

主讲教师：传智.神龙教主

传智播客 PHP学院 <http://php.itcast.cn> 发布

## 回顾

1. 文件域：<input type=’file’ name=’’/>
2. 表单的enctype属性
   1. application/x-www-form-urlencoded：默认值，表示传输的是文本数据
   2. multipart/form-data：复合的数据类型（字符串和文件）
   3. text/plain：传递大量数据，用于电子邮件
3. $\_FILES[][]
   1. 行是文件域的名称，列有5个

$\_FILES[][‘name’]:上传的文件名

$\_FILES[][‘type’]：上传的文件类型

$\_FILES[][‘size’]：上传的文件大小，单位是字节

$\_FILES[][‘tmp\_name’]:上传文件在服务器上的临时保存文件地址

$\_FILES[][‘error’]:错误号

* 1. $\_FILES[][‘error’]

0：正确

1：文件超过了php.ini允许的最大值

2：文件超过了表单允许的最大值

3：只有部分上传

4：没有文件上传

6：找不到临时文件

7：写入失败

1. 与文件上传有关的配置
   1. upload\_max\_filesize=2M：允许上传文件的大小
   2. post\_max\_size=8M：post提交允许的最大值
   3. file\_uploads=on：允许文件上传
   4. upload\_tmp\_dir=’’：临时文件保存的目录
   5. max\_file\_uploads=20：允许同时上传20个文件
2. move\_uploaded\_file(临时目录,保存目录)
3. 如果上传了同名的文件，则后面的文件将前面的覆盖。解决：给文件一个不同名的文件名
   1. 时间戳 time()
   2. uniqid(‘前缀’,是否更加唯一)
4. 验证上传文件的类型
   1. 方法一：通过文件的扩展名（不能识别伪装）
      1. strrchr()
      2. pathinfo()
   2. 方法二：通过开启php\_pathinfo.dll扩展（可以识别伪装）
5. 生成缩略图
   1. imagecreatetruecolor()：创建真彩色图片
   2. imagecopyresampled()：生成缩略图

## 作业：修改商品

### 显示修改商品界面

控制器

public function modifyAction(){

//获取商品类别

$cat\_Model=new CategoryModel();

$cat\_list=$cat\_Model->getCategoryTree();

//获取商品信息

$goodsid=$\_GET['goodsid'];

$goods\_Model=new GoodsModel();

$goods\_info=$goods\_Model->find($goodsid);

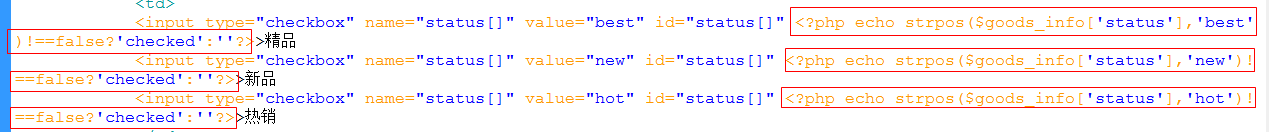
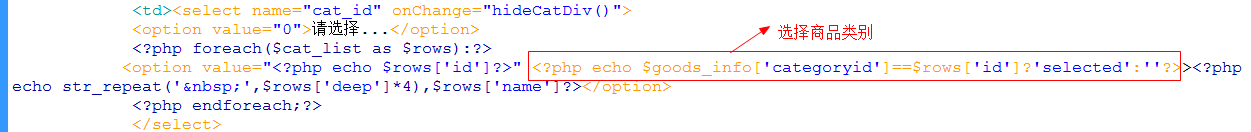
require \_\_VIEW\_\_.'goods\_update.html';

}

模型

没有

视图



### 实现修改的业务逻辑

模型

没有

视图

没有

控制器

public function modifyAction(){

$goodsid=$\_GET['goodsid'];

//获取商品信息

$goods\_Model=new GoodsModel();

$goods\_info=$goods\_Model->find($goodsid);

if(!empty($\_POST)){

$data['name']=$\_POST['goods\_name'];

$data['price']=$\_POST['shop\_price'];

$data['categoryid']=$\_POST['cat\_id'];

$data['status']= isset($\_POST['status'])?implode(',',$\_POST['status']):'';

$data['goods\_desc']=$\_POST['goods\_desc'];

