

十天学会 PHP

下面简单介绍一下 PHP 的语法。

1、嵌入方法：

类似 ASP 的 <% ，PHP 可以是 <?php 或者是 <? ，结束符号是 ?> ，当然您也可以自己指定。

2、引用文件：

引用文件的方法有两种：require 及 include 。

require 的使用方法如 require("MyRequireFile.php"); 。这个函数通常放在 PHP 程序的最前面，PHP 程序在执行前,就会先读入 require 所指定引入的文件，使它变成 PHP 程序网页的一部份。常用的函数，亦可以这个方法将它引入网页中。

include 使用方法如 include("MyIncludeFile.php"); 。这个函数一般是放在流程控制的处理部分中。PHP 程序网页在读到 include 的文件时，才将它读进来。这种方式，可以把程序执行时的流程简单化。

3、注释方法：

```
[php]
1.  <?php
2.
3.  echo "这是第一种例子。\\n" ; //本例是 C++ 语法的注释
4.
5.  /* 本例采用多行的
6.
7.  注释方式 */
8.
9.  echo "这是第二种例子。\\n" ;
10.
11. echo "这是第三种例子。\\n" ; #本例使用 UNIX Shell 语法注释
12.
13. ?>
```

4、变量类型：

```
[php]
1.  $mystring = "我是字符串" ;
2.
3.  $NewLine = "换行了\\n" ;
4.
5.  $int1 = 38 ;
6.
7.  $float1 = 1.732 ;
8.
9.  $float2 = 1.4E+2 ;
10.
11. $MyArray1 = array( "子" , "丑" , "寅" , "卯" );
```

这里引出两个问题，首先 PHP 变量以 \$ 开头，第二 PHP 语句以 ; 结尾，可能 ASP 程序员会不适应。这两个遗漏也是程序上大多错误所在。

5、运算符：

数学运算：符号意义

+ 加法运算

- 减法运算

* 乘法运算

/ 除法运算

% 取余数

++ 累加

-- 递减

字符串运算：

运算符只有一个，就是英文的句号。它可以将字符串连接起来，变成合并的新字符串。类似 ASP 中的 &

```
[php]
1.  <?
2.  $a = "PHP 4" ;
3.  $b = "功能强大" ;
4.  echo $a.$b;
5.  ?>
```

这里也引出两个问题，首先 PHP 中输出语句是 echo ，第二类似 ASP 中的 <%= 变量 %> ， PHP 中也可以 <?= 变量 ?> 。

逻辑运算：

符号意义

< 小于

> 大于

<= 小于或等于

>= 大于或等于

== 等于

!= 不等于

&& 而且 (And)

and 而且 (And)

|| 或者 (Or)

or 或者 (Or)

xor 异或 (Xor)

! 不 (Not)

今天就说到这里，明天说一下流程控制。

学习目的：掌握 php 的流程控制

1、if..else 循环有三种结构

第一种是只有用到 if 条件，当作单纯的判断。解释成 " 若发生了某事则怎样处理 " 。语法如下：

```
if (expr) { statement }
```

其中的 expr 为判断的条件,通常都是用逻辑运算符当判断的条件。而 statement 为符合条件的执行部分程序，若程序只有一行，可以省略大括号 {} 。

范例：本例省略大括号。

```
[php]
1.  <?php
2.  if ($state==1)echo "哈哈" ;
3.  ?>
```

这里特别注意的是，判断是否相等是 == 而不是 = ， ASP 程序员可能常犯这个错误， = 是赋值。



范例：本例的执行部分有三行，不可省略大括号。

```
[php]
1.  <?php
2.
3.  if ($state==1) {
4.  echo "哈哈" ;
5.  echo "<br>";
6.  }
7.  ?>
```

第二种是除了 if 之外，加上了 else 的条件，可解释成 " 若发生了某事则怎样处理，否则该如何解决 " 。语法如下



```
if (expr) { statement1 }else { statement2 }
```

