

PHPer

增刊

09
2008

PHPChina.com
电子期刊

第二届PHP调查圆满结束



PHPChina | PHP中国开源之路

卷首语

第二届 PHP 调查圆满结束

由 PHPChina 主办, CSDN 协办的“第二届 PHP 中国开源与人才发展状况调查”活动于近日正式结束, 共有近三千名 PHP 爱好者及技术管理人员参与了此调查。

本次调查依据统计学理论和行业惯例, 本次调查主要采用互联网调查方式进行, 通过在中国 28 家网络媒体和 47 家重点网站相关频道投放网幅广告和文字链接广告, 将问卷放置在(调查链接: <http://www.phpchina.com/2008php>)在 PHPChina 官方指定网站上, 于 2008 年 7 月 1 日至 2008 年 8 月 1 日期间, 由用户自主填写并提交问卷的方式来获取信息。

主要采用频数分析、交叉列联分析、轮廓图分析等统计分析方法从调研数据中提炼信息, 并通过一系列的条形图、饼图、柱图、折线图和百分比堆积图等直观的展示分析结果。

较首届 PHP 调查结果不同的是, 第二届的调查中不同等级的参与者比例有很大变化。其中 23% 的参与者属于听说过, 但不会使用的朋友, 29% 属于了解, 偶然使用者, 非常熟悉的参与者达到了 45% 的比例。值得一提的是, 今年的调查增加了针对 PHP 应用企业部分的调查, 这部分参与者的比例达到了 3%。虽然今年调查受暑假和奥运会的影响, 样本收集数量方面不如去年。但在质量上较去年有很大改观, 从结果上我们能看出, 国内 PHP 的发展更为成熟了。

下面根据《PHPer》增刊所面向的软件工程师以及项目经理, 整理了一些这方面的调查资料, 以供参考:

开发 PHP 的 IDE 环境

67.7% 的 PHP 从业者是在 Zend Studio 环境下开发 PHP, 另有 27.7% 的从业者在 Eclipse 环境下开发 PHP, 使用过 PHPEdit 的从业者比例是 23.6%, 在 PHP 设计器环境下开发 PHP 项目的有 11.6%。PhpED、Komodo 的使用率分别为 5.1% 和 2.5%, 如图 1 所示。

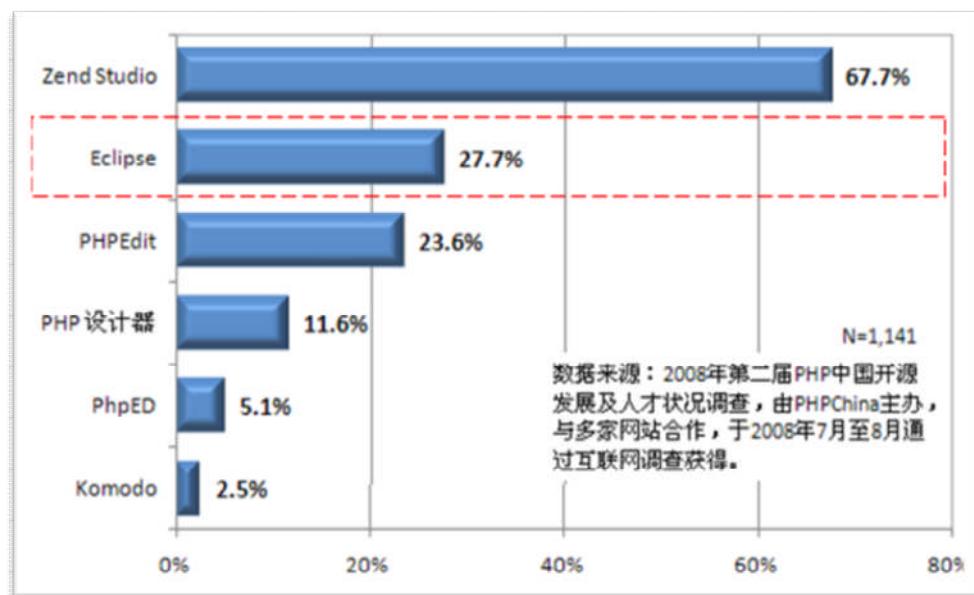


图 1 PHP 从业者使用的 IDE 环境

使用的模板技术

Smarty 模板技术的使用者比例为 80.5%，在 2007 年的调查中该数字是 70.3%，可见 Smarty 模板技术的使用者大幅增长。使用 PHPlib 模板技术的从业者约占 19.1%，使用 Template Lite 的占 10.5%，而使用 XTemplate 模板的从业者比例仅为 3.5%，此外还有 10.3% 的从业者没有使用过模板技术，如图 2 所示。

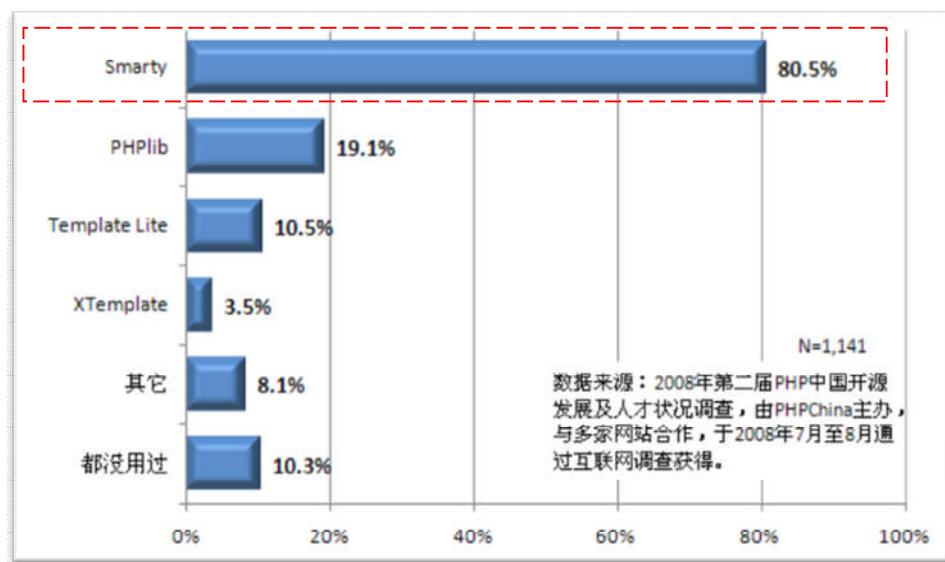


图 2 PHP 从业者使用过的模板技术

使用过的开发框架

在本次调查中发现，有 41.7% 的 PHP 从业者使用 Zend Framework 开发框架，使用自己的框架的从业者比例是 36.6%，FleaPHP/QeePHP 框架的使用者占 27.2%，ThinkPHP 的使用比率在 23.3% 左右。CakePHP、CodeIgniter 和 Symfony 这几种开发框架的使用比率都相对较低，分别为 13.1%、8.9% 和 5.1%，如图 3 所示。

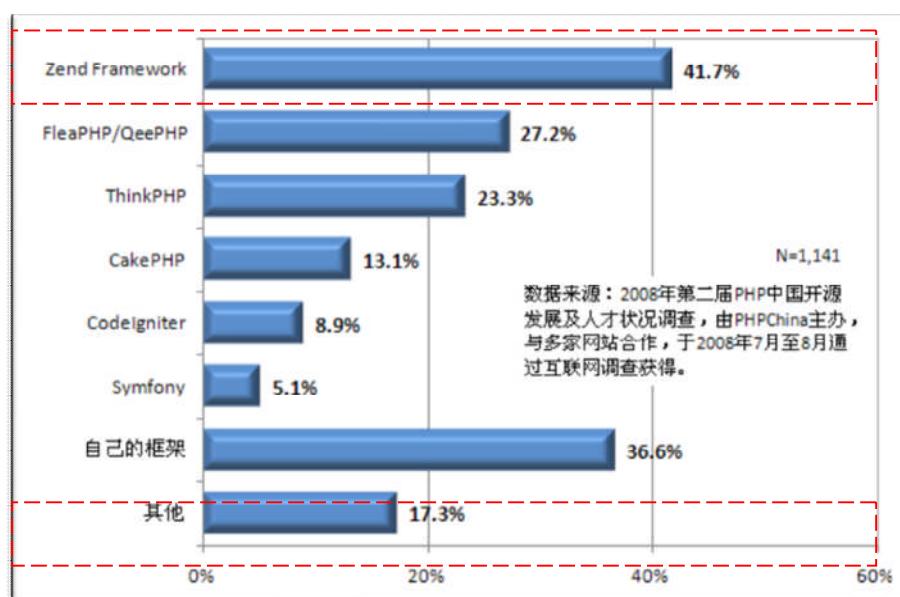


图 3 PHP 从业者使用过的开发框架

PHPPer 贡献者名单

2008 年 9 月 1 日

增刊

主管: PHPChina.com

主办: PHPChina网站会员

网址: <http://www.phpchina.com>

总编: PHPChina

副总编: 廖宇雷

卷首语: 刘 昊

开源项目: 陈 明 / CodeIgniter中国

企业解决方案: 僮燮水

扩展实例: 简张桂

高级应用: 廖宇雷

项目管理: Watts S.Humphrey

校验排版: PHPChina

后期制作: 风吹过的夏天

另外, 欢迎广大PHP程序员踊跃投稿!

请将稿件发送至

phper@phpchina.com

请注明所投栏目, 并附上作者简介。

PHP中国开源社区感谢您的参与!

目录

封面.....	1
卷首语.....	2
第二届 PHP 调查圆满结束.....	2
开发 PHP 的 IDE 环境.....	2
使用的模板技术.....	3
使用过的开发框架.....	3
PHPER 贡献者名单	4
目录.....	5
开源项目介绍.....	7
基于 THINKPHP 的身份证查询系统.....	7
一、项目要求.....	7
二、模块分析.....	7
三、数据库及模型.....	7
四、操作实现.....	9
简化 HTML 页面和表单的设计.....	13
编写视图.....	13
PHP 语法的长格式和短格式	14
嵌套视图	15
网站架构的现实问题.....	17
CI 的表单辅助函数：输入数据.....	18
使用表单辅助函数的好处之一：清楚.....	19
使用表单辅助函数的好处之二：自动化.....	21
我的“Display”模型	22
CI 的验证类：方便地检验数据.....	23
设置校验	24
设置控制器.....	24
设置表单	24
总结	26
企业解决方案.....	27
MooPHP 开发 MANYOU 应用实例讲解.....	27
开发前言	27
关于漫游平台：	27
开发简要	27

开发细节	28
扩展实例	36
PEAR 使用教程（三）	36
1 介绍	36
2 数据库查询	36
3 模板系统	36
高级应用	45
用 VIM 做 PHP 开发环境	45
安装 VIM	45
开始配置 VIM	45
打造 PHP IDE	50
后记	60
项目管理	62
成为团队领导者	62
1 什么是领导力	62
2 成为领导者或管理者	64
3 领导角色	66
4 在领导的同时进行训练	72
5 前路挑战	74
6 小结	76
学专业课程 获权威认证	78
封底	81



开源项目介绍

基于 ThinkPHP 的身份证证查询系统

程 明

大家好，很高兴能够和大家一起分享 ThinkPHP 的魅力。这里以一个非常简单的企业级应用为例，旨在抛砖引玉。

一、项目要求

假设某公司，要求实名审核所有员工的身份信息，包员工姓名、员工编号，并要求上传身份证件图片。之后输入员工编号、员工姓名及姓名拼音的首字母，就能够查询到符合条件的员工信息。

二、模块分析

由于项目很简单，甚至用一个模块就能够完成所有的功能了。但是，我们应该考虑到，使用者至少应该分为操作员和管理员两种，为了很好的把权限区分开来，我们可以将查询、结果显示划分到默认的 index 模块；操作员具备权限，将增加、编辑、删除操作划分到 info 模块，这个是只有管理员才具有操作的权限（至于权限分配请参考 PHPer 第 10 期流年的文章《基于 RBAC 的权限访问控制设计》，这里略过）。

三、数据库及模型

为了更好的体验出 ThinkPHP 中视图模型的强大，这里用了两个表，一个用来保存员工的常规信息（员工编号、员工姓名、姓名拼音的首字母），一个用来保存证件信息。这样对搜索也是有好处的。

SQL 脚本

```
-- 数据库: `membercard`  
-- 表的结构 `member`  
CREATE TABLE `member` (  
    `id` int(11) NOT NULL auto_increment,  
    `cardId` varchar(11) NOT NULL,  
    `name` char(50) NOT NULL,  
    `pyShort` char(18) NOT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;  
  
-- 表的结构 `pic`  
CREATE TABLE `pic` (  
    `memberId` int(11) NOT NULL,  
    `pic` char(50) default NULL,  
    PRIMARY KEY (`memberId`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

因为对这两个表都要进行操作,所以首先要分别为这两个表各建立一个模型。接下来,还要建立一个视图模型。因为当我们通过 member 表查询到了员工的信息后,还需要根据 member 表的主键 id 在 pic 表中获取对应的证件图片文件名。这就是一个简单的双表关联查询,用 ThinkPHP 中的强大的视图模型可以很方便的实现。

PHP 代码 viewMode.class.php

```
class viewModel extends Model{
    //这句申明该模型是一个视图模型
    protected $viewModel = true;
    //规定此视图模型要从以下两个表中各获取哪些字段
    protected $viewFields = array(
        'member' => array('id','cardId','name'),
        'pic'=>array('pic')
    );
    //两个表结合的条件是 pic.memberId = member.id
    protected $viewCondition = array(
        'pic.memberId' => array('eqf','member.id'),
    );
    //返回此模型的主键,其实就是 member.id。
    public function getPk(){
        return 'id';
    }
}
```

这样,我们只要 Action 中 “D(‘view’)->findAll();” 就能得到所有的数据了,如图 1 所示:



图 1 视图模型示例

这里不过是视图模型的冰山一角，更强大的应用，正等着你来发掘！

四、操作实现

限于篇幅，这里只着重说下新增数据的操作。

新增操作因为需要上传图片，并且要实现证件图片的预览，如图 2 和图 3 所示：



图 2 图片上传



图 3 图片预览

我使用了 SWFUpload, 以求提供更好的用户体验。这些只是表现层 (View) 的问题, 控制层 (Controller) 还是和传统的文件上传一样的写:

PHP 代码片段

