**PHP的MySQL扩展**

**回顾：**

**完整文件上传步骤**







**思考：**

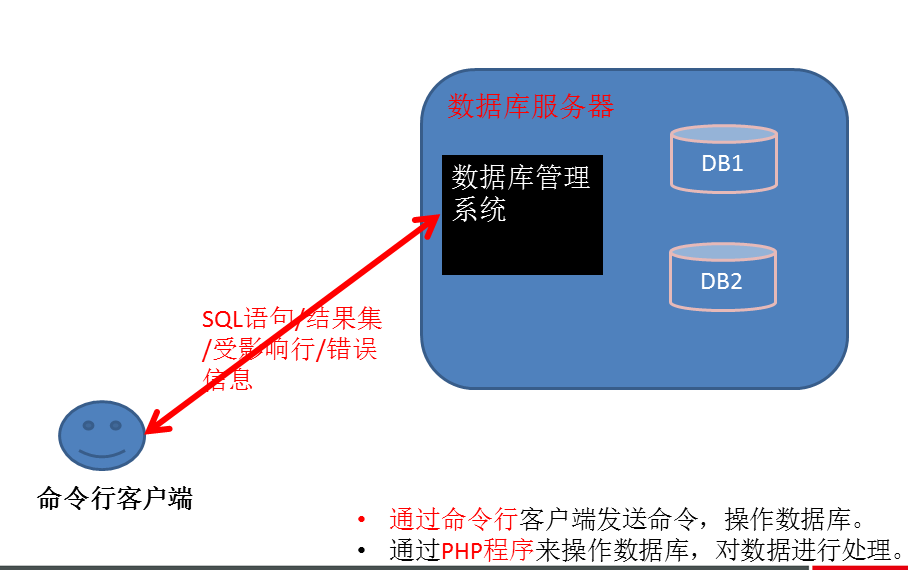
1. **PHP操作MySQL的步骤是什么？每个步骤都用到哪些系统函数？**
2. **增删改语句的执行有什么的共同点？**
3. **查询语句返回的情况有哪几种？如何解析结果集？**
4. **解析结果集的函数有什么差别？如何在表格中展示结果数据？**
5. **学生管理系统核心模块有哪些？修改学生数据如何实现？**

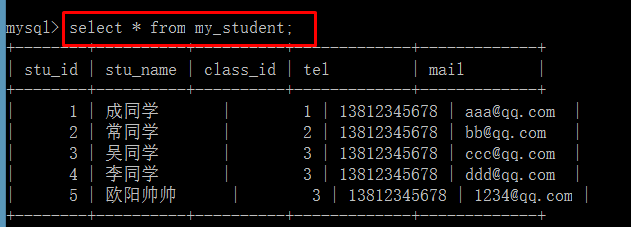
# （回顾）操作MySQL数据库

## 访问数据库方式

### 命令行客户端

通过命令行客户端发送命令及接收返回的结果。仅用于个人测试，用户无法使用。

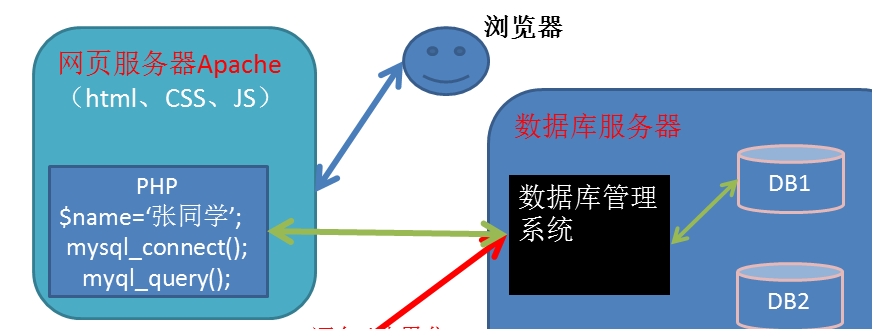


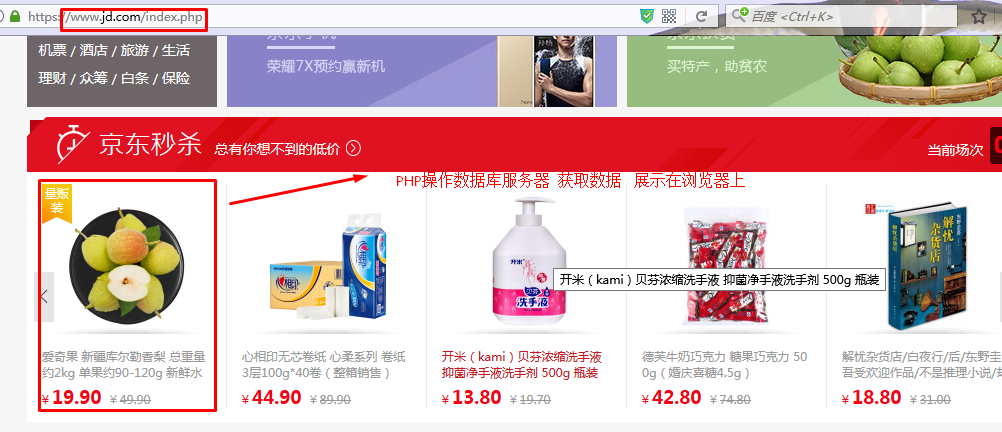


### PHP操作操作数据库

PHP通过专门的函数，建立起数据库连接。然后可以发送命令，对数据库的数据进行操作。PHP可以将执行的结果（包括查询结果集）展示在浏览器上。

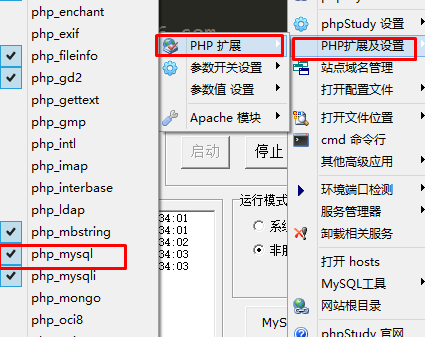
此时，可以将PHP看做是MySQL数据库服务器的一个客户端。





## PHP的MySQL数据库扩展

### 开启MySQL数据库扩展



检查扩展库中的函数是否存在即可检查扩展库是否加载成功。

function\_exists(函数名);

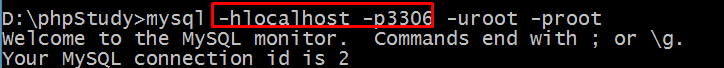
//存在返回true 否则返回false



# PHP操作数据库步骤

**命令行操作数据库的步骤：**

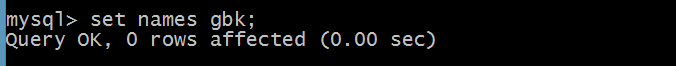
1. 连接数据库:参数：主机名，端口号，用户名，密码



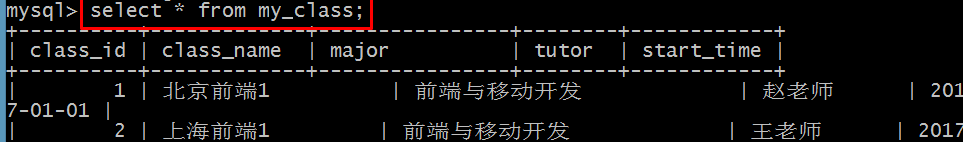
1. 设置字符集：



1. 选择数据库：



1. 发送SQL



1. 检查结果

成功，失败（查询，结果集）



## PHP操作MySQL步骤介绍

1. 建立数据库连接：提供连接参数：主机名，端口号，用户名，密码
2. 设置字符集：

set names utf8;

