

DW 属性中可以选 HTML 和 CSS,html 中可以选择 div,ul,li ,ol。

DW 切图的时候可以在页面属中先“跟踪图像”加载蓝图，设置半透明，更加方便切图。

超链接点击，经过，点击过后颜色的变化，DW 中变化。

锚记的作用

a:hover {color:red;} //经过

a:active {color:green;} //正在点击中

表格居中：`<table align="center"></table>`

超链接 target 中的\_blank 选项：将链接文件加载到未命名的新浏览器窗口中

设置 div 边框颜色：`<div style="border:1px solid #000">这是 DIV1</div>`

`<div style="border:1px dashed #000;"> 黑色虚线边框</div>`

使用<hr style="border:1px dashed #F00;border-bottom:0;border-right:0;border-left:0;width:600px;">

去掉点 li 的点:`<li style="list-style-type:none;">百度</li>`

服务 dos 代码：services.msc

html 头部-----

`<head>`

`<title>我的第一页面</title>`

```
<meta name="description" content="the page of html"><!--描述-->

<meta name="keywords" content="good,better,best"><!--关键字-->

<meta http-equiv="content-type" content="text/html;charset=gb2312"><!--编码-->

<meta name="author" content="zhou runfa"><!--作者姓名-->

<meta http-equiv="refresh" content="3; url=http://www.263.net"><!--refresh , 用
来设置网页的自动更新-->

<!--当 content ="3; url=http://www.263.net"时 , 该网页打开 3 秒钟后 , 就自动的转
到 www.263.net 网站-->

<link type="text/css" rel="stylesheet" href="css/inc.css" />

</head>

html 头部-----
```

通过 GET 方法从表单发送的信息对任何人都是可见的（所有变量名和值都显示在 URL 中）。GET 对所发送信息的数量也有限制。限制在小于 2000 个字符。不过，由于变量显示在 URL 中，把页面添加到书签中也更为方便。

'是直接输出，"是运行之后输出

mysql 数据库：<http://localhost/phpmyadmin/>

日期：<?php echo date("Y-m-d H:i:s ") ?> //php 语言默认的标准时间是格林威治时间，修改方法有二：其一，修改 php.ini 文件设

置，找到[date]下

的";date.timezone="选项，改成"date.timezone=Asia/Hong\_Kong"，然后重启 apache 服务器。其二，在函数前添加函数

```
"date_default_timezone_set("Asia/Hong_Kong");"即可,或者是  
"ini_set('date.timezone','Asia/Shanghai');"
```

获取当前时区：echo date\_default\_timezone\_get();

注释：单行注释//xxxx,多行注释/\*xxx\*/,shell 风格注释#xxx,(注意，单行注释里不能出现?>，否则解析器会认为是 php 脚本结束。)

echo 和 print 之间的差异：

echo - 能够输出一个以上的字符串

print - 只能输出一个字符串，并始终返回 1

提示：echo 比 print 稍快，因为它不返回任何值。

```
<?php  
    $a='xiaojing';  
    $arr=array('one','two','three');  
    echo "$a is a goodboy!";  
    echo "{$arr['1']}"  
?>
```

在 PHP 中，所有用户定义的函数、类和关键词（例如 if、else、echo 等等）都对大小写不敏感。

不过在 PHP 中，所有变量都对大小写敏感。ps:

```
$a=3;  
$A=7;  
  
echo $a; //输出 3  
  
echo $A; //输出 7
```

PHP 变量规则：

变量以 \$ 符号开头，其后是变量的名称

变量名称必须以字母或下划线开头

变量名称不能以数字开头

变量名称只能包含字母数字字符和下划线 ( A-z、0-9 以及 \_ )

变量名称对大小写敏感 ( \$y 与 \$Y 是两个不同的变量 )

```
常量 bool define ( string $name , mixed $value [, bool $case_insensitive = false ] )
```

首个参数定义常量的名称，第二个参数定义常量的值，可选的第三个参数规定常量名是否对大小写敏感。默认是 false。

```
<?php
define("GREETING", "Welcome to W3School.com.cn!");
echo GREETING;
?>
```

用户定义的函数声明以关键字"function" 开头

PHP 默认函数参数值，如果我们调用没有参数的函数，它的参数会取默认值：

```
<?php
function setHeight($minheight=50) {
    echo "The height is : $minheight <br>";
}
setHeight(350);
setHeight(); // 将使用默认值 50
setHeight(135);
```

```
setHeight(80);
```

```
?>
```

如需使函数返回值，请使用 return 语句

```
<?php
```

```
function sum($x,$y){
```

```
    $z=$x+$y;
```

```
    return $z;
```

```
}
```

```
echo sum(3,7);
```

```
?>
```

```
<?php
```

```
var_dump(3+7); //输出结果是：int 10
```

```
var_dump("3"+"7"); //输出结果是：int 10
```

```
var_dump("3"+7); //输出结果是：int 10
```

```
var_dump(3+"7"); //输出结果是：int 10
```

```
?>
```

作用域(begin)-----

PHP 有三种不同的变量作用域：

local ( 局部 )

global ( 全局 )

static ( 静态 )

global 关键词用于访问函数内的全局变量。 ps:

```
<?php
```

```
$x=5;
```

```
$y=10;
```

```
function myTest() {
```

```
    global $x,$y;
```

```
    $y=$x+$y;
```

```
}
```

```
myTest();
```

```
echo $y; // 输出 15
```

```
?>
```

PHP 同时在名为\$GLOBALS[index]的数组中存储了所有的全局变量。下标存有变量名。这个数组在函数内也可以访问，并能够用于直接更新全局变量。

上面的例子可以这样重写

```
<?php
```

```
$x=5;
```

```
$y=10;
```

```
function myTest() {
```

```
    $GLOBALS['y']=$GLOBALS['x']+$GLOBALS['y'];
```

```
}
```

```
myTest();
```

```
echo $y; // 输出 15
```

```
?>
```

通常，当函数完成/执行后，会删除所有变量。不过，有时我需要不删除某个局部变量。实现这一点需要更进一步的工作。

要完成这一点，请在您首次声明变量时使用 static 关键词：

```
<?php
```

```
function myTest() {  
    static $x=0;  
    echo $x;  
    $x++;  
}
```

```
myTest();  
myTest();  
myTest();
```

```
?>
```

作用域(end)-----

整形：数字 0 开头的是八进制，因为十进制是不会有 0 开头的，0x 开头的是十六进制。

php 中不是只有 false 值才为假，这些特殊情况是：0、0.0,"0",空字符串("")和只声明但没有赋值的数组等。

.连接符：.连接符不能直接挨着数字，前后都要空一格，如：echo '传统书写格式：  
3.14159265359= ' . 3.14159265359 .'<p>';

定义简单的字符串是用单引号比较合适，如果用双引号，php 将要话费一些时间来处理。

null 型：被赋予空值的情况有以下 3 种。

a 没有赋值任何值。

b 被赋予 null 值。

c 被 unset()函数处理过的变量。

#### 将指定字符串转换成其他类型:

(boolean)\$num、(boolean)\$str

(string)\$boo、(string)\$flo

(integer)\$boo、(integer)\$str

(float)\$str、(float)\$str

(array)\$str

(object)\$str

```
ps:$a=7; $a=(string)$a;
```

#### PHP 常用的预定义常量

PHP\_VERSION：PHP 版本号

PHP\_OS：操作系统类型

PHP\_SAPI：当前使用的 PHP 接口类型

PHP\_EOL：换行符

DEFAULT\_INCLUDE\_PATH：默认包含目录

\_\_LINE\_\_：当前所在的行号

\_\_FILE\_\_：当前文件的绝对路径

\_\_DIR\_\_：当前文件的目录路径（末尾不包含斜线）

\_\_FUNCTION\_\_：函数名称

\_\_CLASS\_\_：类名称



\_\_NAMESPACE\_\_：命名空间名

### 转义字符；

\a 响铃(BEL)  
\b 退格(BS)，将当前位置移到前一个  
\f 换页(FF)，将当前位置移到下页开头  
\n 换行(LF)，将当前位置移到下一行开头  
\r 回车(CR)，将当前位置移到本行开头  
\t 水平制表(HT)（跳到下一个 TAB 位置）  
\v 垂直制表(VT)  
\\ 代表一个反斜线字符"  
' 代表一个单引号（撇号）字符  
\" 代表一个双引号字符  
\0 空字符(NULL)  
\ddd 1 到 3 位八进制数所代表的任意字符  
\xhh 1 到 2 位十六进制数所代表的任意字符

==是等于，===是全等(数据类型也等)，!==不全等

<?php

```
$a=7;$b='7';
```

```
var_dump($a==$b); //返回 true,因为值相等
```

```
var_dump($a=== $b); //返回 false，因为类型不相等，$a 是 int,$b 是 string
```

```
var_dump($a!== $b); //返回 true
```

?>

do...while 的 while 语句括号后面记得 ";"

表格参数：Cellpadding 和 Cellspacing 属性的使用方法。Cellpadding 的意思是单元格的边距，指

的是字与单元格边框的距离。Cellspacing 的意思是单元格间距，指的是单元格之间的距离。

表格参数：colspan 跨列，rowspan 跨行

```
<table width="200" border="1">
  <tr>
    <td>&nbsp;</td>
    <td colspan="2">&nbsp;</td>
  </tr>
  <tr>
    <td rowspan="2">&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
```

div 居中使用"margin:0 auto; width:1000px;"

输出 php 版本信息代码：echo phpinfo();

include 与 require 的区别:

include 和 require 语句是相同的，除了错误处理方面：

require 会生成致命错误（E\_COMPILE\_ERROR）并停止脚本

include 只生成警告（E\_WARNING），并且脚本会继续

include 与 include\_once 的区别：include\_once 导入时会先检测文件是否在该页面的其他部分被引用过

require 与 require\_once 的区别：require\_once 导入时会先检测文件是否在该页面的其他部分被引用过，

require 如果调用的文件找不到就会立刻停止脚本的运行，include 就继续运行，被显示警告。

print 与 printf 的区别：print 输出，printf 语句格式化输出

switch 语句执行时，即使遇到符合要求的 case 语句段，也会继续往下执行，直到 switch 结束，为了避锅免这种浪时间和资源的情况发生，每个 case 语句段后一定要加上 break；

ps: switc 输出今天是星期几

```
<?php
$a=3;
switch($a){
case "1" : echo "1"; break;
case "2" : echo "2"; break;
case "3" : echo "3"; break;
default: echo '参数错误，请重新输入！'; break;
}
```