```
public function upload(){
    //导入 TP 的 UploadFile 类库, 使用方法详见 TP 官方手册
    import('ORG.Net.UploadFile');
    $upload = new UploadFile();
    $upload->maxSize = 2048000;
    $upload->allowExts=array("jpg","jpeg","gif","png");
    $upload->savePath="../public/uploads/"; //保存在与项目目录同级的 public\uploads 下
    if ($upload->upload()){

        //如果上传成功, 则将图片保存的文件名写入 session, 因为前面说了, 新增时需要预览图片,
        第一次可以通过 Ajax 来实现预览, 但如果用户刷新了页面, 则要靠 session 或 cookie 来保证图片
        的预览了。
        $info=$upload->getUploadFileInfo();
        session::set("savename",$info[0]['savename']);
        echo $info[0]['savename'];
    }else{
        //如果上传失败, 则返回一个提示错误的图片名给 Ajax, 用来显示给用户。
        echo "error.jpg";
        //$$this->error($upload->getErrorMsg());
    }
}
```

现在图片上传好了, 如何入库呢, 这个问题是很多 ThinkPHP 初学者想问的。好, 那么下面来看下将员工信息写入数据库的操作 insertInfo 是如何实现的:

PHP 代码片段

```
public function insertInfo(){
    //前面的 upload 操作设置的 session 变量, 如果不存在, 则说明员工的证件图片没有上传成功
    if (!session::is_set('savename')){
        $$this->error("<a href=\"".__APP__."/info/add".$this->static_suffix.">
            请选择好要上传的证件图片</a>");
    }elseif (!checkGPC('post','num') || !checkGPC('post','name')) { // 检测表单完整性
        $$this->error("<a href=\"".__APP__."/info/add".$this->static_suffix.">
            请将表单填写完整</a>");
    }else{
        $input=Input::getInstance(); //使用 Input 类库过滤用户输入的数据
        $input->filter('addslashes');
        //导入项目目录下 vendor 中的自定义类库, 它的功能是将汉字转为拼音
        vendor('Zh2py','vendor/');
        $info = array(
            'cardId' => $input->post('num'),
            'name' => $input->post('name'),
            'pyShort' => Zh2py::toFirst($input->post('name'))
        );
        if (!checknum()){ //检测员工编号的唯一性, 也可以用 TP 自带的数据验证来处理
            $db = D('member');
```

```
$insert=$db->add($info); //写入数据库
if ($insert){
    $data=array(
        'memberId' => $insert,
        'pic' => session::get('savename') //因为图片名已经在上传成功时写入了 session 了,
    这里直接取出来就可以了。同样，这里也可以在模型里设置自动填充。
    );
    $pic = new picModel($data);
    if ($pic->add()){
        session::set('savename');
        $this->success("信息新增成功!");
    }else{
        unlink("../public/uploads/".$session::get('savename'));
        session::set('savename');
        $this->error("<a href=\"".__APP__."/info/add".$this->static_suffix."\">".
                     .$pic->getError()."</a>");
    }
}else{
    unlink("../public/uploads/".$session::get('savename'));
    session::set('savename');
    $this->error($pic->getError());
}
}else{
    $this->error("错误:员工编号已存在!");
}
}
}
}
```

这样，数据就写入了数据库了。现在可以看下查询出来的结果了，如图 4 所示：



图 4 查询结果

是不是觉得很简单呢? 心动不如行动, 也自己动手写一个吧, 毕竟, 实践才能出真知。

最后想说的是, 欢迎越来越多的 PHPer 加入到 ThinkPHP 的阵营, 相互学习, 共同提高!

注: 文中涉及到例子的完整代码在“附件/memberCard/”中。

作者: 程明

Email: eclanp@qq.com

简化 HTML 页面和表单的设计

David Upton (著)

CodeIgniter 中国 (译)

本章介绍了另一种节约你的时间，并使你的代码更具安全性和逻辑性的方法。

首先，我们将会介绍创建视图的各种不同方法——与你的控制器和模型协同并用来显示结果的页面。

然后，你将会学到如何快速地创建 HTML 表单，并实现内建的保护，而且你还将看到如何校验你的表单。

我假定这本书的读者熟悉 HTML 和 CSS。下面的例子非常简单，因此，我们能把重点放在 CI 代码上。而且我假定我们已经写了一个 CSS 文件并已把它保存在网站的某个目录中。

编写视图

视图控制着你的网站怎样呈现给用户。他们使用一个统一的接口，而且可以根据需要进行修改。MVC 的好处之一是分离了表示层和逻辑层，一切都显得很干净。

到目前为止，我们已经完成了非常简单的“welcome”页面，（记得第 3 章吗？）现在让我们看看如何使它变得更精细。

视图实际上是一组含有你的内容的 HTML 结构。结构中有各种元素，如颜色、字体等；不过视图不关心这些，它要做的只是取来内容，显示出来。

创建视图，首先你要创建一个 HTML 页面的骨架，并保存为 PHP 文件。让我们称它为 basic_view.php。保存在 application/views 目录中。（CI 的装载器会在这个目录寻找视图文件。）

basic_view.php

```
<html>
<head>
</head>
<body>
<p>Hello World!</p>
</body>
</html>
```

然后当你想要从一个控制器装载它时，在函数中使用\$this->load->view():

代码片段

```
function index() {
    $this->load->view('basic_view');
}
```

注意,如果这是一个模型或者一个辅助函数,你应该先装载它,然后根据需要使用它。如果是视图,调用它只需要 1 行代码。

当然,那是一个空的视图。为了让它有用,我们需要内容。因此我们要增加标题和一些文本。首先我们在控制器中定义他们:

代码片段

```
function() {  
    $data['mytitle'] = "A website monitoring tool";  
    $data['mytext'] = "This website helps you to keep track of the other websites you control";  
}
```

注意,我们并没有把它们定义为单独的变量,而是作为数组\$data 的元素。对于第一个元素,键名是“mytitle”,值是“A website monitoring tool”。

然后,我们调用装载函数:

代码片段

```
function index() {  
    $data['mytitle'] = "A website monitoring tool";  
    $data['mytext'] = "This website helps you to keep track of the other websites you control.";  
    $this->load->view('basic_view', $data);  
}
```

我们把\$data 数组作为\$this->load->view()的第二个参数。视图接收到\$data 数组后,使用 PHP 函数 extract() 把数组中的每个元素转换成内存变量,数组的键名即为变量名,值为变量内所包含的值。这些变量的值能直接被视图引用:

代码片段

```
<html>  
<head>  
</head>  
<body>  
    <h1 class="test"><?php echo $mytitle; ?> </h1>  
    <p class="test"><?php echo $mytext; ?> </p>  
</body>  
</html>
```

虽然你只能传送一个变量到视图,但是通过建立数组,你能把大量变量整洁地传入视图。这看起来复杂,但实际上是一种紧凑和优秀的信息传输方式。

PHP 语法的长格式和短格式

在我们继续之前,先了解一下 PHP 标记的两种不同格式。

常用的方式是:

代码片段

```
<?php echo $somevariable?>
```

然而, 如果你不喜欢这种, CI 也支持一个较短的版本:

代码片段

```
<?= $somevariable ?>
```

<?php ?>被<? ?>取代, 而且 echo 由“=”代替, 你也能用短格式来使用 if、for、foreach 和 while 循环。完整的介绍请参考在线《用户指南》。

我个人偏好标准格式, 因为我已习惯使用它了。如果你使用短格式, 注意有些服务器不能正确地解释短格式。如果你仍然想要使用短标签, 可以打开 config 文件, 改变下列设置:

代码片段

```
<?php  
$config['rewrite_short_tags']= FALSE;  
?>
```

如设置为 TRUE, CI 在把它们送到服务器之前, 将把短格式改成标准格式。不过这样做会对调试造成困难。因此, 建议使用标准格式。

CI 也有一个“模板语法分析器”类, 允许你把变量放入 HTML 代码而不需要任何的 PHP 代码。本文不涉及这个内容。如果在与可能被 PHP 代码弄糊涂的 HTML 设计者合作时, 它相当有用, 更多内容请参考《用户指南》。

嵌套视图

为了最大程度地重用代码, 我们可以提取 HTML 页面的公共部分, 例如 header 和 footer, 然后在显示实际视图时把它们组合起来。

让我们创建一个视图的 header 部分, 这是一个符合 W3C 标准的 header, 包含 HTML 声明和 meta 数据。

首先, 我们列出 header 部分的代码:

header_view.php

```
<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 strict//EN' 'http://www.w3.org/TR/xhtml1/DTD/  
/xhtml-strict.dtd'>  
< html xmlns='http://www.w3.org/1999/xhtml'>  
<title><?php echo $mywebtitle?> </title>  
<base href=<?php echo "$base"; ?> >  
<?php echo $myrobots?>  
<link="stylesheet" type="text/css" href="<?php echo "$base"; ?> ">
```

把它保存为 views/header_view.php。下面介绍它包含的变量:

- **\$mywebtitle**, 是标题 (meta 标签; 它不在屏幕上显示, 但是搜索引擎将会读取, 每个页面的 meta 有可能有变化, 因此, 我把它设为一个变量。);
- **\$myrobots**, 这个变量用来告诉机器人当前页面不需要被编入索引;
- **\$base** 和 **\$css**, 描述基本网址的字符串。\$css 包含 CSS 文件的路径信息, 这些信

息会从 CI 的 config 文件读取 (也可以用 CI config 变量 site_url 代替。)

现在我们需要知道:

- 我们如何调用嵌套视图?
- 我们如何获取变量的值?

有两个方法可选择。第一, 在主调用视图中读入其它视图, 因此我们的主视图, 也就是 basic_view, 应该加上一行:

代码片段

```
<html>
<head>
<?php $this->load->view('header_view'); ?>
</head>
<body>
<?php echo $mytitle; ?>
<?php echo $mytext; ?>
</body>
</html>
```

变量可以在控制器中加上两行:

代码片段

```
function index() {
    $data['mytitle'] = "A website monitoring tool";
    $data['mytext'] = "This website helps you to keep track of the other websites you control.";
    $data['myrobots'] = '<meta name="robots" content="noindex,nofollow">';
    $data['mywebtitle'] = 'Web monitoring tool';
    $data['base'] = $this->config->item('base_url');
    $data['css'] = $this->config->item('css');
    $this->load->view('basic_view',$data);
}
```

在这里新的变量 \$myrobots、\$css、\$base、\$mywebtitle 被创建为数组 \$data 的新元素, 当 header_view 被 basic_view 调用时, CI 使用 extract() 解开他们, 在视图中显示出来 (在两个视图中不要出现同名的变量, 否则会引起冲突)。

第二个方法是把视图加入到控制器里面, 给它分配一个变量:

代码片段

```
function index() {
    $data['mytitle'] = "A website monitoring tool";
    $data['mytext'] = "This website helps you to keep track of the other websites you control.";
    $data['myrobots'] = '<meta name="robots" content="noindex,nofollow">';
    $data['mywebtitle'] = 'Web monitoring tool';
    $data['base'] = $this->config->item('base_url');
    $data['css'] = $this->config->item('css');
    $data['header'] = $this->load->view('header_view', "", TRUE);
    $this->load->view('basic_view',$data);
}
```

从严格的 MVC 设计原则来看，这样做似乎更正确。

实际上有三个参数可传给 load->view 函数。

- 第一个，header_view，是要装载的视图名称，这是必选的；
- 第二个，是可选项，是装入视图的数据；
- 第三个是布尔值。如果你不指定它，默认是 FALSE，将视图送到浏览器。然而，在嵌套视图这种情况下，你需要将 header_view 送到主视图 basic_view 中，因此需要将第三项参数设置为 TRUE。

现在我们已经建立了与样式表的关联，我们能够用定义好的 CSS 中的类来更新视图中的显示部分：

代码片段

```
<html>
<head>
    <?php $this->load->view('header_view'); ?>
</head>
<body>
    <h1 class="test"><?php echo $mytitle; ?></h1>
    <p class="test"><?php echo $mytext; ?></p>
</body>
</html>
```

请注意 CI 的 MVC 系统能让你分离显示的内容。视图只为内容提供结构，结构的样式则由 CSS 样式表控制。

视图不关心 \$mytext 的内容是什么，它只是按照正确的格式在正确的位置上显示它。定义 \$mytext 的控制器甚至不知道（也不关心）它产生的数据如何被显示。

因此，如果我们需要改变我们网页的外观，或在一个不同的系统（如 WAP）上显示它们，那么我们只需要修改视图和 CSS 样式表。我们不需要修改控制器。

而且，如果我们想要修改页面上的信息，我们不需要去修改视图，而只需要修改控制器里的变量值。

记得“松耦合”原则吗？这里再一次体会到了这个原则，这使设计、升级和维护你的网站更容易。

网站架构的现实问题

请稍等片刻，我们在 header_view 中动态地生成了 CSS 样式表地址：

代码片段

```
<link="stylesheet" type="text/css" href="<?php echo \"$base/$css\";?>">
```

这意味着控制器必须生成变量的值，这些值与数据如何被显示有关，但是我们在上面说过控制器不应该知道或者关心它们具体的值是什么。这样不就符合了我们刚才提及的

“松耦合”原则了吗？动态地产生这些数据需要这样一些操作：首先，控制器必须在 config 文件中读取它们，然后控制器必须在 \$data 数组中装入它们并传送到视图，然后视图必须解开成为内存变量 \$base 和 \$css，真正使用这两个变量的是 HTML 协议。

似乎这样做太绕圈子了，为什么不直接在视图中静态地插入数据？

代码片段