//php编码为UTF8,数据库服务器的默认编码为UTF8

避免在命令行显示时乱码的发生。

1. 选择数据库：

执行 use db\_name;

指定函数发送执行

1. 发送执行SQL语句

$sql =””;

mysql\_query($sql);

1. 结果（集）、错误信息处理

判断执行结果，给出提示。

1. 断开连接:释放内存。

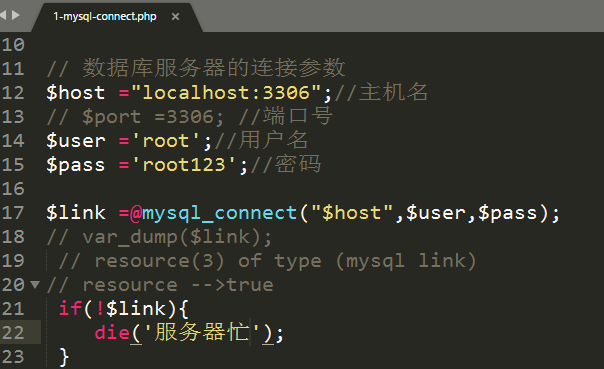
## 连接数据库服务器

### 语法

mysql\_connect(“主机名[：端口号]”,”用户名”,”密码”)；

成功时，返回资源型的数据库连接。失败返回false。

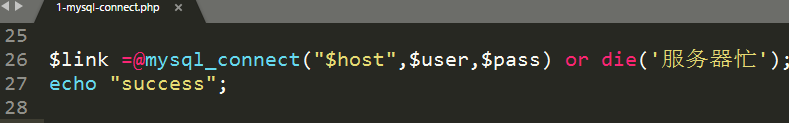
资源型：是一种特殊变量，保存了到外部资源的一个引用(PHP操作的数据)。资源是通过专门的函数来建立和使用的。默认端口为3306，可以省略。



### 判断连接，并给出提示

与运算在第一个表达式为真时，PHP选择不执行第二个表达式，成为与运算的短路运算。可以替换if分支实现连接提示。

不同写法：



## 设置字符集

数据库服务器的默认字符集为UTF8，PHP的编码字符集为UTF8，所以需要执行：

set names utf8;

### 设置字符集的函数mysql\_query()

mysql\_query($sql);

//发送执行SQL语句





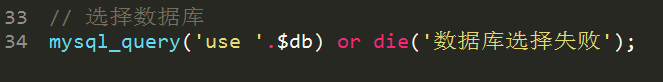
## 选择数据库

对数据进行操作，需要指定数据所在的数据库。

use db\_name;

### 函数一

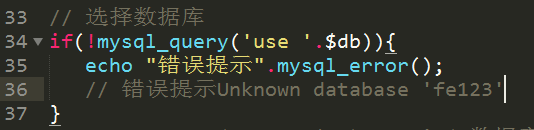
mysql\_query(“use $db”);



### 获取错误信息的相关函数

mysql\_error()

返回上一个SQL语句执行的错误信息。



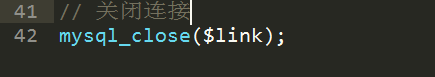
**提示：**

SQL语句执行失败 ，不提示错误信息，只能通过mysql\_error函数获取。建议在调试过程中，多使用mysql\_error（）快速定位错误原因。

## 关闭连接

mysql\_close(resource [$link]);

关闭数据库连接，及时释放内存。没有关闭的连接，会在脚本执行结束时自动销毁。

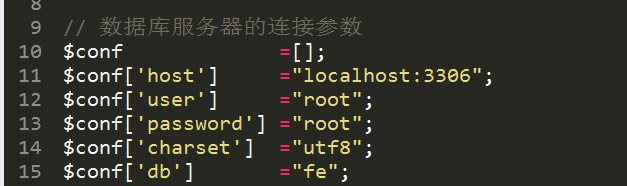


## 应用：封装连接函数db\_connect（）

每个页面几乎都会使用到**数据库连接、字符集设置、数据库选择**。将三步操作封装在函数中，可以在任意页面调用，提高代码复用率。

### 定义my\_connect函数，实现连接数据库、设置字符集及选择数据库

设置配置参数：



**函数封装：**



# 增删改操作

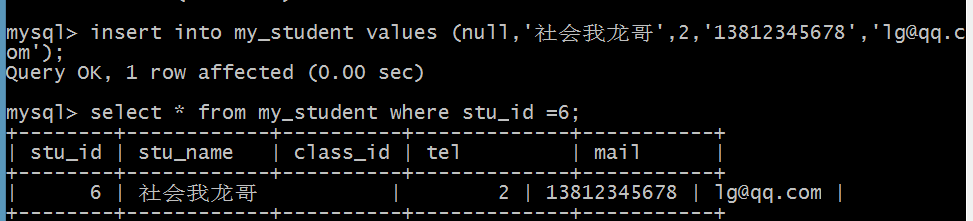
基本的数据操作：CRUD ，对应**insert/update/delete/select**。可以分为两种情况：

* + 增删改语句执行的结果，要么成功（返回true），要么失败（返回false）。
  + 查询语句执行成功时，返回查询结果集（对应资源型），失败时返回false。

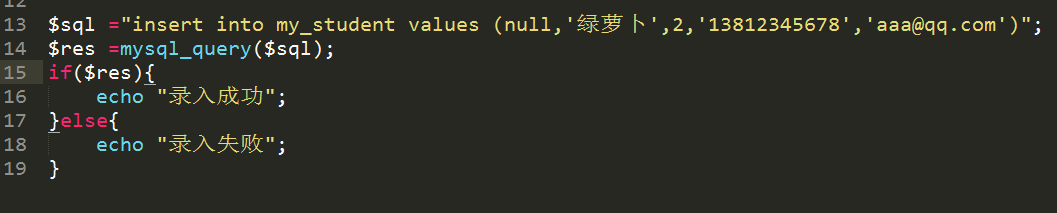
## 插入操作

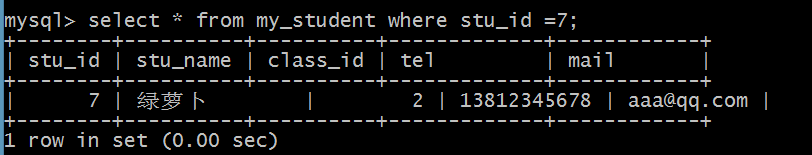
### 增添数据基本语法

insert into table\_name (字段1，字段2) values (值1，值2);