?>

对关键字进行高亮显示代码：\$str="西游记很好看";

```
$keyword="西游记";
```

```
echo str_replace($keyword,"<font  
color=red><b>$keyword</b></font>",$str);
```

判读闰年：if(!empty(\$\_POST['Button']) && isset(\$\_POST['Button'])){\$

```
$year=$_POST['txt_year'];
```

```
$result =($year%4==0 && $year%100!=0)||($year%400==0)?$year."是闰  
年":$year."不是闰年";
```

ps:

```
<body>
```

```
<table width="392" height="101" border="1" align="center" cellpadding="1"  
cellspacing="1" bordercolor="#009900"
```

```
bgcolor="#CCFF00">
```

```
<tr>
```

```
<td height="25" colspan="2" align="center" class="STYLE5">判断指定的年份是否  
为闰年</td>
```

```
</tr>
```

```
<tr>
```

```
<td width="160" height="30" bgcolor="#FFFF99"><span class="STYLE3">请输  
入一个四位数的年份：</span></td>
```

```
<td width="219" bgcolor="#FFFF99">
```

```
<!--表单-->
```

```
<form name="form1" method="post" action="">
```

```
<table width="196" height="29" border="0" cellpadding="0" cellspacing="0">
```

```
<tr>
```



```
<?php
```

界定符<<<内不允许有添加解析，结束符要在第一列，并且所在的列不能有其他的字符和空格，否则 php 继续运行下去。界定符格式：

```
<<<str
```

格式化文本

```
str
```

```
?>
```

.链接字符串，.=是将右边的参数附加到左边形成一个新的参数。比如：\$a=111; \$a.=222; echo \$a; 运行的结果是 111222

var\_dump(\*);会返回变量的数据类型和值(检查属性类型);ps

```
<?php $a=7.3; var_dump($a); ?>
```

```
<?php $a=7; $b='7'; var_dump($a===$b); ?>
```

```
<?php $a=3; $b=7; var_dump($a>$b); ?>
```

addslashes()自动转义字符串，stripslashes()还原字符串

```
<?php
```

```
$str = "我有一个'梦想'。";
```

```
$a = addslashes($str); //对字符串中的特殊字符进行转义
```

```
echo $a."<br>"; //输出转义后的字符
```

```
$b = stripslashes($a); //对转义后的字符进行还原
```

```
echo $b."<br>"; //将字符原义输出
```

```
?>
```

获取字符串长度 strlen()

```
<?php
    echo strlen("xiao jing"); //输出 9，空格也是一个字符
?>
```

判断注册提交的密码是否少于 6 位：

```
<?php
if(isset($_POST['pwd'])){
    if(!empty($_POST['pwd'])&& strlen($_POST['pwd'])>6){
        echo "注册成功！";
    }else{
        echo "<script>alert('用户的密码不能少于 6 位
');history.back();</script>";
    }
}
?>
```

echo substr("xiaojing",1,3); //截取字符串，1 是开始位置，3 是截取长度(1 和 3 可以改变)

echo substr("xiaojing",-4,5); //-4 表示从后 4 位开始，结果是：jing

```
<?php
```

按字节比较字符串：strcmp 区别大小写，strcasecmp 不区别大小写；

按自然排序法比较：strnatcmp(string str1,string str2)将字符串中的数字部分从大到小排序,再比较大小，返回 boolean；如果相

等，返回 0;如果

str1>str2,返回 1;如果 str1<str2,返回-1;

strnatcasecmp()函数作用与 strnatcmp()作用相似，但是不区分大小写。

指定从源字符串的位置比较：int strncmp(string str1,string str2,int len),len 为必选参数，用来指定每个字符串中参与比较字

符的数量.如果字符

串相等返回 0，str1>str2 返回 1，str1<str2 返回-1；

strstr — 查找字符串的首次出现,该函数区分大小写。如果想要不区分大小写，请使用 stristr()。

int substr\_count(string haystack,string needle)，参数 haystack 是指定的字符串，needle 是指定的字符，substr\_count() 返回

子字符串 needle 在

字符串 haystack 中出现的次数。注意 needle 区分大小写。

example:if(substr\_count(\$user,'傻逼')>0 or substr\_count(\$user,'妈的')>0){echo "您输入的信息中含有敏感关键字";}

显示输入的数值的绝对值经典案例，form 表单提交数据的处理

```
<body>
```

```
<?php
```

```
function abso($num){           //定义函数 abso，参数为输入的数字变量$num
```

```
    if($num>=0){                //如果参数值不是负数
```

```
        return $num;            //返回参数值本身
```

```
    }
```

```
    else{
```

```
        return $num*(-1);        //否则如果是负数则返回参数值乘以-1，
```

即它的绝对值

```
    }
```



```

    }
?>

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

<label>请输入数字：</label>

<input type="text" name="num" id="num" value="<?php if(!empty($_POST['num']))
{echo $_POST['num']; }?>"><br>

<input type="submit" name="sub" value="提交"><p>

</form>

<?php
    if(!empty($_POST['num']) && isset($_POST['num'])) {
        echo "您输入的数字是".$_POST['num'].", ";

        echo $_POST['num']. "的绝对值是".abs($_POST['num']);
    }
?>

</body>

```

弹窗点击返回：<script>alert('成功'); history.back();</script>

**strpos() 函数用于检索字符串内指定的字符或文本的位置。**

```
echo strpos("Hello world!", "world");
```

**应用 str\_replace()函数替换子字符串**

```
<body>

<?php

$str="          予人温暖，自己也更温暖。予人快乐，自己也将更快乐。这是一种境界，
也是心灵上的一种拥有。在温暖里，人才

不会迷失，在相互
```

扶持中，婚姻才会更长久。就算偶有寒流来袭，只要两颗心相互取暖，又怎么会给寂寞留下可乘之机呢。";

```
if(!empty($_POST['str1']) && isset($_POST['str1']) && !empty($_POST['str2']) &&
isset($_POST['str2'])){
```

```
    $str2=$_POST["str2"];

    $str1=$_POST["str1"];

    $xiao= str_replace($str2,$str1,$str);
```

```
}
```

```
?>
```

```
<table width="760" border="0" cellspacing="0" cellpadding="0">
```

```
<tr>
```

```
<td width="415" height="490" align="center" valign="top"
background="images/bg.jpg"><table width="400"
```

```
height="462" border="0"
```

```
cellpadding="0" cellspacing="0">
```

```
<tr>
```

```
<td width="161" height="165">&nbsp;  </td>
```

```
<td width="239">&nbsp;  </td>
```

```
</tr>
```

```
<tr>
```

```
<td height="112" colspan="2" align="center"><table width="360"
border="0" cellspacing="0" cellpadding="0">
```

```
<tr>
```

```
<td width="360" align="left"><span class="STYLE4">原字符串：<br>
```

```
&nbsp;  &nbsp; <?php echo $str;?></span></td>
```

```
</tr>
```

```
</table></td>
```

```
</tr>
```

```
<form name="form1" method="post" action="index.php">
```

```
<tr>
```

```
<td height="32" align="right"><span class="STYLE3">被替换的字符
</span></td>

<td align="left"><input name="str2" type="text" id="str2" size="24"></td>
</tr>
<tr>
<td height="30" align="right"><span class="STYLE3">替换的字符
</span></td>
<td height="30"><input name="str1" type="text" id="str1"
size="24"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td align="center" valign="top"><input name="Submit" type="submit"
id="Submit" value="提交">
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
</tr>
</form>
</table></td>
<td width="345" align="center" valign="middle"
background="images/bg1.jpg"><table width="300" border="0"
cellspacing="0"
cellpadding="0">
<tr>
<td class="STYLE4"><?php if(!empty($_POST["str2"])) &&
isset($_POST["str2"])){ echo "$xiao";} ?></td>
</tr>
</table>
</td>
</tr>
</table>
</body>
```

数组(start)-----

### foreach:遍历数组

有两种语法：

```
foreach (array_expression as $value)
```

```
    statement
```

```
foreach (array_expression as $key => $value)
```

```
    statement
```

案例：

```
<?php
```

```
    $xiao=array("a","b","c");
```

```
    $jing=array("一","二","三");
```

```
    $price=array("10","20","50");
```

```
    print_r($xiao);
```

```
    echo "<br>";
```

```
    print_r($jing);
```

```
    echo "<br>";
```

```
    foreach($xiao as $key=>$value){
```

```
        echo $value;
```

```
        echo $jing[$key];
```

```
        echo "$price[$key]". "<br />";
```

```
    }
```

```
?>
```

### 用 for 语句遍历数组

```
<?php
```

```
    $arr=array("xiao","haining","lanpei");
```

```
$len=count($arr);  
for($i=0;$i<$len;$i++){  
    echo "$arr[$i] ";  
}  
?>
```

遍历关联数组

```
<?php  
$arr=array("xiaojing"=>"萧劲","haining"=>"海宁","lanpei"=>"蓝佩");  
foreach($arr as $key=>$value){  
    echo "key=$key value=$value";  
    echo '<br>';  
}  
?>
```

**explode() : 应用于将字符串转换成数组。** ps:

```
<?php  
$str="xiao,jing,jia,you";  
$strs=explode(",",$str);  
print_r($strs);  
?>
```

**implode():应用于将数组转换为一个新的字符串。** ps:

```
<?php  
$str=array("xiao","jing","jia","you");  
$string=implode(" ",$str);  
print_r($string);  
?>
```

应用 count()函数对数组中的元素个数进行统计(获取数组的长度)。 ps:

```
<?php
    $array=array("PHP","asp","JSP");
    $count=count($array);
    echo $count; //输出结果是：3
?>
```