```
<link="stylesheet" type="text/css" href="http://www.mysite.com/mystylesheet.css">
```

用变量方式做这件事情的好处是：如果你迁移网站，或移动你的 CSS 文件，你只需要在 config 文件中更改设置，然后每个控制器和视图将会立刻反映变化。而如果把 CSS 位置硬编码到每个视图的后果是一旦需要修改，你必须到每个视图中去修改它们，明白这样做的好处了吧？

CI 的表单辅助函数：输入数据

让我们把目光转向如何使用你的 HTML 页面。任何动态网站最重要的部份之一是和用户互动，而且这通常意味着使用 HTML 表单。编写表单是重复和无聊的。

CI 的表单辅助函数是非常有用的代码片段。它有一个稍稍不同的语法，使表单创建起来更容易。让我们建立一个表单，这个表单允许我们在浏览器中输入数据。在 websites 数据库的 sites 表中，我们想要输入网站的名字、类型和网址，还有更新的日期。

你可以用简单的 HTML 代码建立表单，或者你可以在一个控制器内建立它，把它赋给一个变量，然后调用视图，并且传送该变量到视图。我正在按第二种方式做。

首先，我们必须装载表单辅助函数到我们需要使用它的控制器内。然后，我们把下列代码放入控制器的构造函数中：

代码片段

```
$this->load->helper('form');
```

然后，我们开始编写表单。

现在，为了生成表单的输入项，我们不用这样写：

代码片段

```
$variable .= "<input type='text' name='name' value='>";
```

CI 让你这样做：

代码片段

```
$variable .= form_input('name','');
```

（记住“name”是输入项的名称，“value”是你想输入的内容。在这里可以设定 value 的初始值，或你能动态地从表单中获取。）

可以看到，使用 CI 的表单辅助函数很方便。

使用表单辅助函数的好处之一：清楚

使用 CI 表单辅助函数的第一个好处是你的代码绝对清楚。如果你想要一个比较精细的输入框，如果用 HTML 是这样的：

代码片段

```
$variable = '
```

`type` 是你的输入项的类型—`text`、`hidden` 等；

`name` 是将在`$_POST` 数组中取得的变量名称；

`id` 是在网页上定位这个输入框的标识符，如果你使用 JavaScript 的话；

`value` 是输入框里显示的值，它一开始是一个默认值，用户也可以输入一个新的值；

`maxlength` 和 `size` 顾名思义是最大输入长度和输入框的长度；

`style` 一组 HTML 格式或者在 CSS 样式表中定义。

CI 用一个数组代替上述的 HTML 代码：

代码片段

```
$data = array(  
    'name' => 'url',  
    'id' => 'url',  
    'value' => 'www.mysite.com',  
    'maxlength' => '100',  
    'size' => '50',  
    'style' => 'yellow',  
);  
$variable = form_input($data);
```

它看上去蛮长的，实际上并不比 HTML 代码长，而且，它非常清楚，容易理解和维护。而且是动态的。

“隐藏”表单输入项非常简单。如果我们想要自动地记录我们的数据库被更新的日期。我们把日期放入一个`$date` 变量，然后：

代码片段

```
form_hidden('updated', $date);
```

如果你想要一个“文本”输入框，给你的使用者提供一个可以输入超过一行的地方，可以使用 CI 的 `form_textarea()` 函数，下面的代码使用默认的长度，在网页上显示一个文本输入框：

代码片段

```
$data = array(  
    'name' => 'url',  
    'id' => 'url',  
    'value' => 'www.mysite.com',
```

```

    );
$variable = form_textarea($data);

```

CI 的表单辅助函数在你编写下拉框、多选框和单选框时特别有用，如果我们要改变我们的 URL 输入框为一个下拉框，允许用户从下拉列表中选取一个 URL。首先，把下拉列表的选项存入一个数组，然后调用 `form_dropdown()` 函数：

代码片段

```

$urlarray = array(
    '1' => 'www.this.com',
    '2' => 'www.that.com',
    '3' => 'www.theother.com',
);
$variable = form_dropdown('url'、 $urlarray, '1');

```

被传给表单中 `url` 下拉框的第一个参数是输入框的名字；第二个是包含下拉列表的数组，第三个默认选项。换句话说，如果使用者接受默认值，你的 `$_POST` 数组将会包含值 “`url=>1`”，但是你的用户将会见到选项 “`www.this.com`”。

如果使用 HTML 代码编写：

代码片段

```

<select name="type">
<option value="1" selected>www.this.com</option>
<option value="2">www.that.com</option>
<option value="3">www.theother.com</option>
</select>

```

CI 实现的代码实际上比较短，很容易学会。

如果你在一个数据库表（'urls'）中储存可能选择的网址的目录，那么生成一个动态下拉框很容易。首先把数据从表中读出放到一个数组中：

代码片段

```

$urlarray = array();
$this->db->select('id,url');
$query = $this->db->get('urls');
if ($query->num_rows() > 0) {
    foreach($query->result() as $row) {
        $urlarray[$row->id] = $row->url;
    }
}

```

然后重复我们以前用过的 CI `form_dropdown()` 函数：

代码片段

```

echo form_dropdown('type', $urlarray, '1');

```

只有 \$urlarray 的内容会发生变化；代码总是一样的。

如果你正在更新一个表中的记录而不是插入，你不想为你的用户显示默认值。你想要为那条记录显示已存在的值。你应该已经知道你想要修改的记录的 id 值，因此，你需要先读取数据库中“sites”表中的相关记录。把查询结果赋给一个变量，使用第二个变量取出第一个变量中的相关记录，再调用 CI 的 form_dropdown 函数，把第二个变量和对应的列名作为参数传入：

代码片段

```
$this->db->select('id, url, name');  
$this->db->where('id', '$id');  
$sitequery = $this->db->get('sites');  
$siterow = $sitequery->row();
```

然后 CI 下拉框函数会从中读取相关信息：

代码片段

```
echo form_dropdown('url', $urlarray, $siterow->url);
```

本文没有太多的篇幅讨论所有的表单辅助函数。它还能编写单选框、隐藏区域、多选框和一些其它内容，完整的资料请参考 CI 《用户指南》。

使用表单辅助函数的好处之二：自动化

使用表单辅助函数的第二个好处是可以自动化实现一些功能，不然的话，你只能自己编写相关的脚本了。

首先，它拦截 HTML 的一些字符，比如用户输入的引号，并且转义它们以免破坏表单。

其次，它自动链接。当你打开一个表单时，你必须声明目标页，它将会接受表单的数据并且处理它。（在 CI 中，这是一个控制器里面的一个功能而不是一个静态页。比如它指向控制器的更新函数。）因此，如果你用纯 HTML 代码，你将会这样写：

代码片段

```
<form method="post" action="http://www.mysite.com/index.php/websites/update" />
```

如果你用 CI，你只需要这样做：

代码片段

```
form_open('websites/update');
```

CI 自动地在你的 config 文件中取出基本 URL 并定位到对应的控制器函数。再次强调，如果你迁移你的网站，你只需修改 config 文件，而不是去一个一个地修改代码文件。

顺便提一下，CI 假定你的表单总是以 POST 的方式提交数据而不是 GET 方式。CI 普
网站：<http://www.phpchina.com> 投稿：phper@phpchina.com 《PHPer》 21/81

遍使用 URL 本身，因此，不要搞错。

我的“Display”模型

作为示范（稍微简化了一下），这里是我的 Display 模型：

代码片段

```
<?php
class Display extends Model {
    /*create the array to pass to the views*/
    var $data = array();
    /*two other class variables*/
    var $base;
    var $status = '';
    /*the constructor function: this calls the 'model' parent class, loadsthe CI libraries and helpers it
     * requires, and dynamically setsvariables*/
    function Display(){
        parent::Model();
        $this->load->helper('form');
        $this->load->library('user_agent');
        $this->load->library('errors');
        $this->load->library('menu');
        $this->load->library('session');
        /*now set the standard parts of the array*/
        $this->data['css'] = $this->config->item('css');
        $this->data['base'] = $this->config->item('base_url');
        $this->base      = $this->config->item('base_url');
        $this->data['myrobots'] = '<meta name="robots" content="noindex,nofollow" />';
        /*note that CI's session stuff doesn't automatically recall the extra variables you have added,
         * so you have to look up the user's status in the ci_sessions table*/
        $sessionid = $this->session->userdata('session_id');
        $this->db->select('status');
        $this->db->where('session_id', $sessionid);
        $query = $this->db->get('ci_sessions');
        if ($query->num_rows() > 0){
            $row = $query->row();
            $this->status = $row->status;
        }
    }
    /*function to assemble a standard page. Any controller can call this.
     * Just supply as $mydata an array, of key/value pairs for the contents
     * you want the view to display. Available variables in this view are:
     * mytitle, menu, mytext, diagnostic
     */
    function mainpage($mydata){
        $this->data['mytitle'] = 'Monitoring website';
        $this->data['diagnostic'] = $diagnostic;
        foreach($mydata as $key => $variable){
            $this->data[$key] = $variable;
        }
    }
    /*here's the menu class we looked at in Chapter 3*/
}
```

```
$fred = new menu;  
$this->load->library('session');  
$mysess = $this->session->userdata('session_id');  
if(isset($this->status) && $this->status > 0){  
    $this->data['menu'] = $fred->show_menu($this->status);  
}  
$this->load->view('basic_view', $this->data);  
}  
?  
>
```

我可以用下面的代码在任何控制器中调用这个页面：

代码片段

```
$this->load->model('display');  
$this->display->mainpage($data);
```

我也知道我的视图正在被动态地装配，完全符合我的需要。

CI 的验证类：方便地检验数据

在你编写 HTML 表单时一个重要的工作是检查输入。我们都知道我们应该这样做，但是直到现在为止，我们已经编写了一个简单的表单，它会信任地接受任何用户输入的任何数据。你应该意识到可能有一些用户是不怀好意的，并且剩下的也不都是负责任的。（别直接告诉他们。）如果他们有可能犯一个简单的错误，他们就会犯。确保你始终检查用户输入的数据，并使它们符合你的要求。

你能在客户端用 JavaScript 做到这一点，但是这样做的作用有限，用户能轻易地绕过它。而在服务器端的校验需要一个额外的信息来回，这点额外的开销是值得的。

编写校验代码也相当复杂，但是，你一定猜到了，CI 提供了一个校验类可以使这项工作变得非常容易。

让我们改变我们自己的表单处理过程来实现校验。你需要在表单里作一些调整，还要在它指向的函数里作一些调整。

如果你的表单由 `form_open('sites/update')` 开始，你需要修改的函数是“sites”控制器里的“update”函数。如果你没有使用 CI 的表单辅助函数，HTML 等价代码是：

代码片段

```
<form method="post" action="http://www.mysite.com/index.php/sites/update"/>
```

你需要做三件事情：

1. 设置校验
2. 设置控制器
3. 设置表单

设置校验

在你的表单指定的那个函数中装载验证类，并声明你的校验规则：

代码片段

```
$this->load->library('validation');
$rules['url'] = "required";
$rules['name'] = "required";
$this->validation->set_rules($rules);
```

“url” 和 “name” 输入框一定要有输入内容。CI 提供了各种操作，确保一些操作一定要进行，《用户指南》全面地解释了这些内容。他们的含义非常明了：min_length[6]显然意味着输入的信息长度一定要大于等于 6 个字符。numeric 意味着只能输入数字，等等。你还能组合规则，用 “|” 把它们连接起来：

代码片段

```
$rules['name'] = "required|alpha|max_length[12]";
```

上面代码意味着不能为空，字母、长度至少 12 个字符。你甚至能编写你自己的规则。

设置控制器

仍然在相同的函数中，创建一个 “if/else” 语句：

代码片段

```
if ($this->validation->run() == FALSE) {
    $this->load->view('myform');
} else {
    $this->load->view('success');
}
```

你进行确认测试，如果输入内容不能通过测试的话，就再返回到输入页面。（如果你在一个控制器内的一个函数中生成你的视图，则使用 `$this->myfunction` 代替 `$this->load->view('myform')`。如果校验成功，就生成 `view("success")`，告诉用户输入的信息已被接受，然后给出一个链接让他前进到下一步。

设置表单

输入信息的表单也要做相应的调整。每次校验没有通过的话，你不但要让系统返回到输入界面，而且必须说明哪一项出错，以及为什么出错。因此你必须在表单的某处给出一个附加信息：

代码片段

```
$this->validation->error_string;
```

这行代码显示适当的信息，避免用户在那里犯嘀咕。

你也需要自动地填写用户已正确输入的那些内容，否则，用户必须再次输入上一次他网站：<http://www.phpchina.com> 投稿：phper@phpchina.com 《PHPer》 24/81

们已经正确输入的信息。

首先，你需要在控制器里增加更多的代码。而且是立刻加在校验规则之后，加入一个数组来存放给用户的提示信息。数组的键名是你表单中的输入框名，值是给出的错误提示信息：

代码片段

```
$fields['url'] = '你的网址';
```

然后，增加一行代码：

代码片段

```
$this->validation->set_fields($fields);
```

现在你已经在控制器里声明了一个存有信息的数组，你只需要在表单内加入显示它们的代码。对于 HTML 代码，这会是：

代码片段

```
<input type="text" name="url" value=<?php echo $this->validation->url; ?>/>
```

或者，如果你正在使用 CI 的表单辅助函数：

代码片段

