# Mysql扩展

PHP针对MySQL数据库操作提供的扩展：允许PHP当做MySQL的一个客户端连接服务器进行操作。

## 连库基本操作

### 连接数据库服务器

1）资源 mysql\_connect(服务器地址，用户名，密码)



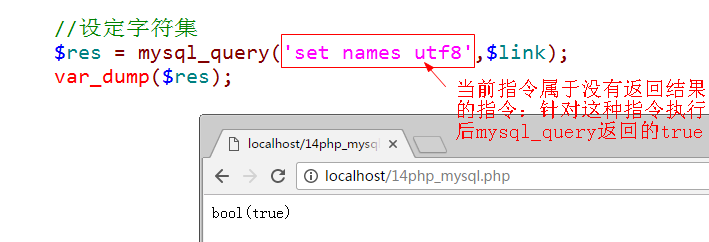
连接资源默认也是超全局的，任何地方都可以使用该资源进行数据库的其他操作

补充：mysql\_connect默认是对一个服务器只连接一次（如果再次运行，返回的是同一个资源），但是如果有需要连接同一个资源两次，那么可以使用第四个参数：TRUE

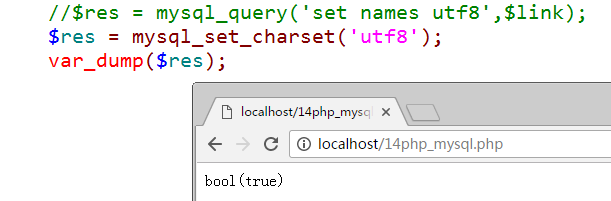
### 设置连接编码

保持客户端与服务器之间的沟通顺畅：同一“语言”（字符集）

1）形式1：mysql\_query("set names XXX")



2）形式2：mysql\_set\_charset("XXX")



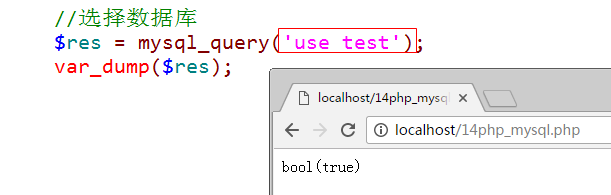
3）怎么确定要使用何种编码

客户端当前执行脚本的界面是什么字符集，就设定成什么字符集

### 选定要使用的数据库

1）形式1：mysql\_query("use XXX")

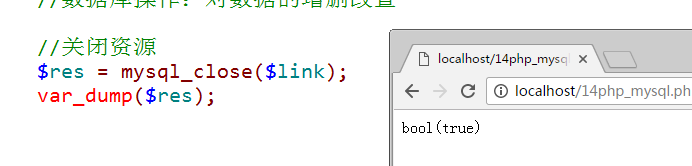
2）形式2：mysql\_select\_db("XXX")



### 关闭连接

主动释放连接：MySQL服务器的连接资源是有限的，不用了需要释放（脚本执行结束系统也会自动释放）

1）mysql\_close()



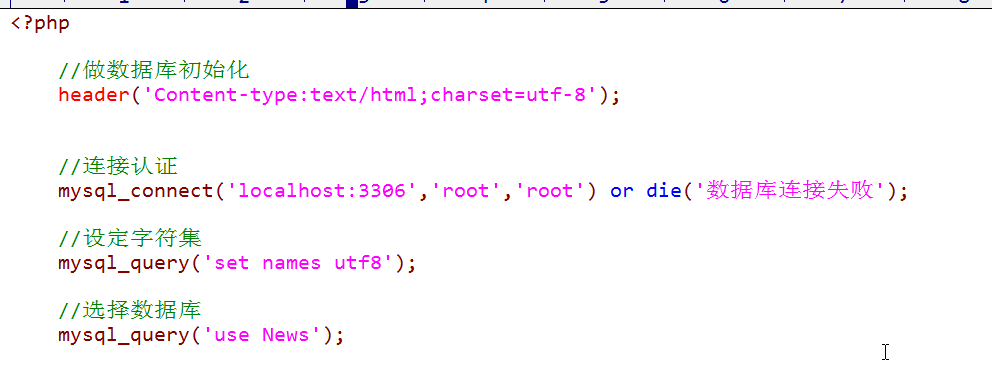
## 执行增删改操作

### 执行增删改操作

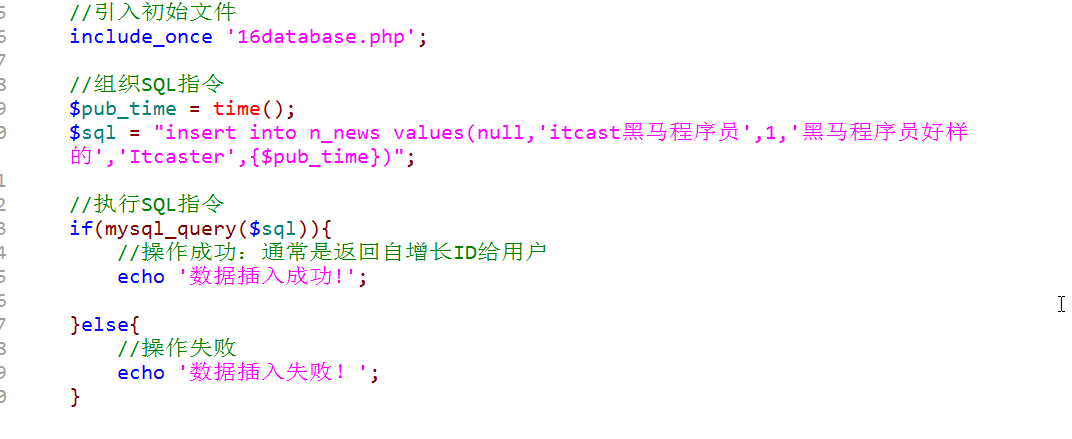
Mysql\_query函数执行结果返回的是true或者false，true代表执行成功，false代表执行失败：失败原因分两种：一种是SQL指令本身语法错误；第二种是执行失败；