### PHP - 数组的排序函数

在本节中，我们将学习如下 PHP 数组排序函数：

sort() - 以升序对数组排序

rsort() - 以降序对数组排序

asort() - 根据值，以升序对关联数组进行排序

arsort() - 根据值，以降序对关联数组进行排序

ksort() - 根据键，以升序对关联数组进行排序

krsort() - 根据键，以降序对关联数组进行排序

bool sort ( array &\$array [, int \$sort\_flags = SORT\_REGULAR ] )

sort()：对字符串进行比较

```
<?php
    $arr=array("xiaojing","haining","lanpei");
    sort($arr);
    var_dump($arr);
?>
```

```
<?php  
$array=array(1,4,8,25,0,42,88,66);  
  
$array1=sort($array); //用 sort()函数对$array 数组进行升序排序，并返回新数组  
$array1  
  
for($i=0;$i<count($array);$i++){ //应用 for 循环语句输出数组元素  
  
    echo $array[$i]."        "; //输出结果：0 1 4 8 25 42 66 88  
  
}  
?>
```

```
<?php  
    $array=array(1,4,8.25,0,42,88,66);  
    $array2=sort($array,SORT_STRING);  
    for($i=0;$i<count($array);$i++){  
        echo $array[$i]."&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&"; //输出结果：0 1 25 4 42 66 88  
    }  
?>
```

[illegible]

**关联数组**：数组中下标不是数字的数组，ps:

```
$a=array("x"=>"xiao");
```

ksort():对关联数组的关键字进行排序

```
<?php
    $array = array ("p"=>"php","a"=>"asp","j"=>"jsp");
声明数组
    asort($array);
    print_r($array);
?>
```

asort():对关联数组的元素进行排序

```
<?php
    $array = array ("p"=>1,"a"=>2,"j"=>3);
    asort($array);
    print_r($array);
?>
```

**二维数组**

```
<?php
$cars = array(
    array("电脑",33,20),
    array("手机",17,15),
    array("牙膏",5,2),
    array("单反",15,11));
for($i=0;$i<4;$i++){
    for($j=0;$j<3;$j++){
```



```

        echo $cars[$i][$j]."\t";
    }
    echo "<hr />";
}
?>

```

数组(end)-----

超全局(start)-----

PHP 在名为 \$GLOBALS[index] 的数组中存储了所有全局变量。变量的名字就是数组的键。

```

<?php
    $a=3;$b=7;
    function mytest(){
        $GLOBALS['z']=$GLOBALS['a']+$GLOBALS['b'];
    }
    mytest();

    echo $z; //在上面的例子中，由于 z 是 $GLOBALS 数组中的变量，因此在函数
    之外也可以访问它。
?>

```

\$\_SERVER 这种超全局变量保存关于报头、路径和脚本位置的信息。

```

<?php
    foreach($_SERVER as $key=>$value){
        echo "$key:    ($value)<hr />";
    }
?>

```

超全局(end)-----

**str\_ireplace 说明**

```
mixed str_ireplace ( mixed $search , mixed $replace , mixed $subject [, int  
&$count ] )
```

该函数返回一个字符串或者数组。该字符串或数组是将 subject 中全部的 search 都被 replace 替换（忽略大小写）之后的结果。

如果没有一些特殊的

替换规则，你应该使用该函数替换带有 i 修正符的 preg\_replace() 函数。

```
<?php
```

```
    $str2="xiaojing";
```

```
    $str1="追求自我";
```

```
    $str="xiaojing 是走向成功的一条捷径";
```

```
    echo str_ireplace($str2,$str1,$str); //输出的结果是：追求自我是走向成功的一  
    条捷径
```

```
?>
```

gettype() 用来取得变量的类型。返回的类型字符串可能为下列字符串其中之一：

integer、double、string、array、object、

unknown type

is\_numeric ( mixed var ): //检验测定变量是不是为数码或数码字符串

is\_bool(): //检验测定变量是不是是布尔型

is\_float(): //检验测定变量是不是是浮点型 和 is\_double , is\_real() 都同样的用法

is\_int(): //检验测定变量是不是是整儿 is\_integer() 同样的用法

is\_string(): //检验测定变量是不是是字符串

is\_object(): //检验测定变量是不是是一个对象

is\_array(): //检验测定变量是不是是数组

is\_null(): //检验测定变量是不是为 灭荏 值是不是是灭荏巨细写敏感

```
bool is_bool ( mixed $var )
```

如果 var 是 boolean 则返回 TRUE。

Example #1 is\_bool() 示例

```
$a = false;
```

```
$b = 0;
```

```
// 因为 $a 是布尔型，所以结果为真
```

```
if (is_bool($a)) {
```

```
    print "Yes, this is a boolean";
```

```
}
```

```
// 因为 $b 不是布尔型，所以结果为非真
```

```
if (is_bool($b)) {
```

```
    print "Yes, this is a boolean";
```

```
}
```

```
?>
```

trim()函数用于去除字符串左右两边的空白字符和特殊字符，

ltrim()函数去除字符串左边的空白字符和特殊字符，

rtrim()函数去除字符串中右边的空白字符和特殊字符。

```
ps1:
```

```
<?php
```

```
$str=" (:@_@ 有一条路走过了总会想起！ @_@:) ";
```

```
echo ltrim($str);
```

```
//去除字符串左边的空白字符
```

```
echo "<br>"; //执行换行

echo ltrim($str, "(:@_@ "); //去除字符串左边的特殊字符@_@:)

?>
```

```
ps2:

<?php

$str="(:@_@ 有一段感情经历了就再难忘记! @_@:) ";

echo rtrim($str); //去除字符串右边的空白字符

echo "<br>"; //执行换行

echo rtrim($str, "@_@:)"); //去除字符串右边的特殊字符@_@:)

?>
```

```
ps3:

<?php

$str="\r\r(:@_@有一条路走过了总会想起! 有一段感情经历了就再难忘记! @_@:)
";

echo trim($str); //去除字符串左右两边的
空白字符

echo "<br>"; //执行换行

echo trim($str, "\r\r(:@_@ @_@:)"); //去除字符串左右两边的空白字符
和特殊字符\r\r(: :)

?>
```

number\_format:以千位分隔符方式格式化一个数字,number 为格式化后的字符串, 该函数可以有一个, 两个, 四个, 但不能是三个。

```
<?PHP

$number=2668.764;

echo number_format($number);
```

```

echo " <br> ";
echo number_format($number,2);
echo " <br> ";
$number2=123456.7832;
echo number_format($number2,2,"*", "*");
?>

```

**array explode(string separator,string str,[int limit]):按照指定的规则对一个字符串进行分割，返回值为数组。**

print\_r()显示关于一个变量的易于理解的信息。如果给出的是 string、integer 或 float，将打印变量值本身。如果给出的是

array，将会按照一定

格式显示键和元素。object 与数组类似。

记住，**print\_r()将把数组的指针移到最后边。使用 reset()可让指针回到开始处。**

ps:

```

<?php

    $str="品牌电脑@品牌手机@男士衬衫@女士挎包";

    $str_arr=explode("@",$str);

    print_r($str_arr);

?>

```

输出为：Array ( [0] => 品牌电脑 [1] => 品牌手机 [2] => 男士衬衫 [3] => 女士挎包 )

**htmlentities:把字符转换为 HTML 实体。**

ps:

```

<?php

    $a="<table>

        <tr><td>sljo</td>

        </tr>

```

```

        <tr><td>xiao</td><td>jing</td></tr>
    </table>";
    echo htmlentities($a);
?>

```

输出结果为：数组\$a 双引号里的内容：

```

<table><tr><td>sljo</td><tr><td>xiao</td><td>jing</td></tr></table>

```

### preg\_match — 执行一个正则表达式匹配

```

<?php
//模式分隔符后的"i"标记这是一个大小写不敏感的搜索
if (preg_match("/php/i", "PHP is the web scripting language of choice.")) {
    echo "A match was found.";
} else {
    echo "A match was not found.";
}
?>

```

正则表达式(start)-----  
-----

### 正则表达式语法规则：

**行定位符：**“^”表示行的开始；“\$”表示行的结尾。ps:^mr，匹配mr开头的字符串，例如mrsoft,mrbook.

**字符类：**正则表达式区分大小写，如果想要忽略大小写，可以通过“i”来完成，但一个[]只能匹配一个字符。ps：不分大小写输入

abc:[Aa][Bb][Cc]

**选择字符**：“|”该字符可以理解为“或”的意思，ps:不分大小写输出 abc : (a | b | c)

**连字符**：“-”表示字符的范围，ps:[a-zA-Z]表示所有字母的大小

**排除字符**：“^”这个字符在开头是表示行的开始，如果是在中间出现是“非”的意思。

ps:[^a-zA-Z]的意思是匹配非字母字符串

**限定符**：限定符用来指定正则表达式的一个给定组件必须要出现多少次才能满足匹配。有\*或+或?或{n}或{n,}或{n,m}

共 6 种。

\*、+和?限定符都是贪婪的，因为它们会尽可能多的匹配文字，只有在它们的后面加上一个?就可以实现非贪婪或最小匹配。

正则表达式的限定符有：

字符 描述

**\*** 匹配前面的子表达式零次或多次。例如，zo\* 能匹配 "z" 以及 "zoo"。\* 等价于 {0,}。

**+** 匹配前面的子表达式一次或多次。例如，'zo+' 能匹配 "zo" 以及 "zoo"，但不能匹配 "z"。+ 等价于 {1,}。

**?** 匹配前面的子表达式零次或一次。例如，"do(es)?" 可以匹配 "do" 或 "does" 中的 "do"。? 等价于 {0,1}。

**{n}** n 是一个非负整数。匹配确定的 n 次。例如，'o{2}' 不能匹配 "Bob" 中的 'o'，但是能匹配 "food" 中的两个 o。

**{n,}** n 是一个非负整数。至少匹配 n 次。例如，'o{2,}' 不能匹配 "Bob" 中的 'o'，但能匹配 "fooooood" 中的所有

o。'o{1,}' 等价于

'o+'。'o{0,}' 则等价于 'o\*'。

**{n,m}** m 和 n 均为非负整数，其中 n <= m。最少匹配 n 次且最多匹配 m 次。例如，"o{1,3}" 将匹配 "fooooood" 中的前

三个 o。'o{0,1}'