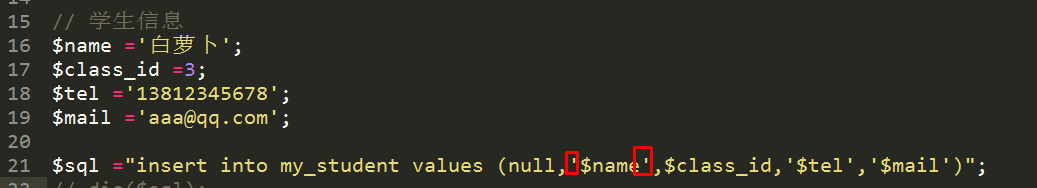
执行结果，成功返回true，失败返回false。





### 构建包含变量的SQL语句

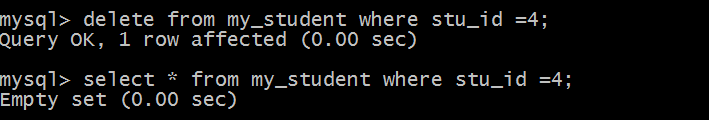
当值对应变量的类型是字符串时，必须在外面添加引号。新构建的SQL语句，为确保语句的正确性，建议在命名行执行检查一下。



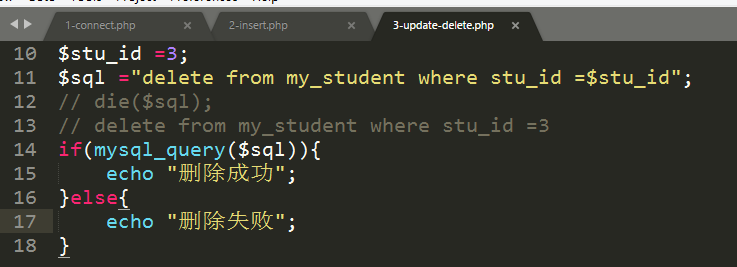
## 删除操作

### 基础语法

delete from tb\_name where条件；



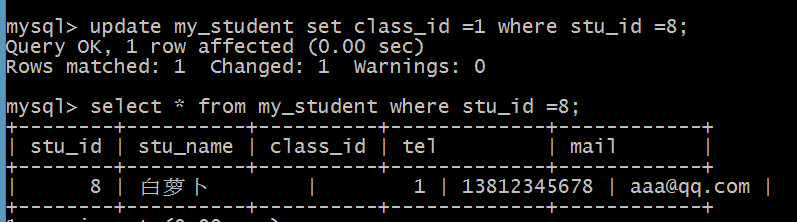
### 执行删除语句



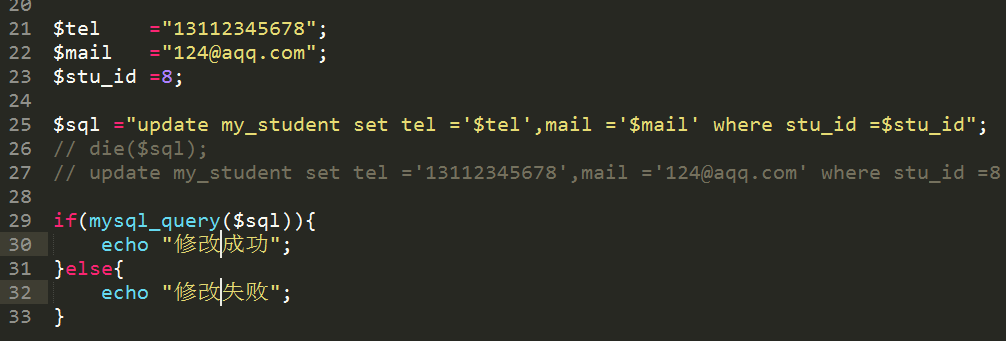
## 更新操作

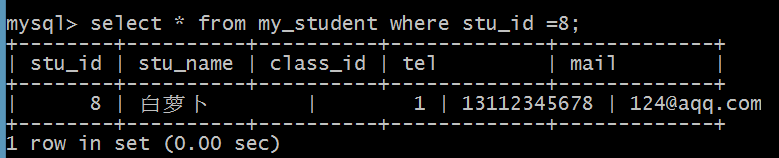
### 基础语法

update tb\_name set 字段1 =值1，字段2 =值2 where 条件；



### 执行更新语句





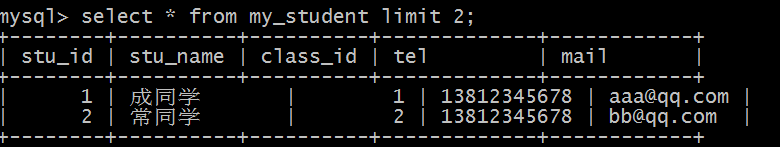
## 总结：增删改注意事项

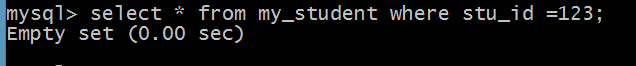
* 使用PHP发送SQL语句前，可以先打印SQL语句，检查语句的正确性。
* 使用变量拼接SQL语句时，字段为字符串类型，需要在变量的两侧使用单、双引号包裹。可以将所有的字段外面都使用双引号包含。

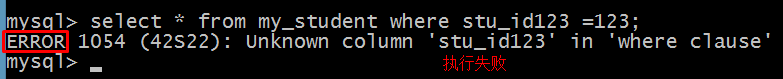
# 查询操作

select 字段列表 from tb where 条件；

查询语句执行成功,返回资源型resource（结果集）。失败为false。







## 查询结果集

$sql =”select 字段列表 from tb where 条件”；

mysql\_query($sql);