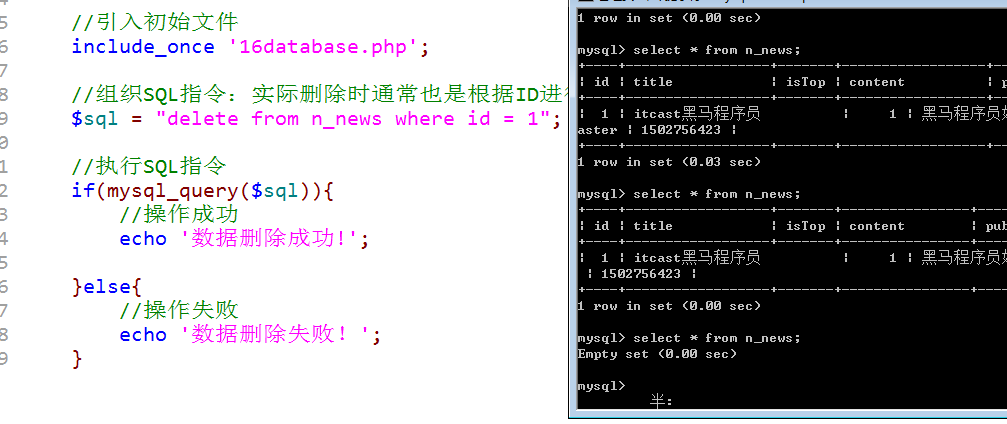
做个专门负责数据库初始化的脚本：之后要操作数据库包含该脚本即可



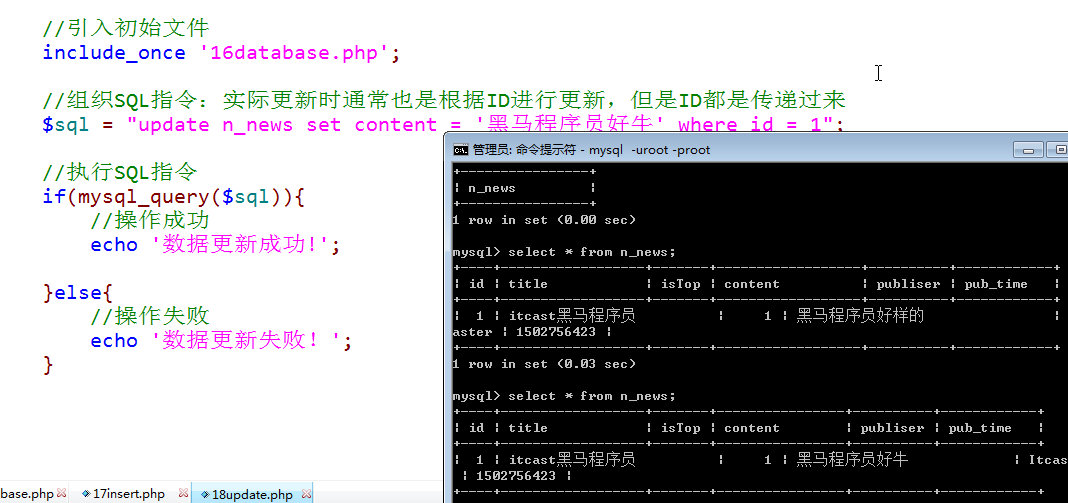
1）mysql\_query("insert ....")：通过insert指令向指定表插入对应的数据



2）mysql\_query("delete ....")：根据删除指令操作数据库（表）



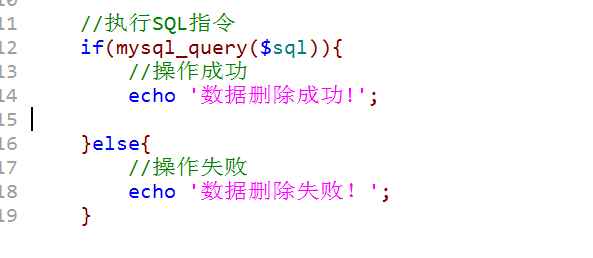
3）mysql\_query("update ....")：根据更新指令操作数据库



### 判断执行结果

1）执行成功，结果为true，只代表SQL指令执行成功

2）执行失败，结果为false，代表SQL指令错误或者操作对象不存在



## 执行查询操作

### 执行查询语句

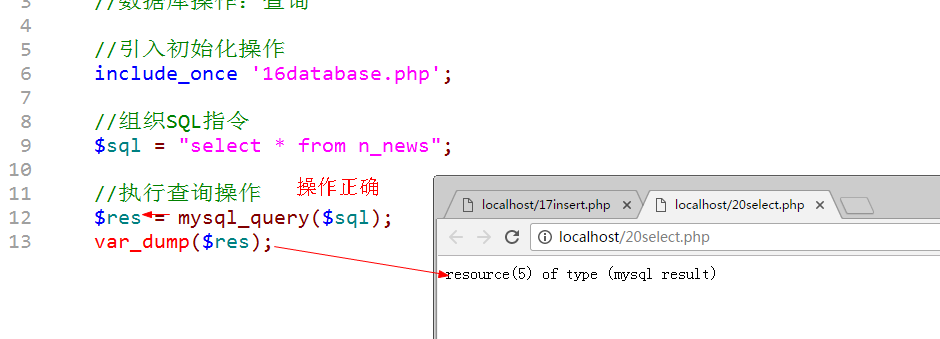
1）mysql\_query("select ...")