等价于 'o?'。请注意在逗号和两个数之间不能有空格。

## 非打印字符 字符 含义

`\cx` 匹配由 `x` 指明的控制字符。例如，`\cM` 匹配一个 Control-M 或回车符。`x` 的值必须为 A-Z 或 a-z 之一。否则，将 `c` 视

为一个原义的 'c'

字符。

`\f` 匹配一个换页符。等价于 `\x0c` 和 `\cL`。

`\n` 匹配一个换行符。等价于 `\x0a` 和 `\cJ`。

`\r` 匹配一个回车符。等价于 `\x0d` 和 `\cM`。

`\s` 匹配任何空白字符，包括空格、制表符、换页符等等。等价于 `[ \f\n\r\t\v]`。

`\S` 匹配任何非空白字符。等价于 `[^ \f\n\r\t\v]`。

`\t` 匹配一个制表符。等价于 `\x09` 和 `\cI`。

`\v` 匹配一个垂直制表符。等价于 `\x0b` 和 `\cK`。

## 特别字符 说明

`$` 匹配输入字符串的结尾位置。如果设置了 `RegExp` 对象的 `Multiline` 属性，则 `$` 也匹配 `'\n'` 或 `'\r'`。要匹配 `$` 字符本身，请

使用 `\$`。

`()` 标记一个子表达式的开始和结束位置。子表达式可以获取供以后使用。要匹配这些字符，请使用 `\(` 和 `\)`。

`*` 匹配前面的子表达式零次或多次。要匹配 `*` 字符，请使用 `\*`。

`+` 匹配前面的子表达式一次或多次。要匹配 `+` 字符，请使用 `\+`。



. 匹配除换行符 `\n` 之外的任何单字符。要匹配 `.`，请使用 `\.`。

[ 标记一个中括号表达式的开始。要匹配 `[`，请使用 `\[`。

? 匹配前面的子表达式零次或一次，或指明一个非贪婪限定符。要匹配 `?` 字符，请使用 `\?`。

\ 将下一个字符标记为或特殊字符、或原义字符、或向后引用、或八进制转义符。例如，  
'`n`' 匹配字符 '`n`'。'`\n`' 匹配换行符。序列 '`\\`' 匹配 "`\`"，"`\'`" 则匹配 "`"`"。

^ 匹配输入字符串的开始位置，除非在方括号表达式中使用，此时它表示不接受该字符集合。要匹配 `^` 字符本身，请使用 `\^`。

{ 标记限定符表达式的开始。要匹配 `{`，请使用 `\{`。

| 指明两项之间的一个选择。要匹配 `|`，请使用 `\|`。

### 反斜杠指定的预义字符集

/d 任意一个十进制数字，相当于 `[0-9]`

/D 任意一个非十进制数字

/s 任意一个空白字符（空格，换行符，换页符，回车符，水平制表符），相当于 `[\f\n\r]`

/S 任意一个

/w 任意一个

/W 任意一个

正则表达式(end)-----  
-

PHP 表单(start)(经典)-----  
-

index.php 文件，共一个文件

<?php

```
// 定义变量并设置为空值

$name = $email = $gender = $comment = $website = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = test_input($_POST["name"]);
    $email = test_input($_POST["email"]);
    $website = test_input($_POST["website"]);
    $comment = test_input($_POST["comment"]);
    $gender = test_input($_POST["gender"]);
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

?>
```

```
<?php
```

```
// 定义变量并设置为空值

$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "名字不能为空！";
    } else {
        $name = test_input($_POST["name"]);

        // 检查名字是否包含字母和空格
        if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
            $nameErr = "名字格式为字母或空格！";
        }
    }
}
```

```
}  
}
```

```
if (empty($_POST["email"])) {  
    $emailErr = "电子邮箱不能为空！";  
} else {  
    $email = test_input($_POST["email"]);  
    // 检查电邮地址语法是否有效  
    if (!preg_match("/([w\-\]+\@[w\-\]+\.[w\-\]+)/",$email)) {  
        $emailErr = "电子邮箱格式有误！";  
    }  
}
```

```
if (empty($_POST["website"])) {  
    $website = "";  
} else {  
    $website = test_input($_POST["website"]);  
    // 检查 URL 地址语言是否有效（此正则表达式同样允许 URL 中的下划线）  
    if (!preg_match("/\b(?:?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?~_!:,.;]*[-a-z0-9+&@#\/%  
    =~_!]/i",$website)) {  
        $websiteErr = "无效的地址格式！";  
    }  
}
```

```
if (empty($_POST["comment"])) {  
    $comment = "";  
} else {  
    $comment = test_input($_POST["comment"]);
```

```

}

if (empty($_POST["gender"])) {
    $genderErr = "性别不能为空！";
} else {
    $gender = test_input($_POST["gender"]);
}
}
?>

<form name="form1" method="POST" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']); ?>" >

姓名：<input type="text" name="name" value="<?php echo $name;?>" />

<span class="error"><font color='red'>*</font><?php echo
$nameErr;?></span><br><br>

电邮：<input type="text" name="email" value="<?php echo $email; ?>" />

<span class="error"><font color='red'>*</font><?php echo
$emailErr;?></span><br><br>

网址：<input type="text" name="website" value="<?php echo $website; ?>"/>

<span class="error"><?php echo $websiteErr;?></span><br><br>

评论：<textarea name="comment" rows="4" cols="20"><?php echo
$comment; ?></textarea><br><br>

性别：<input type="radio" name="gender" checked>男

        <input type="radio" name="gender">女

<span class="error"><font color='red'>*</font><?php echo
$genderErr;?><br><br>

```

```
<input type="submit" name="submit" value="提交">&nbsp;&nbsp;&nbsp;<input  
type="reset" name="reset" value="清除" />  
</form>
```

PHP 表单(end)(经典)-----

date()(start)-----

星期一 Monday

星期二 Tuesday

星期三 Wednesday

星期四 Thursday

星期五 Friday

星期六 Saturday

星期天 Sunday

一月 : January

二月 : February

三月 : March

四月 : April

五月 : May

六月 : June

七月 : July

八月 : August

九月 : September

十月 : October

十一月：November

十二月：December

获取当前时间戳

```
<?php
```

```
    echo "时间戳：" . time() . "<br>";
```

//返回当前的时间

戳

```
    echo "日期：" . date("Y-m-d", time()) . "<br>";
```

//使用 date 函数输

出格式化后的日期

```
    echo "时间：" . date("H:i:s", time());
```

//使用 date

函数输出格式化后的时间

```
?>
```

输出下个星期日期：

```
<?php
```

```
    date_default_timezone_set("Asia/Shanghai");
```

```
    $nextweek=time()+(7*24*60*60);
```

```
    echo time()."<br>";
```

```
    echo "Now:".date("Y-m-d")."<br>";
```

```
    echo "nwxtweek:".date("Y-m-d",$nextweek);
```

```
?>
```

将英文文本的日期时间描述解析为 UNIX 时间戳。

```
<?php
```

```
    date_default_timezone_set("Asia/Shanghai");
```

```
    echo strtotime("now"),"\r";
```

```

echo "输出时间:".date("Y-m-d H:i:s",strtotime("now")), "<br>";
echo strtotime("27 February 2013"), "\n";
echo "输出时间:".date("Y-m-d H:i:s",strtotime("27 February 2013")), "<br>";
echo strtotime("+3 day"), "\n";
echo "输出时间:".date("Y-m-d H:i:s",strtotime("+3 day")), "<br>";
echo strtotime("+1 week")."<br>";
echo strtotime("+1 week 2 days 3hours 4seconds")."<br>";
echo strtotime("next Thursday")."<br>";
echo strtotime("last Monday"), "\n";
?>

```

格式字串可以识别以下 format 参数的字符串

format 字符

d 月份中的第几天，有前导零的 2 位数字 01 到 31

D 星期中的第几天，文本表示，3 个字母 Mon 到 Sun

j 月份中的第几天，没有前导零 1 到 31

l ("L"的小写字母) 星期几，完整的文本格式 Sunday 到 Saturday

N ISO-8601 格式数字表示的星期中的第几天 (PHP 5.1.0 新加) 1 (表示星期一) 到 7 (表示星期天)

S 每月天数后面的英文后缀，2 个字符 st, nd, rd 或者 th。可以和 j 一起用

w 星期中的第几天，数字表示 0 (表示星期天) 到 6 (表示星期六)

z 年份中的第几天 0 到 365

星期 --- ---

W ISO-8601 格式年份中的第几周，每周从星期一开始 (PHP 4.1.0 新加的) 例如：  
42 (当年的第 42 周)

月 --- ---

F 月份，完整的文本格式，例如 January 或者 March January 到 December

m 数字表示的月份，有前导零 01 到 12

M 三个字母缩写表示的月份 Jan 到 Dec

n 数字表示的月份，没有前导零 1 到 12

t 给定月份所应有的天数 28 到 31

年 --- ---

L 是否为闰年 如果是闰年为 1，否则为 0

o ISO-8601 格式年份数字。这和 Y 的值相同，只除了如果 ISO 的星期数 (W) 属于前一年或下一年，则用那一年。( PHP 5.1.0 新

加) Examples:

1999 or 2003

Y 4 位数字完整表示的年份 例如：1999 或 2003

y 2 位数字表示的年份 例如：99 或 03

时间 --- ---

a 小写的上午和下午值 am 或 pm

A 大写的上午和下午值 AM 或 PM

B Swatch Internet 标准时 000 到 999

g 小时，12 小时格式，没有前导零 1 到 12

G 小时，24 小时格式，没有前导零 0 到 23

h 小时，12 小时格式，有前导零 01 到 12

H 小时，24 小时格式，有前导零 00 到 23

i 有前导零的分钟数 00 到 59>

s 秒数，有前导零 00 到 59>



u 毫秒 ( PHP 5.2.2 新加 ) 。需要注意的是 date() 函数总是返回 000000 因为它只接受 integer 参数 , 而 DateTime::format()