//文件上传

if($\_FILES['goods\_img']['name']!=''){

$upload=new UploadLib();

if(!$path=$upload->upload($\_FILES['goods\_img'])){

$this->error('index.php?p=Admin&c=Goods&a=add',$upload->getError());

exit;

}

$this->delRubbishImage($goods\_info); //删除垃圾文件

$data['img']=$path; //源图的路径

//生成缩略图

$image=new ImageLib();

$src\_path=$GLOBALS['config']['app']['upload\_path'].$path;

$data['img\_thumb\_sm']=$image->thumb($src\_path, 100, 100,'sm\_');

$data['img\_thumb\_mid']=$image->thumb($src\_path, 300, 300,'mid\_');

}

$data['goodsid']=$goodsid;

//更改数据库

$model=new GoodsModel();

if($model->update($data))

$this->success ('index.php?p=Admin&c=Goods&a=list');

else

$this->error ('index.php?p=Admin&c=Goods&a=add','修改失败');

exit;

}

//获取商品类别

$cat\_Model=new CategoryModel();

$cat\_list=$cat\_Model->getCategoryTree();

require \_\_VIEW\_\_.'goods\_update.html';

}

//删除垃圾图片

private function delRubbishImage($goodsinfo){

$img=$GLOBALS['config']['app']['upload\_path'].$goodsinfo['img'];

$img\_thumb\_sm=$GLOBALS['config']['app']['upload\_path'].$goodsinfo['img\_thumb\_sm'];

$img\_thumb\_mid=$GLOBALS['config']['app']['upload\_path'].$goodsinfo['img\_thumb\_mid'];

if(file\_exists($img))

unlink ($img); //删除文件

if(file\_exists($img\_thumb\_sm))

unlink ($img\_thumb\_sm);

if(file\_exists($img\_thumb\_mid))

unlink ($img\_thumb\_mid);

}

## 作业：删除商品

控制器

public function delAction(){

$goodsid=(int)$\_GET['goodsid'];

$model=new GoodsModel();

$goodsinfo=$model->find($goodsid);

if($model->del($goodsid)){

$this->delRubbishImage($goodsinfo);

$this->success('index.php?p=Admin&c=Goods&a=list', '删除成功',1);

}else{

$this->error('index.php?p=Admin&c=Goods&a=list', '删除失败');

}

}

模型

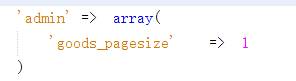
没有

视图



## 作业：分页

### 在配置文件中加入页面大小的参数



### 在基础模型（Model）中封装分页和获取总记录数的方法

/\*

\* 封装一个万能的获取当前页数据的方法

\* @param $pageno int 当前页码

\* @param $pagesize int 页面大小

\*/

public function getPageList($pageno,$pagesize){

$table= $this->getTable();

$pk= $this->getPrimaryKey($table);

$startno=($pageno-1)\*$pagesize;

$sql="select \* from `{$table}` order by `{$pk}` desc limit $startno,$pagesize";

return $this->db->fetchAll($sql);

}

/\*

\* 封装一个万能的获取总记录数的方法

\*/

public function count(){

$table= $this->getTable();

$sql="select count(\*) from `{$table}`";

return $this->db->fetchColumn($sql);

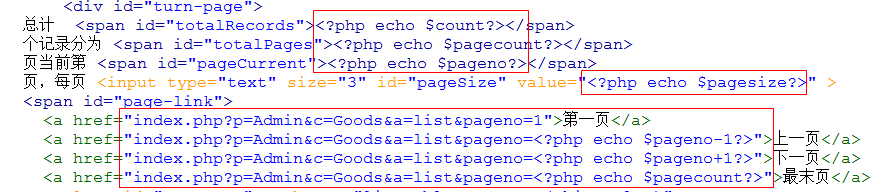
}

### 实现分页

表模型

无

视图



控制器

public function listAction(){

$model=new GoodsModel();

$pagesize=$GLOBALS['config']['admin']['goods\_pagesize']; //页面大小

$count=$model->count(); //总记录数

$pagecount= ceil($count/$pagesize); //总页数

$pageno=isset($\_GET['pageno'])?$\_GET['pageno']:1; //当前页

if($pageno<1)

$pageno=1;

if($pageno>$pagecount)

