**phplib template中文教程（一）**

第一章 如何进行安装配置

PHP世界中比较流行的模板处理工具有PHPLIB Template和FastTemplate两种。据笔者看有关资料介绍，PHPLIB Template的技术易用性和速度较为理想，所以俺就学习使用了PHPLIB Template。

phplib template 的安装其实非常简单。现在说简单，呵呵，其实刚开始上手的时候竟然花了好几天的功夫才知道大概是怎么一回事儿。那我们首先来认识一下phplib吧。

随着php的应用和推广，众多开发人员和很多爱好者在php的基础上开发了很多php的程序库，这些程序库大大简化了程序设计和提高了代码的可重用 性。其中 phplib就是这些库中相当优秀的产品。phplib有五大功能：将数据库驱动和对数据库操作完全分离；支持session；权限许可；模板；购物车。 我们要用的模板功能只是phplib这个库的诸多功能中的一种。笔者刚开始看phplib 的网方官站，就看得一头雾水，怎么还有数据库、session等等，最后才明白原来俺需要的只是template。

要说安装，其实就是下载 phplib库文件。你可以到phplib的官方网站下载最新的版本：http://phplib.sourceforge.net 这是一个 tar.gz 压缩包，下载完之后就是解压了。打开解压后的文件夹，里边有一个php文件夹，将该php文件夹复制到任何你想要的其它位置，但不要放在网页服务器的目录 下。比如：我的网页服务器根目录是：/var/www，我可以将php文件夹放在/ var下，与www文件夹平行。然后打开php的配置文件php.ini进行编辑。更改include\_path,使其指向刚刚复制的php文件夹的位 置。比如：include\_path = “.:/var/www/php”。phplib的官方站quick guide to installation讲了十个步骤，我们只用phplib template这一项功能的话，其它的安装步骤其实根本不用理会。保存退出，重启 apache2服务。然后就可以尝试编写你的第一个模板页了。

**phplib template中文教程（二）**

第二章 第一个简单的模板页  
我理解，所谓”模板”，就是将界面和实现代码分离开来，这样做的目的一是将美工和程序员的工作分离开来，提高工作效率，二是方便维护。下面我们就开始尝试 写一个简单的模板了，文件名first.htm。

< HTML>  
< HEAD>  
< TITLE> 我的第一个模板文件 < /TITLE>  
< /HEAD>  
< BODY>  
你知道吗？ {man}真是一个好人。  
< font color=”#0000FF”> 作者：{author}</font>  
< /BODY>  
< /HTML>

first.html模板中的{man}{author}{date}可以称为”模板变量”，用花括号({})括起来。

上面我们定义三个变量的原因是我们想随时改变它们的值，下面我们就来做这个工作。新建first.php文件，内容如下:

<?php  
include (’template.inc’); //包含进模板类 template.inc  
$tpl = new Template; //创建一个新模板  
$tpl->set\_file(’main’, ‘first.html’); //把模板文件加载进来  
$tpl->set\_var(’man’, ‘fuyatao’); // //给文件中的模板变量赋值  
$tpl->set\_var(’author’, ‘fuyatao’);  
$tpl->parse(’mains’, ‘main’); // //完成替换  
$tpl->p(’mains’); // //输出替换的结果  
?>

你也可以一次完成给变量赋值，这样

$tpl->set\_var(  
array(’man’=>’fuyatao’, ‘author’=>’fuyatao’)  
);

浏览器中浏览这个文件,你就会发现输出：你知道吗？fuyatao真是一个好人。作者：fuyatao.

初次尝试还算顺利吧？如果没有如你所愿地看到上述文字，那一定是你哪里弄错了。浏览器会有提示。还有如果你将这些代码复制到本机不能正确执行，有可 能是标点符号的问题，必须用英文的标点符号，程序中用中文的引号或分号都会引起程序不能正常运行，本人就曾经吃过这个亏，反复检查程序没有错，可就是执行 不了，后来才发现是一个地方用错了标点。

**phplib template中文教程（三）**

第三章 稍复杂点的应用

现在我们设想稍微复杂一点的情况，新建一个second.htm模板文件，内容为：

<HTML>  
<HEAD>  
<TITLE>第二个模板文件</TITLE>  
</HEAD> <BODY>  
下面是一个列表  
<UL>  
<li>谷军涛 的身高是 170cm  
<li>武剑 的身高是 171cm  
<li>青竹 的身高是 180cm  
<li>王聪聪 的身高是 172cm  
</UL>  
</BODY>  
</HTML>  
以上页面是纯静态的，假设要使用数据库，用PHP代码嵌入HTML的话，是这样写的:  
<HTML>  
<HEAD>  
<TITLE>第二个模板文件</TITLE>  
</HEAD> <BODY>  
下面是一个列表  
<?php  
$link=mysql\_connect(’localhost’,'fuyatao’,’ passwd’);  
$result=mysql\_db\_query(’gfs’,’select name,tall from my\_table’);  
echo “<UL>”;  
while ($a = mysql\_fetch\_array($result)){  
echo “<li>$a[name]的身高是$a[tall]”;  
}  
?>  
</UL>  
</BODY>  
</HTML>

PHP 新手一般都这样写，但你很快发现：你或者美工改这个列表样式的时候需要多么的小心翼翼,特别是美工，如果他不懂PHP代码，那么他会终日在恐惧之中！因为 稍微的不小心，可能就会带来程序的运行错误。对于页面的维护来说，非常不方便。即便是你自己写的代码，过一段时间你也可能有所遗忘，修改起来一样要备加小 心。