范例：上面的例子来修改成更完整的处理。其中的 else 由于只有一行执行的指令，因此不用加上大括号。

```
[php]    
1. <?php  
2.     if ($state==1) {  
3.         echo "哈哈" ;  
4.         echo "<br>" ;  
5.     }  
6.     else{  
7.         echo "呵呵" ;  
8.         echo "<br>" ;  
9.     }  
10. ?>
```

第三种就是递归的 if..else 循环，通常用在多种决策判断时。它将数个 if..else 拿来合并运用处理。

直接看下面的例子

```
[php]    
1. <?php  
2.     if ( $a > $b ) {  
3.         echo "a 比 b大" ;  
4.     } elseif ( $a == $b ) {  
5.         echo "a 等于 b" ;  
6.     } else {  
7.         echo "a 比 b小" ;  
8.     }  
9. ?>
```



上例只用二层的 if..else 循环，用来比较 a 和 b 两个变量。实际要使用这种递归 if..else 循环时，请小心使用，因为太多层的循环容易使设计的逻辑出问题，或者少打了大括号等，都会造成程序出现莫名其妙的问题。

2、for 循环就单纯只有一种，没有变化，它的语法如下

```
for (expr1; expr2; expr3) {statement }
```

其中的 expr1 为条件的初始值。expr2 为判断的条件，通常都是用逻辑运算符 (logicaloperators) 当判断的条件。expr3 为执行 statement 后要执行的部份，用来改变条件，供下次的循环判断，如加一 .. 等等。而 statement 为符合条件的执行部分程序，若程序只有一行，可以省略大括号 {}。

下例是用 for 循环写的例子。

```
[php]    
1. <?php  
2.     for ( $i = 1 ; $i <= 10 ;$i ++ ) {  
3.         echo "这是第".$i."次循环<br>" ;  
4.     }  
5. ?>
```

3、switch 循环,通常处理复合式的条件判断,每个子条件,都是 case 指令部分。在实作上若使用许多类似的 if 指令,可以将它综合成 switch 循环。

语法如下

```
switch (expr) { case expr1:statement1; break; case expr2: statement2; break; default: statementN; break; }
```

其中的 expr 条件，通常为变量名称。而 case 后的 exprN，通常表示变量值。冒号后则为符合该条件要执行的部分。注意要用 break 跳离循环。

```
[php]    
1. <?php  
2.     switch ( date ("D" ) ) {  
3.         case "Mon" :  
4.             echo "今天星期一" ;  
5.             break;  
6.         case "Tue" :  
7.             echo "今天星期二" ;  
8.             break;  
9.         case "Wed" :  
10.            echo "今天星期三" ;  
11.            break;  
12.         case "Thu" :  
13.            echo "今天星期四" ;  
14.            break;  
15.         case "Fri" :  
16.            echo "今天星期五" ;  
17.            break;  
18.         default:  
19.            echo "今天放假" ;  
20.            break;  
21.     }  
22. ?>
```

这里需要注意的是 break; 别遗漏了，default，省略是可以的。

很明显的,上述的例子用 if 循环就很麻烦了。当然在设计时,要将出现机率最大的条件放在最前面,最少出现的条件放在最后面,可以增加程序的执行效率。上例由于每天出现的机率相同,所以不用注意条件的顺序。

今天就说到这里,明天开始说数据库的使用。

学习目的：学会连接数据库

PHP 简直就是一个函数库,丰富的函数使 PHP 的某些地方相当简单。建议大家 down 一本 PHP 的函数手册,总用的到。

我这里就简单说一下连接 MYSQL 数据库。

1、mysql_connect

打开 MySQL 服务器连接。

语法 : int mysql_connect(string[hostname] [:port], string [username], string [password]); 返回值 : 整数