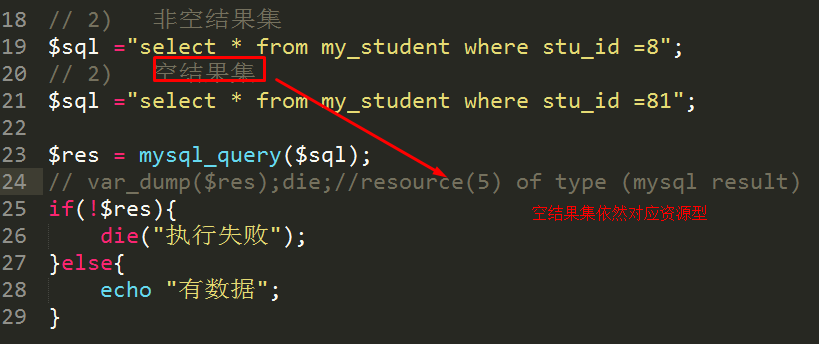
### 失败的情况



### 非空结果集

### 空结果集情况

查询不到内容，返回空结果集，但是返回的类型仍然是资源型。

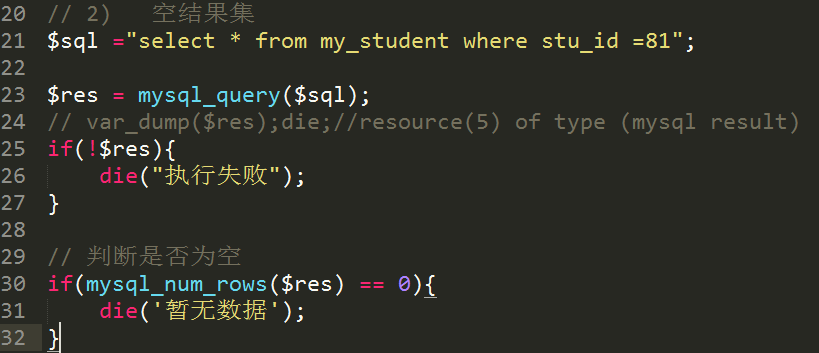


### 判断结果集相关函数

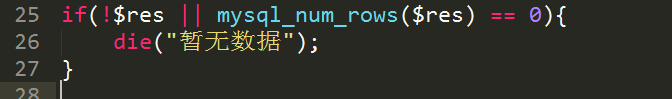
展示数据之前，需要筛选空结果集及非空结果集。

mysql\_num\_rows(resource $res);//查询结果集资源

返回查询结果集中的行数，如果为0行，则为空结果集。



升级：

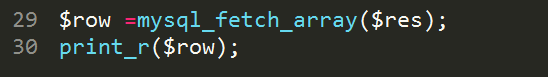


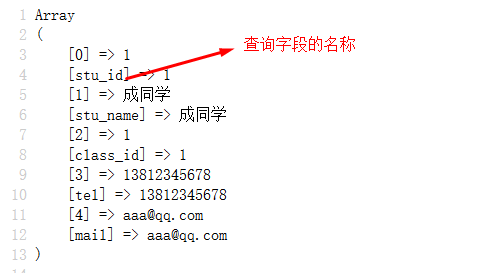
## 解析结果集的三种方式

查询结果集数据类型为资源型。

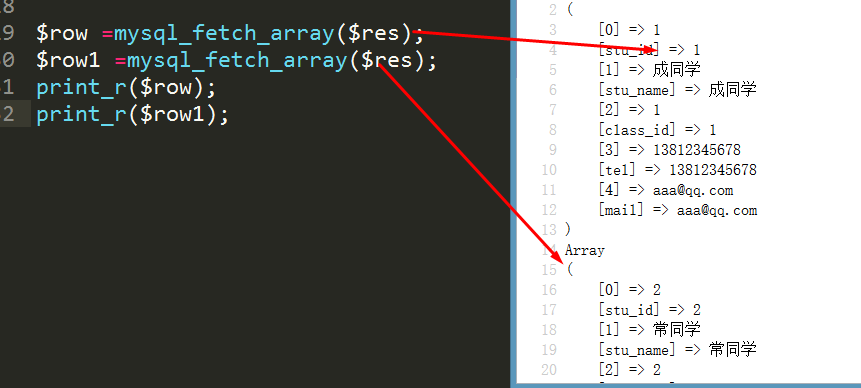
### mysql\_fetch\_array(resource $res);

* 返回结果集中的一行，以混合数组形式返回。既有数字下标形式，又有字符下标形式（查询各个字段名称）。
* 函数的参数必须为查询结果集
* 每次执行，返回一行数据。指针向下移动一次，指向下一行数据。



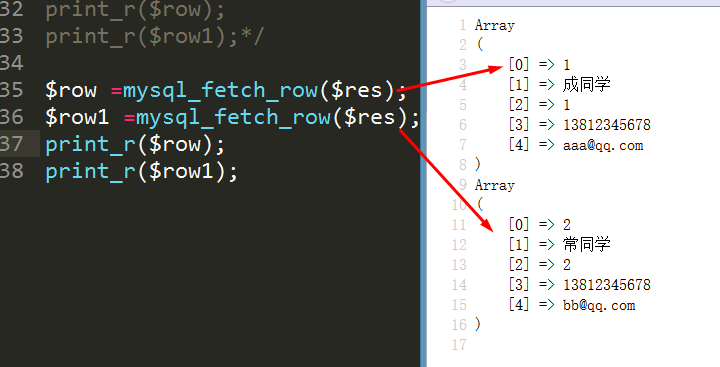


**获取多行：**



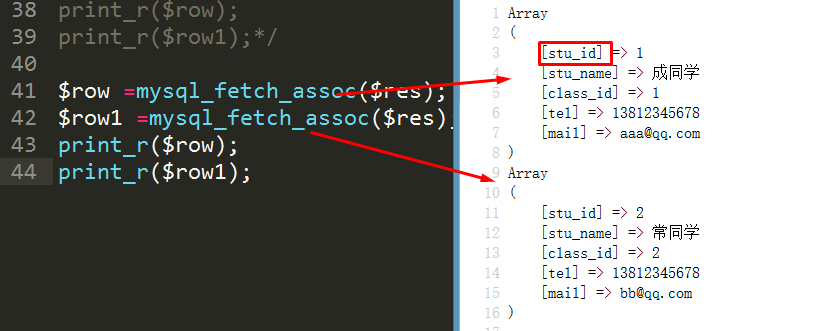
### mysql\_fetch\_row($res);

* 执行一次返回一行，以索引数组形式返回。



### mysql\_fetch\_assoc($res);

返回关联数组，元素下标为查询字段的名称。



### 三个函数的区别与相同点

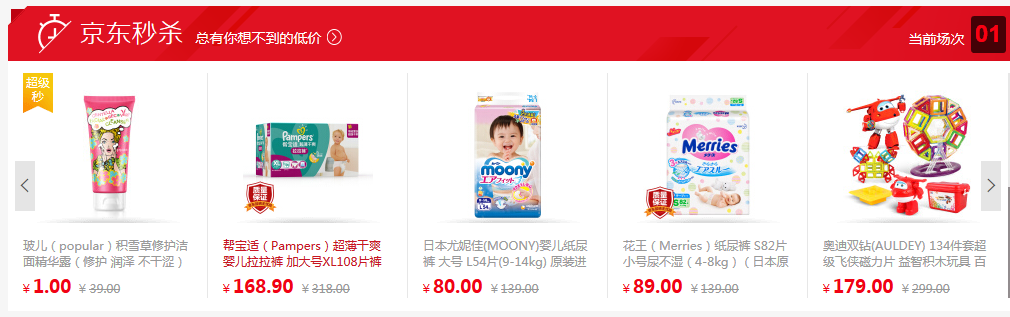
1. **相同点**：

每次执行，取出结果集中的一行（以数组的形式），并且将指针向下移动一次。指针非法时，返回false。

1. **不同点**：

* mysql\_fetch\_row()：返回索引数组
* mysql\_fetch\_assoc()：返回关联数组。
* mysql\_fetch\_array()：返回混合数组。