$pageno=$pagecount;

$list=$model->getPageList($pageno, $pagesize); //分页

require \_\_VIEW\_\_.'goods\_list.html';

}

## PHP错误处理

在默认情况下，当错误发生的时候显示错误。

可以使用@来屏蔽非致命的错误。



### PHP错位级别

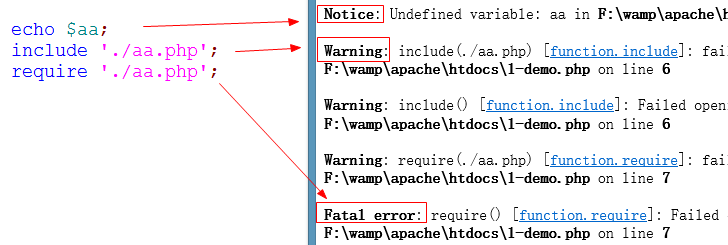
按照错位的严重程度，将错误划分为：

notice：提示性错误

warning：警告性错误

error：致命性错误

发生了非致命性错误继续执行，致命性错误终止执行。



### 处理错误的三种方式

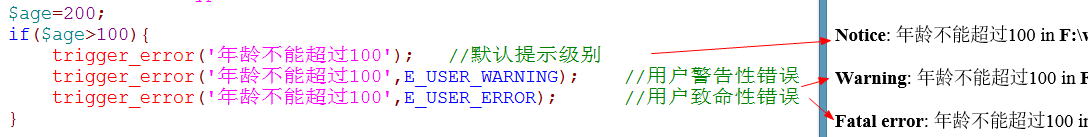
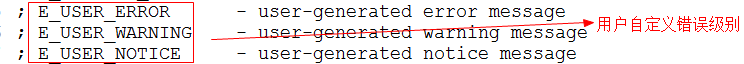
1. 在浏览器上显示错误报告（默认的方式）
2. 将错误记录在日志中
3. 交给用户自定义处理错误脚本完成

### 错误的触发两种方法

方式一：PHP程序触发

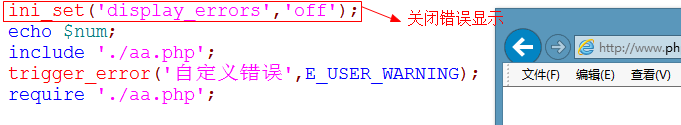
在PHP程序运行的过程中，发现并触发错误，比如include,require，…

方式二：用户触发



## 错误处理

### 在浏览器上显示|关闭错误报告



脚下留心：更改php.ini配置可以在3个地方

第一：php.ini文件，对所有的虚拟目录有效

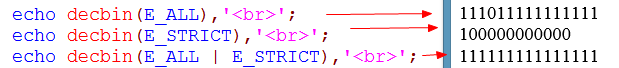
第二：分布式部署文件，在当前目录及子目录中有效

第三：在php页面中通过ini\_set()函数，在当前php页面有效

### 显示指定级别的错误



多学一招：在php5.3中，E\_ALL | E\_STRICT表示所有错误。



多学一招：按位非（~）：1变成0,0变成1



例题

//ini\_set('error\_reporting',(E\_ALL | E\_STRICT) & ~E\_NOTICE); //系统的提示错误不显示

//ini\_set('error\_reporting',(E\_ALL | E\_STRICT) & ~E\_NOTICE & ~ E\_USER\_NOTICE); //所有的提示错误都不显示

//ini\_set('error\_reporting',E\_ERROR | E\_USER\_ERROR); //只显示致命错误信息

ini\_set('error\_reporting',(E\_ALL | E\_STRICT) & ~E\_ERROR & ~ E\_USER\_ERROR); //致命性错误不显示

echo $aa;

trigger\_error('myerror');

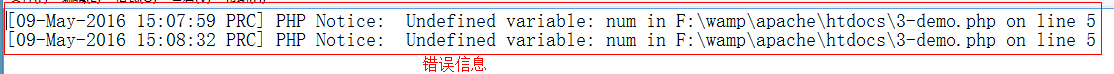
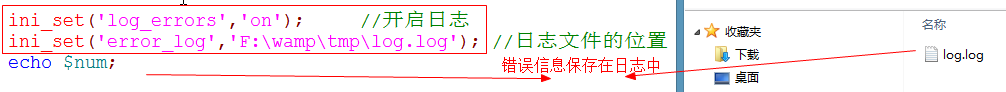
include './aa.php';

mysql\_co();