2）其他类似查询语句，比如：show语句，desc语句

以上两种情况的总结：凡是执行操作希望拿到数据库返回的数据进行展示的（结果返回：数据结果）

3）执行结果的处理：成功为结果集，失败为false

成功返回结果：SQL指令没有错误，但是如果查询结果本身为空返回也是true（结果集是一种资源：转换成bool永远为TRUE）

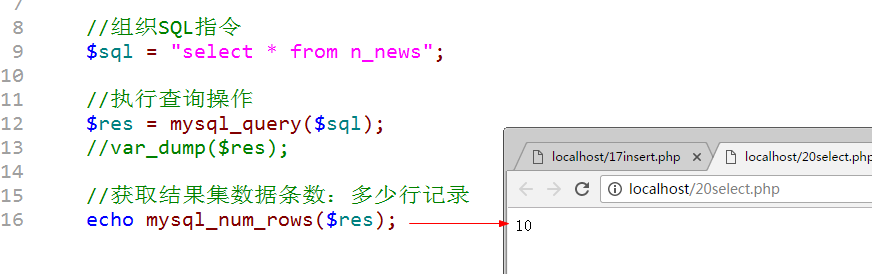


失败为false：SQL指令有错误



### 获取结果集行数

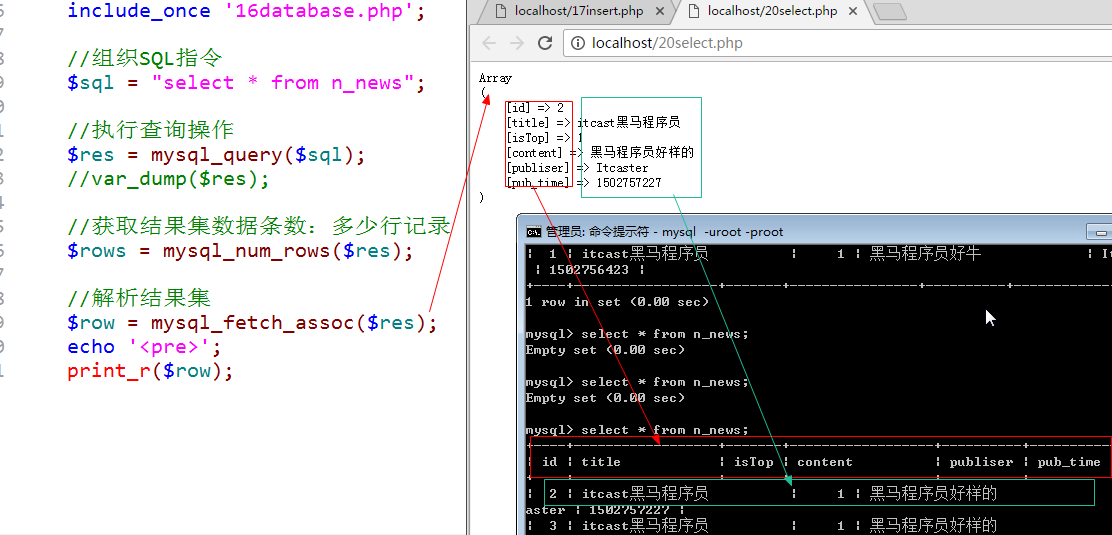
1）mysql\_num\_rows()：获取结果集中到底有多少行记录



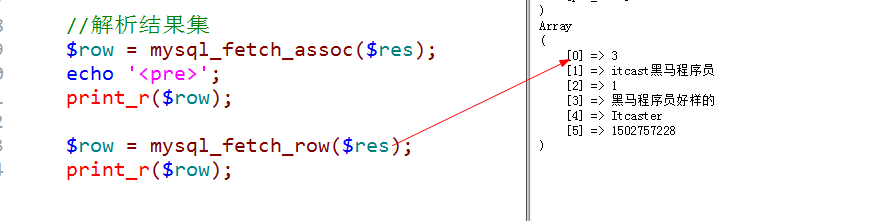
### 解析结果集

将一种结果集资源（PHP不能直接使用），转换成一种PHP能够解析的数据格式：通过从结果集中（结果集指针：类似数组指针），按照结果集指针所在位置取出对应的一条记录（一行），返回一个数组，同时指针下移…直到指针移出结果集。

