

QT之信号与槽



啥是信号? 啥是槽?

- **信号** ,就是在特定情况下被发射的事件, 例如 `PushButton` 最常见的信号就是鼠标单击时发射的 `clicked()`信号
- **槽函数** ,就是对信号响应的函数。槽就是一个函数, 与一般的C++函数是一样的, 可以定义在类的任何部分 (`public`、`private` 或 `protected`) , 可以具有任何参数, 也可以被直接调用。
- 槽函数与一般的函数**不同**的是: 槽函数可以与一个信号关联, 当信号被发射时, 关联的槽函数被自动执行。

为啥要用?

- 信号与槽 (Signal & Slot) 是 Qt 编程的基础, 也是 Qt 的一大创新。因为有了信号与槽的编程机制, 在 Qt 中处理界面各个组件的交互操作时变得更加直观和简单

怎么用?

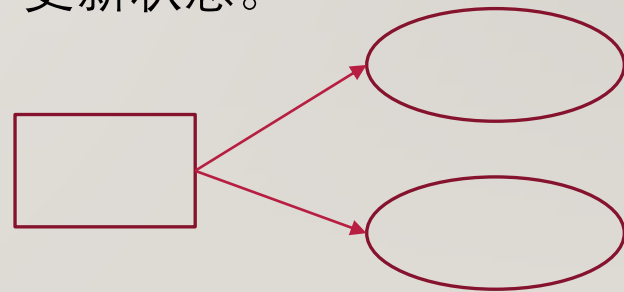
- 使用 `connect` 必须要是 `QObject` 子对象, 必须包含 `Q_OBJECT`
- 信号与槽关联是用 `QObject::connect()` 函数实现的, 其基本格式是:
- `QObject::connect(sender, SIGNAL(signal()), receiver, SLOT(slot()));`



- `sender` 是发射信号的对象名称, `signal()` 是信号名称。信号可以看做是特殊的函数, 需要带括号, 有参数时还需要指明参数。
- `receiver` 是接收信号的对象名称, `slot()` 是槽函数的名称, 需要带括号, 有参数时还需要指明参数。

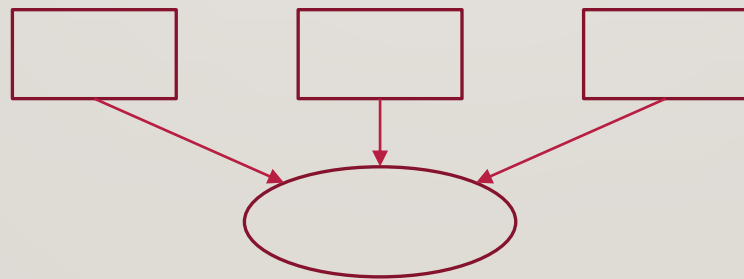
注意什么?

- 一个信号可以连接多个槽，例如：
- `connect(spinNum, SIGNAL(valueChanged(int)), this, SLOT(addFun(int)));`
- `connect(spinNum, SIGNAL(valueChanged(int)), this, SLOT(updateStatus(int)));`
- 当一个对象 `spinNum` 的数值发生变化时，所在窗体有两个槽进行响应，一个 `addFun()` 用于计算，一个 `updateStatus()` 用于更新状态。
- 执行的时候按照建立连接时的顺序依次执行。



注意什么?

- 多个信号可以连接同一个槽，例如让三个选择颜色的 RadioButton 的 clicked() 信号关联到相同的一个自定义槽函数 setTextFontColor()。这样，当任何一个 RadioButton 被单击时，都会执行 setTextFontColor() 函数。
- `connect(ui->rBtnBlue,SIGNAL(clicked()),this,SLOT(setTextFontColor()));`
- `connect(ui->rBtnRed,SIGNAL(clicked()),this,SLOT(setTextFontColor()));`
- `connect(ui->rBtnBlack,SIGNAL(clicked()),this,SLOT(setTextFontColor()));`

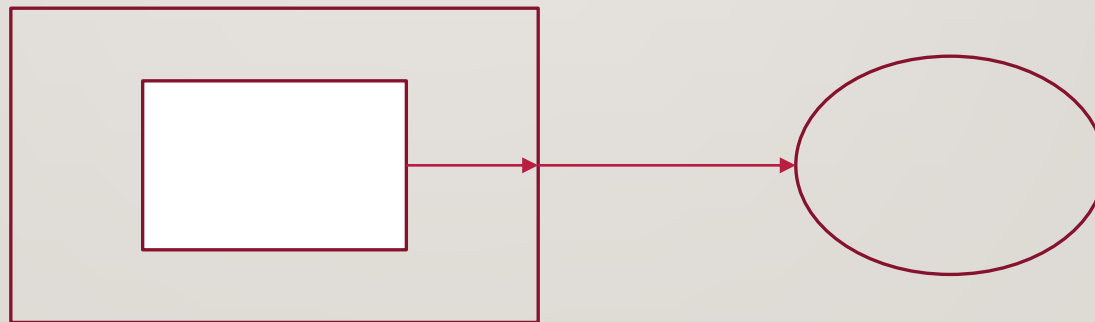


注意什么?