才支持毫秒。 示

例: 654321

时区 --- ---

e 时区标识 ( PHP 5.1.0 新加 ) 例如 : UTC , GMT , Atlantic/Azores

I 是否为夏令时 如果是夏令时为 1 , 否则为 0

O 与格林威治时间相差的小时数 例如 : +0200

P 与格林威治时间 ( GMT ) 的差别 , 小时和分钟之间有冒号分隔 ( PHP 5.1.3 新加 )

例如 : +02:00

T 本机所在的时区 例如 : EST , MDT ( 【译者注】在 Windows 下为完整文本格式 , 例如 "Eastern Standard Time" , 中文版会显示 "中

国标准时间" ) 。

Z 时差偏移量的秒数。UTC 西边的时区偏移量总是负的 , UTC 东边的时区偏移量总是正的。 -43200 到 43200

完整的日期 / 时间 --- ---

c ISO 8601 格式的日期 ( PHP 5 新加 ) 2004-02-12T15:19:21+00:00

r RFC 822 格式的日期 例如 : Thu, 21 Dec 2000 16:01:07 +0200

U 从 Unix 纪元 ( January 1 1970 00:00:00 GMT ) 开始至今的秒数 参见 time()

**getdate — 取得日期 / 时间信息**.ps

```
<?php
    date_default_timezone_set("Asia/Shanghai");
    $arr=getdate();
    print_r($arr);
?>
```

输出结果：

```
Array (  
[seconds] => 38  
[minutes] => 37  
[hours] => 20  
[mday] => 5  
[wday] => 6  
[mon] => 12  
[year] => 2015  
[yday] => 338  
[weekday] => Saturday  
[month] => December  
[0] => 1449319058 )
```

**setlocale — 设置地区信息。** ps:

```
<?php  
  
    setlocale(LC_ALL,"en_US");                                //设置为美国本地  
环境  
  
    echo "美国格式：".strftime("Today is %A");  
    echo "<p>";  
  
    setlocale(LC_ALL,"chs");                                    //设置为简体中文  
  
    echo "中文简体格式：".strftime("今天是%A");  
    echo "<p>";  
  
    setlocale(LC_ALL,"cht");                                    //设置为繁体中文  
  
    echo "<p>";  
  
    echo "繁体中文格式：".strftime("今天是%A");  
  
?>
```

**checkdate** — 验证一个格里高里日期,检查由参数构成的日期的合法性。如果每个参数都正确定义了则会被认为是有效的。 ps:

```
<?php
var_dump(checkdate(12, 31, 2000));
var_dump(checkdate(2, 29, 2001));
?>
```

**mktime()** 函数返回日期的 Unix 时间戳,Unix 时间戳包含 Unix 纪元 ( 1970 年 1 月 1 日 00:00:00GMT ) 与指定时间之间的秒数。

mktime(hour,minute,second,month,day,year)

```
<?php
$d=mktime(9, 12, 31, 6, 10, 2015);
echo "创建日期是 " . date("Y-m-d h:i:sa", $d);
?>
```

**strtotime** — 将任何英文文本的日期时间描述解析为 Unix 时间戳

```
<?php
$d=strtotime("10:38pm April 15 2015");
echo "创建日期是 " . date("Y-m-d h:i:sa", $d);
?>
```

```
<?php
$d=strtotime("tomorrow");
echo date("Y-m-d h:i:sa", $d) . "<br>";
```

```
$d=strtotime("next Saturday");
echo date("Y-m-d h:i:sa", $d) . "<br>";
```

```
$d=strtotime("+3 Months");  
echo date("Y-m-d h:i:sa", $d) . "<br>";  
?>
```

PHP 提示 - 自动版权年份,使用 date() 函数在您的网站上自动更新版本年份：

```
? 2010-<?php echo date("Y")?>
```

date()(end)-----

phpmyadmin

使用 SQL 语句删除数据库：delete from db\_xiaojingtest where name="萧劲"

phpmyadmin 的界面，创建数据库编码格式一般选择 utf8\_general\_ci

添加数据：选择数据库》》结构》》表》》选择“插入”

修改数据：选择数据库》》选择数据表》》浏览》》选择“编辑”，如果没有“编辑”选项，那是字段没有设主键。

1 ) VARCHAR，文本类型

2 ) INT，整数类型

3 ) FLOAT，浮点数类型

4 ) DATE，日期型

选择 INT 类型，在后面的额外中选择 auto\_increment 就可以自动添加 id

如果您还不能察觉 \$exec 有什么错误的话，可以复制这个 sql 语句到 phpmyadmin 中执行，看看它的出错信息。

php 与 mysql 编码问题乱码问题（ phpMyAdmin 编码乱码问题 ），网页设为为：utf-8, 保存时选择 utf-8，但连接数据库一定要用 utf8,因

为 phpmyadmin 不能识别 utf-8。

phpMyAdmin 中创建数据库时选择 utf8\_general\_ci

mysql 中使用：mysql\_query 设 utf8。ps;("set names gbk"); 改成 mysql\_query("set names utf8");

PDO 中使用：\$pdo->query('set names utf8');

或者用 \$pdo->exec('set character set utf8');

PHP 操作 MySQL 数据库的过程分为 5 步：

1 连接 MySQL 服务器

2 选择数据库

3 执行 SQL 语句

4 关闭结果集

5 断开与 MySQL 服务器的连接

mysql\_connect — 打开一个到 MySQL 服务器的连接.ps:

连接数据库(判断数据库是否连接成功)

```
<?php
```

```
    $host="127.0.0.1"; //MySQL 服务器地址
```

```
    $userName="root"; //用户名
```

```
    $password=""; //密码
```

```
    if($connID= mysql_connect($host,$userName,$password)){
```

```
        //建立与 MySQL 数据库的连接，并弹出提示对话框
```

```

        echo "<script language='javascript'>alert('数据库连接成功
');</script>";
    }else{
        echo "<script language='javascript'>alert('数据库连接失败
');</script>";
    }
?>

```

mysql\_select\_db — 选择 MySQL 数据库。 ps

判断数据库是否选择成功！

```

<?php
    $host="127.0.0.1"; //MySQL 服务器地址

    $userName="root"; //用户名

    $password=""; //密码

    $xiao="$xiaojing"; //数据库

    $connID=mysql_connect($host,$userName,$password); //建立与 mysql 数
数据库服务器的连接

    if(mysql_select_db($xiao,$connID)){ //选择数据
库

        echo "数据库选择成功！";
    }else{
        echo "数据库选择失败！";
    }

?>

```

mysql\_query — 发送一条 MySQL 查询

```

<?php

```

```

$host = "localhost";
$username = "root";
$password = "";
$dbName = "xiaojing";

$connID = mysql_connect($host, $username, $password);
mysql_select_db($dbName, $connID);
mysql_query("set names utf8");
echo "<table border=\"1px\" align=\"center\">
    <tr>
        <td>a</td>
        <td>b</td>
        <td>c</td>
        <td>d</td>
        <td>e</td>
        <td>f</td>
    </tr>";

$query = mysql_query("select sno, sname, class, chinese, math,english from
tb_student", $connID);
while($result = mysql_fetch_array($query))
{
    echo "<tr>
        <td>".$result["sno"]."</td>
        <td>".$result["sname"]."</td>
        <td>".$result["class"]."</td>
        <td>".$result["chinese"]."</td>
        <td>".$result["math"]."</td>
        <td>".$result["english"]."</td>
    </tr>";
}

```

```
echo "</table>";  
mysql_close($connID);  
?>
```

mysql\_fetch\_array()函数获取结果集。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
<title>无标题文档</title>  
</head>  
  
<body>  
  
<?php  
$host = "127.0.0.1";  
$userName = "root";  
$password = "";  
$dbName = "db_database14";  
  
$connID = mysql_connect($host, $userName, $password);  
mysql_select_db($dbName, $connID);  
mysql_query("set names utf8");  
echo "<table border=\"1px\" align=\"center\">  
    <tr>  
        <td>学号</td>  
        <td>姓名</td>  
        <td>班级</td>
```



```

        <td>语文</td>

        <td>数学</td>

        <td>英语</td>

    </tr>";

$query = mysql_query("select sno, sname, class, chinese, math ,english from
tb_student", $connID);
while($result = mysql_fetch_array($query))
    echo "<tr>

        <td>".$result["sno"]."</td>

        <td>".$result["sname"]."</td>

        <td>".$result["class"]."</td>

        <td>".$result["chinese"]."</td>

        <td>".$result["math"]."</td>

        <td>".$result["english"]."</td>

    </tr>";

echo "</table>";
mysql_close($connID);
?>

</body>
</html>

```

**mysql\_close():断开与数据库的连接（关闭连接）**

```

<?php

$host = "127.0.0.1";    //MySQL 数据库服务器

$username = "root";    //用户名

$password = "111";    //密码

```

```

$dbName = "db_database13"; //数据名

$connID = mysql_connect($host, $userName, $password); //连接 MySQL 数据库

mysql_select_db($dbName, $connID); //选择数据库

mysql_close($connID); //断开与数据库的连接

mysql_select_db($dbName, $connID); //

?>

```

PDO(start)-----

```

<?php

    $dbms='mysql';

    $host='localhost';

    $user='root';

    $pwd="";

    $dbName="";

    $dsn="$dbms:host=$host;dbname=$dbName";

    try{

        $pdo= new PDO($dsn,$user,$pwd);

        echo 'PDO 连接成功!!!!';

    }catch(exception $e){

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

    }

?>

```

PDO 添加数据,用 exec()

```

<html>

<head>

    <title>xiaojingtest</title>

```

```

        <meta http-equiv="Content-Type" content="text/html; charset=utf8" />
</head>
<body>

    <form name="form" method="POST" action="index.php">

        姓名：<input type="text" name="user" /><br />

        成绩：<input type="text" name="grade" /><br />

        <input type="submit" name="submit" value="插入"
/>&nbsp;&nbsp; <input type="reset" name="reset"
        value="清除">

    </form>

</body>
</html>
<?php
    if(!empty($_POST['submit']) && isset($_POST['submit'])){
        if(($_POST['user'])==""){
            echo '姓名不能为空'.<br>;
        }
        if(($_POST['grade'])==""){
            echo '成绩不能为空'.<br>;
        }
        if((($_POST['user'])!=" && ($_POST['grade'])!=")){
            $dbms='mysql';
            $host='localhost';
            $user='root';
            $pass="";
            $dbName='xiaojingtest';
            $dsn="$dbms:host=$host;dbname=$dbName";

```