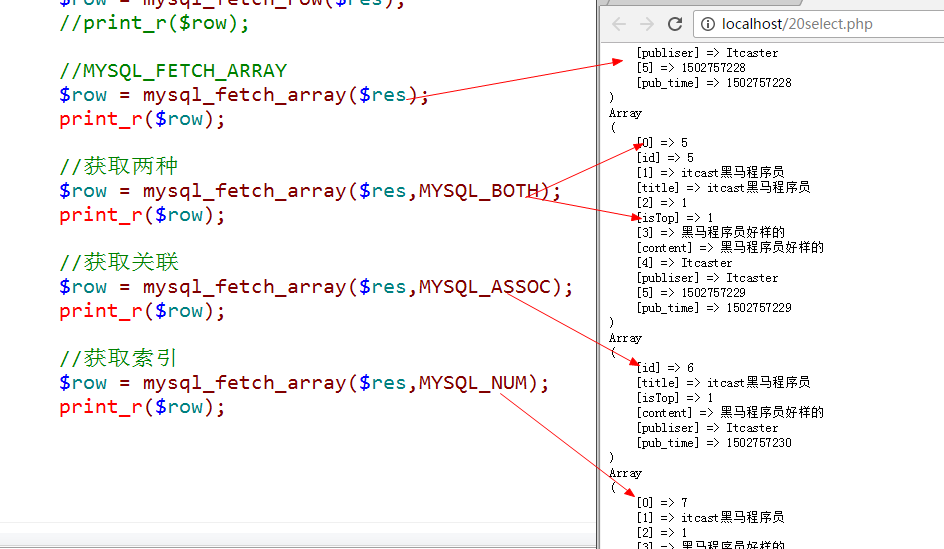
1）mysql\_fetch\_assoc()：获取关联数组，表的表单名字作为数组下标，元素值作为数组元素值



2）mysql\_fetch\_row()：获取索引数组，只获取数据的值，然后数组的下标从0开始自动索引



3）mysql\_fetch\_array()：获取关联或者索引数组，但是默认是同时存在：一个记录取两次，形成一组是关联数组，一组是索引数组；但是可以通过第二个参数来决定获取的方式：MYSQL\_ASSOC只获取关联数组；MYSQL\_NUM只获取索引数组；MYSQL\_BOTH获取全部两种。