# 应用--列表形式显示结果集

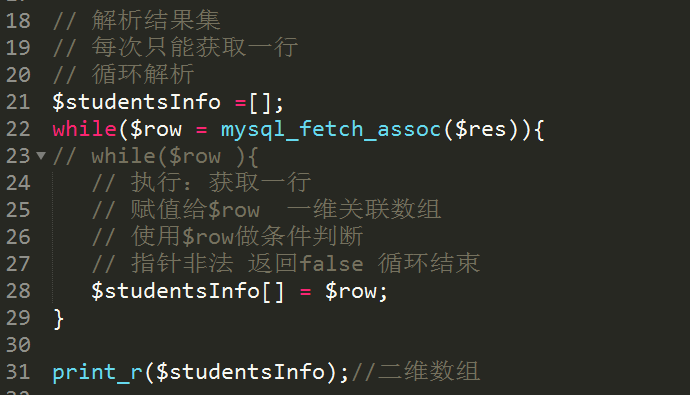


## 遍历结果集原理

每次执行，自动获取一行，指针自动移动。指针非法，返回false。使用while，获取结果集的所有行，并在指针非法时停止循环。



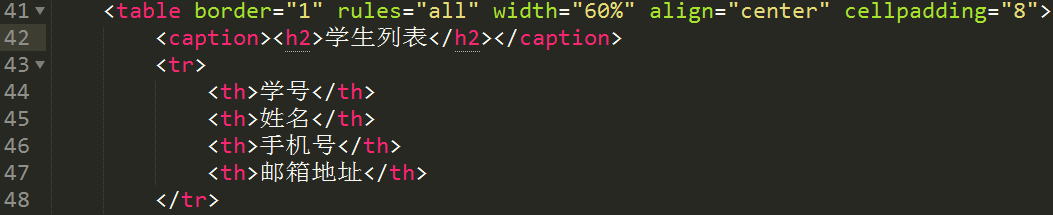
**循环遍历结果集：**



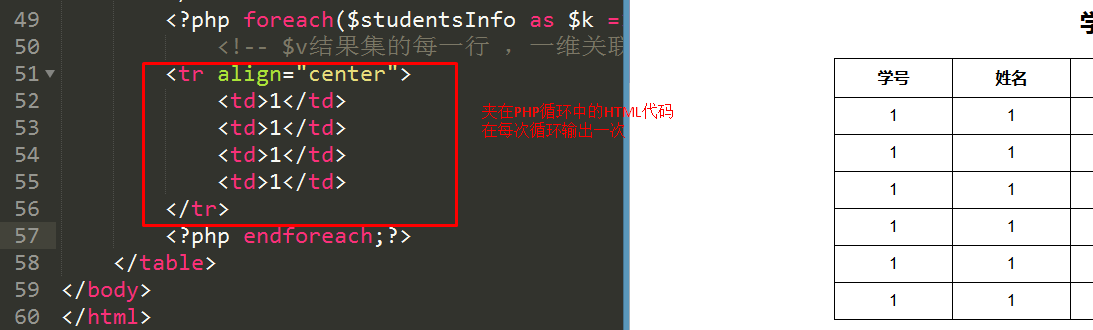


## 结果集显示在表格中

### 绘制表格



### 使用固定的数据初步填充



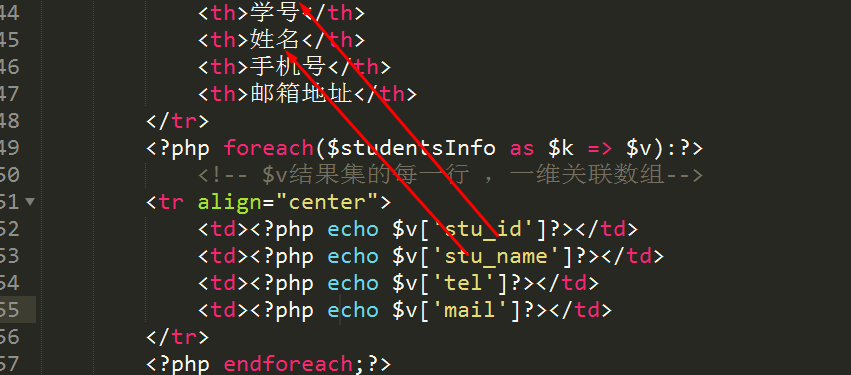
语法：

<?php foreach(…) :?>

HTML代码

<?php endforeach;?>

### 使用学生信息填充

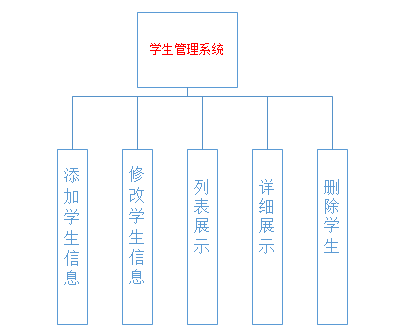




# 学生管理系统

## 功能模块概述

主要实现学生管理系统的增删改查功能。后台是指管理员使用身份信息账号信息验证登录后，才能进行操作。对应的前台是供用户使用的。



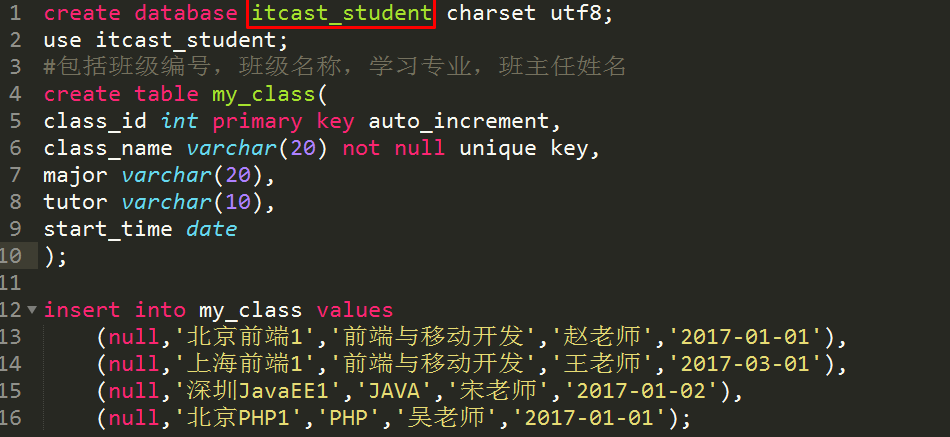
|  |  |  |
| --- | --- | --- |
| **模块** | **脚本名称** | **功能描述** |
| 工具函数 | connect.php | 数据库连接、字符集设置、数据库选择 |
| functions.php | 打印数组、跳转函数 |
| 添加 | add.php | 添加学生信息的表单页面 |
| insert.php | 表单数据提交到的页面，完成添加学生数据处理 |
| 查看 | index.php | 后台首页，列表展示数据（简要的展示） |
| detail.php | 查看学生详细信息（**按照学号**） |
| 更新 | edit.php | 展示原先的数据（表单）（**按照学号**） |
| update.php | 完成数据更新（**按照学号**） |
| 删除 | delete.php | 删除操作（**按照学号**） |