```

try{

    $pdo=new PDO($dsn,$user,$pass);
    $pdo->query('set names utf8');
    $query="insert into

student(user,grade)values('".$_POST['user']. "','".$_POST['grade']. "')";
    $result=$pdo->exec($query);
    echo "数据添加成功，受影响的行数为：".$result;

}catch(PDOException $e){
    die("Error!: ".$e->getMessage()."<br>" );
}
}
}
?>

```

PDO:用预处理语句 prepare()和 execute()添加数据（注：多次使用 SQL，推荐使用 PDO::prepaare）

```

<html>
<head>
    <title>xiaojingtest</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf8" />
</head>
<body>

    <form name="form" method="POST" action="index.php">
        姓名：<input type="text" name="user" /><br />
        成绩：<input type="text" name="grade" /><br />
    </form>

```

```
        <input type="submit" name="submit" value="插入"
/>&nbsp;&nbsp;&nbsp;<input type="reset" name="reset"
        value="清除">
    </form>
```

```
</body>
```

```
</html>
```

```
<?php
```

```
    if(!empty($_POST['submit']) && isset($_POST['submit'])){
        if(($_POST['user'])==""){
            echo '姓名不能为空'.<br>';
        }
        if(($_POST['grade'])==""){
            echo '成绩不能为空'.<br>';
        }
        if(($_POST['user'])!=" && ($_POST['grade'])!="){
            try{

                $pdo=new
PDO("mysql:host=localhost;dbname=xiaojingtest",'root','");

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

                $query="insert into

student(user,grade)values('".$_POST['user']. "','".$_POST['grade']. "')";

                $result=$pdo->prepare($query);

                if($result->execute()){

                    echo "数据添加成功！";

                }else{

                    echo "数据添加失败！";

                }
            }
```

```

        }catch(PDOException $e){
            die("Error!: ".$e->getMessage()."<br>" );
        }
    }
}
?>

```

PDO 查询数据,query()返回执行查询后的结果集,(注:使用一次 SQL 语句,推荐使用 PDO::query)

```

<?php
    try{
        $pdo=new PDO("mysql:host=localhost;dbname=xiaojingtest",'root','');
        $pdo->query('set names utf8');
        $query="select * from student limit 5";
        $result=$pdo->query($query);
        foreach($result as $home){
            echo '<p><p>';
            echo $home['id']."\t";
            echo $home['user']."\t";
            echo $home['grade']."\t";
        }
    }catch(PDOException $e){
        die("Error!:".$e->getMessage().'\t<br>');
    }
}
?>

```

PDO 查询数据:预处理语句-prepare()方法做查询的准备工作,和 execute()执行查询。

PDO:fetch 方法获取结果集中下一行的数据库。

```

<?php
    try{

```

```

        $pdo=new PDO("mysql:host=localhost;dbname=xiaojingtest",'root','');
        $pdo->query('set names utf8');
        $query="select * from student limit 10";
        $result=$pdo->prepare($query);
        $result->execute();

        while($res=$result->fetch(PDO::FETCH_ASSOC)){ //fetch 方法获取结果集中下一行的数据库

            echo $res['id'].' ';
            echo $res['user'].' ';
            echo $res['grade'].' ';
            echo "<p>";

        }
    }catch(PDOException $e){
        die("Error!:".$e->getMessage()."<br>");
    }
?>

```

**PDO:fetchAll()方法用于获取结果集中的所有行**

```

<?php
    try{
        $pdo=new PDO("mysql:host=localhost;dbname=xiaojingtest",'root','');
        $pdo->query('set names utf8');
        $query="select * from student";
        $result=$pdo->prepare($query);
        $result->execute();
        $res=$result->fetchAll(PDO::FETCH_ASSOC);
        for($i=0;$i<count($res);$i++){
            echo $res[$i]['id'];
            echo $res[$i]['user'];
            echo $res[$i]['grade'];
        }
    }
}

```

```

        echo "<p>";
    }
}catch(PDOException $e){
    die("Error!:".$e->getMessage().'<br>');
}
?>

```

PDO:fetchColumn()方法获取结果集中下一行指定列的值

```

<?php
    try{
        $pdo=new PDO("mysql:host=localhost;dbname=xiaojingtest",'root','');
        $pdo->query('set names utf8');
        $query="select * from student";
        $result=$pdo->prepare($query);
        $result->execute();
        echo $result->fetchColumn(2);
        echo "<br>";
        echo $result->fetchColumn(2);
        echo "<br>";
        echo $result->fetchColumn(2);
        echo "<br>";
        echo $result->fetchColumn(2);
    }catch(PDOException $e){
        die("Error!:".$e->getMessage().'<br>');
    }
?>

```

PDO:errorCode()方法用于获取操作数据库句柄时所产生的错误代码，(注：推荐使用

PDO:使用警告模式--PDO::ERRMODE\_WARNING)

```

<?php

```



```

try{
    $pdo=new PDO("mysql:host=localhost;dbname=xiaojingtest",'root','');
    $pdo->query('set names utf8');
    $query="select * from student limit 10"; //可以故意把数据库名写错看
一下效果

    $result=$pdo->query($query);
    echo 'errorCode 为 : '.$pdo->errorCode();
    foreach($result as $home){
        echo '<p><p>';
        echo $home['id']."\t";
        echo $home['user']."\t";
        echo $home['grade']."\t";
    }
}catch(PDOException $e){
    die("Error!:".$e->getMessage().'\<br>');
}

?>

```

PDO:errorinfo()方法用于获取操作数据库句柄时所产生的错代码，(注：推荐使用 PDO: 使用警告模式--PDO::ERRMODE\_WARNING)

```

<?php
try{
    $pdo=new PDO("mysql:host=localhost;dbname=xiaojingtest",'root','');
    $pdo->query('set names utf8');
    $query="select * from student limit 5";
    $result=$pdo->query($query);
    print_r($pdo->errorinfo()); //可以故意把数据库名写错看一下效果
    foreach($result as $home){
        echo '<p><p>';
    }
}

```

```

        echo $home['id']."\t";
        echo $home['user']."\t";
        echo $home['grade']."\t";
    }
} catch(PDOException $e){
    die("Error!:".$e->getMessage()."<br>");
}
?>

```

**PDO:使用警告模式--PDO::ERRMODE\_WARNING** , (注:推荐使用 PDO:使用警告模式 --PDO::ERRMODE\_WARNING)

```

<?php
    try{
        $pdo=new PDO("mysql:host=localhost;dbname=xiaojingtest",'root','');
        $pdo->query('set names utf8');
        $pdo->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_WARNING);
        $query="select * from student limit 5";
        $result=$pdo->prepare($query);
        $result->execute();
        while($res=$result->fetch(PDO::FETCH_ASSOC)){
            echo "xiajing"." ";
            echo $res['id'].' ';
            echo $res['user'].' ';
            echo $res['grade'].' ';
            echo "<p>";
        }
    } catch(PDOException $e){
        die("Error!:".$e->getMessage()."<br>");
    }
}

```

?>

PDO:使用异常模式--PDO::ERRMODE\_EXCEPTION , (注：推荐使用 PDO:使用警告模式--PDO::ERRMODE\_WARNING)

```
<?php
    try{
        $pdo=new PDO("mysql:host=localhost;dbname=xiaojingtest",'root','');
        $pdo->query('set names utf8');
        $pdo-
>setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
        $query="select * from students limit 20";
        $result=$pdo->prepare($query);
        $result->execute();

        while($res=$result->fetch(PDO::FETCH_ASSOC)){ //fetch 方法获取结
果集中下一行的数据库

            echo $res['id'].' ';
            echo $res['user'].' ';
            echo $res['grade'].' ';
            echo "<p>";

        }
    }catch (PDOException $e) {
        echo 'PDO Exception Caught.';
        echo 'Error with the database:<br/>';
        echo 'SQL Query: '.$query;
        echo '<pre>';

        echo "Error: " . $e->getMessage(). "<br/>";
        echo "Code: " . $e->getCode(). "<br/>";
        echo "File: " . $e->getFile(). "<br/>";
        echo "Line: " . $e->getLine(). "<br/>";
        echo "Trace: " . $e->getTraceAsString(). "<br/>";
```

```
        echo '</pre>';
    }
?>
```

PDO(end)-----

:hover 选择器用于选择鼠标指针浮动在上面的元素。

Cookie(start)-----

**创建 Cookie:setcookie(name, value, expire, path, domain);**

cookie 有设时间的，在时间到达之后会自动删除，没设时间的，会在关闭浏览器之后

ps:<?php setcookie("cookieName","cookievalue",time()+60); ?>

读取 cookie

ps1:<?php echo \$\_COOKIE["cookieName"]; ?>

ps2:<?php print\_r(\$\_COOKIE); ?>

**用 isset()函数检测 Cookie 是否存在**

```
if (isset($_COOKIE["user"]))
    echo "Welcome " . $_COOKIE["user"] . "!"<br />";
else
    echo "Welcome guest!"<br />";
?>
```

**当删除 cookie 时，您应当使过期日期变更为过去的时间点。**

<?php setcookie("cookieName","",time()-60); ?>

Cookie(end)-----

Session(start)-----

启动 Session:<?php session\_start(); ?>

存储 Session 变量:

```
<?php
    session_start();
    $_SESSION["xiao"]="xiaojingtest";
?>
```

读取 Session 变量:

```
<?php
    session_start();
    $_SESSION["xiao"]="xiaojingtest";
    echo $_SESSION["xiao"];
?>
```

可以用 isset 判断 Session 是否已定义, 用 empty 判断 Session 是否为空

删除 Session

删除单个会话: unset(\$\_SESSION["name"]);

删除多个会话: \$\_SESSION=array();

结束会话: session\_destroy();

Session(end)-----

mail("收件人地址","主题","正文","From:发件人\r\nReply-to:发件人的地址");

<?php

\$Name = "Da Duder"; //senders name

\$email = "email@adress.com"; //senders e-mail adress

\$recipient = "PersonWhoGetsIt@emailadress.com"; //recipient

\$mail\_body = "The text for the mail..."; //mail body

\$subject = "Subject for reviever"; //subject