```
$variable = form_input('url', $this->valication->url);
```

如果使用这个表单插入一个新的记录到数据库表中，上面的代码已经够用了。如果你正在使用表单更新一个已经输入过的记录，当表单第一次显示时，应该在输入框中显示数据库表中的实际信息，这个时候，它的值应该是从数据库里取回来的（记得前面的例子吗？我们把代码再显示在这里：

代码片段

```
$this->db->select('id, url, name');
$this->db->where('id', $id);
$sitequery = $this->db->get('sites');
$siterow = $sitequery->row();
echo form_dropdown('url', $urlarray, $siterow->url);
```

如果你在更新一个现有的记录时，上一次的输入内容由于一个输入框内容没有输入而无法通过校验，在重新回到表单之前，你需要在通过校验的输入框中填写用户刚输入的信息，而在校验出错的输入框里再次放入从数据库表中读入的信息，否则，你就需要再次输入已经校验通过的信息了。

还好，这可以通过一个简单的“if/else”语句来实现：

代码片段

```
if (isset($_POST['url'])) {
    $myvalue = $this->validation->url;
} else {
```

```
$myvalue = $siterow->url;  
}
```

第一次表单显示的时候，`$_POST` 数组中没有内容；因此你应该从数据库的相关表中读取信息。但当你提交一次以后，`$_POST` 数组中就有数据存在了，所以你应该取 validation 函数中的返回的值。

查阅 CI 《用户指南》，了解表单验证的其它内容，你还可以做到：

- 自动地准备你的数据，举例来说，通过它消除可能产生的跨站脚本攻击；
- 编写你自己的复杂验证标准，举例来说，用户输入的值不能已存在于数据库中；
- 编写你自己的错误信息。

CI 的验证类非常有用而又功能强大，值得花时间好好研读并掌握。

总结

我们已经学习了 CI 中生成视图的方法，以及它如何让你创建“迷你一视图”，你可以把视图嵌套到其它视图中去。这意味你能建立共用的 HTML header 和 HTML footer，实现视图的重用。

我们也已经见到了 CI 如何帮助你编写 HTML 输入表单，通过表单辅助函数简化 HTML 表单的编写工作。

最后，我们学习了 CI 的验证类，这是检查用户输入信息的有用工具。没有什么是完美的，但是这个工具的确能阻止你的用户输入垃圾或企图进行的攻击。它也使你的网站看起来更专业，能够有效地捕捉用户造成的各种输入错误，而不是一味地接受无意义的输入。

在整个学习过程中，我们也再次玩味了 MVC 的原则，而且有时稍稍地做一些变通会让生活变得更容易。CI 有一种非常有柔性的哲学：如果要有效率地解决问题，就要学会灵活地使用工具。

CodeIgniter 中国简介

CodeIgniter 中国成立于 2007 年 10 月，由 Hex 一手创办而成。其后由沧蓝代表 CodeIgniter 中国与官方取得联系，得到了官方的支持。CodeIgniter 中国自成立以来，目标一直锁定在中国范围内推广和普及 CodeIgniter 的使用。我们希望通过推广和普及 CodeIgniter 框架，使中国程序员的 Web 开发水平达到一个新的高度。

CodeIgniter 中国以开发者社区为基础，面向所有的 PHP 爱好者，提供一个具有良好氛围的交流平台，提供 CodeIgniter 相关英文资料的中文化支持，也将为企业提供各种增值服务。我们将一如既往的为广大 PHP 爱好者提供最优质的服务！

Web: <http://codeigniter.org.cn>



企业解决方案

MooPHP 开发 ManYou 应用实例讲解

僭變水

开发前言

Manyou Open Platform (Manyou 开放平台/MYOP) 服务是由 Comsenz 公司为应用开发者提供的开放平台。

值得应用开发者注意的是，所有基于 MYOP 开发的应用程序，事实上逻辑层都是运行于应用开发者自己的服务器，经过 MYOP 的代理，或 iframe 方式，展现在各种 UCenter Home 用户网站上。因此 MYOP 提供的 API，是一套真正实现一次编写，到处运行的协议。

得益于 MYOP 与 UCenter Home 的紧密整合特性，对于访问者而言，在访问各种不同类型的 UCenter Home 用户网站时，例如股票 SNS、旅游 SNS、高校 SNS、地区 SNS 等等，均会感觉到这些 MYOP 应用程序是运行在 UCenter Home 用户网站上，从而获得十分良好的使用体验，而实际上这些应用程序所实现的功能，是由 UCenter Home+MYOP+应用服务器共同提供的。

关于漫游平台：

我的理解就是：Comsenz 提供的一个连通我们应用开发者和 Comsenz 上千万用户的纽带。

可以更简单的理解，Comsenz 给我们提供上千万的用户给你利用，只要的应用足够吸引用户，最主要 Comsenz 是义务免费的给我们提供。你更可以这样想想，你做一个新站的时候，你要从一个个 IP 做起来，而从漫游可以两周内上 100 万 PV，你可以想想这是什么概念。

当然我上面的理解完全是从一个开发小众，一个平常小站长去理解一个漫游平台对于自己的意义。实际漫游平台给予我们的机遇远远不止此。

开发简要

- TA 请客插件基于 MooPHP、0.93.322 alpha、manyou_api_client_0.1 开发的；
- ManYou api 接口使用到的有：user.getLoggedInUser、user.isAppAdded、friend.get、feed.publishTemplatizedAction、notification.send；
- ManYou MYML 使用到的有：my:tabs、my:editor、my:dialog、my:success、my:title、my:redirect；
- ManYou MYJS 使用到的有：setStyle、addEventListene、Dialog。

开发细节

1. 获取开发许可

获得漫游开发许可，这一步是我们进行 ManYou 应用开发的先决条件。注册并登录 <http://uchome.developer.manyou.com/uchome/>，在开始菜单找到“开发者”，点击进入，开始申请 ManYou 开发资格，如图 1 所示。

The screenshot shows the 'Developer Application Submission' form. It includes fields for application name, developer email, customer service email, callback URL, execution mode, service terms URL, developer name, application introduction, default MYML, personal homepage module size, developer mode, privacy URL, and secret installation. A 'Submit' button is at the bottom.

应用程序名称
（限制：8个汉字
中文计算为3个字符）
TA请客 *必填

我已阅读并接受 [MYOP应用服务协议](#).

开发者E-mail
（限制：100个字符）
kimi@ismole.com
如果有任何问题或更新，manyou将会通过这个email与您联系

客户服务E-mail
（限制：100个字符）
kimi@ismole.com
显示在该应用的介绍页面，用户可发邮件到此邮箱寻求帮助

Callback URL
（限制：100个字符）
<http://xxxx.ismole.com/temp/>
服务器上的应用所在目录的网址，必须以斜线结尾

应用页面的执行方式
 使用MYML 使用iframe

应用的服务条款的URL
[http://www.phpchina.com](#)
应用服务条款的网址，用户必须同意这些条款

开发者
潜变水

应用的简介
[http://www.phpchina.com](#)
将显示在用户安装操作的页面上

默认MYML
[http://www.phpchina.com](#)
将显示在用户个人主页上应用盒子里，如果用[Profile.setMYML](#)用盒子，这里的MYML将会重新解析。

默认个人主页模块大小
 宽 窄

开发者模式
 只允许您自己安装此应用

隐私URL
（限制：100个字符）
[http://www.phpchina.com](#)
链接到此应用的隐私设置或条款

秘密安装
 在好友动态、个人动态里不显示安装消息

提交

图 1 申请 ManYou 开发

申请完成后的界面如图 2 所示，取得 API Key 和 Secret 这两项是我们以后应用开发所需要的。

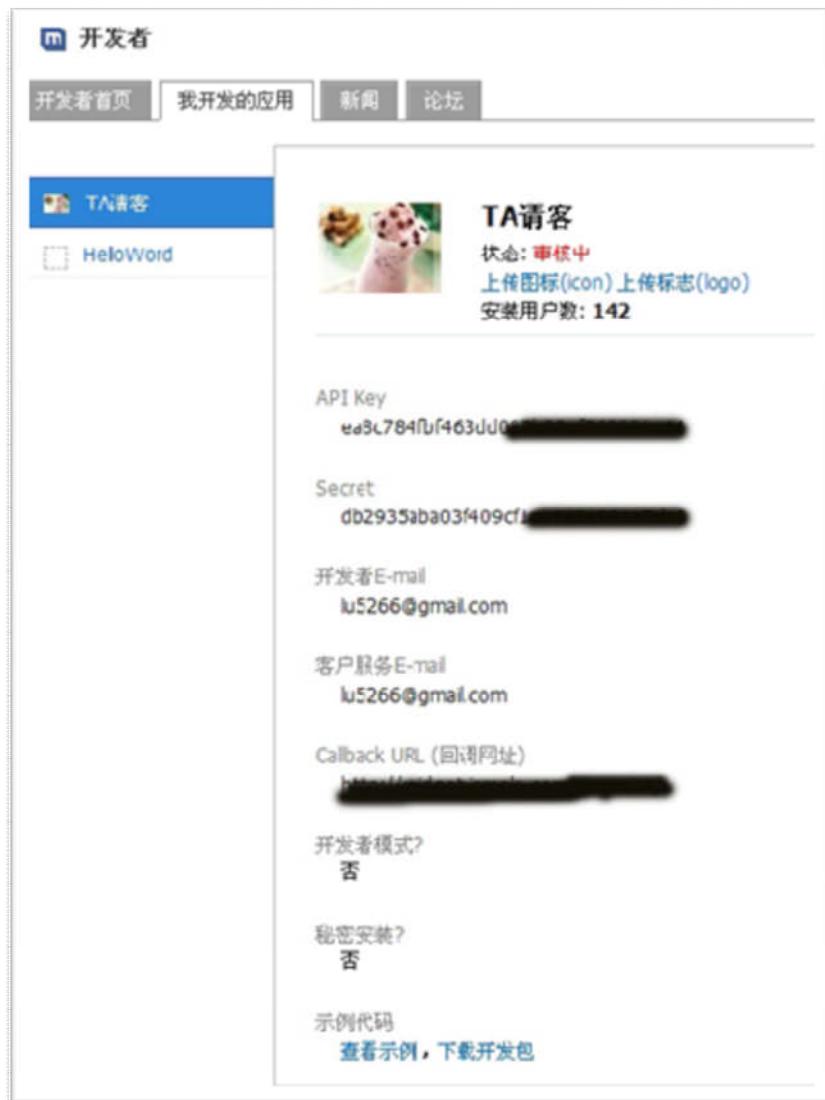


图 2 取得 API Key 和 Secret

2. 应用构思设计

做这个应用是基于这样一个想法的：当时一个同事发了一个网址，是关于谁请吃冰淇淋的小游戏。具体是这样一个样子的，一个人发起一个谁请冰淇淋的随机事件，然后找同事来参加，最后从参与的人当中随机抽出一个人来实现大家想吃冰淇淋的愿望。

挺有意思的一个构想，所有打算做到漫游应用上面。而我想我得更宽广点，不只是请冰淇淋而已，就变成了 TA 请客了。当然这些都得有我创建的记录、我参与记录、我朋友的参与记录。

3. 数据库设计

因为本来比较简单的逻辑，所以数据库的设计也比较简单。创建游戏，参与游戏就两个表即可。具体的结构见附件带的 sql.sql。

值得注意的“uid”为所有 Comsenz 的 uchome 用户在漫游的唯一标识 id，“site”为 Comsenz 的 uchome 在漫游上的站点 id。具体可查看如图 3、图 4 所示。

服务器: localhost ▶ 数据库: widget_icecream ▶ 表 : icecream_events

	字段	类型	整理	属性	Null	默认	额外
<input type="checkbox"/>	id	int(10)		UNSIGNED	否		auto_increment
<input type="checkbox"/>	site	int(10)		UNSIGNED	否	0	
<input type="checkbox"/>	subject	char(100)	utf8_general_ci		否		
<input type="checkbox"/>	description	char(255)	utf8_general_ci		否		
<input type="checkbox"/>	dateline	int(10)		UNSIGNED	否	0	
<input type="checkbox"/>	uid	int(10)		UNSIGNED	否	0	
<input type="checkbox"/>	nums	smallint(5)		UNSIGNED	否	0	
<input type="checkbox"/>	resultid	int(10)		UNSIGNED	否	0	
<input type="checkbox"/>	expiry	int(10)		UNSIGNED	否	0	
<input type="checkbox"/>	status	tinyint(1)			否	0	

↑ 全选 / 全部不选 选中项:

打印预览 规划表结构 ②
添加 1 字段 子表结尾 子表开头 于 id 之后 执行

索引: ② 已使用空间
 健名 类型 基数 操作 字段 类型 用法 语句
 PRIMARY PRIMARY 7 id 数据 7,651 字节 格式
 uid INDEX 无 site 索引 3,072 字节 整理
 uid INDEX 无 uid 统计 10,723 字节 行数

图 3 events 表

服务器: localhost ▶ 数据库: widget_icecream ▶ 表 : icecream_joins

	字段	类型	整理	属性	Null	默认	额外	操作
<input type="checkbox"/>	eid	int(10)		UNSIGNED	否	0		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	site	int(10)			否	0		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	uid	int(10)			否	0		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	dateline	int(10)			否	0		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	subject	char(255)	utf8_general_ci		否			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

↑ 全选 / 全部不选 选中项:

打印预览 规划表结构 ②
添加 1 字段 子表结尾 子表开头 于 eid 之后 执行

索引: ② 已使用空间 行统计
 健名 类型 基数 操作 字段 类型 用法 语句 值
 uid INDEX 无 site 数据 10,166 字节 格式
 uid INDEX 无 uid 索引 2,048 字节 整理 utf
 eid INDEX 无 eid 统计 12,214 字节 行数
 在第 1 列创建索引 执行
 行长度 ø
 行大小 ø
 创建时间 2008 年 07 月
 最后更新时间 2008 年 07 月

图 4 joins 表

4. 应用代码实现

和 ManYou 通讯: 这个极度的方便, 因为 ManYou 平台的主要开发语言为 PHP, 所以你会感觉所有的一切都是那么的熟悉。而且 ManYou 官方直接提供了 api client, 通过 client 那是非常方便的使用所有的 api, 我们把该 client 集成到了 MooPHP 的一个插件, 所以使用只需一句代码即可方便调用。

功能实现: 根据前面构思里面有要实现的功能模块有: 首页、新游戏、我参与的游戏、朋友的游戏, 查看某一个游戏的详情, 结果页面, 那我们只要分开模块各个实现即可。

具体的代码实现可以看源码, 附件提供的源码的基本的代码块都有简要的注释。下面给出一些基本的 MYJS MYML api 出现的文件列表:

使用到的漫游 MYJS:

setStyle 、 addEventListener、 Dialog, 出现的文件为./icecream/templates/new.htm

使用到的漫游 MYML:

my:tabs、 my:title, 出现的文件为./icecream/templates/header.htm

my:editor, 出现的文件为./icecream/templates/new.htm

my:success、 my:redirect, 出现的文件为./icecream/include/global.function.php

使用到的漫游 api 接口:

user.getLoggedInUser、 user.isAppAdded、 friend.ge, 出现的文件为./icecream/index.php

feed.publishTemplatizedAction、 notification.send, 出现的文件为

./icecream/include/new.inc.php

5. 具体代码举例

index.php 片段