## 数据库、数据表设计

创建项目数据库。为了减少数据的冗余，将学生的基本信息存储在学生表中，将班级信息存储在班级表中。

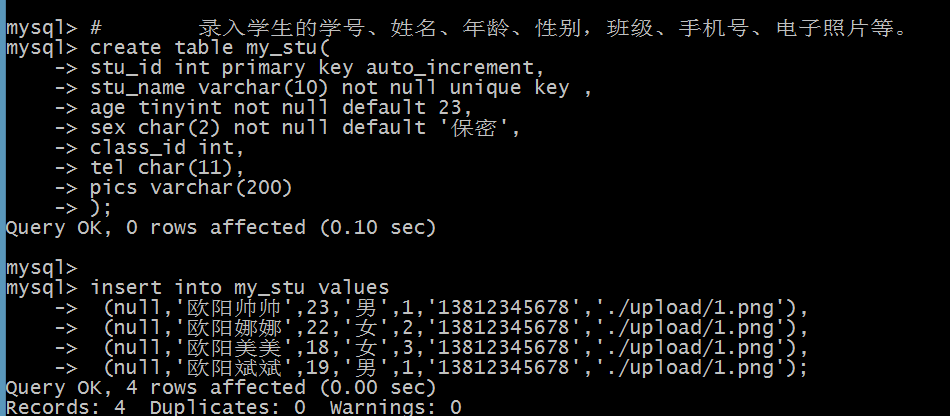
### 班级表设计

包括班级编号，班级名称，学习专业，班主任姓名



### 学生表设计

录入学生的学号、姓名、年龄、性别，班级、手机号、电子照片等。



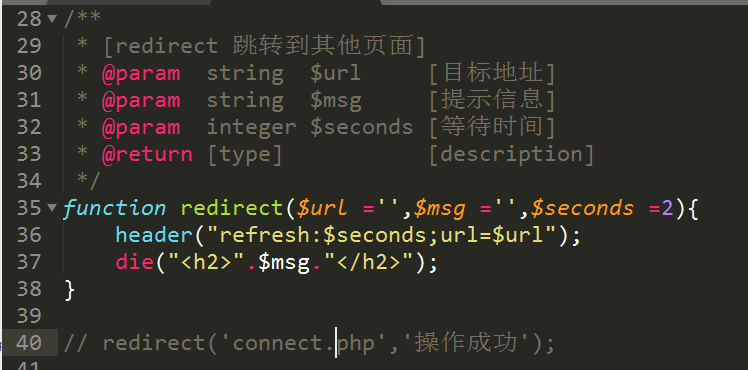
## 工具函数封装

### 数据库连接函数



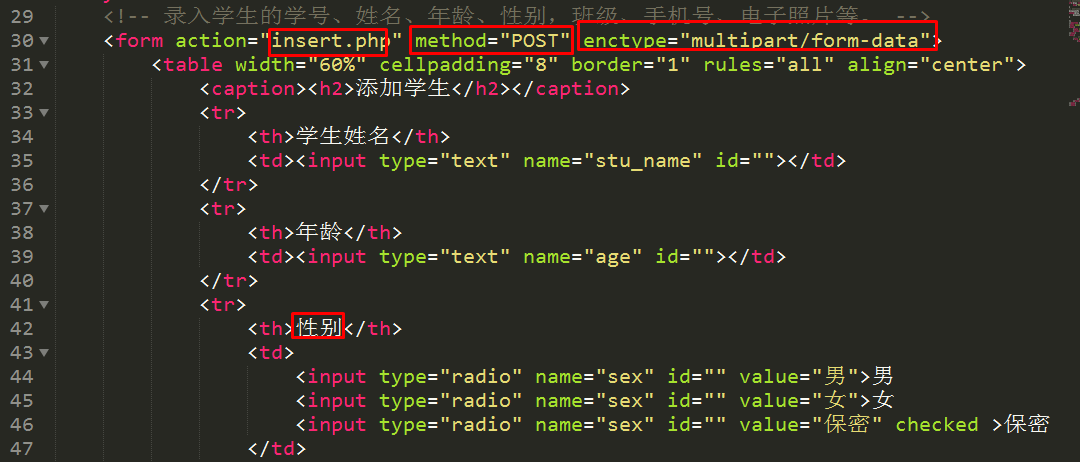
### 跳转函数封装

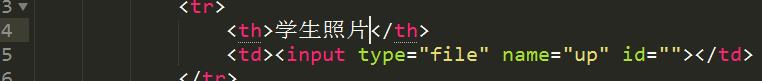




## 添加学生信息

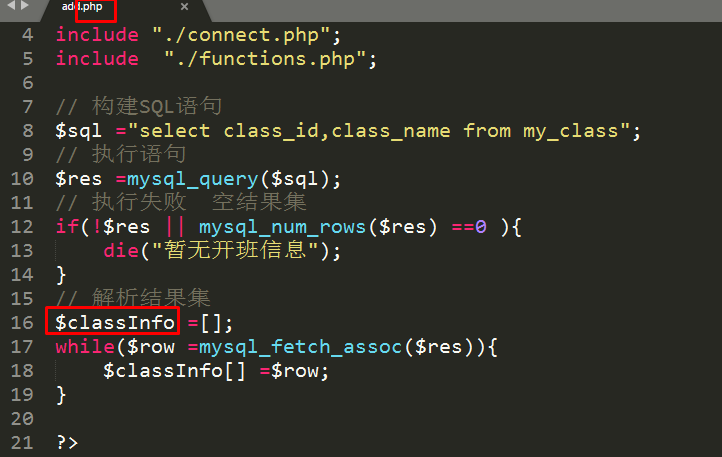
### 表单设计



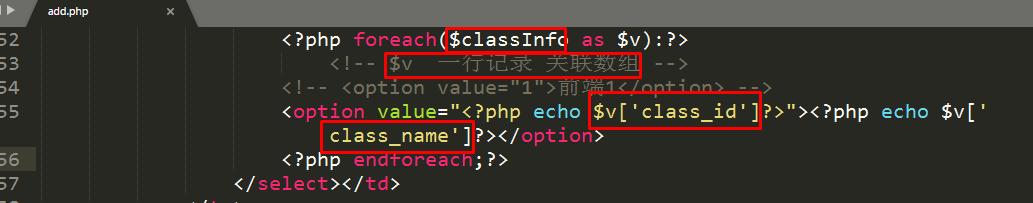


### 查询并展示班级信息

先查询：



遍历二维数组：



### 接收、判断提交的数据

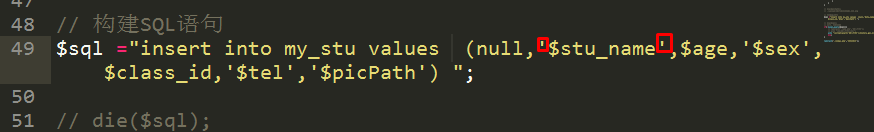


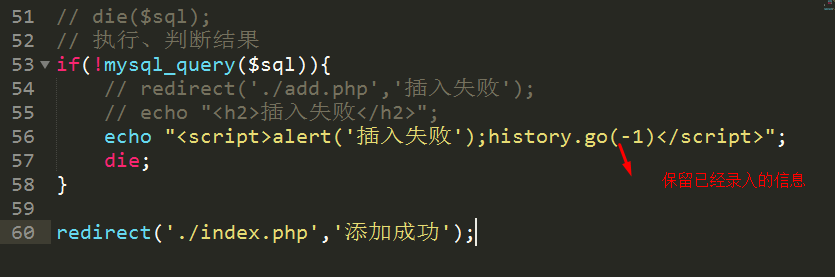
