目前的按钮点击没有任何反应

需求：点击按钮之后可以关闭窗口

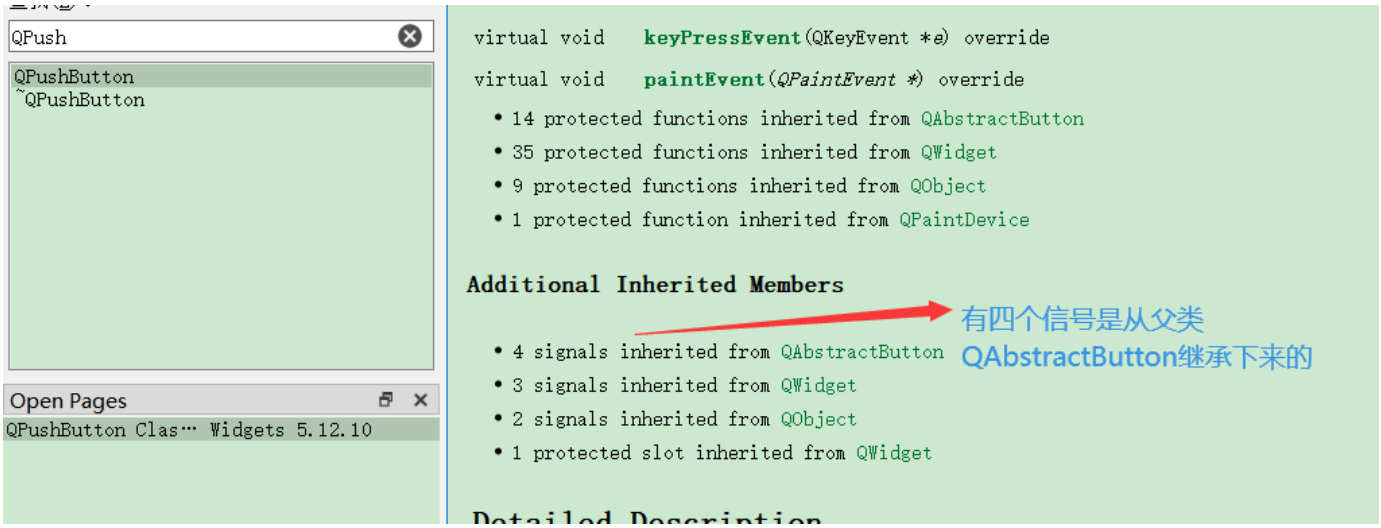
做法：

1. 按钮
2. 点击
3. 窗口
4. 关闭窗口

函数原型：

连接 ： connect(信号发送者，发送的具体信号，信号的接收者，信号的处理(槽函数))；

#信号处理就是使用槽函数



信号槽的优点：松散耦合（发送端 和 接收端 本身是没有关联的，通过connet连接，将两端耦合在一起）

那么接下来我们引入槽的概念

Slots:槽

```
// connet(信号发送者，发送的具体信号，信号的接收者，处理的槽函数)

//Slots:槽

connect(tbn,&MyPushButton::clicked,this,&myweight::close);
connect(tbn,&QpushButton::clicked,this,&Qweight::close);
                                                    //
//    两种方式都是可以的
```

signals(信号)

- clicked(bool checked=false) //点击
- pressed() //按下
- released() //释放
- toggled(bool checked) //切换状态

信号与槽是QObject的机制，因此，因此在创建相关的类时，基类必须是QObject

自定义信号写在 Signals 下（返回值为void，可以有参数，可以重载，仅需要声明，不需要实现

自定义槽函数写在public slots下或者全局函数（返回类型void,需要声明，需要实现，可以有参数，可以重载）

步骤：在父类的signal下声明自定义信号函数（无需实现）

1. 在子类的槽函数--->public slots 中声明自定义槽函数
2. 在子类源文件中实现该槽函数
3. 创建对象
4. 连接对象
5. 触发信号函数

附录：

```
//mainwindows.h
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include"students.h"
#include"teacher.h"
QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT//Q_OBJECT宏，允许使用信号和槽

public:
```

```
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private:
    Ui::MainWindow *ui;
    Students *st;
    Teacher *zt;
    void Eat();
};
#endif // MAINWINDOW_H
```

```
//students.h
#ifndef STUDENTS_H
#define STUDENTS_H

#include <QObject>

class Students : public QObject
{
    Q_OBJECT
public:
    explicit Students(QObject *parent = nullptr);

signals:

public slots:
    void treat();

};

#endif // STUDENTS_H
```

```
//teachers.h
#ifndef TEACHER_H
#define TEACHER_H

#include <QObject>

class Teacher : public QObject
{
    Q_OBJECT
public:
    explicit Teacher(QObject *parent = nullptr);

signals:
    void hungry();

};

#endif // TEACHER_H
```

```cpp
//students.cpp
#include "students.h"
#include"QDebug"
Students::Students(QObject *parent) : QObject(parent)
{

}
void Students::treat()
{
    qDebug()<<"学生请客吃饭";
}
```

```cpp
//teacher.cpp
#include "teacher.h"

Teacher::Teacher(QObject *parent) : QObject(parent)
{

}
```

```cpp
//mainwindows.cpp
#include "mainwindow.h"
#include "ui_mainwindow.h"


MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)//初始化列表
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    this->st=new Students(this);
    this->zt=new Teacher(this);

    //连接connet
    connect(zt,&Teacher::hungry,st,&Students::treat);
    Eat();

}
void MainWindow::Eat()
{
    emit zt->hungry();
}
MainWindow::~MainWindow()
{
    delete ui;
}
```

```cpp
//main.cpp
#include "mainwindow.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    //应用程序对象  a
    QApplication a(argc, argv);
    //窗口对象 w
    MainWindow w;
    w.show();//显示窗口
    return a.exec();//让a 进入消息循环机制，，，阻塞，后续代码不执行
}
```