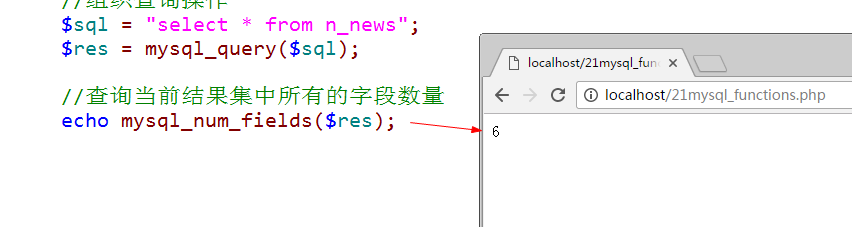


通常结果集的操作：一般是获取里面的所有（全部记录）

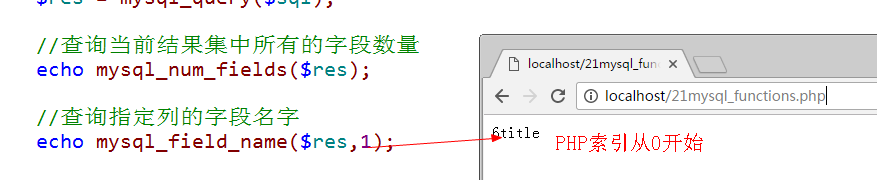
## 其他相关函数

### 有关字段信息

1）mysql\_num\_fields()：获取一个指定结果集中所有的字段数



2）mysql\_field\_name()：获取一个指定结果集中指定位置字段的名字

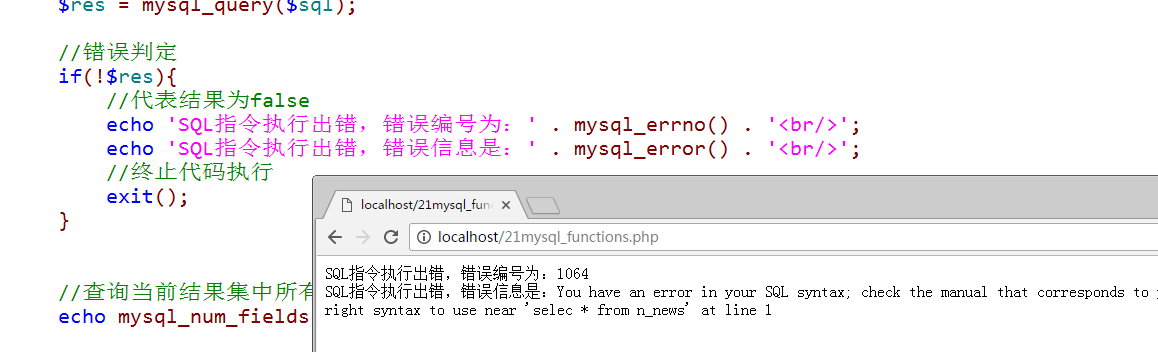


### 有关出错信息

1）mysql\_error()：获取出错对应的提示信息

2）mysql\_errno()：获取出错对应的错误提示代号

错误的判断：基于Mysql\_query这个函数执行的结果：结果返回false就代表执行错误。



### 其他函数

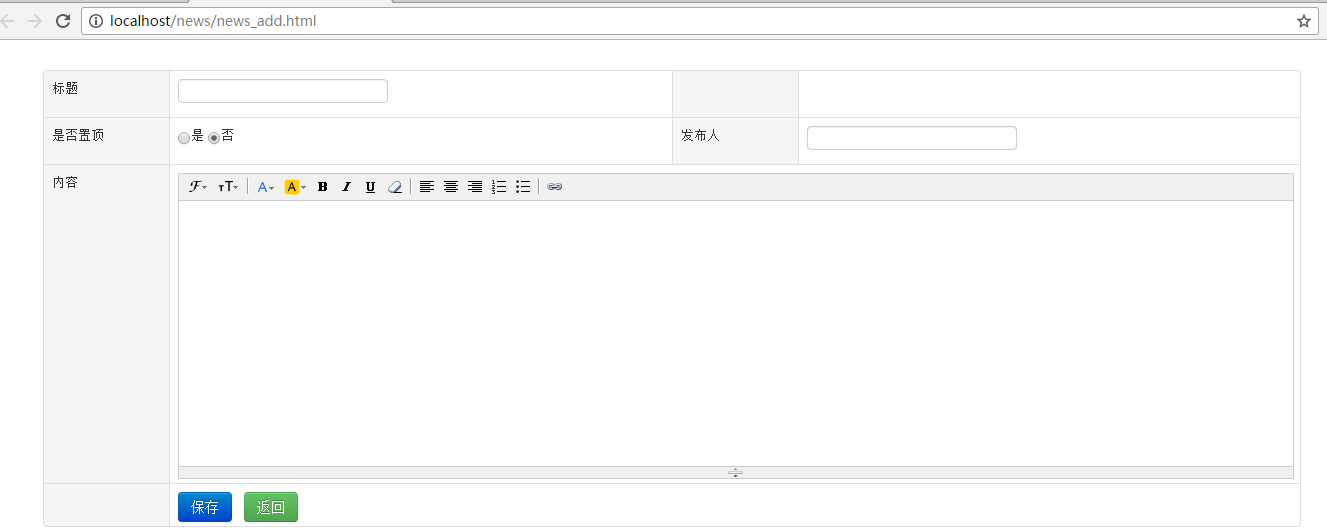
1. mysql\_insert\_id()：获取上次插入操作所产生的自增长ID，如果没有自增长ID返回0