### 处理图像上传

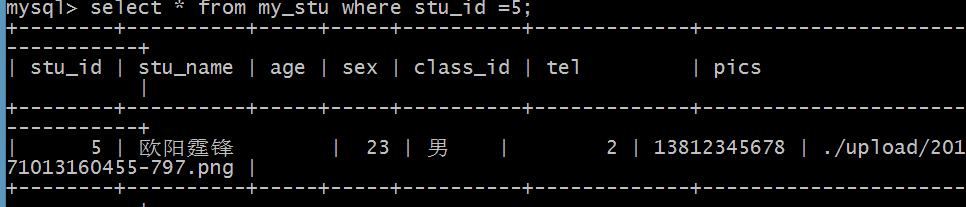
可以调用封装好的文件上传函数完成文件上传。



### 构建执行SQL语句





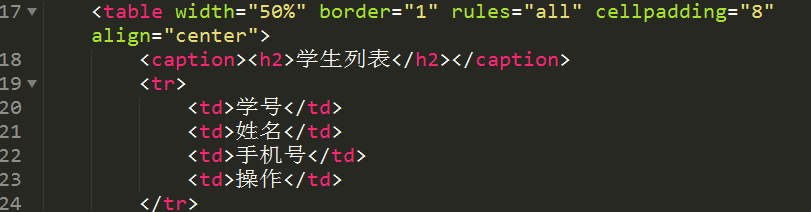


## 列表（index.php）展示学生信息

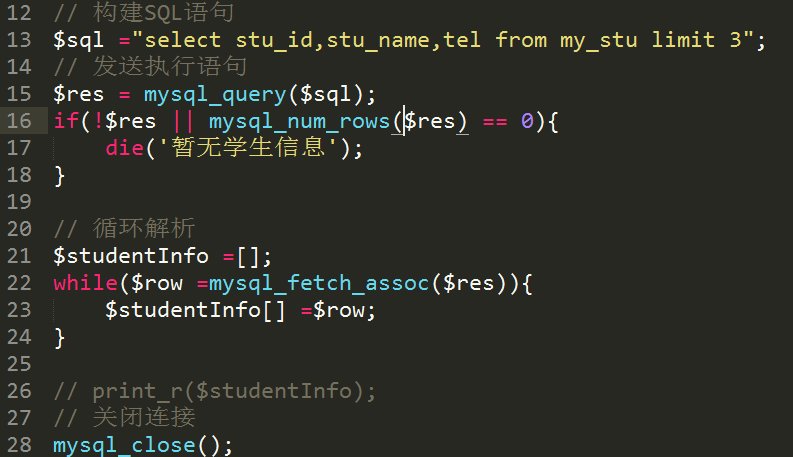
列表展示学生的概要信息，详细信息需要在详情页面进行查看。



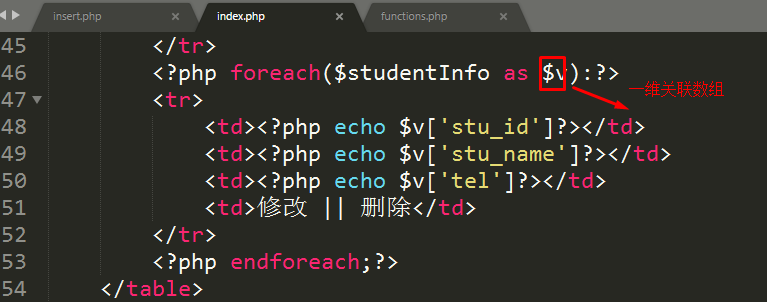
### 绘制表格



### 查询数据



### 将数据循环打印在表格中



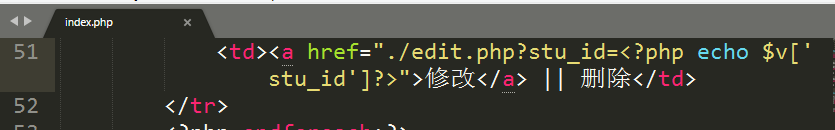


## 修改学生信息

修改学生信息，执行的是UPDATE语句，需要添加where条件（学号信息）

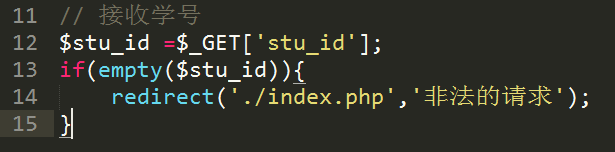
### 添加入口链接

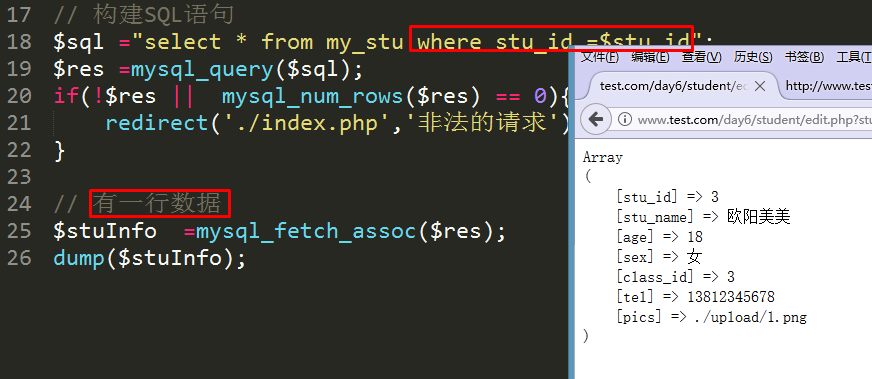
在首页页面，点击传递**学号参数**，其他页面**按照学号，**展示学生的原来的信息。