```
$header = "From: ". $Name . " <" . $email . ">\r\n"; //optional headerfields  
mail($recipient, $subject, $mail_body, $header); //mail command :)  
?>
```

### 定义类，实类化类

```
<?php  
    class Student{  
        private $name;  
        private $age;  
  
        public function __construct($name,$age){  
  
            $this->name=$name;  
            $this->age=$age;  
        }  
        public function getNameAndAge(){  
            return "学生".$this->name."今年".$this->age."周岁";  
        }  
    }  
  
    $student= new Student("海宁",20);  
    echo $student->getNameAndAge();  
?>
```

### 对 url 传递的参数进行编码

```
<a href="index.php?picname=<?php urlencode("xiaojing") ?>">goodboy</a>
```

JavaScript(start)-----

用关键字 var 声明变量与赋值：

声明变量 m,变量 n:var m,n;

JavaScript 注释：//当行注释，(/\* \*/)多行注释

JavaScript(end)-----

图形图像(begin)-----

imagecreate();创建画布

```
<?php
    $im=imagecreate(200,100);
    $white=imagecolorallocate($im,200,126,29);
    imagegif($im);
?>
```

imagecolorallocate();为一幅图像分配颜色

header — 发送一个自定义的 http 报文,告知浏览器所要输出图像的类型。

```
<?php
    header("Content-type: image/png"); //将图像输出到浏览器

    $img = imagecreate(200, 200); //创建一个 200×200 的图像

    $bg = imagecolorallocate($img, 12,135, 20); //设置图像背景色

    $white = imagecolorallocate($img, 255, 255, 255); //设置绘制图像的颜色为白色

    imagearc($img, 100, 100, 150, 150, 0, 360, $white); //绘制一个白色的圆

    imagepng($img); //输出图像

    imagedestroy($img); //释放内存
```

?>

imageline()画线 , imagearc()画圆 , imagerectangle()画矩形

<?php

```
header("Content-type: image/png"); //将图像输出到浏览器
```

```
$img = imagecreate(560, 200); //创建一个 560×200 的图像
```

```
$bg = imagecolorallocate($img, 0, 0, 255); //设置图像背景色
```

```
$white = imagecolorallocate($img, 255, 255, 255); //设置绘制图像的颜色为白色
```

```
imageline($img, 20, 20, 150, 180, $white);
```

```
imagearc($img, 250, 100, 150, 150, 0, 360, $white); //绘制一个白色的圆
```

```
imagerectangle($img, 350, 20, 500, 170, $white);
```

```
imagepng($img);
```

```
imagedestroy($img);
```

?>

imagefilledellipse()画一个填充的圆 , imagefilledrectangle()画一个填充的正方形

<?php

```
header("Content-type: image/png"); //将图像输出到浏览器
```

```
$img = imagecreate(400, 200); //创建一个 400×180 的图像
```

```
$bg = imagecolorallocate($img, 0, 0, 255);
```

```
$white = imagecolorallocate($img, 255, 0, 255);
```

```
imagefilledellipse($img, 100, 100, 150, 150, $white); //绘制圆形
```

```
imagefilledrectangle($img, 200, 50, 300, 150, $white); //绘制正方形
```

```
imagepng($img);
```

```
imagedestroy($img);
```



?>

图形图像(end)-----

文件系统 ( Filesystem)-----

**fopen()**打开一个文件，fclose()关闭文件资源，

fopen() 函数也用于创建文件。ps:\$myfile = fopen("testfile.txt", "a")

读取一个字符：

```
<?php
    $fopen=fopen('./xiaojing.txt','r');
    while(false!==( $chr=fgetc($fopen))){
        echo $chr;
    }
    fclose($fopen);
?>
```

**fgetc()**读取文件

```
<?php
    $fopen = fopen('./xiaojing.txt','rb');                                //
    创建文件资源

    while(false !== ( $chr = fgetc($fopen))){                            //使用 fgetc 函数
    取得一个字符，判断是否为 false

        echo $chr;                                                         //
    如果不是，输出该字符。

    }

    fclose($fopen);
    //关闭文件资源
?>
```

fgets — 从文件指针中读取一行, fgetss — 从文件指针中读取一行并过滤掉 HTML 标记

```
<?php
    $myfile=fopen("xiaojing.txt",'r') or die('Unable to open file.');
```

echo fgets(\$myfile);

fclose(\$myfile);

```
?>
```

PHP 检查 End-Of-File - feof()

feof() 函数检查是否已到达 "end-of-file" (EOF)。

feof() 对于遍历未知长度的数据很有用。

下例逐行读取 "webdictionary.txt" 文件，直到 end-of-file

```
<?php
    $fopen = fopen('./count.php','rb');
```

while(!feof(\$fopen)){ //使用 feof

测试指针是否到了文件结束的

位置

echo fgetss(\$fopen); //输出当前

行

}

fclose(\$fopen);

```
?>
```

fread()读取任意长度的字串

```
<?php
    $filename = "count.txt"; //要读取的文件
    $fp = fopen($filename,"rb"); //打开文件
    echo fread($fp,100); //使用 fread()函数
```

读取文件内容的前 100 个字节

```
        echo "<hr />";

        echo fread($fp,filesize($filename));           //输出其余的文件内容

?>
```

**readfile** — 输出一个文件,

file — 把整个文件读入一个数组中,

**file\_get\_contents** — 将整个文件读入一个字符串

<!-- 使用 readfile()函数读取 xiaojing.txt 文件的内容 -->

<?php readfile('xiaojing.txt'); ?>

<hr />

<!-- 使用 file()函数读取 xiaojing.txt 文件的内容 -->

<?php

\$f\_arr = file('xiaojing.txt');

foreach(\$f\_arr as \$cont){

echo \$cont."<br>";

}

?>

<hr>

<!-- 使用 file\_get\_contents()函数读取 xiaojing.txt 文件的内容 -->

<?php

\$f\_chr = file\_get\_contents('xiaojing.txt');

echo \$f\_chr;

?>

**fwrite()**写文件

<?php

\$filename="xiaojing.txt";

\$fopen=fopen(\$filename,'wb') or die("文件不存在");

```
$str="xiaojingtest";  
fwrite($fopen,$str);  
fclose($fopen);  
readfile($filename);  
?>
```

#### file\_put\_contents()写入文件

```
<?php  
  
    $str="吉林省 11 明日科技有限公司是一家以计算机软件技术为核心的高科技型  
企业，公司创建于 1999 年 12 月，是专业的应用  
  
软件开发和服务提供商。";  
  
    file_put_contents("xiaojing.txt",$str);  
  
    readfile("xiaojing.txt");  
  
?>
```

#### \$\_FILES --HTTP 文件上传变量

```
<form name="form" method="POST" action="index.php" enctype="multipart/form-  
data">  
  
    请选择上传文件：<input name="upfile" type="file" /><input type="submit"  
name="submit" value="提交" />  
  
</form>  
  
<?php  
    if(!empty($_FILES)){  
        foreach($_FILES['upfile'] as $name=>$value)  
            echo $name.'='.$value.'<br>';  
    }  
  
?>
```

#### move\_uploaded\_file — 将上传的文件移动到新位置

```
<form name="form" method="post" action="index.php" enctype="multipart/form-data">
```

```
    请选择上传的文件：<input type="file" name="upfile"><input type="submit"
name="submit" value="提交">
```

```
</form>
```

```
<?php
```

```
    if(!empty($_FILES["upfile"]["name"])){
        $fileinfo=$_FILES["upfile"];
        $type=substr($fileinfo['name'],'.');
        if($type!=".png"){
            echo "您上传的文件格式不正确！";
        }else{
            if($fileinfo['size']<2097152 && $fileinfo['size']>0){
                $path="upfile/".$_FILES["upfile"]["name"];
                move_uploaded_file($fileinfo['tmp_name'],$path);
                if(is_dir("upfile")){
                    $dir=scandir("upfile/");
                    foreach($dir as $value){
                        echo $value."<br>";
                    }
                }else{
                    echo "目录路径错误！";
                }
            }else{
                echo '文件大小不符合要求';
            }
        }
    }
}
```

```
?>
```

## 文件上传

```
<!-- 上传文件的 form 表单，必须有 enctype 属性 -->

<form action="" method="post" enctype="multipart/form-data">

    请选择上传文件：

        <!-- 上传文件域，type 类型为 file -->

        <input type="file" name="up_file"/>

        <!-- 提交按钮 -->

        <input type="submit" name="submit" value="上传" />

    </form>

    <!-- 处理表单返回结果 -->

<?php

/* 判断是否有上传文件 */

    if(!empty($_FILES[up_file][name])){

/* 将文件信息赋给变量$fileinfo */

        $fileinfo = $_FILES[up_file];

        $type=strstr($fileinfo['name'],".");           //获取文件类型

        if($type!=".JPG"){

            echo "您上传文件格式不正确！";

        }else{

            if($fileinfo['size'] < 2097152 && $fileinfo['size'] > 0){           /*
判断文件大小 */

                $path="upfile/".$_FILES["up_file"]["name"];
                //定义上传文

                件的路径

                move_uploaded_file($fileinfo['tmp_name'],$path);

                //上传文件
```

```

//上传成功后，浏览该目录

if(is_dir("upfile/")){                                     //判断文件
名是否为目录

    $dir = scandir("upfile/");

    //使用 sacndir 函数取得所有文件及目录
    foreach($dir as $value){

        //使用 foreach 循环
        echo $value."<br>";

        //循环输出文件及目录名称
    }
}else{
    echo "目录路径错误！";
}
}else{
    echo '文件大小不符合要求';
}
}
}
?>

```

\$ FILES["file"]["name"] - 被上传文件的名称

\$ FILES["file"]["type"] - 被上传文件的类型

\$ FILES["file"]["size"] - 被上传文件的大小，以字节计

\$ FILES["file"]["tmp\_name"] - 存储在服务器的文件的临时副本的名称

\$ \_FILES["file"]["error"] - 由文件上传导致的错误代码

## PHP 文件上传案例 2

