象 window 销毁的时候, 子对象也会被销毁
# 添加按钮, 并传入 ID
gender_group.addButton(r_male, 1)
gender_group.addButton(r_female, 2)
# 查看所有按钮: buttons
print(gender_group.buttons())
print('-' * 50)
# 按ID查看按钮: button(id)
print('ID为2的按钮: ', gender_group.button(2))
# 查看哪个按钮被选中
```

```

print('被选中的按钮: ', gender_group.checkedButton())

ans_group = QButtonGroup(window)
ans_group.addButton(r_yes)
ans_group.addButton(r_no)

# 2.3 展示控件
window.show()

# 3. 应用程序的执行, 进入到消息循环
sys.exit(app.exec_())

```

106. Python-GUI编程-PyQt5-QButtonGroup-移除按钮



```

# 0. 导入需要的包和模块
import sys
from PyQt5.Qt import *

# 1. 创建一个应用程序对象
app = QApplication(sys.argv)

# 2. 控件的操作
# 2.1 创建控件
window = QWidget()
# 2.2 设置控件
window.setWindowTitle('移除按钮')
window.resize(500, 500)

# 创建4个单选按钮
# 分别是男、女和是、否
r_male = QRadioButton('男', window)
r_male.move(100, 100)
# 默认选中男
r_male.setChecked(True)
r_female = QRadioButton('女', window)
r_female.move(100, 150)

```

```

r_yes = QRadioButton('是', window)
r_yes.move(300, 100)
r_no = QRadioButton('否', window)
r_no.move(300, 150)

# 创建按钮组 QButtonGroup
gender_group = QButtonGroup(window) # 父对象 window 销毁的时候, 子对象也会被销毁
# 添加按钮, 并传入 ID
gender_group.addButton(r_male, 1)
gender_group.addButton(r_female, 2)

ans_group = QButtonGroup(window)
ans_group.addButton(r_yes)
ans_group.addButton(r_no)

# 移除女按钮
# 此时女只有一个父控件, 没有组
gender_group.removeButton(r_female)
# 查看组内按钮
print('组内成员: ', gender_group.buttons())
# 按ID为2查看按钮
print('ID为2的按钮: ', gender_group.button(2))

# 2.3 展示控件
window.show()

# 3. 应用程序的执行, 进入到消息循环
sys.exit(app.exec_())

```

107. Python-GUI编程-PyQt5-QButtonGroup-ID的操作



```
# 0. 导入需要的包和模块
```

```

import sys
from PyQt5.Qt import *

# 1. 创建一个应用程序对象
app = QApplication(sys.argv)

# 2. 控件的操作
# 2.1 创建控件
window = QWidget()
# 2.2 设置控件
window.setWindowTitle('ID操作')
window.resize(500, 500)

# 创建4个单选按钮
# 分别是男、女和是、否
r_male = QRadioButton('男', window)
r_male.move(100, 100)
# 默认选中男
r_male.setChecked(True)
r_female = QRadioButton('女', window)
r_female.move(100, 150)
r_yes = QRadioButton('是', window)
r_yes.move(300, 100)
r_no = QRadioButton('否', window)
r_no.move(300, 150)

# 创建按钮组 QButtonGroup
gender_group = QButtonGroup(window) # 父对象 window 销毁的时候, 子对象也会被销毁
# 添加按钮, 并传入 ID
gender_group.addButton(r_male, 1)
gender_group.addButton(r_female, 2)

ans_group = QButtonGroup(window)
ans_group.addButton(r_yes)
ans_group.addButton(r_no)

# 设置是为1, 否为2
ans_group.setId(r_yes, 1)
ans_group.setId(r_no, 2)
# 查看 ID
print('是按钮的ID: ', ans_group.id(r_yes))
print('否按钮的ID: ', ans_group.id(r_no))
# 查看选中的按钮对应的 ID 是多少
# 如果没有选中的按钮, 结果为 -1
# 注意: 所以千万不要把 ID 设置为 -1 !!!
print('回答组选中按钮的ID为: ', ans_group.checkedId())
# 设置否被选中
r_no.setChecked(True)
print('再次查看回答组选中的按钮ID为: ', ans_group.checkedId())

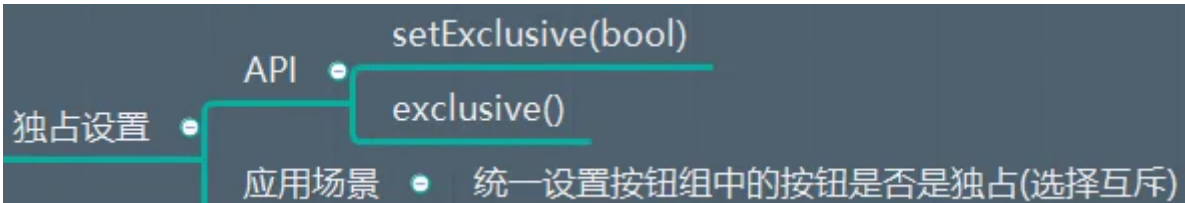
# 2.3 展示控件

```

```
window.show()
```

```
# 3. 应用程序的执行, 进入到消息循环  
sys.exit(app.exec_())
```

108. Python-GUI编程-PyQt5-QButtonGroup-独占操作