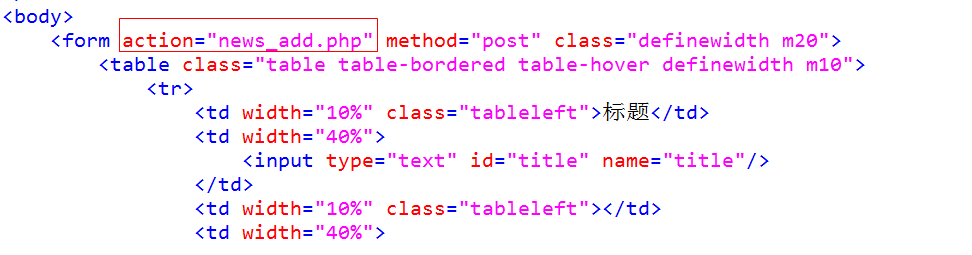
## Mysql扩展应用案例

### 添加新闻表单

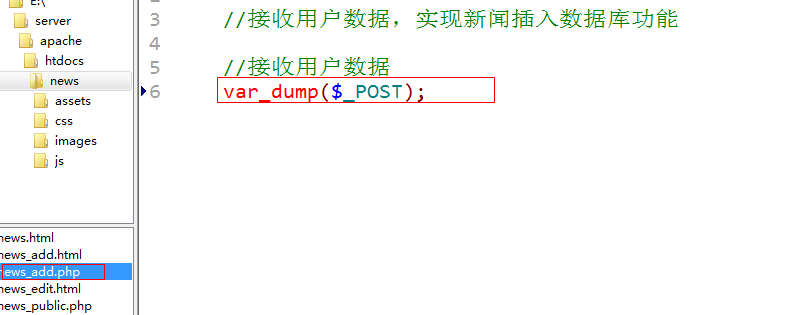
1） 用户获取到一个可以添加新闻表单的HTML文件



修改表单项：method提交对象

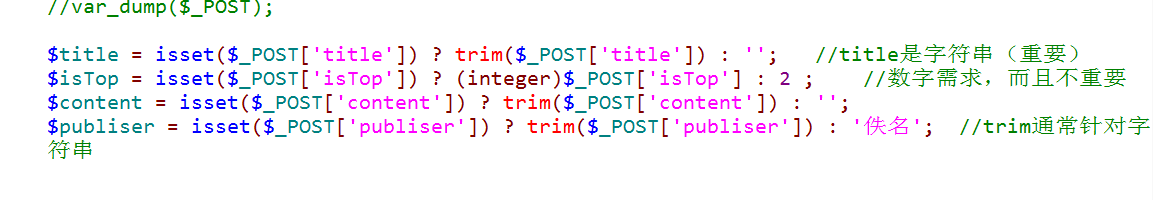


2） 用户提交数据到后台指定的PHP脚本（POST）

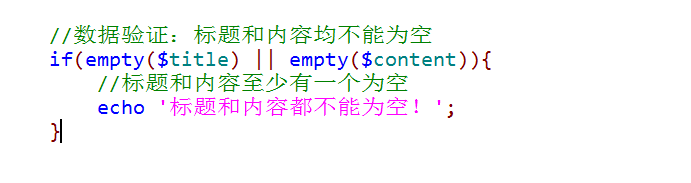


3） 后台脚本处理数据，插入到数据库：处理指PHP在拿到数据放入数据库之前要做很多验证工作保证数据是有效的。

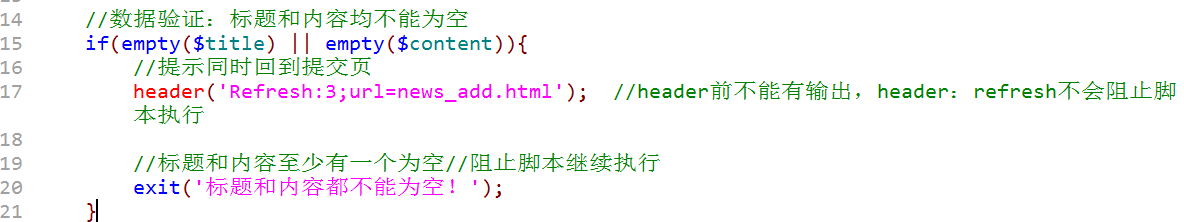
3.1 接收数据



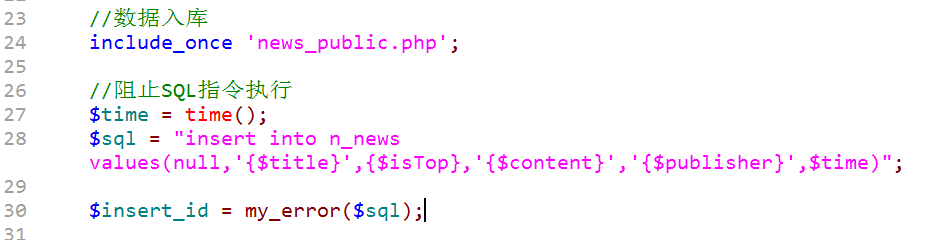
3,2 数据验证：合法性验证。标题不能为空，内容不能为空



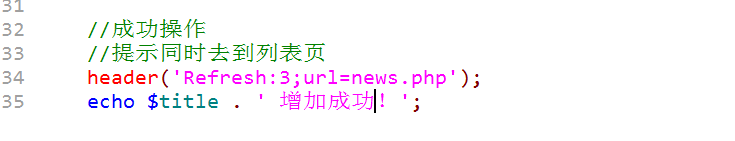
3.3 提示用户，同时让用户重新再来（回到新增表单页面）



3.4 数据入库



4） 提示用户操作结果，并跳转到列表页（跳转到详情页）



## Mysql扩展应用案例

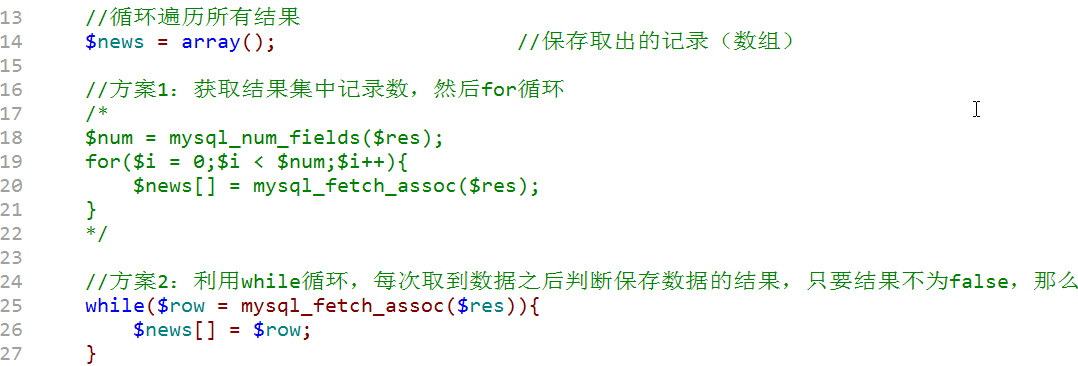
### 显示新闻列表

将数据库中所有的新闻信息获取出来，然后讨要HTML提供的显示模板把数据全部按照指定的位置和格式输出显示。

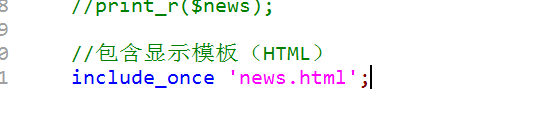
1、 从数据库获取所有的新闻数据：PHP脚本



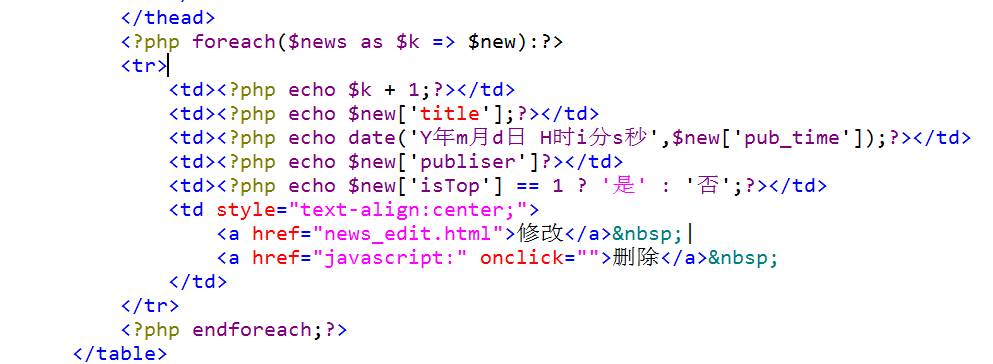
从结果集中取出所有的记录：一次取一条，一条一个数组；然后将所有的记录放到一个数组中：形成一个典型的二维数组