### 在日志中记录错误信息

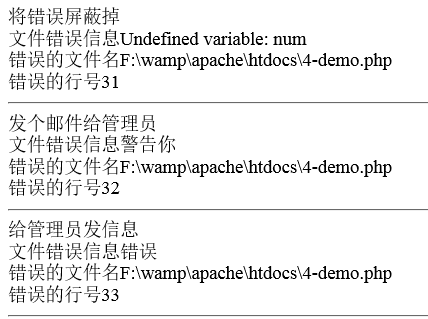


例题



### 自定义错误处理

1. <?php
2. header('content-type:text/html;charset=utf-8');
3. /\*\*
4. \*自定义错误
5. \*@param $level 错误级别
6. \*@param $msg 错误信息
7. \*@param $file 错误的文件
8. \*@param $line 错误的行号
9. \*/
10. function customerError($level,$msg,$file,$line) {
11. switch($level) {
12. case E\_NOTICE:
13. case E\_USER\_NOTICE:
14. echo '将错误屏蔽掉<br>';
15. break;
16. case E\_WARNING:
17. case E\_USER\_WARNING:
18. echo '发个邮件给管理员<br>';
19. break;
20. case E\_ERROR:
21. case E\_USER\_ERROR:
22. echo '给管理员发信息<br>';
23. break;
24. }
25. echo '文件错误信息'.$msg,'<br>';
26. echo '错误的文件名'.$file,'<br>';
27. echo '错误的行号'.$line,'<hr>';
28. //return false; //在自定义错误处理完毕后再交个PHP处理
29. }
30. set\_error\_handler('customerError');
31. echo $num;
32. trigger\_error('警告你',E\_USER\_WARNING);
33. trigger\_error('错误',E\_USER\_ERROR);



## 在项目中封装错误提示

在开发的时候，将错误信息显示在浏览器中，在运行的时候将错误信息保存到日志中。

在配置文件中用一个变量定义开发模式和运行模式



在framework.class.php中，定义日志目录



在Framework类中定义初始化显示出错方式

private static function initError(){

ini\_set('error\_reporting', E\_ALL | E\_STRICT); //报所有错误

if($GLOBALS['config']['app']['app\_debug']){ //开发模式

ini\_set('display\_errors', 'on'); //在浏览器上显示

ini\_set('log\_errors', 'off'); //关闭日志功能

}else{ //运行模式

ini\_set('display\_errors', 'off'); //关闭在浏览器显示

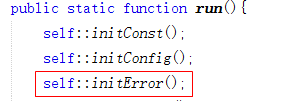
ini\_set('log\_errors', 'on'); //开启日志功能

ini\_set('error\_log', LOG\_PATH.'error.log'); //日志地址

}

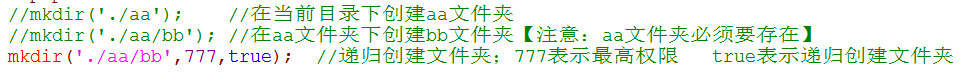
}

调用初始化错误显示

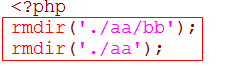


## 文件夹操作

### 创建文件夹（mkdir）



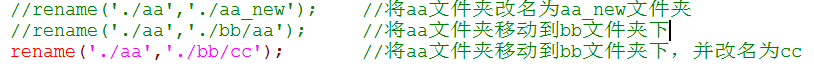
### 删除文件夹（rmdir）



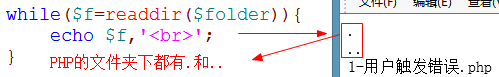
脚下留心：如果文件夹下有内容，此文件夹是不允许删除的。



### 重命名、移动文件夹(rename)



### 打开文件夹（opendir）并读取文件夹(readdir)



通过递归的方式遍历文件夹

<?php

//通过递归实现遍历文件夹

function getFile($path) {

$folder=opendir($path); //打开文件夹

echo '<ul>';

while($f=readdir($folder)){

if($f=='.' || $f=='..')

continue;

echo '<li>',$f,'</li>';

if(is\_dir("{$path}/{$f}")){ //如果当前是文件夹继续递归

getFile("{$path}/{$f}");