```
# 0. 导入需要的包和模块  
import sys  
from PyQt5.Qt import *  
  
# 1. 创建一个应用程序对象  
app = QApplication(sys.argv)  
  
# 2. 控件的操作  
# 2.1 创建控件  
window = QWidget()  
# 2.2 设置控件  
window.setWindowTitle('独占操作')  
window.resize(500, 500)  
  
# 创建4个单选按钮  
# 分别是男、女和是、否  
r_male = QRadioButton('男', window)  
r_male.move(100, 100)  
# 默认选中男  
r_male.setChecked(True)  
r_female = QRadioButton('女', window)  
r_female.move(100, 150)  
r_yes = QRadioButton('是', window)  
r_yes.move(300, 100)  
r_no = QRadioButton('否', window)  
r_no.move(300, 150)  
  
# 创建按钮组 QButtonGroup  
gender_group = QButtonGroup(window) # 父对象 window 销毁的时候, 子对象也会被销毁  
# 添加按钮, 并传入 ID
```

```

gender_group.addButton(r_male, 1)
gender_group.addButton(r_female, 2)

ans_group = QButtonGroup(window)
ans_group.addButton(r_yes)
ans_group.addButton(r_no)

# 设置非独占
gender_group.setExclusive(False)
# 获取独占状态
print('当前按钮组是否独占: ', gender_group.exclusive())

# 2.3 展示控件
window.show()

# 3. 应用程序的执行, 进入到消息循环
sys.exit(app.exec_())

```

109. Python-GUI编程-PyQt5-QButtonGroup-信号使用

信号

- buttonClicked(int/QAbstractButton) • 当按钮组中的按钮被点击时, 发射此信号
- buttonPressed(int/QAbstractButton) • 当按钮组中的按钮被按下时, 发射此信号
- buttonReleased(int/QAbstractButton) • 当按钮组中的按钮被释放时, 发射此信号
- buttonToggled(QAbstractButton/int, bool) • 当按钮组中的按钮被切换状态时, 发射此信号

如果一个对象向外界提供的信号名称一样, 但参数不一样

重点注意 • 外界在使用信号时, 可以使用如下格式进行选择

signal_name[type] • signal_name • 信号名称
type • 参数类型

```

# 0. 导入需要的包和模块
import sys
from PyQt5.Qt import *

# 1. 创建一个应用程序对象
app = QApplication(sys.argv)

# 2. 控件的操作
# 2.1 创建控件
window = QWidget()
# 2.2 设置控件
window.setWindowTitle('信号操作')

```



```

window.resize(500, 500)

# 创建4个单选按钮
# 分别是男、女和是、否
r_male = QRadioButton('男', window)
r_male.move(100, 100)
# 默认选中男
r_male.setChecked(True)
r_female = QRadioButton('女', window)
r_female.move(100, 150)
r_yes = QRadioButton('是', window)
r_yes.move(300, 100)
r_no = QRadioButton('否', window)
r_no.move(300, 150)

# 创建按钮组 QButtonGroup
gender_group = QButtonGroup(window) # 父对象 window 销毁的时候, 子对象也会被销毁
# 添加按钮, 并传入 ID
gender_group.addButton(r_male, 1)
gender_group.addButton(r_female, 2)

ans_group = QButtonGroup(window)
ans_group.addButton(r_yes)
ans_group.addButton(r_no)

# 信号操作
def test(val):
    print(val)
    # 如果不使用 signal_name[type] 格式
    # 可以根据按钮来获取ID
    print('被点击的按钮ID是: ', gender_group.id(val))

# 按钮状态改变信号 buttonToggled
# gender_group.buttonToggled.connect(test)

# 按钮点击信号 buttonClicked
gender_group.buttonClicked.connect(test)
# 默认是这样的, 跟上面一样, 打印了按钮对象
# gender_group.buttonClicked[QAbstractButton].connect(test)
# 让信号发射 ID, 而不是 QAbstractButton 对象
# 需要使用 signal_name[type] 格式
# gender_group.buttonClicked[int].connect(test)

# 2.3 展示控件
window.show()

# 3. 应用程序的执行, 进入到消息循环
sys.exit(app.exec_())

```

完成于 201810241936