**PHP超全局变量**

$GLOBALS：包含一个引用指向每个当前脚本的全局范围内有效的变量。

$\_SERVER：由 web 服务器设定或者直接与当前脚本的执行环境相关联。

$\_GET：由 URL 请求提交至脚本的变量。

$\_POST：由 HTTP POST 方法提交至脚本的变量。

$\_COOKIE：由 HTTP Cookies 方法提交至脚本的变量。

$\_FILES：由 HTTP POST 文件上传而提交至脚本的变量。

$\_ENV：执行环境提交至脚本的变量。

$\_REQUEST：由 GET，POST 和 COOKIE 机制提交至脚本的变量。

**来自 PHP 之外的变量**

当一个表单提交给 PHP 脚本时，表单中的信息会自动在脚本中可用。有很多方法访问此信息，例如：

<form action="foo.php" method="POST">  
   Name:  <input type="text" name="username"><br />  
   Email: <input type="text" name="email"><br />  
   <input type="submit" name="submit" value="Submit me!" />

</form>

**在PHP中以如下方式引用这些变量**

<?php

echo $\_POST['username'];

echo $\_REQUEST['username'];

echo $HTTP\_POST\_VARS['username'];

echo $username;

?>

**用PHP处理表单：一个用户登录的表单**

Loginform.php

<form name="fmlog" method="post”action=“logcheck.php”>

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

密　码:<input type="password" name="password"/><br>

<input type="hidden" name="action" value="login" />

<input type="submit" value="登 录" />

</form>

**处理用户登录的php代码**

<?php

session\_start();

if (isset($action) && $\_POST["action"] == "login"){

// 处理登录的代码

$user=$\_POST["username"];

$pwd =$\_POST["password"];

// 记录用户id和password

$\_SESSION["USERID"] = strtoupper($user);

$\_SESSION["USERPSW"] = $psw;

echo "$user logged in. Your are welcome. ";

echo '<form name="fmlog" method="post">';

echo '<input type="hidden" name="action" value="logout" />';

echo '<input type=“submit” value=“注销" />';

echo '</form>';

}

?>

**文件上传的例子**

<?php

//filename:upldfile.php

if($\_POST["ifupload"]=="1") {

$path=AddSlashes(dirname(\_\_FILE\_\_))."\\\\upload\\\\";

for($i=1;$i<=8;$i++) {

$files="afile$i";

if (is\_uploaded\_file($\_FILES[$files]['tmp\_name'])) {

$filename = $\_FILES[$files]['name'];

$localfile = $path.$filename;

move\_uploaded\_file($\_FILES[$files]['tmp\_name'], $localfile);

}

}

echo "<br><b>You have uploaded files successfully</b><br>";

exit;

}

?>

<body>

**商品图片上传的例子**

<table id="divContainer" style="HEIGHT: 100%; WIDTH: 380" border="0">

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

商品图片上传必考，这个enctype=“multipart/form-data属性必考！必须有！不写会扣分！

<tr height="35"><td align="right" valign="bottom">多文件上传 </td></tr>

<tr><td align="center" valign="top">

<table class="Transex" border="1" cellspacing="0" cellpadding="0" >

<tr>

<td nowrap><label class="FieldLabel">文件1</label></td>

<td><input type="file" class="TxtInput" tabindex="1" name="afile1" ></td>

</tr>

…

<tr>

<td colspan="2" align="right">

abc()用于校验。类似于submit按钮，但是它直接提交了。用普通按钮更常见一些。

<button name=“btnOk”id=“btnOk” onclick=“javascript:abc()”>确定</button>

<input type="hidden" name="ifupload" value=1>&nbsp;&nbsp;

<button name="btnCancel" id="btnCancel" onclick="window.close();">取消</button></td>

</tr>

</table>

</td>

</tr>

</form>

</table>

</body>

**逻辑与和逻辑或的短路原则**

在逻辑判断语句中我们还需要注意的是逻辑与和逻辑或的短路问题，在平常编程中我们常用的是（&&）和（||），他们和（&）（|）的区别就是有短路原则。也就是说，逻辑与，只要第一个操作数的值为FALSE，它就不会再去验证或者执行第二个表达式了，因为表达式结果已经确定为FALSE了。

**利用php输出九九乘法表**

**使用for循环实现输出九九乘法表。**

<?php

for ($i = 1; $i <= 9; $i++) {

for ($j = 1; $j <= $i; $j++) {

echo "$j \* $i = " . ($j \* $i) . "\t";