}

}

echo '</ul>';

}

$path=$\_SERVER['DOCUMENT\_ROOT']; //根目录

getFile($path);

closedir(); //关闭文件夹

is\_dir()：是否是文件夹

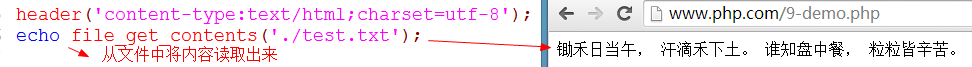
closedir()：关闭文件夹

## 文件操作

### file\_put\_contents()：将内容写到文件中



### file\_get\_contents()：将文件内容读取出来



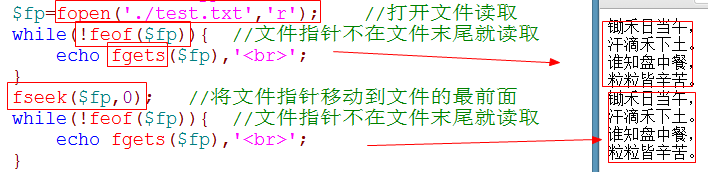
### 打开文件（fopen）

语法：$fp=fopen(地址，模式)，返回的是文件指针（file pointer）

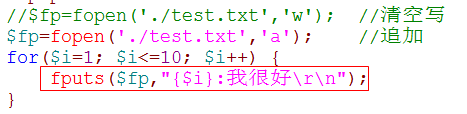
|  |  |
| --- | --- |
| 模式 | 含义 |
| r | 读 |
| w | 写（清空重写） |
| a | 追加 |

### fgets()、feof()、fseek()

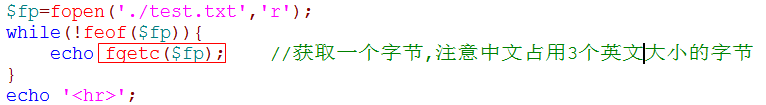
1. fgets()：读取一行，指针下移一行
2. feof()：判断文件指针是否在文件末尾，如果文件指针在末尾返回true，否则返回false
3. fseek()：移动文件指针



### fputs()：向文件中写一行

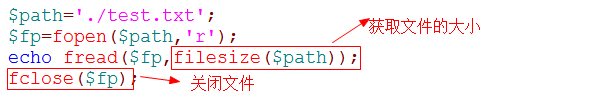


### getc()：获取一个字符



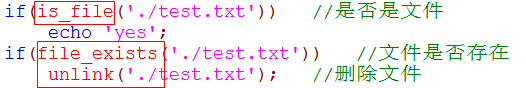
### read(文件指针，文件大小)：二进制读取

二进制文件没有明确的结束标记，通过文件大小来判断文件是否读取完毕



### is\_file()、file\_exists()、unlink()

1. is\_file()：是否是文件
2. file\_exists()：文件夹或文件是否存在
3. unlink()：删除文件



### 诗歌分页

<?php

header('content-type:text/html;charset=utf-8');

$fp=fopen('./shige.txt','r');

$str='';

while(!feof($fp)) {

$str.=fgets($fp).'<br>';

}

//通过'--'来分割

$array=explode('--',$str);

$pageno=isset($\_GET['pageno'])?$\_GET['pageno']:1;

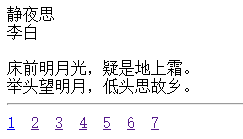
echo $array[$pageno-1];

echo '<hr>';

for($i=1,$n=count($array); $i<=$n; $i++) {

echo "<a href='?pageno={$i}'>{$i}</a>&nbsp;";

}



### 新闻分页

$fp=fopen('./news.txt','r');

$str='';

$n=0;

while(!feof($fp)) {

$str.=fgets($fp);

if(++$n%20==0)

$str.='<<<<<';

}

$array=explode('<<<<<',$str);

$pageno=isset($\_GET['pageno'])?$\_GET['pageno']:1;

echo $array[$pageno-1];

echo '<hr>';

for($i=1,$n=count($array); $i<=$n; $i++) {

echo "<a href='?pageno={$i}'>{$i}</a>&nbsp;";

}

### 动态转静态（页面的静态化）

<?php

$url="http://www.php.com/index.php";