2、 有一个已经做好的HTML模板能够显示数据：前端



3、 PHP想办法将数据在HTML中显示（二者结合）：PHP包含HTML



注意细节：

1、 数据的显示要通过循环遍历输出：需要明确PHP是如何保存数据

2、 数据的格式需要按照需求来显示：列表、时间、是否置顶

## Mysql扩展应用案例

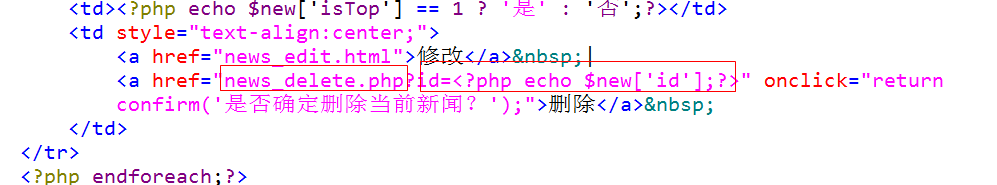
### 删除新闻

删除新闻就是通过用户选中某一个新闻点击删除实现从数据库剔除。

1、 让用户看到所有的新闻：已经完成（新闻显示列表：news.php🡪news.html）

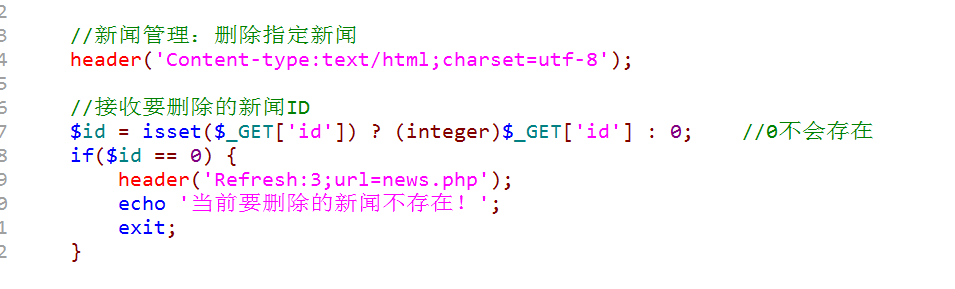
2、 给用户提供一个对应的点击按钮：已经存在

3、 实现点击针对指定的新闻

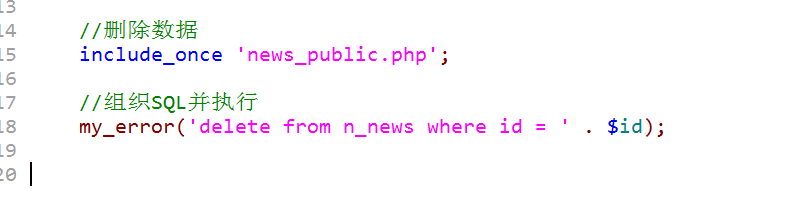


4、 有一个专门的PHP脚本负责响应用户的点击：点哪条删哪条

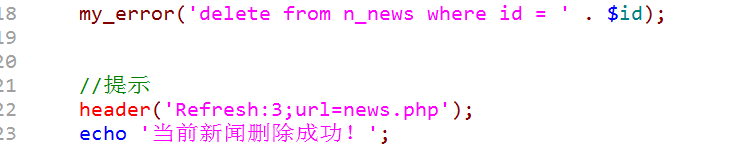
4.1 接收要删除的新闻ID



4.2 调用数据库操作删除数据



4.3 提示删除成功，并且刷新当前界面



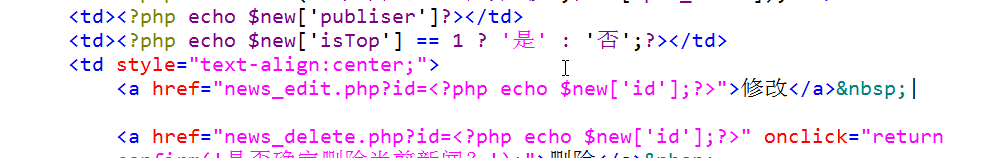
## Mysql扩展应用案例

### 修改新闻

修改新闻指用户通过某个渠道看到想修改的新闻，然后点击之后能够修改对应的新闻信息，最后点击保存实现修改更新到数据库。

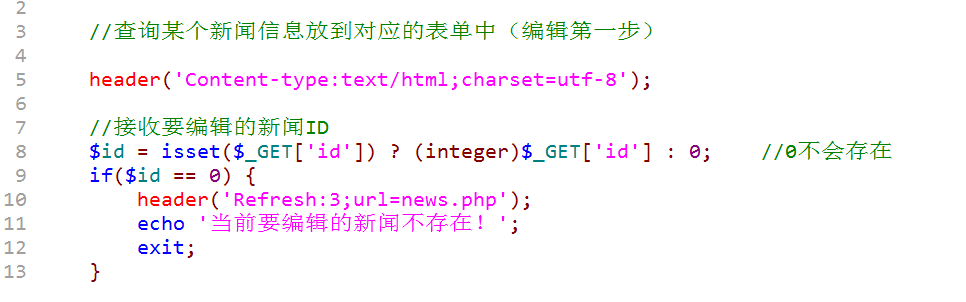
1、 需要看到想编辑新闻：已经完成：新闻列表news.php🡪news.html