- 一个信号可以连接另外一个信号,

例如:

```
connect(spinNum, SIGNAL(valueChanged(int)), this, SIGNAL (refreshInfo(int)));
```



注意什么?

- 严格的情况下，信号与槽的参数个数和类型需要一致
- 在使用信号与槽的类中，必须在类的定义中加入宏 `Q_OBJECT`。
- (单线程)当一个信号被发射时，与其关联的槽函数通常被立即执行，就像正常调用一个函数一样。只有当信号关联的所有槽函数执行完毕后，才会执行发射信号处后面的代码。

特殊知识

- QT的控件只能是主线程控制, 如果新建的线程控制控件会产生问题, 崩溃
- QT新建的线程 (如网络recv线程), 如果要对界面控件进行控制, 只能通过信号槽完成

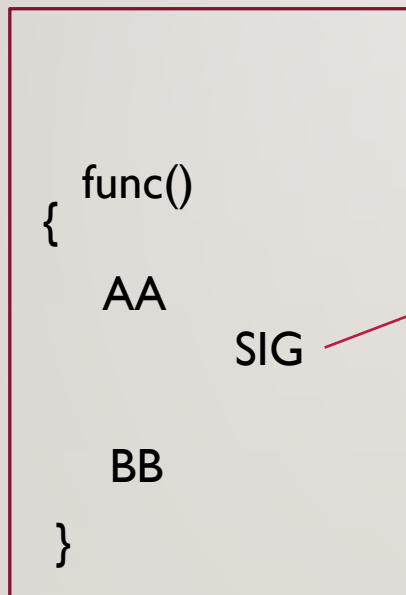
CONNECT函数的第5个参数

- Qt::ConnectionType

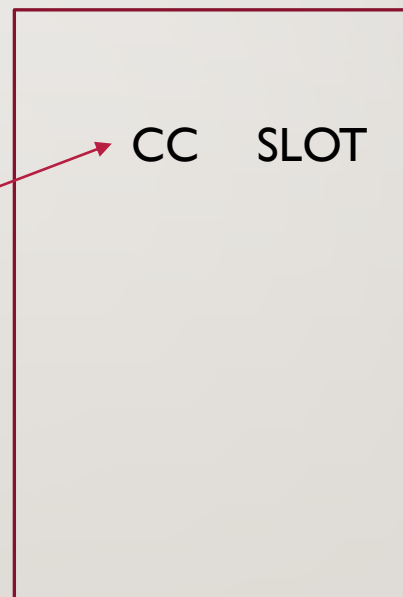
Qt::AutoConnection	0	Default) If the receiver lives in the thread that emits the signal, Qt::DirectConnection is used. Otherwise, Qt::QueuedConnection is used. The connection type is determined when the signal is emitted.
Qt::DirectConnection	1	The slot is invoked immediately when the signal is emitted. The slot is executed in the signalling thread.
Qt::QueuedConnection	2	The slot is invoked when control returns to the event loop of the receiver's thread. The slot is executed in the receiver's thread.
Qt::BlockingQueuedConnection	3	Same as Qt::QueuedConnection, except that the signalling thread blocks until the slot returns. This connection must <i>not</i> be used if the receiver lives in the signalling thread, or else the application will deadlock.
Qt::UniqueConnection	0x80	This is a flag that can be combined with any one of the above connection types, using a bitwise OR. When Qt::UniqueConnection is set, QObject::connect() will fail if the connection already exists (i.e. if the same signal is already connected to the same slot for the same pair of objects).

多线程

线程1



线程2



Qt::DirectConnection
Qt::QueuedConnection
Qt::BlockingQueuedCo
nnection

自定义参数

- 在connect 连接前添加如下声明:
- `#include <QMetaType>`
- `qRegisterMetaType<Myclass>(" Myclass ");`

总结

- 严格的情况下，信号与槽的参数个数和类型需要一致
- 使用信号槽, 必须是QObject 子类
- 在使用信号与槽的类中，必须在类的定义中加入宏 Q_OBJECT
- 一个信号可以连接多个槽
- 多个信号可以连接同一个槽
- 一个信号可以连接另外一个信号