```
if(!$localDebug) {
    //**本地测试时候, 不加载漫游环境
    //note 加载漫游插件
    MooPlugins('manyous');
    //note 初始化 Manyou 类库
    $manyous = & new Manyou($icecreamConfig['api_key'], $icecreamConfig['api_secret']);
    //note 不是在漫游应用内
    if(!$manyous->session_key) {
        exit('Error: Not In Manyou');
    }
    //note 检测用户是否已添加应用
    $manyous->require_add();
    //note 获取登录用户 id
    $uid = intval($manyous->api_client->user_getLoggedInUser());
    //note 站点 id
    $site = intval($manyous->current_site());
    //note 当前用户的所有信息
```

```

//UserFileds = 'uid, uch_id, name, handle, site, birthday, blood_type, relationship_status, current_location, hometown_location, has_added_app';
//UserInfo = $manyou->api_client->user_getinfo($uid, $userFileds);
if(in_array($a, array('friend', 'new'))) {
    //note 当前用户的所有好友
    $userFriends = $manyou->api_client->friend_get();
}
//*/
}else {
    //note 站点 id
    $site = 100;
    $uid = 1;
}
//note 链接 mysql
$db = MooAutoLoad('MooMySQL');
$db->connect($icecreamConfig['dbHost'], $icecreamConfig['dbUser'], $icecreamConfig['dbPasswd'], $icecreamConfig['dbName'], $icecreamConfig['dbPconnect'], $icecreamConfig['dbCharset']);
//note 根据预先定义, 加载相应处理的模块文件
if(in_array($a, $actions)) {
    require ICECREAM_ROOT."/include/{$a}.inc.php";
}else {
    showMessage('Sorry, this page was not available!');
}

```

view.inc.php 片段

```

if(!MooSubmit('icecreamSubmit')) {
    //note 已经到开结果的时间
    if($timestamp > $event['expiry']) {
        showMessage('你要查看的游戏已经结束, 请直接查看结果',
                    "index.php?a=result&eid=$eid", true);
    }
    //note 取得所有参加者
    $sql = "SELECT * FROM {$icecreamConfig['dbTablePre']}joins WHERE eid='$eid';";
    $joins = $db->getAll($sql);
    //note 已经参加的童鞋
    $joinUids = array();
    foreach($joins as $key=>$join) {
        $joins[$key]['dateline'] = sgodate("{$join['dateline']}");
        $joinUids[] = $join['uid'];
    }
    //note 毫秒时间
    $millisecondTimestamp = $timestamp * 1000;
    $millisecondExpiry = $event['expiry'] * 1000;
    $formatExpiry = sgodate("{$event['expiry']}");
    //note 到期跳转地址
    $jumpUrl = $manyou->get_url("index.php?a=result&eid=$eid");
    //note 邀请好友参加的地址
    $inviteUrl = $manyou->get_url("index.php?a=view&eid=$eid");
}

```

```

//note 是否已经参加
$joined = in_array($uid, $joinUids) ? true : false;
//note 包含解析模板
include MooTemplate('view');
}else {
    //note 过滤所需变量
    $yourSubject = MooHtmlspecialchars(trim(MooGetGPC('yourSubject', 'string')));
    //note 变量检查
    if(empty($yourSubject)) {
        showMessage("您的期待不可以为空,请返回", "index.php?a=view&eid=$eid");
    }
    //note 数据入库, 变量已经在 MooPHP 转义
    //note 更新参加人数
    $sql = "UPDATE {$icecreamConfig['dbTablePre']}events SET nums=nums+1 WHERE id='$eid'";
    $db->query($sql);
    //note joins 表
    $sql = "INSERT INTO {$icecreamConfig['dbTablePre']}joins (eid, site, subject, dateline, uid)
            VALUES ($eid, '$site', '$yourSubject', '$timestamp', '$uid')";
    $db->query($sql);
    //note 发送一个 feed, 让好友知道我参加游戏的动态
    $addFeed = MooGetGPC('addFeed');
    if($addFeed) {
        $feed = array();
        $feedDescription = $event['description'] ? $event['description'].',' : '';
        $feed['title_template'] = '{actor} {act} {view} ';
        $feed['title_data'] = "{$act}:"."参加了游戏", "view": "<a href=\"index.php?a=view&eid=$eid.\">
                                ".$event['subject']."</a>"';
        $feed['body_template'] = '{description} {join}游戏';
        $feed['body_data'] = "{$description}:".$feedDescription.", "join": "我想<a href=\"index.php?a=
                                view&eid=$eid.\">参加</a>"';
        $feed['body_general'] = "";
        $manyou->api_client->feed_publishTemplatizedAction($feed['title_template'], $feed['title_data'],
                                                            $feed['body_template'], $feed['body_data'], $feed['body_general'], ", , , , , , ");
    }
    //note 成功提示
    showMessage('参与游戏成功,点击跳转到查看游戏页面', "index.php?a=view&eid=$eid");
}

```

new.inc.php 片段

```

//note 过滤所需变量
$subject = MooHtmlspecialchars(trim(MooGetGPC('subject', 'string')));
$yourSubject = MooHtmlspecialchars(trim(MooGetGPC('yourSubject', 'string')));
$description = MooHtmlspecialchars(trim(MooGetGPC('description', 'string')));
$expiry = MooGetGPC('expiry');
$hiddenExpiry = MooGetGPC('hiddenExpiry');
//note 变量检查
if(empty($subject)) {

```

```
showMessage('游戏主题不可以为空,请返回', 'index.php?a=new');
}elseif(empty($yourSubject)) {
    showMessage('您的期待不可以为空,请返回', 'index.php?a=new');
}elseif(!$expiry || ($expiry == -1 && !$hiddenExpiry)) {
    showMessage('揭晓答案时间不可以为空,请返回', 'index.php?a=new');
}
//note 有效期处理
$expiry = ($expiry != -1 ? $expiry : $hiddenExpiry) * 60 + $timestamp;
$expiry = $expiry > 4294967295 ? 4294967295 : $expiry;
//note 数据入库, 变量已经在 MooPHP 转义
//note events 表
$sql = "INSERT INTO {$icecreamConfig['dbTablePre']}events (site, subject, description, nums,
dateline, uid, expiry) VALUES ('$site', '$subject', '$description', '1', '$timestamp', '$uid', '$expiry')";
$db->query($sql);
//note 取得本次游戏 id
$eid = $db->insertId();
//note joins 表
$sql = "INSERT INTO {$icecreamConfig['dbTablePre']}joins (eid, site, subject, dateline, uid)
VALUES ('$eid', '$site', '$yourSubject', '$timestamp', '$uid')";
$db->query($sql);
//note 通知我的好友
$notice = MooGetGPC('notice');
if($notice) {
    $comma = $toUids = '';
    foreach($userFriends as $friendUid) {
        $toUids .= $comma . $friendUid;
        $comma = ',';
    }
    $noticeMessage = "创建了游戏<a href=\"index.php?a=view&eid=$eid\">$subject</a>,
并邀请你 <a href=\"index.php?a=view&eid=$eid\">参加</a>。 ";
    $manyou->api_client->notification_send($toUids, $noticeMessage);
}
//note 发送一个 feed
$feedDescription = $description ? $description.'.' : '';
$feed = array();
$feed['title_template'] = '{actor} {act} {view} ';
$feed['title_data'] = '{"act":"创建了游戏", "view":"<a href=\"index.php?a=view&eid='.$eid.'\">' .
    '$subject.</a>"}';
$feed['body_template'] = '{description} {join},{new}游戏';
$feed['body_data'] = '{"description":'.$feedDescription.', "join":"我想<a href=\"index.php?a=' .
    'view&eid='.$eid.'\">参加</a>", "new":"<a href=\"index.php?a=new\">创建</a>"};
$feed['body_general'] = ""; $manyou->api_client->feed_publishTemplatizedAction($feed['title_template'],
$feed['title_data'],
$feed['body_template'], $feed['body_data'], $feed['body_general'], " ", " ", " ", " ", " ", " ");
//note 成功提示
showMessage('创建游戏成功,点击跳转到查看游戏详细', "index.php?a=view&eid=$eid");
```

6. 最终页面效果, 如图 5 所示

The screenshot shows a web page titled "用户系统与discuz.net整合已完成，您可以使用discuz.net帐号登录了。" (The user system has been integrated with discuz.net, you can log in with your discuz.net account.). Below the title is a section titled "今天TA请客了吗？" (Did TA invite you today?) with four tabs: "创建游戏" (Create Game), "所有游戏" (All Games), "我的游戏" (My Games) (which is selected), and "好友的游戏" (Friend's Games). The main content area displays a list of game logs:

- dev_lulu于 2008-7-22 12:02 创建了 [谁请冰淇淋](#), 现共有12参与, 将于2010-3-11 17:22揭晓。
- dev_lulu于 2008-7-24 11:17 创建了 [谁去买绿舌头](#), 共有1人参与, dev_lulu获得殊荣。
- lu5266于 2008-7-24 12:10 创建了 [谁请冰淇淋](#), 共有1人参与, lu5266获得殊荣。
- qiubowinter于 2008-7-25 18:32 创建了 [谁请冰淇淋](#), 共有1人参与, qiubowinter获得殊荣。
- qiubowinter于 2008-7-25 18:33 创建了 [谁请冰淇淋](#), 共有1人参与, qiubowinter获得殊荣。
- fengxue于 2008-7-27 16:27 创建了 [谁请冰淇淋](#), 共有1人参与, fengxue获得殊荣。
- fengxue于 2008-7-27 16:30 创建了 [谁请冰淇淋](#), 共有1人参与, fengxue获得殊荣。
- fengxue于 2008-7-27 16:33 创建了 [谁请冰淇淋](#), 共有1人参与, fengxue获得殊荣。
- caosir于 2008-7-27 17:49 创建了 [谁送300积分](#), 共有1人参与, caosir获得殊荣。

At the bottom right of the content area is a page navigation bar with buttons labeled 26, 1, 2, 3, and >>. At the very bottom of the page is a footer note: "该页面由TA请客创建(举报)" (This page was created by TA Please Guest (Report)).

图 5 最终页面效果

7. 源代码下载

本教程案例地址: <http://www.ismole.net/thread-1405-1-1.html>

本教程源码下载: <http://www.ismole.net/attachment.php?aid=753>

注: 文中涉及到例子的完整代码在“附件/widget/”中。



扩展实例

PEAR 使用教程 (三)

——重要的 PEAR 包

Important PEAR Packages

简张桂

1 介绍

Introduction

在这章中, 你将看到一些流行的 PEAR 包的例子。本文不会介绍每一个 PEAR 包的例子, 但是它至少给你做了一个介绍。

2 数据库查询

Database Queries

关于这块感兴趣的朋友可以查看《PHP 权威编程》第 6 章“使用 PHP5 访问数据库”, 来查看 PEAR DB 的介绍。

3 模板系统

Template Systems

模板系统是 PHP 中让你分离应用逻辑与显示逻辑的组件, 而且提供一个比 PHP 本身更为简单的模板格式。

从本质上说 PHP 就是从一个模板语言开始的, 你可能会觉得奇怪的是它又用来实现模板系统。但是, 对此除了代码与显示的分离, 还有一些很好的理由可以说明, 例如给 Web 设计者一个更加简单的标记格式, 以便他们可以在其页面编写工具中使用, 而且开发工程师可以在页面生成后更好地进行控制。例如, 一个模板系统可以自动把文本片段转变为另外一种语言, 或者往一个表单填入默认值。

PHP 中可以使用许多种模板系统。这是因为事实上和数据库抽象层一起, 模板系统是在开发中需求最强并且很少放弃的 PHP 组件之一。所以, 许多人都编写了他们自己的模板系统, 结果造成一个令人惊奇的多样性与标准化不足。

3.1 模板术语

Template Terminology

在你深入到各种各样的模板系统之前, 你可能需要熟悉模板的专用语。

表 1 模板术语表

词语	意思
模板	输出的设计页；包含占位符和代码块
编译	把一个模板转换为 PHP 代码
占位符	在执行的时候被替代的有分界线的字符串
模块或者片断	一个模板中可以重复显示不同数据的部分

3.2 HTML_Template_IT

你将熟悉的第一个 PEAR 模板系统就是 HTML_Template_IT，或者只是 IT。这是最流行的 PEAR 模板包，但是它也是运行最慢的，因为它在每次请求时都解析模板而且没有把它们编译到 PHP 代码中。

提示：HTML_Template_Sigma 包提供一个兼容 HTML_Template_IT 的 API，但是它把模板编译为 PHP 代码。

占位符语法

IT 使用大括号作为占位符分界线，如下：

代码片段

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

这是最常用的占位符语法，所以机会就是一个只使用占位符的模板如今可以与不同的模板包搭配使用。

例子：基本的 IT 模板

这个例子是用 HTML_Template_IT 实现的“Hello World”：

代码片段