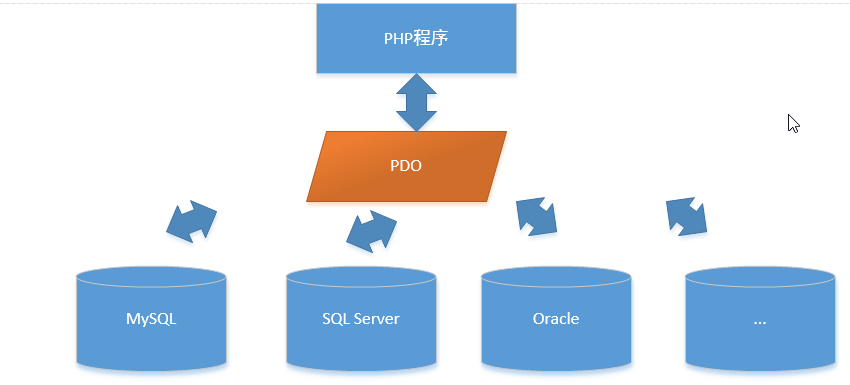
$str=file\_get\_contents($url);

file\_put\_contents('./index.html',$str);

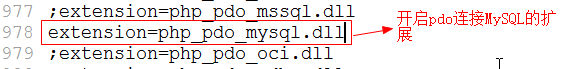
echo 'success';

## PDO

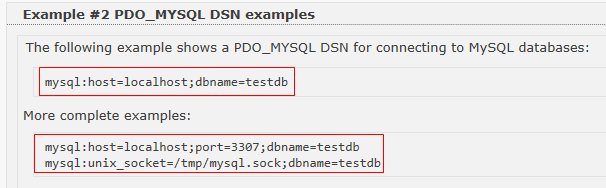
PDO（PHP Data Object）PHP数据对象，它为PHP访问数据库定义了一个轻量级的、一致性的接口。无论访问什么数据库，PDO都支持，都可以通过统一的函数去获取或执行数据。



PDO只有在php5以上的版本才支持。要使用pdo，必须开启相应的扩展



### pdo驱动



### 执行SQL语句

$pdo->exec()：执行数据操作语句，错误返回false，正确返回受影响的记录数。

$pdo->query()：执行数据查询语句

<?php

$pdo=new PDO('mysql:host=localhost;port=3306;dbname=php4','root','aa'); //pdo驱动见帮助文档

$pdo->exec('set names utf8');

if(isset($\_POST['button'])) {

$name=$\_POST['name'];

$pwd=$\_POST['pwd'];

$ip=ip2long($\_SERVER['REMOTE\_ADDR']);

$time=time();

$sql="insert into admin values (null,'$name','$pwd',$ip,$time)";

echo $pdo->exec($sql),'<br>'; //返回受影响的记录数

echo '生成的自动增长编号：'.$pdo->lastInsertId(),'<br>';

}

//显示所有管理员

$rs=$pdo->query('select \* from admin');

//var\_dump($rs); //PDOStatement

?>

<table>

<tr>

<th>编号</th> <th>姓名</th> <th>密码</th> <th>最后登录IP</th> <th>最后登录时间</th>

</tr>

<?php foreach($rs as $rows):?>

<tr>

<td><?php echo $rows['id']?></td>

<td><?php echo $rows['name']?></td>

<td><?php echo $rows['pwd']?></td>

<td><?php echo long2ip($rows['last\_login\_ip'])?></td>

<td><?php echo date('Y-m-d H:i:s',$rows['last\_login\_time'])?></td>

</tr>

<?php endforeach;?>

</table>

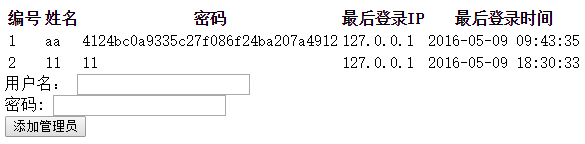
<form method="post" action="">

用户名： <input type="text" name="name"> <br />

密码: <input type="text" name="pwd"> <br />

<input type="submit" name="button" value="添加管理员">

</form>



## 预处理语句（位置占位符）

<?php

$pdo=new PDO('mysql:host=localhost;port=3306;dbname=php4','root','aa');