```
<form name="form1" method="post" action="upload_file.php"
enctype="multipart/form-data">
```

<!--enctype 属性规定了在提交表单时要使用哪种内容类型。在表单需要二进制数据时，比如文件内容，请使用 "multipart/form-data"。 -->

```
<input type="file" name="file" /><br>
```

```
<input type="submit" name="submit" value="上传" />
```

```
</form>
```

## upload\_file.php 文件

```
<?php
```

```
if(
```

```
($_FILES['file']['type'])=="image/gif" ||
```

```
($_FILES['file']['type'])=="image/jpeg" ||
```

(\$\_FILES['file']['type'])=="image/pjpeg" || //对于 IE 识别 jpg 文件的类型必须是 pjpeg，对于 FireFox，必须是 jpeg。

```
($_FILES['file']['type'])=="image/png" &&
```

```
($_FILES['file']['size'])<=(1*1024*1024)
```

```
){
```

```
if((($_FILES['file']['error'])>0)){
```

```
    echo "Error :". $_FILES['file']['error'];
```

```
}else{
```

```
    echo "名字 :". $_FILES['file']['name'].'<br>';
```

```
    echo "型形 :". $_FILES['file']['type'].'<br>';
```

```
    echo "大小 :". ($_FILES['file']['size'])/1024 . 'kb<br>';
```



```
echo "临时副本:" . $_FILES['file']['tmp_name'].'<br>';

if(file_exists("upload/" . $_FILES["file"]["name"])){

    echo $_FILES["file"]["name"]."已存在！";

}else{

    move_uploaded_file($_FILES["file"]["tmp_name"],"upload/" . $_FILES["file"]["name"]);

    echo "上传成功：upload/" . $_FILES["file"]["name"];

}

}

}else{

    echo "上传的图片有误！图片格式为 gif 或 jpg 或 png，且小于 1M";

}

?>
```

**scandir** — 列出指定路径中的文件和目录,浏览目录

```
<?php
    $path = 'F:/AppServ/www/tm/sl/14/11';
        //要浏览的目录

    if(is_dir($path)){
        //判断文件名是否为目

        录

        $dir = scandir($path);
        //使用 sacndir 函
        数取得所有文件及目录

        foreach($dir as $value){
            //使用 foreach 循环
```

```

                echo $value."<br>";
            //循环输出文件及目录
            名称
        }
    }else{
        echo "目录路径错误！";
    }
?>

```

**closedir — 关闭目录句柄**

**flock — 轻便的咨询文件锁定**

```

<?php
    $filename = 'count.txt';                                //声明要打
    开的文件的名称

    $fd = fopen($filename,'w');                             //以 w 形式打开文件

    flock($fd, LOCK_EX);                                    //锁定文件（独占共享

    fwrite($fd, "<strong>PHP 从基础到项目实战</strong>");
    //向文件中写入数据

    flock($fd, LOCK_UN);                                    //解除锁定

    fclose($fd);                                            //关闭文件指针

    readfile($filename);                                    //输出文件内容

?>

```

操作文件

复制文件:bool copy(string path1,string path2)

重命名文件：bool rename(string filename1,string filename2)

删除文件：bool unlink(string filename)

注：更多"操作文件"请查询 Filesystem 或者 php 中文函数手册

## 操作目录

新建目录：bool mkdir('temp')

删除指定空目录：bool rmdir(string dirname)

注：更多"操作目录"请查询 Filesystem

文件系统 ( end)-----

Smarty 模板技术(begin)-----

- 1.下载最新版 Smarty,解压包中 demo 是一个案例,libs 是核心程序
- 2.把 libs 里的程序复制复制到根目录，新建 4 个目录 templates,templates\_c,configs 和 cache
- 3.创建配置文件 config.php,配置文件的内容参考 demo 中的案例。

assign 用于在模板被执行时为模板变量赋值

display 显示模板，需要指定一个合法的模板资源的类型和路径

修饰变量：变量和方法之间使用"|"分隔，多个参数用":" 分开，

ps:{ $\$$ str|capitalize} //首字母大写

从配置文件中读取数据，可以通过##来读取配置中的数据，可以通过\$smarty.config 来读取配置中的数据

./config/xiaojingtest.conf 文件

title="xiaojingtest"

bgcolor="\$fofofo"

border="2px"

type="计算机"

name="php 从入门到精通"

./template/index.tpl 文件

```
{config_load file="./config/xiaojingtest.conf"}
```

```
<html>
```

```
<head><title>{$title}</title></head>
```

```
<body bgcolor="{#bgcolor#}"> //可以通过##来读取配置中的数据
```

```
    <table border="{#border#}">
```

```
        <tr>
```

```
            <td>{$smarty.config.type}</td> //可以通过$smarty.config 来  
读取配置中的数据
```

```
            <td>{$smarty.config.name}</td>
```

```
        </tr>
```

```
    </table>
```

```
</body>
```

```
</html>
```

./config.php 文件

```
<?php
```

```
    require("Smarty.class.php");
```

```
    $smarty=new Smarty;
```

```
    //$smarty->force_compile = true;
```

```
    $smarty->debugging = true;
```

```
    $smarty->caching = true;
```

```
    $smarty->cache_lifetime = 120;
```

```
?>
```

./index.php 文件

```

<?php
    include_once("config.php");

    $smarty->assign("title","The one test!");

    $smarty->assign("top","导航");

    $smarty->assign("content","Hello,Smarty!");

    $smarty->assign("foot","页脚");

    $smarty->display("index.tpl");
?>

```

## 流程控制

if...elseif...else,语句必须以/if 为结束标记

```

<body>
{if $smarty.get.type == 'tm'}
欢迎光临, {$smarty.get.type}
{else}
对不起, 您不是本站 VIP, 无权访问此栏目。
{/if}
</body>

```

foreach 循环控制,

```

{foreach name=foreach_name key=key item=item from=arr_name}
... //name 为循环的名称, key 为元素键值, item 为元素的变量名, from 为数组
{/foreach}

```

//模板文件

```

<html>

```

```

<head>

```

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
```

```
<title>{$title}</title>
```

```
</head>
```

```
<body>
```

```
    {foreach key=key item=item from=$infobook}
```

```
    {$key}=>{$item}<br />
```

```
    {/foreach}
```

```
</body>
```

```
</html>
```

//index.php 文件

```
<?php
```

```
    include_once './config.php';
```

```
    $infobook=array('one'=>'萧劲','two'=>'蓝佩','three'=>'海宁');
```

```
    $smarty->assign('title','xiaojingtest');
```

```
    $smarty->assign('infobook',$infobook);
```

```
    $smarty->display('index.tpl');
```

```
?>
```

section 循环控制{section name="sec\_name" loop=\$arr\_name start=num step=num}

//name 为名称，loop 为数组，start 为循环的初始位置，step 为步长

//index.tpl 文件

```
<body>
```

```
    <table>
```

```
        {section name=sec1 loop=$obj}
```

```
            <tr>
```

```
                <td style="color:red;font-size:30px;">{$obj[sec1].bigclass}</td>
```

```
            </tr>
```

```
        {section name=sec2 loop=$obj[sec1].smallclass}
```

```
            <tr>
```

```

        <td>{$obj[sec1].smallclass[sec2].s_type}</td>

    </tr>

    {/section}

{/section}

</table>

</body>

//index.php 文件

<?php

    require"config.php";

    $obj = array(

        array("id" => 1, "bigclass" => "计算机图书", "smallclass" =>
array(array("s_id" => 1, "s_type" =>
        "PHP"))),

        array("id" => 2, "bigclass" => "历史传记", "smallclass" =>
array(array("s_id" => 2, "s_type" => "中国
        历史"), array("s_id" => 3, "s_type" => "世界历史"))),

        array("id" => 3, "bigclass" => "畅销小说", "smallclass" =>
array(array("s_id" => 4, "s_type" => "网络
        小说"), array("s_id" => 5, "s_type" => "科幻小说")))

    );

    $smarty->assign("title", "xiaojingtest");

    $smarty->assign("obj", $obj);

    $smarty->display("index.tpl");

```

?>

Smarty 模板技术 ( end)-----

<meta http-equiv="content-Type" content="text/html" charset="utf-8" />

<link href="css/style.css" rel="stylesheet">

html 中插入一条直线 : <hr />

ThinkPHP 框架 ( begin ) -----

下载 ThinkPHP 框架代码 , 解压到根目录

www WEB 部署目录 ( 或者子目录 )

├─index.php     入口文件

├─README.md     README 文件

├─Application     应用目录

├─Public     资源文件目录

└─ThinkPHP     框架目录

ThinkPHP 框架系统目录 ( 可以部署在非 web 目录下面 )

├─Common     核心公共函数目录

├─Conf     核心配置目录

├─Lang     核心语言包目录

├─Library     框架类库目录

|   ├─Think     核心 Think 类库包目录

|   ├─Behavior   行为类库目录



- | |─Org     Org 类库包目录
- | |─Vendor   第三方类库目录
- | |─ ...    更多类库目录
- ─Mode       框架应用模式目录
- ─Tpl        系统模板目录
- ─LICENSE.txt 框架授权协议文件
- ─logo.png   框架 LOGO 文件
- ─README.txt 框架 README 文件
- └─index.php  框架入口文件

## Application

- ─Common     应用公共模块
  - | |─Common   应用公共函数目录
  - | |─Conf     应用公共配置文件目录
- ─Home       默认生成的 Home 模块
  - | |─Conf     模块配置文件目录
  - | |─Common   模块函数公共目录
  - | |─Controller 模块控制器目录
  - | |─Model    模块模型目录
  - | |─View     模块视图文件目录
- └─Runtime   运行时目录
  - ─Cache     模版缓存目录
  - ─Data     数据目录
  - ─Logs     日志目录

└─Temp      缓存目录

未完成案例 30.2

未完成案例 30.4

未完成案例 30.5

ThinkPHP(end)-----

相对定位和绝对定位

绝对定位：<div style=" position:absolute; left:100px; top:150px;">xiaojingtest</div>

相对定位：<div class="div1" style="background-color:#ccc;"><div class="div2" style="position:relative;top:50px;">test</div><div>//div2 相对 div1 定位