```
<?php
require_once "HTML/Template/IT.php";
$tpl = new HTML_Template_IT('./templates');
$tpl->loadTemplateFile('hello.tpl');
$tpl->setVariable('title', 'Hello, World!');
$tpl->setVariable('body', 'This is a test of HTML_Template_IT!');
$tpl->show();
```

首先，你创建一个 HTML_Template_IT 对象，把模板的目录作为一个参数进行传递。接下来，模板文件被加载进来而且设置一些变量。变量的名字对应于模板文件中的占位符，所以 {title} 模板占位符就会被 “title” 变量的值代替。最后，show()方法处理所有的替代并且显示模板的输出。

这个模板文件是用在这个例子中的：

代码片段

```
<html>
<head>
```

```
<title>{title}</title>
</head>
<body>
<h1>{title}</h1>
<p>{body}</p>
</body>
</html>
```

图 1 显示了结果。

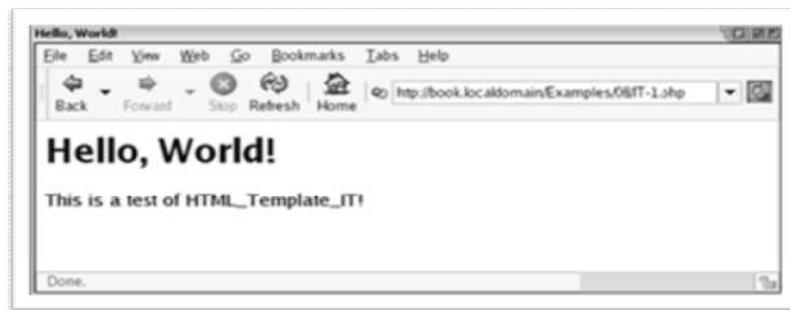


图 1 基础 IT 模板输出

模块语法

对于模块，IT 使用了 HTML 注释的开始与结束标志，如下：

代码片段

```
<!-- BEGIN blockname -->
<li>{listitem}
<!-- END blockname -->
```

模块可以嵌套使用，但是重要的是你从最里面的模块开始处理而且自己分布好嵌套。

例子：IT 中使用模块

首先，安装 HTML_Template_IT：

如下所示

```
$ pear install HTML_Template_IT
downloading HTML_Template_IT-1.1.tgz ...
Starting to download HTML_Template_IT-1.1.tgz (18,563 bytes)
.....done: 18,563 bytes
install ok: HTML_Template_IT 1.1
```

这个例子使用模块来在模板中实现一个简单的类似 foreach 的循环：

代码片段

```
<?php
require_once "HTML/Template/IT.php";
$list_items = array(
'Computer Science',
'Nuclear Physics',
'Rocket Science',
```

```

);
$tpl = new HTML_Template_IT('./templates');
$tpl->loadTemplateFile('it_list.tpl');
$tpl->setVariable('title', 'IT List Example');
foreach ($list_items as $item) {
$tpl->setCurrentBlock("listentry");
$tpl->setVariable("entry_text", $item);
$tpl->parseCurrentBlock("cell");
}
$tpl->show();

```

这个例子与先前一样启动 IT 对象，但是它调用了 `setCurrentBlock()` 来指定接下来的 `setVariable()` 调用应用的模块。当 `parseCurrentBlock()` 被调用时，模块就被解析，占位符被替代，而且结果被缓冲起来直到模板显示出来。

下面表示如何写模块模板

代码片段

```

<html>
<head>
<title>{title}</title>
</head>
<body>
<h1>{title}</h1>
<ul>
<!-- BEGIN listentry -->
<li>{entry_text}</li>
<!-- END listentry -->
</ul>
(End of list)
</body>
</html>

```

图 2 显示了输出结果。



图 2 使用模块的 IT 的输出

最后，IT 让你包含你的模板中其他任何位置的模板文件，如下：

代码片段

```
<!-- INCLUDE otherfile.tpl -->
```

在这个模块的例子中，你可以只用一个包含标记来替换模块内容，而 HTML_Template_IT 将在每一次遍历模块时包含该文件。

通过仔细地使用包含功能，你可以设计你的模板以便你可以复用子模板。

3.3 HTML_Template_Flexy

下一个模板包是 HTML_Template_Flexy，或者简称 Flexy。虽然为 IT 编写的纯占位符的模板可以在 Flexy 中即装即用，但是这两个模板包差别非常大。

首先，不像 IT 中顺序存放在联合数组变量中，Flexy 操作的是对象和对象成员变量。这本身并不是一个很大的差别，但是 Flexy 的优势是你可以给它任何类的对象，而你的模板就可以访问它们的公共成员变量。

例 1：基本的 Flexy 模板

这是一个使用 Flexy 的“Hello,World!”例子

代码片段

```
<?php
require_once 'HTML/Template/Flexy.php';
$tpldir = 'templates';
$tpl = new HTML_Template_Flexy(array('templateDir' => 'templates','compileDir' => 'compiled'));
$tpl->compile('hello.tpl');
$view = new StdClass;
$view->title = 'Hello, World!';
$view->body = 'This is a test of HTML_Template_Flexy';
$tpl->outputObject($view);
```

启动 Flexy 需要稍微多点的代码，因为你需要同时指定模板的目录和编译目录。编译目录 Compile Directory 是指编译后的模板文件存放的地方。这个目录必须可以让 Web 服务器写入。默认情况下，编译目录与模板目录是相关的。

接下来，编译 hello.tpl 模板。你应该注意到这个模板与最开始的 IT 例子的模板是一样的；这是因为模板只包含了两个简单的占位符所以有效。

编译是很耗时的，但是它只会被操作一次或者当模板文件更改时再操作一次。所以，你将注意到第一次你加载页面的时候，时间会长些。接下来的页面加载将非常快。

在编译一个模板时，编译的版本是存放在 compileDir 中的。在先前的例子中，这是相对于当前目录的“编译”目录。这个目录必须能够被 Web 服务器写入，因为当一个用户点击页面的时候，模板是由于 PHP 需要而编译的。

最后，创建一个保存视图数据的对象并且传递到 outputObject()方法中，由它执行模板并打印输出。

例 2：Flexy 中使用模块

这个例子对应于“IT 中使用模块”的例子：

代码片段

```
<?php
require_once 'HTML/Template/Flexy.php';
$tpldir = 'templates';
$tpl = new HTML_Template_Flexy(array(
'templateDir' => 'templates',
'compileDir' => 'compiled',
));
$tpl->compile('flexy_list.tpl');
$view = new StdClass;
$view->title = 'Flexy Foreach Example';
$view->list_entries = array(
'Computer Science',
'Nuclear Physics',
'Rocket Science',
);
$tpl->outputObject($view);
```

这次，模板文件是不一样的，因为它不止使用了占位符而且不再与 IT 兼容：

PHP 代码

```
<html>
<head>
<title>{title}</title>
</head>
<body>
<h1>{title}</h1>
<ul>
{foreach:list_entries,entry_text}
<li>{entry_text}
{end:}
</ul>
(End of list)
</body>
</html>
```

如果你比较这个例子中的 PHP 代码和对应的 IT 例子，会发现所有模块解析的问题都不存在了。这是因为模板已经被编译了；与自己处理流控制不同的是，Flexy 让 PHP 的执行者来处理这个问题。请看通过 Flexy 编译器生成的 PHP 文件：

代码片段

```
<html>
<head>
<title><?php echo htmlspecialchars($t->title);?> </title>
</head>
<body>
<h1><?php echo htmlspecialchars($t->title);?> </h1>
```

```

<ul>
<?php if(is_array($t->list_entries) || is_object($t->list_entries))foreach($t->list_entries as $entry_text){?>
<li><?php echo htmlspecialchars($entry_text);?> </li>
<?php }?>
</ul>
(End of list)
</body>
</html>

```

Flexy 标记格式

目前为止，你已经看到了占位符的例子和 Flexy 中的{foreach:}结构。表 2 给出了一个 Flexy 支持的结构的完整的列表。

表 2 Flexy 标记的标记符

标记符	描述
{ variable}	这是正则占位符。默认情况下，占位符是通过 htmlspecialchars() 编码的。:h 修饰符禁止它以便传递原始的值，而:u 修饰符则表示用 urlencode() 来编码
{method()}	这个标记符调用一个在视图对象中的方法并且使用返回值。就跟变量一样，默认下使用 htmlspecialchars() 来处理，而且你可以同时使用:h 和:u 修饰符
{if:variable}	可以使用 If 语句，但是只能用布尔值来验证而且不能用其他任意复杂的逻辑。（有限的 If 语句验证变量、方法调用和否定操作）
{else:}	Else 标记符必须要与{If:}一起使用
{end:}	{end:}标记符用来结束{foreach:}或者{If:}
{foreach:arr,val}	与 PHP 的 foreach 相对应。第一种形式遍历 arr 并且按顺序把每一个元素赋值给 val。第二种形式中还会把数组的索引赋值给 ind
{foreach:arr,ind,val}	

Flexy 的 HTML 属性处理

关于 Flexy 的一个有趣的事情之一是如何处理 HTML/XML 元素和模板中的属性。为了给你一个例子，这里再次使用上一个例子并且模板中变为使用一个 Flexy 的 HTML/XML 属性来控制模块：

代码片段

```

<html>
<head>
<title>{title}</title>
</head>
<body>
<h1>{title}</h1>
<ul>

```

```

<li flexy:foreach="list_entries,text">{text}
</ul>
(End of list)
</body>
</html>

```

{foreach:}结构不在了；它被代替为一个将被重复输出的元素的属性：``。这个看起来有点像 XML 的命名空间，但是它不是；Flexy 编译器会在编译过程中把 `flexy:foreach` 属性去掉，并且生成与{foreach:}形式相同的 PHP 代码。这个版本编译后的版本如下：

代码片段

```

<html>
<head>
<title><?php echo htmlspecialchars($t->title);?> </title>
</head>
<body>
<h1><?php echo htmlspecialchars($t->title);?></h1>
<ul>
<?php if (is_array($t->list_entries) || is_object($t->list_entries))
    foreach($t->list_entries as $entry_text)
{?>
<li><?php echo htmlspecialchars($entry_text);?></li>
<?php
}?>
</ul>
(End of list)
</body>
</html>

```

表 3 概要介绍了 Flexy 支持的 XML 或 HTML 的属性。

表 3 Flexy 支持的 HTML/XML 属性

属性	描述
flexy:if="variable" flexy:if="method()" flexy:if="!variable" flexy:if="!method()"	这是简化的{if:}。条件应用到 XML/HTML 元素和它的子元素，而且没有{else:}。如果验证失败，当前元素和所有其子元素将被忽略
flexy:start="here"	flexy:start 属性可以用来忽略当前元素以外的任何内容。如果你有子模板但是仍然想把他们当作完整的 HTML 文件来查看或者编辑时你可以使用它
flexy:startchildren="here"	与 flexy:start 类似，但是它忽略所有的内容并且包含当前的元素
flexy:ignore="yes"	忽略当前的元素和所有的子元素。它可以用来在一些可视的 Web 设计工具编辑的模板中放置实验数据
flexy:ignoreonly="yes"	忽略所有的子元素，但是包含当前元素

Flexy 的 HTML 元素处理

最后，Flexy 可以从元素中解析 HTML 并且用正确的数据填充它们。这可以轻松地在一些 Web 设计工具中创建一个表单，而且不需要在你的站点使用它以前仔细研究该模板。

Flexy 能够处理下面的 4 种 HTML 元素（见表 4）。

表 4 HTML 元素

<form name="xxx">
<input name="xxx">
<select name="xxx">
<textarea name="xxx">

当 Flexy 在模板中找到任何这些 HTML 元素时，该元素会被替换为 PHP 代码并输出正确属性的元素：

代码片段

```
<html>
<head><title>{title}</title></head>
<body bgcolor=white>
<form name="myform">
{user_label} <input type="text" name="user">
<br>
{pw_label} <input type="password" name="pw">
</form>
</body>
</html>
```

在这个模板中，<form>和<input>元素将被 Flexy 替换，填入参数。

作者介绍



简张桂，前 PHPChina 开源社区负责人，美国 Zend 公司（PHP 的创始公司）中国技术支持中心技术总监，多迪网络（广州 PHP 授权培训认证中心）特聘讲师。

毕业于北京邮电大学计算机专业。拥有多年的 Web 开发经验，精通基于 LAMP 架构的 PHP 开发。作为美国 Zend 公司（PHP 的创始公司）全球的技术支持工程师之一，一直在国内负责开源 PHP 的推广。同时，作为 PHPChina 的 PHP 培训最早的讲师之一，负责 Zend 公司 PHP 培训课程本地化项目。

作者翻译图书：《PHP5 权威编程》 购买地址：[中国互动出版社](#)

作者 BLOG：[hosander's PHP house](#)



高级应用

用 VIM 做 PHP 开发环境

廖宇雷

虽然 VIM 本质上只是一个编辑器。但只要配合一些适当的插件，VIM 也能变成一个全功能的 IDE。笔者使用 VIM 已经有挺长一段时间了，经过反复的试验，配置了一个高效的 PHP 开发环境，实为居家旅行、谋财害命、杀人越货必备之良品。

安装 VIM

鉴于大多数读者都是使用 Windows 环境，所以本文就以 Windows 作为运行环境。不过由于 VIM 出色的跨平台特性，配置文件只需简单修改就可以在 Linux 版本的 VIM 中使用。

目前 VIM 的最新版本是 7.2，如果没有特别原因建议安装最新的版本。

下载地址：<http://www.vim.org/download.php#pc>

下载文件：<ftp://ftp.vim.org/pub/vim/pc/gvim72.exe>

安装时不要装到带有空格、中文的目录中，其他保持默认即可。笔者的安装目录是 c:\apps\office\vim，后文也以此为准。为了叙述方便，笔者用\$VIM 来表示 VIM 的安装目录。例如您的 VIM 安装在 d:\vim\ 中，那\$VIM 就代表 d:\vim\。

开始配置 VIM

我们对 VIM 的配置分为几个步骤。

修改_vimrc 后自动生效

打开\$VIM 目录，可以看到其中有一个_vimrc 文件，用 VIM 打开此文件，删除所有内容后，在最后插入两行：

如下所示