现在有了模板，你可以把PHP代码抽取出来。 我们可以这样修改second.htm文件。

<HTML>  
<HEAD>  
<TITLE>第二个模板文件</TITLE>  
</HEAD> <BODY>  
下面是一个列表  
<UL>  
<li>{ name1} 的身高是 {tall1}  
<li>{name2} 的身高是 {tall2}  
<li>{name3} 的身高是 {tall3}  
<li>{name4} 的身高是 {tall4}  
</UL>  
</BODY>  
</HTML>

建立second.php：

<?php  
include (’template.inc’);  
$tpl = new Template;  
$tpl->set\_file(’main’, ’second.htm’);  
$link=mysql\_connect(’localhost’,'fuyatao’,'passwd’);  
$result=mysql\_db\_query(’fyt’,’select name,tall from my\_table’);  
$i=0;  
while ( $a = mysql\_fetch\_array($result))  
{  
$i++;  
$tpl->set\_var(’name’ . $i, $a[name]);  
$tpl->set\_var(’tall’ . $i, $a[tall]);  
}  
$tpl->parse(’mains’, ‘main’);  
$tpl->p(’mains’, ‘main’);  
?>

在浏览器里执行 second.php 一样会得到正确的结果。这样以后不管是你还是美工修改页面的时候，再也不用胆颤心惊、担心会把程序弄坏了。

**phplib template中文教程（四）**

第四章 block的应用

在上一章中，我们使用模板把html和php代码分开。其中second.htm模板中关于列表部分是这样写的：

<UL>  
<li>{name1} 的身高是 {tall1}  
<li>{name2} 的身高是 {tall2}  
<li>{name3} 的身高是 {tall3}  
<li>{name4} 的身高是 {tall4}  
</UL>

是不是有点烦琐？本来没用模板之前，我们用了一句

while ($a = mysql\_fetch\_array($result)){  
echo “<li>$a[name]的身高是$a[tall]”;  
}

就搞定了，用了模板，反而要写4次，如果是100人的列表，岂不是我们要写100次？天啊！够吓人的。好在phplib templete为我们提供了block，将其真译为“块”。通过定义块，也可以用一行轻松搞定。

<UL>  
<!– BEGIN list –>  
<li>{name} 的身高 {tall}  
<!– END list –>  
</UL>

需要注意的是，BEGIN list与前后的中划线–之间必须有空格，前后中划都是两个，此处可能由于显示太小，两个中划线连到一块儿了。

在 second.php中需要加载块。

<?php  
include (’template.inc’);  
$tpl = new Template;  
$tpl->set\_file(’main’, ’second.htm’);  
$tpl->set\_block(’main’, ‘list’, ‘nlist’); //加载模板main中的块list,并给其一个名字nlist  
$link=mysql\_connect(’localhost’,'fuyatao’,'passwd’);  
$result=mysql\_db\_query(’fyt’,’select name,tall from my\_table’);  
while ( $a = mysql\_fetch\_array($result))  
{  
$tpl->set\_var(’name’ , $a[name]);  
$tpl->set\_var(’tall’ , $a[tall]);  
$tpl->parse(’nlist’, ‘list’, true);  
}  
$tpl->parse(’mains’, ‘main’);  
$tpl->p(’mains’);  
?>

**phplib template中文教程（五）**

第五章 模板嵌套

在写PHP页面的时候，我们会发现像页面头部分和尾部分，有好多页都要用，每页写一遍就太麻烦了，这些”公用代码”我们可以把它单独写在一个文件 里，如果这个公用部分要有所改动,无需再去改每一个页面，能减少非常多的工作量。用Template模板可以很方便的把一个页面随意插入另一个模板的任意 地方。  
新建3个文件third.htm、header.htm、footer.htm，内容分为如下  
third.htm

<!– 这是页面头部 –>  
{header}  
<BODY>  
下面是一个列表  
<UL>  
<!– BEGIN list –>  
<li>{name} 的身高是 {tall}  
<!– END list –>  
</UL>  
<!– 这是页脚部分 –>  
{footer}  
</BODY>  
</HTML>

header.htm

<HTML>  
<HEAD>  
<TITLE> {title} </TITLE>  
</HEAD>

footer.htm

<P>author © fuyatao

下面我们开始我们的PHP程序:

<?php  
include (’template.inc’);  
$tpl = new Template;  
$tpl->set\_file(’main’, ‘third.htm’);  
$tpl->set\_file(’my\_header’, ‘header.htm’);  
$tpl->set\_file(’my\_footer’, ‘footer.htm’);  
$tpl->set\_var(’title’, ‘这个是网页标题’);  
$tpl->set\_block(’main’, ‘list’, ‘nlist’);  
$link=mysql\_connect(’localhost’,'fuyatao’,'passwd’);  
$result=mysql\_db\_query(’fyt’,’select name,tall from my\_table’);  
while ( $a = mysql\_fetch\_array($result))  
{  
$tpl->set\_var(’name’ , $a[name]);  
$tpl->set\_var(’tall’ , $a[tall]);  
$tpl->parse(’nlist’, ‘list’, true);  
}  
$tpl->parse(’header’, ‘my\_header’);  
$tpl->parse(footer, my\_footer);  
$tpl->parse(’mains’, ‘main’);  
$tpl->p(’mains’);  
?>