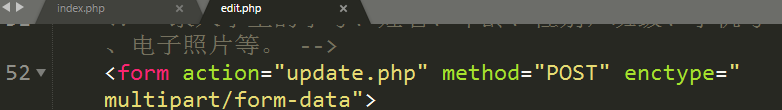


### 接收学号，查询学生对应信息

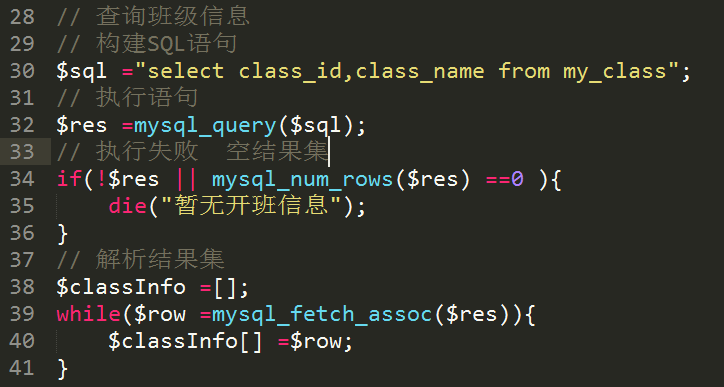




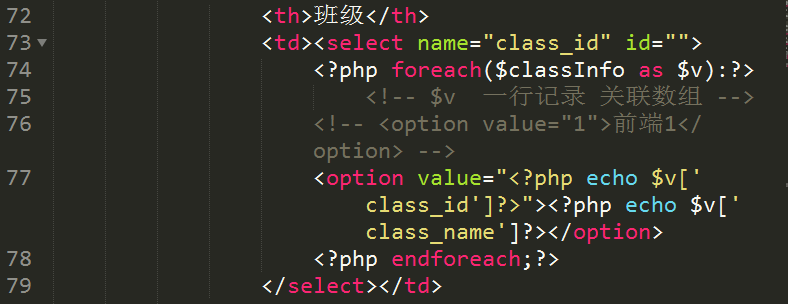
### 绘制表单



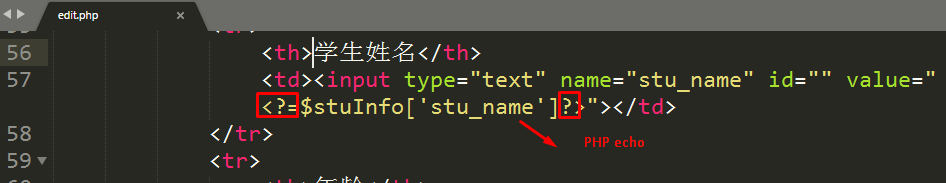
**查询班级信息：**

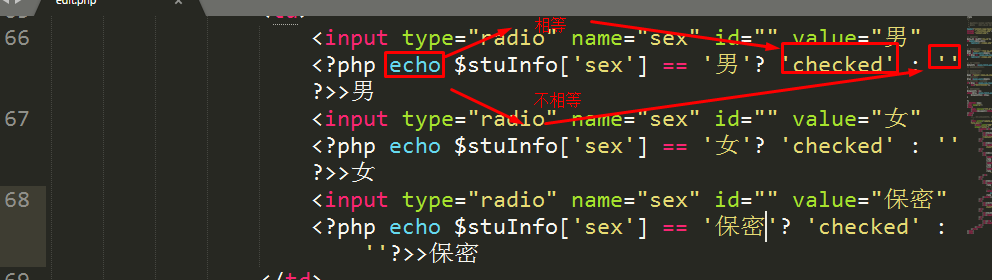


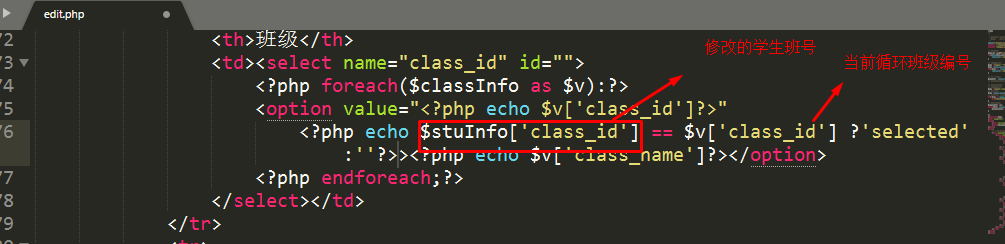
**填充在下拉菜单中：**



### 学生原始信息的填充

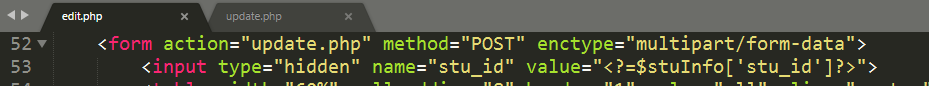






传递学号：

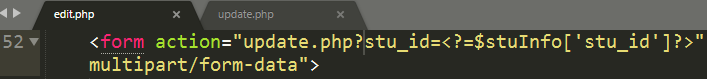
1. 隐藏域

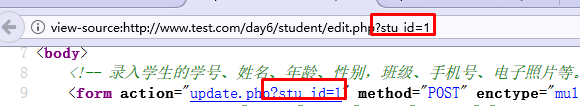


以POST方式提交

1. 地址栏方式

**GET方式提交**

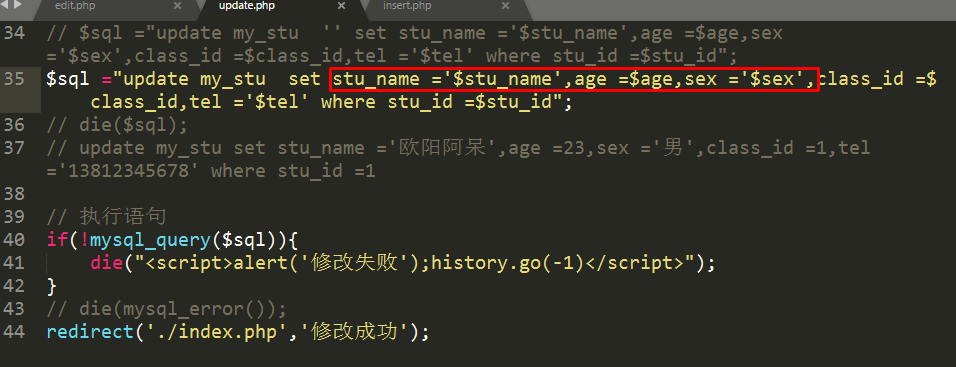




### 接收、筛选处理数据



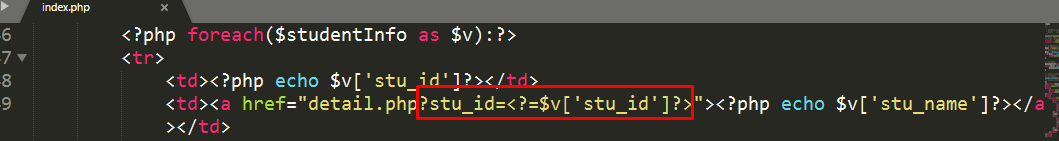
### 执行更新操作



## 展示学生详细信息

在列表点击链接，以GET方式传递学生学号信息，按照学号查询展示学生的完整信息。

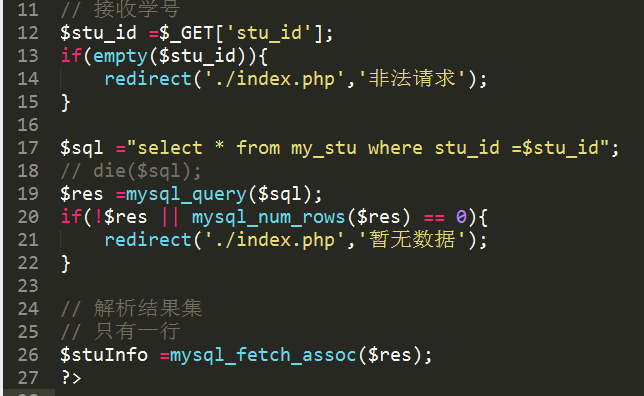
### 传递学号到详情页



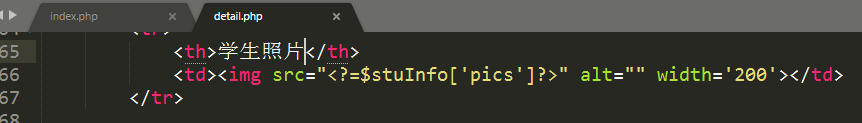


### 判断、接收学号参数

### 按照学号查询学生信息

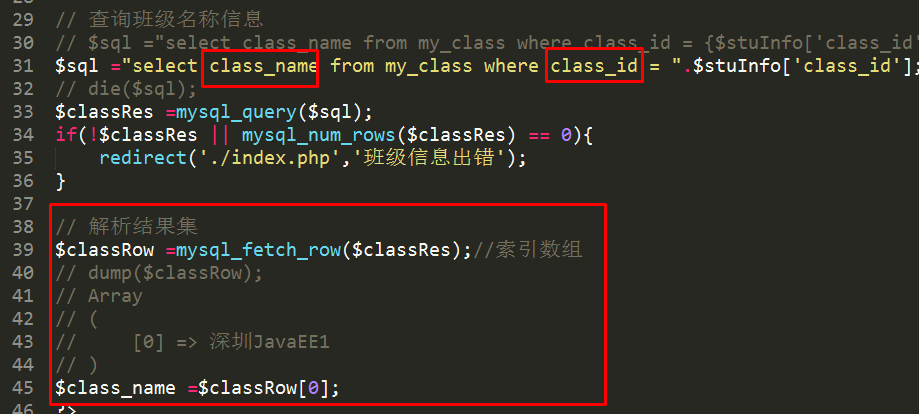


### 展示数据





### 处理班级信息



## 删除学生信息

按照学号删除。