$pdo->exec('set names utf8');

if(isset($\_POST['button'])) {

$sql="insert into admin values (null,?,?,?,?)"; //?是位置占位符

$stmt=$pdo->prepare($sql); //预处理语句

$ip=ip2long($\_SERVER['REMOTE\_ADDR']);

$time=time();

//var\_dump($stmt); //object(PDOStatement)

$stmt->bindParam(1,$\_POST['name']); //在第一个位置上绑定name的值

$stmt->bindParam(2,$\_POST['pwd']);

$stmt->bindParam(3,$ip);

$stmt->bindParam(4,$time);

echo $stmt->execute()?'成功':'失败'; //执行预处理

}

?>

<form method="post" action="">

用户名： <input type="text" name="name"> <br />

密码: <input type="text" name="pwd"> <br />

<input type="submit" name="button" value="添加管理员">

</form>

## 预处理语句（参数占位符）

$pdo=new PDO('mysql:host=localhost;port=3306;dbname=php4','root','aa');

$pdo->exec('set names utf8');

if(isset($\_POST['button'])) {

$sql="insert into admin values (null,:name,:pwd,:ip,:time)"; //:name参数占位符

$stmt=$pdo->prepare($sql); //预处理语句

$ip=ip2long($\_SERVER['REMOTE\_ADDR']);

$time=time();

//var\_dump($stmt); //object(PDOStatement)

$stmt->bindParam(':name',$\_POST['name']); //在第一个位置上绑定name的值

$stmt->bindParam(':pwd',$\_POST['pwd']);

$stmt->bindParam(':ip',$ip);

$stmt->bindParam(':time',$time);

echo $stmt->execute()?'成功':'失败'; //执行预处理

}

## 预处理读取

<?php

$pdo=new PDO('mysql:host=localhost;port=3306;dbname=php4','root','aa');

$pdo->exec('set names utf8');

$sql="select \* from admin order by id desc limit 0,:n";

$stmt=$pdo->prepare($sql);

$stmt->bindParam(':n',$n);

$n=3;

$stmt->execute();

?>

<table>

<tr>

<th>编号</th> <th>姓名</th> <th>密码</th> <th>最后登录IP</th> <th>最后登录时间</th>

</tr>

<?php

while($rows=$stmt->fetch(PDO::FETCH\_NUM)) {

?>

<tr>

<td><?php echo $rows[0]?></td>

<td><?php echo $rows[1]?></td>

<td><?php echo $rows[2]?></td>

<td><?php echo $rows[3]?></td>

</tr>

<?php

}

?>

</table>

## 异常处理

异常处理是OOP中非常重要的知识点，由于PHP中从错误处理比异常处理功能强大，所以在PHP中，很少用异常处理。

关键字：

1. try：检测代码块
2. throw：抛出异常
3. catch：捕获异常
4. Exception：异常，这是一个内置的类

### 语法

try

{

throw 异常

}

Catch(异常){

//处理异常

}

例题1

<?php

if(isset($\_POST['button'])) {

try{

$age=$\_POST['age'];

if($age>100)

throw new Exception('年龄不能超过100'); //抛出异常对象

echo "你的年龄是：{$age}<br>";

}catch(Exception $e){ //捕获异常

echo $e->getMessage();

}

}

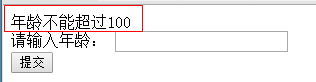
?>

<form method="post" action="">

请输入年龄： <input type="text" name="age"> <br />

<input type="submit" name="button" value="提交">

</form>



例题二：

<?php

if(isset($\_POST['button'])) {

try

{

$num1=$\_POST['num1'];

$num2=$\_POST['num2'];

if(!is\_numeric($num1))

throw new Exception('第一个不是数字');

elseif(!is\_numeric($num2))

throw new Exception('第二个不是数字');

elseif($num2==0)

throw new Exception('除数不能为零');

else {

echo $num1/$num2;

}

}

catch(Exception $e){

echo $e->getMessage(),'<br>';

echo '行号:'.$e->getLine();

}

}

?>

<form method="post" action="">

第一个数： <input type="text" name="num1"> <br />

第二个数： <input type="text" name="num2"> <br />

<input type="submit" name="button" value="求商">

</form>

## 作业

1. 写一个删除垃圾图片的功能（在数据库中找不到的图片）