2、 可以点击实现查看：列表页中为每个新闻增加点击链接

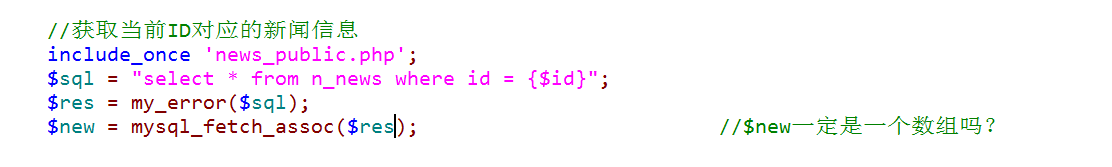


3、 需要一个PHP脚本能够处理这件事

3.1 创建PHP文件，同时接受要编辑的ID

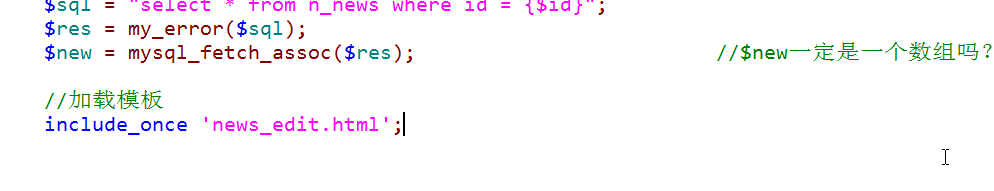


3.2 通过ID获取对应的新闻信息

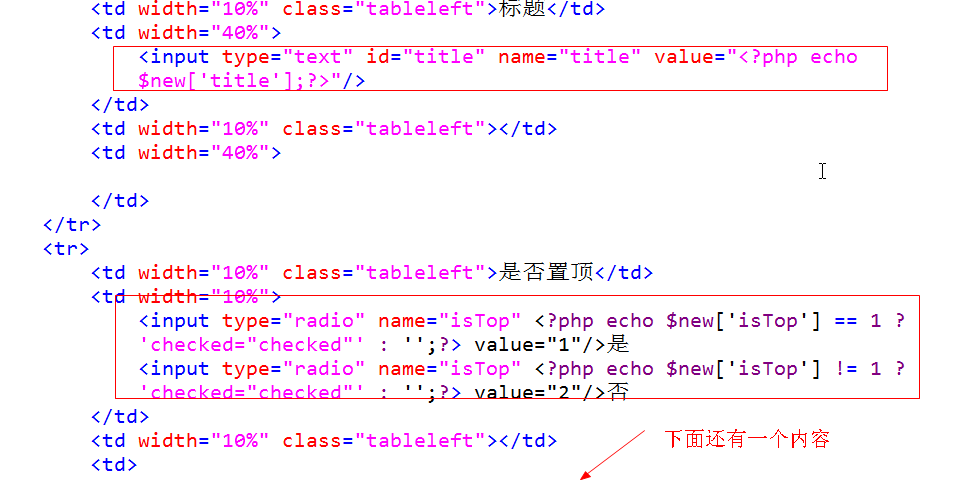


4、 需要一个HTML模板来展示当前要编辑的数据：PHP包含HTML

4.1 包含模板HTML

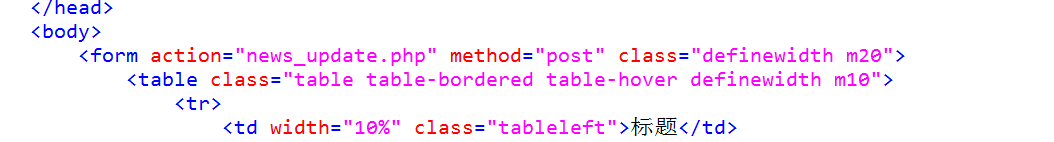


4.2 在模板表单中显示对应的数据

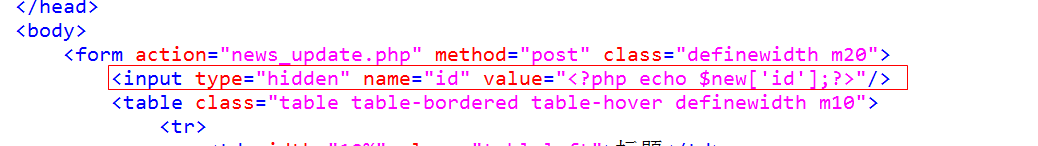


5、 需要修改前端提供的更新表单：表单提交对象（不限）

5.1 需要修改表单提交对象

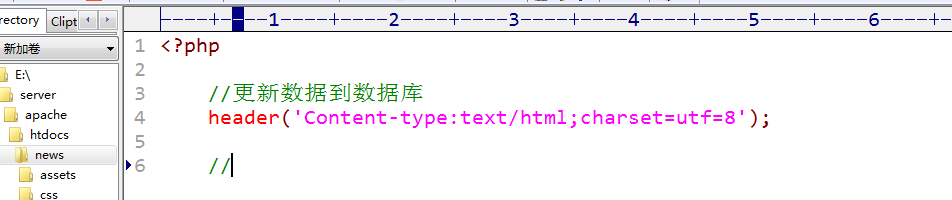


5.2 更新操作，是后台根据表单提交的数据来选择某个新闻进行更新：表单中必须包含一个唯一能确定记录的新闻：ID，而ID不需要给用户看，需要增加隐藏表单



6、 需要一个PHP脚本能够接受修改后的数据并保存到数据库

6.1 增加更新数据的PHP脚本



6.2 接收用户提交的数据，同时进行数据验证：标题和内容均不能为空



6.3 组织SQL更新到数据库



6.4 提示结果