}

echo PHP\_EOL; // 换行

}

?>

**使用while循环实现输出九九乘法表。**

<?php

$i = 1;

while ($i <= 9) {

$j = 1;

while ($j <= $i) {

echo "$j \* $i = " . ($j \* $i) . "\t";

$j++;

}

echo PHP\_EOL; // 换行

$i++;

}

?>

**常用的字符串输出函数**

|  |  |
| --- | --- |
| **echo()** | 输出字符串 |
| **print()** | 输出一个或者多个字符串 |
| **die()** | 输出字符串并退出当前脚本 |
| **printf()** | 输出格式化字符串 |
| **sprintf()** | 把格式化的字符串写入一个变量中 |

**SQL语言**

**Where子句：**select \* from authors where au\_lname ='Smith'

**算术操作符(+)：**select discounttype,stor\_id,lowqty, highqty,discount+0.5 from discounts

**建立虚拟列：**select discounttype,stor\_id,lowqty, highqty, discount 旧折扣,discount+0.5 新折扣 from discounts

**算术操作符(-)：**select discounttype,stor\_id,lowqty, highqty, discount 旧折扣, discount+4 新折扣, discount+4-discount 折扣差 from discounts

**算术操作符(/)：**select discounttype,stor\_id,lowqty, highqty, discount 旧折扣, discount+4 新折扣, (discount+4)/discount 折扣比 from discounts

**算术操作符(%)：**select discounttype,stor\_id,lowqty, lowqty%3 模, highqty, discount from discounts

**比较操作符：**比较操作符比较两个表达式并返回如下三个值之一，TRUE，FALSE或NULL。不能将空值用于区分表中两行所需的信息（例如，外键或主键）如果数据出现空值，则逻辑运算符和比较运算符有可能返回 TRUE 或 FALSE 以外的第三种结果 UNKNOWN。需要三值逻辑是导致许多应用程序出错之源。

**Is null 操作符:** select \* from discounts where lowqty is null

**比较操作符(=):**select \* from discounts where discounttype = 'Volume Discount'

**比较操作符(>和>=):**select \* from discounts where discounttype >= 'Vo'

**比较操作符(<>或!=):** select \* from discounts where discount <> 6.7

select \* from discounts where discount != 6.7

**字符操作符(like和%):**select \* from authors where au\_lname like 'St%'

**字符操作符(\_):**select \* from authors where phone like '%9\_6\_8%'

**字符操作符(+):**select au\_id, au\_lname, au\_fname, au\_lname+au\_fname from authors

**逻辑操作符(and):** select \* from discounts where discount>=5 and discount<10

**逻辑操作符(or):** select \* from discounts where discount=5 or discount=6.7

**逻辑操作符(not):** select \* from discounts where not discount = 5

**操作符(in):** select \* from discounts where discount in (5,6.7)

**操作符(between):** select \* from discounts where discount between 5 and 10

**聚集函数(count):** select count(\*) from discounts where discount>=5 and discount<10

**聚集函数(sum):** select sum(discount) from discounts where discount>=5 and discount<10

**聚集函数(avg):** select avg(discount) from discounts where discount>=5 and discount<10

**聚集函数(max):** select max(discount) from discounts where discount>=5 and discount<10

**聚集函数(min):** select min(discount) from discounts where discount>=5 and discount<10

**聚集函数(var):** select var(discount) from discounts

**聚集函数(stdev):** select stdev(discount) from discounts

**日期函数(年、月、日):** SELECT "Year Number" = YEAR('03/12/2003'),"Month Number" = MONTH('03/12/2003'),DAY('03/12/2003') AS 'Day Number'

SELECT "Year Number" = DATEPART(yy, '03/12/2003'), "Month Number" = DATEPART(mm, '03/12/2003'), "Day Number" = DATEPART(dd, '03/12/2003')

**从系统日期中取年、月、日:**SELECT "Year Number" = YEAR(getdate()),"Month Number" = MONTH(getdate()),DAY(getdate()) AS 'Day Number'

**从日期字段中取年、月、日:**SELECT "Year Number" = YEAR(hire\_date),"Month Number" = MONTH(hire\_date),DAY(hire\_date) AS 'Day Number' from employee

**数学函数**SELECT discount, exp(discount) from discounts

还有很多数学函数，例如：ABS、CEILING、DEGREES、FLOOR、POWER、RADIANS、SIGN、EXP、LOG、LOG10、SQUARE、SQRT和三角函数。

**字符函数(chr)：**SELECT char(65)

**字符函数(len)：**SELECT discount, len(discount) from discounts

**Select语句的子句**

SELECT select\_list  
[ INTO new\_table ]   
FROM table\_source   
[ WHERE search\_condition ]   
[ GROUP BY group\_by\_expression ]   
[ HAVING search\_condition ]   
[ ORDER BY order\_expression [ ASC | DESC ] ]