```
" autoload _vimrc  
autocmd! bufwritepost _vimrc source %
```

上述命令让我们通过 VIM 编辑_vimrc 文件并保存时，可以自动载入_vimrc 文件，使得我们对 VIM 的定制可以立即生效（无需重启 VIM）。

使 VIM 中文化

在_vimrc 顶部增加下列文本：

如下所示

```
" disable VI's compatible mode..
```

```
set nocompatible
" set encoding=utf-8
set fileencodings=ucs-bom,utf-8,gbk,default,latin1
" use chinese help
set helplang=cn
```

上述代码的作用是禁用 vi 兼容模式（原始的 vi 功能太少了，没必要考虑兼容）、按照 utf-8、gbk 的顺序来检测文件编码，并设置帮助为中文。不过设置 `set helplang=cn` 并不能马上看到中文帮助，我们还得下载中文帮助文件。

下载地址: <http://vimcdoc.sourceforge.net/>

下载文件: vimcdoc-1.6.0.tar.gz

得到中文帮助文件压缩包后，将压缩包中 `doc` 子目录的所有文件复制到 `$VIM\vimfiles\doc` 目录中。此时再输入`:help` 命令就可以看到中文帮助了，如图 1 所示。

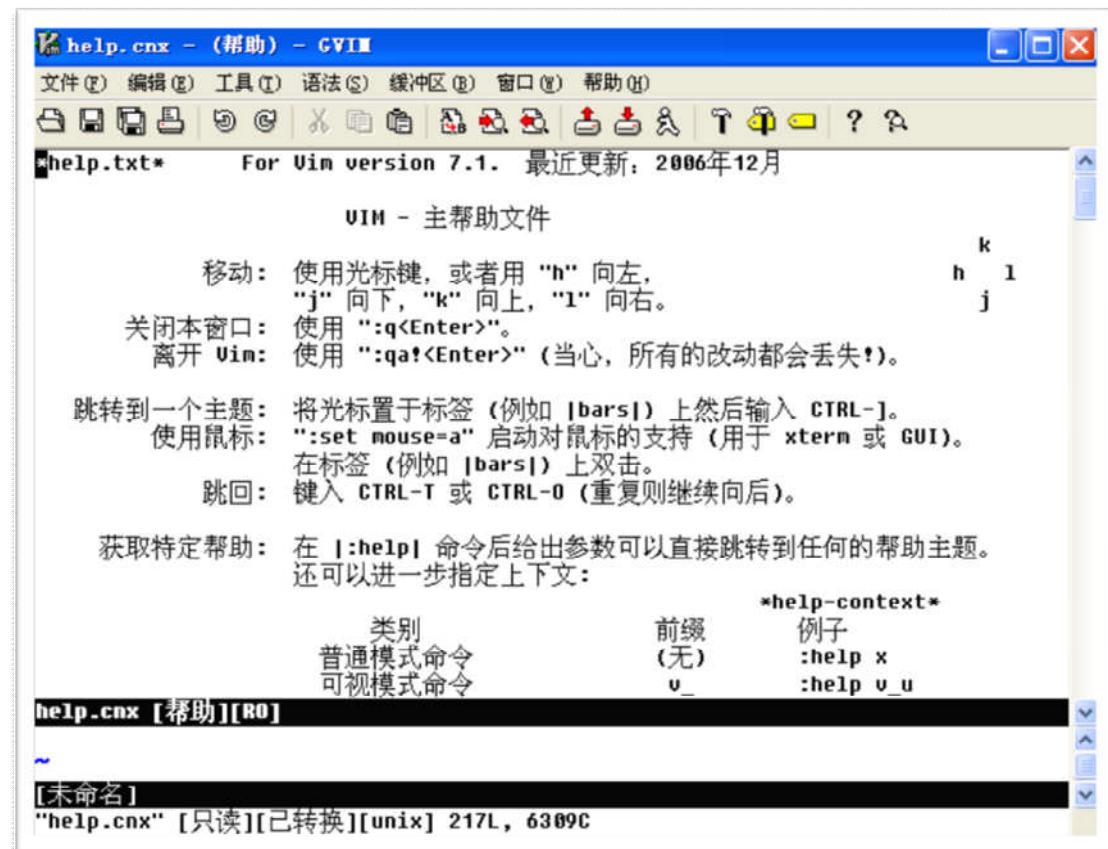


图 1 帮助文档界面

中文帮助还是 7.1 版的，不过不影响我们使用。

设置字体

选择 VIM 菜单的“编辑”->“选择字体”，就可以为 VIM 指定喜欢的显示字体了。笔者使用的是 Consolas 字体，大小设置为 9pt。这个设置显示代码非常漂亮，不过中文就有点变形了。

设置好后，输入命令：**:set guifont** 可以查看到当前的字体设置，将该设置写入 `_vimrc` 文件。

如下所示

```
"set gui options  
if has("gui_running")  
    set guifont=Consolas:h9  
endif
```

上述代码中的 `if...endif` 是一个条件判断结构。指示只有当我们使用图形界面版的 VIM 时才设置字体。

为编辑 PHP 代码进行基本设置

现在用 VIM 打开.php 文件看上去是很难看的，别说代码高亮，连行号都没有显示。所以还要给 `_vimrc` 添加以下内容：

如下所示

```
" Enable syntax highlight  
syntax enable  
  
" Show line number  
set nu  
  
" show matching braces  
set showmatch  
  
" Basic editing options  
set expandtab  
set shiftwidth=2  
  
au FileType html,python,vim,javascript setl shiftwidth=2  
au FileType html,python,vim,javascript setl tabstop=2  
au FileType java,php setl shiftwidth=4  
au FileType java,php setl tabstop=4  
  
set smarttab  
set lbr  
set tw=0  
  
"Auto indent  
set ai  
  
" Smart indent  
set si  
  
" C-style indenting  
set cindent  
  
" Wrap lines  
set wrap
```

上述设置启用了格式化高亮、行号显示,以及括号匹配、自动缩进等编辑功能,对于大多数情况都可以获得理想的编辑体验。不过此时对.php 文件的支持还不完善,需要下载专门的 PHP 插件。

下载地址: http://www.vim.org/scripts/script.php?script_id=1571

下载文件: php.tar.gz

将其中的 php.vim 复制到\$VIM\vimfiles\syntax 目录中即可。

设置喜欢的配色方案

默认的配色方案相信没几个人会喜欢,可以通过下面的网址查看配色方案并下载(有几百个哦)。

下载网址: <http://www.cs.cmu.edu/~maverick/VimColorSchemeTest/index-c.html>

这个网址列出了 300 多个配色方案以及实际的显示效果,点击方案名称即可下载到一个.vim 文件。将该文件放入\$VIM\vimfiles\colors 目录,然后在_vimrc 中添加:

如下所示

```
" set color schema  
colorscheme oceandee
```

这两行代码需要加到 if has("gui_running") ...endif 代码块中,例如:

如下所示

```
if has("gui_running")  
    set guifont=Consolas:h9  
    " set color schema  
    colorscheme oceandee  
endif
```

设置后显示效果就漂亮多了,效果如图 2, :-)

```
228     static final function loadClass($class_name, $dirs = null)  
229     {  
230         if (class_exists($class_name, false) || interface_exists($class_name  
, false))  
231         {  
232             return;  
233         }  
234  
235         $class_name = strtolower($class_name);  
236         if (strpos($class_name, '@') !== false)  
237         {  
238             list ($class_name, $module) = explode('@', $class_name);  
239             QContext::instance($module_name);  
240         }  
241         $filename = str_replace('_', DS, $class_name);  
242  
243         if ($filename != $class_name)  
244         {
```

图 2 颜色设置效果

更多有用的编辑设置

虽然不是专门针对编辑.php 文件的设置,但这些选项可以让 VIM 更好用,所以建议加上:

如下所示

```
" Sets how many lines of history VIM has to remember  
set history=400  
  
" Set to auto read when a file is changed from the outside  
set autoread  
  
" Have the mouse enabled all the time:  
set mouse=a  
  
" Do not redraw, when running macros.. lazyredraw  
set lz  
  
" set 7 lines to the cursor - when moving vertical..  
set so=7  
  
" The commandbar is 2 high  
set cmdheight=2  
  
" Change buffer - without saving  
set hid  
  
" Ignore case when searching  
" set ignorecase  
set incsearch  
  
" Set magic on  
set magic  
  
" No sound on errors.  
set noerrorbells  
set novisualbell  
set t_vb=  
  
" How many tenths of a second to blink  
set mat=4  
  
" Highlight search things  
set hlsearch  
  
" Turn backup off  
set nobackup  
set nowb  
set noswapfile  
  
" smart backspace  
set backspace=start,indent,eol  
  
" switch buffers with Tab  
map <C-Tab> :bn<CR>  
map <S-Tab> :bp<CR>
```

经过一番改造，我们的 VIM 已经可以很顺手的编辑.php 文件了。不过要打造 PHP 开发环境还有很多工作要做呢。

打造 PHP IDE

下图是配置完成的 PHP IDE，看上去很不错哦，如图 3 显示，^_^

```

meta.php (d:\www\qeephp\library\qdb\activerecord) - GVIM
文件(①) 编辑(②) 工具(③) 语法(④) 缓冲区(⑤) 窗口(⑥) 插件(⑦) Snippets 帮助(⑧)

265 }
266 /**
267 * 返回一个根据 $data 数组构造的 ActiveRecord 隶属类实例
268 */
269 *
270 * @return QDB_ActiveRecord_Abstract
271 */
272 function newObject(array $data = null)
273 {
274     return new $this->class_name($data);
275 }
276 /**
277 * 打开一个查询
278 */
279 *
280 * @return QDB_Select
281 */
282 function find()
283 {
284     return $this->findByArgs(func_get_args());
285 }
286 /**
287 * 打开一个查询，并根据提供的参数设置查询对象
288 */
289 *
290 * @param array $args
291 *
292 * @return QDB_Select
293 */
294 function findByArgs(array $args)
295 {
296     $select = new QDB_Select($this->table->getConn());
297     $select->asObject($this->class_name)->from($this->table);
298     if (count($args))
299     {
d:\www\qeephp\library\qdb\activerecord\meta.php
Function find()

```

图 3 配置效果演示

IDE 左侧是目录导航，中间是编辑区域，而右侧则是方法列表，用于在已经打开的文件中快速跳转。在编辑区域按下“CTRL+X”键，还会显示已打开文件的列表，如图 4 所示。

```

[BufExplorer] - (d:\www) - GVIM
文件(①) 编辑(②) 工具(③) 语法(④) 缓冲区(⑤) 窗口(⑥) 插件(⑦) Snippets 帮助(⑧)

Buffer Explorer (7.1.7)
-----
* <F1> : toggle this help
* <center> or Mouse-Double-Click : open buffer under cursor
* <shift-enter> or t : open buffer in another tab
* d : wipe buffer
* D : delete buffer
* p : toggle splitting of file and path name
* q : quit
* r : reverse sort
* R : toggle showing relative or full paths
* u : toggle showing unlisted buffers
* s : select sort field [ 'number', 'name', 'fullpath', 'mrn', 'extension' ]
* f : toggle find active buffer
* = : Sorted by mrn | Locate buffer | Absolute Split path
* =
5 Ma  meta.php  d:\www\qeephp\library\qdb\activerecord 第 282 行
4 en  abstract.php d:\www\qeephp\library\qdb\activerecord 第 1 行
1 h   q.php    d:\www\qeephp\library 第 16 行

d:\www\qeephp\library\qdb\activerecord
class
QDB_ActiveRecord_Meta

function
__construct
instance
newObject
find
findByArgs
updateWhere
updateDbWhere
destroyWhere
deleteWhere
validate
getRelatedObjects
bindBehaviors
unbindBehaviors
addDynamicMethod
removeDynamicMethod
addStaticMethod
removeStaticMethod
setPropSetter
unsetPropSetter
setPropGetter
unsetPropGetter
addEventHandler
removeEventHandler
addProp

```

图 4 文件列表显示

其他诸如自动补全、代码模板等功能，都应有尽有。看过了漂亮的截图，我们就来一步步打造 PHP IDE 吧。

用 NERDTree 实现目录导航

在进行 PHP 应用开发时, 同时编辑多个文件是很正常的事情。所以必须有一个方便的目录导航工具, 以便在目录结构间快速切换, 找到需要编辑的文件。

VIM 中提供该类功能的插件很多, 比较知名的有 project、winmanager 等。但笔者个人认为最好用的还是 The NERD Tree 这个插件。NERDTree 不但可以显示完整的目录树结构, 还可以将任何一个目录设置为根目录。并且提供了目录导航的书签功能, 可谓非常方便。

下载地址: http://www.vim.org/scripts/script.php?script_id=1658

下载文件: NERD_tree.zip

解压缩时, 要把压缩包中的目录结构完整的解压缩到\$VIM\vimfiles 目录中。完成后, 要分别找到\$VIM\vimfiles\doc\NERD_tree.txt 文件和\$VIM\vimfiles\plugin\NERD_tree.vim 文件。然后在 VIM 中输入命令:**:helptags \$VIM\vimfiles\doc**, 将 NERDTree 的帮助文档添加到 VIM 中。

最后在_vimrc 添加如下内容:

如下所示