本函数建立与 MySQL 服务器的连接。其中所有的参数都可省略。当使用本函数却不加任何参数时,参数 hostname 的默认值为 localhost、参数 username 的默认值为 PHP 执行行程的拥有者、参数 password 则为空字符串(即没有密码)。而参数 hostname 后面可以加冒号与端口号,代表使用哪个端口与 MySQL 连接。当然在使用数据库时,早点使用 mysql_close() 将连接关掉可以节省资源。

2、mysql_select_db

选择一个数据库。

语法 : int mysql_select_db(stringdatabase_name, int [link_identifier]); 返回值 : 整数

本函数选择 MySQL 服务器中的数据库以供之后的资料查询作业 (query) 处理。成功返回 true,失败则返回 false。

最简单的例子就是：

```
[php]
1. $conn=mysql_connect ("127.0.0.1", "", "");
2. mysql_select_db("shop");
```

连接机 MY SQL 数据库,打开 SHOP 数据库。在实际应用中应当加强点错误判断。

今天就说到这里,明天再说一下数据库的读取。

学习目的：学会读取数据

先看两个函数：

1、mysql_query

送出一个 query 字符串。语法 : int mysql_query(string query, int [link_identifier]); 返回值 : 整数

本函数送出 query 字符串供 MySQL 做相关的处理或者执行。若没有指定 link_identifier 参数,则程序会自动寻找最近打开的 ID。当 query 查询字符串是 UPDATE、INSERT 及 DELETE 时,返回的可能是 true 或者 false;查询的字符串是 SELECT 则返回新的 ID 值,当返回 false 时,并不是执行成功但无返回值,而是查询的字符串有错误。

2、mysql_fetch_object 返回类资料。语法 : object mysql_fetch_object(int result, int [result_type]); 返回值 : 类

本函数用来将查询结果 result 拆到类变量中。若 result 没有资料,则返回 false 值。

看一个简单的例子：

```
[php]
1. <?
2. $exec="select * from user";
3. $result=mysql_query($exec);
4. while($rs=mysql_fetch_object($result))
5. {
6.     echo "username:".$rs->username."<br>";
7. }
8. ?>
```

当然,表 user 中有一个 username 的字段,这就类似 asp 中的

```
[csharp]
1. <%
2. exec="select * from user"
3. set rs=server.createObject("adodb.recordset")
4. rs.open exec,conn,1,1
5. do while not rs.eof
6. response.write"username:"&rs("username")&"<br>"
7. rs.movenext
8. loop
9. %>
```

当然先要连接数据库，一般我们 require_once('conn.php'); 而 conn.php 里面就是上一次说的连接数据库的代码。

小小的两条命令可以完成读取数据的工作了，今天说到这里下一次说数据的添加删除修改。

学习目的：学会添加删除修改数据

mysql_query(\$exec);

单这个语句就可以执行所有的操作了，不同的就是 \$exec 这个 sql 语句

添加：\$exec="insert into tablename(item1,item2) values ('".\$_POST['item1'].","".\$_POST['item1'].")";

删除：\$exec="delete from tablenameewhere...";

修改：\$exec="update tablename setitem1='".\$_POST['item1']."' where ...";

说到这里就要说一下表单和 php 变量传递，如果表单中的一个 <inputname="item1" type="text" id="item1">

表单以 POST 提交的，那么处理表单文件就可以用 \$_POST['item1'] 得到变量值，同样以 GET 提交的就是 \$_GET['item1']

是不是很简单？但是通常 \$exec 会有问题，因为可能您的 SQL 语句会很长，您会遗漏 . 连接符，或者 ' 来包围字符串字段。

我们可以注释 mysql_query(\$exec); 语句用 echo \$exec; 代替来输出 \$exec 以检查正确性。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 phpmy admin 中执行，看看它的出错信息。还有需要注意的是，我们不要使用一些敏感的字符串作为字段名字，否则很可能会出现问，比如说 date 什么的。变量的命名，字段的命名遵循一点规律有的时候对自己是一种好处，初学者并不可忽视其重要性。