**连接两个表：**SELECT dbo.学生档案.学号, dbo.学生成绩.学习科目 FROM dbo.学生成绩 CROSS JOIN dbo.学生档案

**内连接**：SELECT dbo.学生档案.学号, dbo.学生档案.姓名, dbo.学生档案.性别, dbo.学生档案.班级, dbo.学生成绩.学习科目, dbo.学生成绩.学习成绩 FROM dbo.学生成绩 INNER JOIN dbo.学生档案 ON dbo.学生成绩.s\_id = dbo.学生档案.s\_id

**将一个表连接到它自身是个经常使用的技术，用来找出有重复字段的记录。**

SELECT dbo.学生档案.学号, dbo.学生档案.姓名 FROM dbo.学生档案 INNER JOIN dbo.学生档案 学生档案\_1 ON dbo.学生档案.姓名 = 学生档案\_1.姓名 AND dbo.学生档案.学号 <> 学生档案\_1.学号

**INSERT语句**:INSERT INTO 学生档案 (学号,姓名,性别) VALUES ('009','李刚','男')

**UPDATE语句:** UPDATE 学生档案 SET 班级='初二三班' WHERE s\_id=2

**DELETE语句:** DELETE FROM 学生档案 WHERE s\_id=2

**一个神奇的插入**

mysql> create table set \_test(e\_set set('乒乓球','羽毛球','篮球'，'足球'））;

Query OK, 0 rows affected (0.04 sec);

mysql>insert into set\_test value('乒乓球,篮球');

Query OK, 1 row affected (0.01 sec)

mysql>insert into set test value(9);//插入9，对应1001，1代表选中，0代表 不选中

Query OK, 1 row affected (0.01 sec

mysql>select\* from set test;

**PHP操作MySQL数据库**

**连接数据库**

mysql\_connect($server,$username,$password,$new\_link,$client\_flags)

**选择数据库**

mysql\_select\_db($database\_name,$lin\_identifier)

**使用mysql\_query()发送一条SQL查询语句，查询成后，该函数返回资源类型的结果来供其它函数处理**

mysql\_query($query,$link\_identifier)

**mysql\_fetch\_array()用于从结果集中取得一行作为关联数组，或数字数组，或二者兼有**

mysql\_fetch\_array($result,$result\_type)

**断开与数据库的连接**

mysql\_close($link\_identifier)

mysql\_num\_rows()用于取得结果集中行的数目，此函数仅对SELECT语句有效。

mysql\_affected\_rows()用于记录上一次MySQL操作影响的行数。

**HTML**

**HTML语法**

<标记 属性1=属性值 属性2=属性值…>

受标记影响的内容

</标记>

**HTML文件的基本结构**

<HTML>

<HEAD>

<TITLE>网页的标题</TITLE>

</HEAD>

<BODY>

文档主体，正文部分

</BODY>

</HTML>

**强制换行标记<BR>:**文字<BR>

**图片标记<IMG>:**

<IMG src="图片文件名" alt="简单说明" width="图片的宽度"height="图片的高度" hspace="水平方向空白" vspace="垂直方向空白" border="边框宽度" align="对齐方式">

**建立简单表格:**

<TABLE>

<CAPTION align=left | right | center | top | bottom valign= top | bottom>标题

</CAPTION>

<TR><TH>表头1</TH> <TH>表头2</TH>…<TH>表头n</TH></TR>

<TR><TD>表项11</TD><TD>表项12</TD><TD>表项1n</TD></TR>

<TR><TD>表项m1</TD><TD>表项m2</TD>…<TD>表项mn</TD></TR>

</TABLE>

**表单的标记<Form>**

<form>表单标记的主要作用是设定表单的起止位置，并指定处理表单数

据程序的URL地址。其基本格式为：

<FORM action="url" method="get|post" name="value" onsubmit=

"function" onreset="function" target="window">

……

</FORM>

<INPUT>标记

在表单中使用最频繁，大部分表单内容需要用到此标记。它主要用来设计表单中提供给用户的输入形式。其基本格式为：

<INPUT name="value" type="form area type" value="value" src="url" checked maxlength="n" size="n">

**Cookie与Session技术**

**创建Cookie：**

setcookie($name,$value,$expire,$path,$domain,$secure,$httponly)

**读取Cookie:**

在PHP中可以使用超全局变量$\_COOKIE来读取浏览器端的Cookie值.

**删除Cookie:**