

信号重载

```
signals:
    //自定义信号 写到signals下
    //返回值是void，只需要声明，不需要实现
    //可以有参数，可以重载
    void hungry();

    void hungry(QString foodName);|
```

槽函数重载

```
public slots:
    //早期Qt版本 必须要写到public slots，高级版本可以写到 public或者全局下
    //返回值 void，需要声明，也需要实现
    //可以有参数，可以发生重载
    void treat();

    void treat(QString foodName);

};
```

槽函数实现

```
void Student::treat(QString foodName)
{
    qDebug() << "请老师吃饭，老师要吃：" << foodName ;
}
```

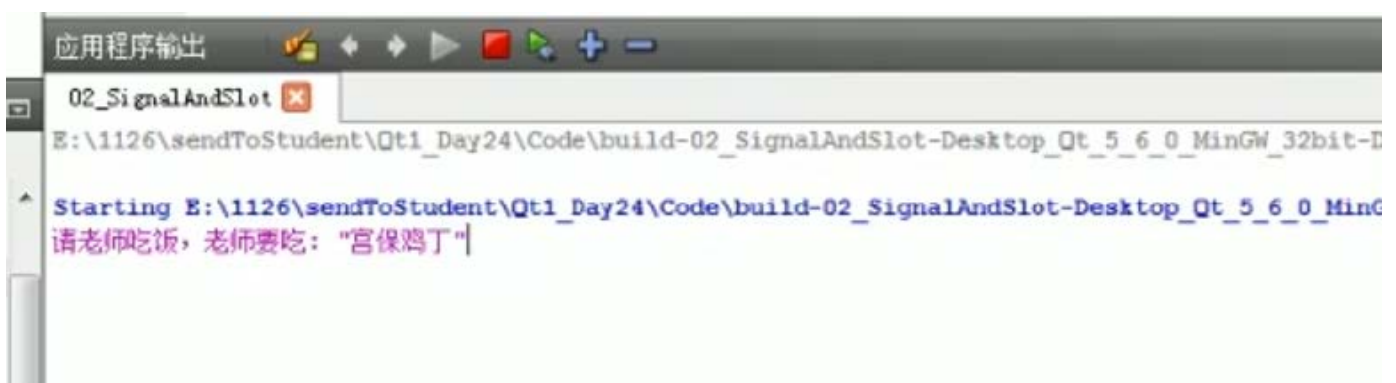
连接两个对象

```

28
29 //连接带参数的 信号和槽
30 //指针 -> 地址
31 // 函数指针 -> 函数地址
32
33 void(Teacher:: *teacherSignal)(QString) = &Teacher::hungry;
34 void(Student:: *studentSlot)(QString) = &Student::treat;
35 connect(zt,teacherSignal,st,studentSlot);
36 classIsOver();
37
38 }
39
40 void Widget::classIsOver()
41 {
42     //下课函数，调用后 触发老师饿了的信号
43     //emit zt->hungry();
44     emit zt->hungry("宫保鸡丁");
45 }
46

```

运行结果



使用方法：

1. 使用函数指针
2. 定义Teacher作用域下的函数指针 teacherSignal
3. 定义Students作用域下的函数指针 studentSlot
4. 然后分别把对应作用域下的地址赋值给这两个指针
5. 在connect中使用这两个指针代替 **具体信号** 和 **槽函数**
6. 完成

可以看到，以上方法输出的是 "宫保鸡丁"，多了引号

是因为我们使用了QStringz字节数组

我们需要将QString----->QByteArray----->char*

操作方法：

```
qDebug()<<"请老师吃饭，老师要吃："<<foodName.toutf8().data();
```

解释：

```
.toutf8() //转化为字节数组
```

```
.data() //转化为char
```