今天就说到这里，大家可以 DOWN 一个 SQL 语句的参考手册，再研究研究。明天继续说 SESSION

学习目的：学会 SESSION 的使用

SESSION 的作用很多，最多用的就是站点内页面间变量传递。在页面开始我们要 session_start(); 开启 SESSION ；

然后就可以使用 SESSION 变量了，比如说要赋值就是：\$_SESSION['item']="item1"; 要得到值就是 \$item1=\$_SESSION['item']; ，很简单吧。这里我们可能会使用到一些函数，比如说判断是不是某 SESSION 变量为空，可以这么写：empty(\$_SESSION['inum']) 返回 true or false 。

下面综合一下前面所说的我们来看一个登陆程序，判断用户名密码是否正确。

登陆表单是这样：login.php

```
[html]
1. <table width="100%" height="100%"border="0" align="center" cellpadding="0" cellspacing="0">
2. <tr>
3. <form action="checklogin.php"method="post"><td align="center" valign="middle"><table width="400"border="0" cellpadding="5" cellspacing
="1" class="tablebg">
4. <tr class="tdbg">
5. <td colspan="2"><div align="center">Administrators Login</div></td>
6. </tr>
7. <tr class="tdbg">
8. <td><div align="center">Username</div></td>
9. <td><div align="center">
10. <input name="username" type="text" id="username">
11. </div></td>
12. </tr>
13. <tr class="tdbg">
14. <td><div align="center">Password</div></td>
15. <td><div align="center">
16. <input name="password" type="password" id="password">
17. </div></td>
18. </tr>
19. <tr class="tdbg">
20. <td colspan="2"><div align="center">
21. <input type="submit" name="Submit" value="Submit">
22. <input type="reset" name="Submit2" value="Clear">
23. </div></td>
24. </tr>
25. </table></td></form>
26. </tr>
27. </table>
```

处理文件是这样

```
[php]
1.  <?
2.  require_once('conn.php');
3.  session_start();
4.  $username=$_POST['username'];
5.  $password=$_POST['password'];
6.  $exec="select * from admin where username='".$username."'";
7.  if($result=mysql_query($exec))
8.  {
9.      if($rs=mysql_fetch_object($result))
10.     {
11.         if($rs->password==$password)
12.         {
13.             $_SESSION['adminname']=$username;
14.             header("location:index.php");
15.         }
16.     }
17.     else
18.     {
19.         echo "<script>alert('Password CheckError!');location.href='login.php';</script>";
20.     }
21.     else
22.     {
23.         echo "<script>alert('Username CheckError!');location.href='login.php';</script>";
24.     }
25.     }
26.     else
27.     {
28.         echo "<script>alert('Database ConnectionError!');location.href='login.php';</script>";
29.     }
30. }
31. ?>
```

conn.php 是这样：

```
[php]
1.  <?
2.  $conn=mysql_connect ("127.0.0.1", "", "");
3.  mysql_select_db("shop");
4.  ?>
```

由于 \$_SESSION['adminname']=\$username; 我们可以这样写验证是否登陆语句的文件： checkadmin.asp

```
[php]
1.  <?
2.
3.  session_start();
4.
5.  if($_SESSION['adminname']=='')
6.  {
7.      echo "<script>alert('Please LoginFirst');location.href='login.php';</script>";
8.  }
9.  ?>
```

呵呵，今天说到这里，明天说一下怎么弄一个分页。

学习目的：做一个分页显示

关键就是用到了 SQL 语句中的 limit 来限定显示的记录从几到几。我们需要一个记录当前页的变量 \$page，还需要总共的记录数 \$num

对于 \$page 如果没有我们就让它 =0，如果有 <0 就让它也 =0，如果超过了总的页数就让他 = 总的页数。

```
[php]
1.  $execc="select count(*) from tablename ";
2.  $resultc=mysql_query($execc);
3.  $rsc=mysql_fetch_array($resultc);
4.  $num=$rsc[0];
```

这样可以得到记录总数

ceil(\$num/10)) 如果一页 10 记录的话，这个就是总的页数

所以可以这么写

```
[php]
1.  if(empty($_GET['page'])){
2.  $page=0;
3.  }
4.  else{
5.  $page=$_GET['page'];
6.  if($page<0)$page=0;
7.  if($page>=ceil($num/10))$page=ceil($num/10)-1;//因为page是从0开始的，所以要-1
8.  }
```

这样 \$exec 可以这么写

```
$exec="select * fromtablename limit ".$page*10.",10";
```

// 一页是 10 记录的

最后我们需要做的就是几个连接：

```
[html]
1.  <a href="xxx.php?page=0">FirstPage</a>
2.  <a href="xxx.php?page=<?=( $page-1) ?>">PrevPage</a>
3.  <a href="xxx.php?page=<?=( $page+1) ?>">NextPage</a>
4.  <a href="xxx.php?page=<?=(ceil($num/10)-1)>">LastPage</a>
```

这是一个大致的思路，大家可以想想怎么来优化？今天说到这里，明天说一下一些注意的问题。

学习目的：注意事项

因为我是先学 ASP 的，所以再做 PHP 的时候会发现很多地方需要适应。

- 1、注意不要漏了分号
- 2、注意不要漏了变量前的 \$
- 3、使用 SESSION 的时候注意不要遗漏 session_start();

如果发生错误的时候，可以采用以下方法：

- 1、如果是 SQL 语句出错，就注释了然后输出 SQL 语句，注意也要注释调后续的执行 SQL 语句
- 2、如果是变量为空，大多是没有传递到位，输出变量检查一下，检查一下表单的 id 和 name
- 3、如果是数据库连接出错，检查是否正确打开 MY SQL 和是否遗漏了连接语句
- 4、注意缩进，排除括号不区配的错误

在做大网站的时候，我的思路是先构建数据库，确定每一个字段的作用，和表之间的关系。然后设计后台界面，从添加数据开始做起，因为添加是否成功可以直接到数据库里面验证，做好了添加再做显示的页面，最后才是两者的结合。一般来说后台就包括添加删除修改和显示，后台没有问题了，前台也没有什么大问题。前台还需要注意安全性和容错还有就是输出格式。

好了，今天说到这里，明天说一下如果用 PHP 上传文件和发邮件。

学习目的：学会用 PHP 上传文件和发邮件

上传文件表单必须加上 enctype="multipart/form-data"

和 <input type="file"name="file">

下面看一下代码：

```
[php]
1.  $f=$_HTTP_POST_FILES['file'];
2.  $dest_dir='uploads';//设定上传目录
3.  $dest=$dest_dir.'/'.date("ymd")." " . $f['name'];//我这里设置文件名为日期加上文件名避免重复
4.  $r=move_uploaded_file($f['tmp_name'],$dest);
5.  chmod($dest, 0755);//设定上传的文件的属性
```

上传的文件名为 date("ymd")." " . \$f['name']，可以在以后插入到数据库的时候用到，PHP 实际上是你上传的文件从临时目录移动到指定目录。move_uploaded_file(\$f['tmp_name'],\$dest); 这是关键

至于发邮件就更加简单，可以使用 mail() 函数

```
[php]
1.  mail("收件人地址", "主题", "正文", "From:发件人\r\nReply-to:发件人的地址");
```

不过 mail() 需要服务器的支持，在 WINDOWS 下还需要配置 SMTP 服务器，一般来说外面的 LINUX 空间都行。

好像上传文件和发邮件比 ASP 简单很多，只要调用函数就可以了。ASP 还需要用到服务器的不同组件比如 FSO、JMAIL 什么的。

文章标签：

php

教程

数据

服务器

个人分类：[纯编程](#)

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com