```
" NERDTree
map <F10> :NERDTreeToggle<CR>
```

重启 VIM 后, 按下 F10 键, 就可以在左侧看到一个目录树了。在目录树窗口中按下“?键”可以查看详细的帮助信息, 如图 5 所示。

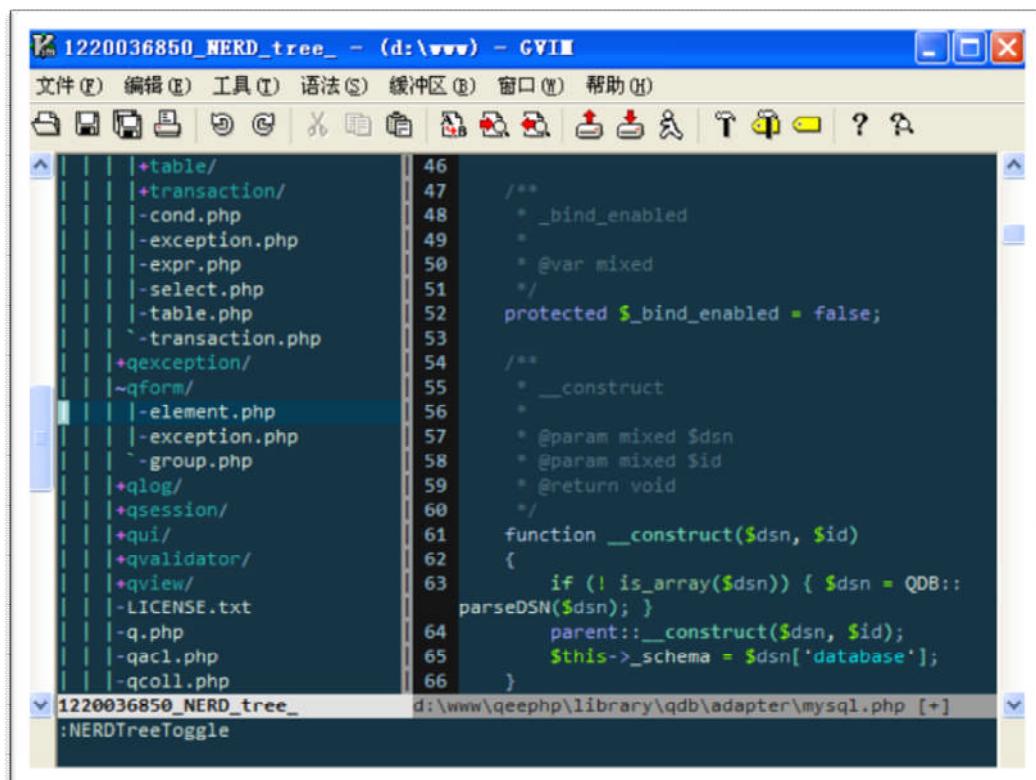


图 5 VIM 目录树显示

最常用的操作键（见表 1）：

表 1 常用操作键

按键	作用
C (大写 C 键)	将光标所在目录设置为根目录
u (小写 u 键)	转到上一级目录
o (小写 o 键, 不是“零”)	展开 (或折叠) 光标所在目录的子目录。如果光标所在位置是一个文件, 则在编辑窗口中打开该文件

此外在目录树窗口中输入目录:**Bookmark 收藏名**还可以将光标所在目录添加到收藏夹中。下次使用:**BookmarkToRoot 收藏名**可以直接转到该目录, 并且以该目录作为根目录。更多命令可以参考 NERDTree 的帮助文档。

用 taglist 实现代码导航

解决了目录和文件导航问题, 我们还要为代码之间的跳转提供辅助手段, taglist 就是这样一个插件。taglist 可以列出已打开文件中定义的类、函数、常量, 甚至变量。

下载地址: http://www.vim.org/scripts/script.php?script_id=273

下载文件: taglist_45.zip

压缩包需要完整解压缩到\$VIM\vimfiles 目录, 并且用:**:helptags \$VIM\vimfiles\doc** 命令索引 taglist 插件的帮助文档。taglist 插件需要依赖 ctags 程序才能工作。目前常用的 ctags 版本是 Exuberant Ctags。

下载地址: <http://ctags.sourceforge.net/>

下载文件: ec57w32.zip

只需要把压缩包中的 ctags.exe 复制到\$VIM\vim72 目录中即可。ctags.exe 应该和 gvim.exe 在一个目录。

最后在_vimrc 添加下列内容, 设置好 taglist 插件:

如下所示

```
" => Plugin configuration
=====
" taglist
let Tlist_Auto_Highlight_Tag = 1
let Tlist_Auto_Open = 1
let Tlist_Auto_Update = 1
let Tlist_Close_On_Select = 0
let Tlist_Compact_Format = 0
let Tlist_Display_Prototype = 0
let Tlist_File_Fold_Auto_Close = 0
let Tlist_GainFocus_On_ToggleOpen = 1
let Tlist_Hightlight_Tag_On_BufEnter = 1
```

```

let Tlist_Inc_Winwidth = 0
let Tlist_Max_Submenu_Items = 1
let Tlist_Max_Tag_Length = 30
let Tlist_Process_File_Always = 0
let Tlist_Show_Menu = 0
let Tlist_Show_One_File = 0
let Tlist_Sort_Type = "order"
let Tlist_Use_Horiz_Window = 0
let Tlist_Use_Right_Window = 1
let Tlist_WinWidth = 40
let tlist_php_settings = 'php;c:class;i:interfaces;d:constant;f:function'

```

这里的设置是笔者个人习惯的设置，大家在熟悉 taglist 的具体设置之前可以先尝试一下。

设置完成后，在编辑.php 文件时按 F8 键右侧就会出现代码导航窗口，如图 6 所示。

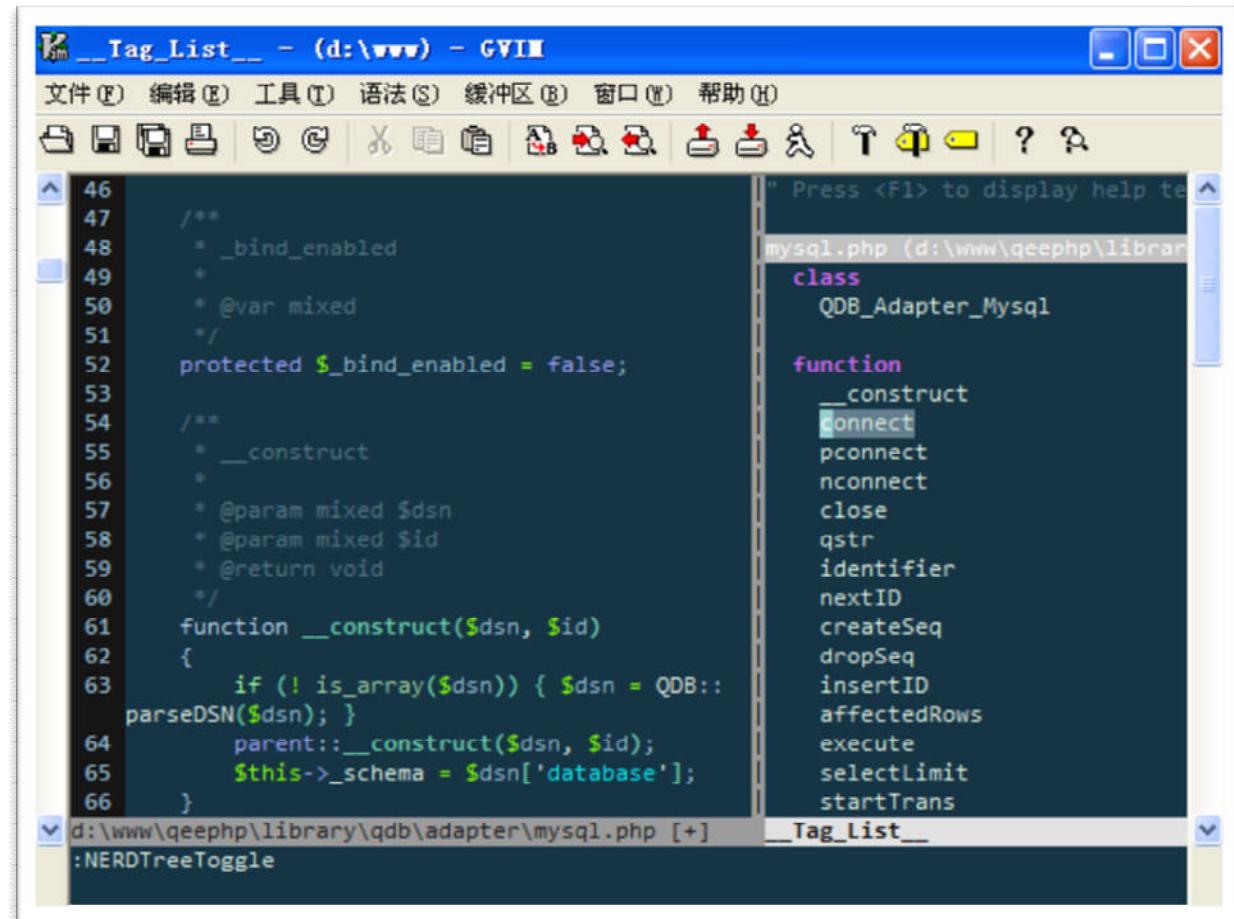


图 6 代码导航窗口

在代码导航窗口中，按回车键即可转到光标所在位置显示的函数或类名称，非常方便。如果打开了多个文件，还可以看到文件列表。

快速文件切换

打开多个文件编辑时，可以使用 bufexplorer 插件进行快速切换。

下载地址: http://www.vim.org/scripts/script.php?script_id=42

下载文件: bufexplorer.zip

完整解压缩到\$VIM\vimfiles 目录, 并添加如下内容到_vimrc 中:

如下所示

```
" bufexplorer
let g:bufExplorerDefaultHelp=1
let g:bufExplorerDetailedHelp=0
let g:bufExplorerSortBy='mru'
nmap <C-X> :BufExplorer<CR>
```

现在打开多个文件后, 在正常模式按“CTRL+X”就可以查看文件列表。在列表项目上按 d (小写 d 键) 即可关闭一个打开的文件。在文件列表窗口按 F1 可以查看详细帮助信息。

进一步提高.php 编辑体验

在 Windows 下面编写.php 文件, 最常见的问题就是使用了 Windows 格式的换行符。使用 Windows 换行符的文件在 Linux/Unix 环境中有时会出现问题, 所以我们需要用一些工具软件将 Windows 换行符转换为 Linux/Unix 换行符。

不过最新的 VIM 已经自带了该功能, 只需要在_vimrc 中添加如下内容即可:

如下所示

```
" php editing
" remove CR at end of lines
let PHP_removeCRwhenUnix = 1
" Set up automatic formatting
set formatoptions+=tcqlro
" Set maximum text width (for wrapping)
set textwidth=110
```

为代码添加注释是一种好习惯, phpDocumentor for Vim 插件可以大大简化添加猪似的操作。

下载地址: http://www.vim.org/scripts/script.php?script_id=1355

下载文件: php-doc.vim

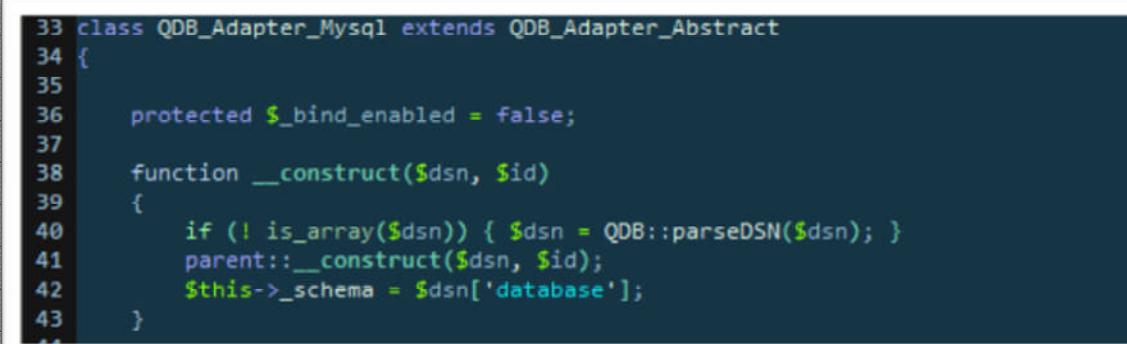
先将 php-doc.vim 文件复制到\$VIM\vimfiles\plugin 目录中, 然后在_vimrc 添加下列内容:

如下所示

```
" phpDocumentor for VIM
imap <A-/> <ESC>:call PhpDocSingle()<CR>i
nmap <A-/> :call PhpDocSingle()<CR>
vmap <A-/> :call PhpDocRange()<CR>
```

```
" set pdv
let g:pdv_cfg_Type = "mixed"
let g:pdv_cfg_Package = ""
let g:pdv_cfg_Version = "$Id$"
let g:pdv_cfg_Author = "dualface <dualface@gmail.com>"
let g:pdv_cfg_Copyright = "2006 - 2009 QeeYuan.com Ltd."
let g:pdv_cfg_License = "New BSD {@link http://www.opensource.org/licenses/bsd-license.php}"
let g:pdv_cfg_php4always = 0
let g:pdv_cfg_php4guess = 0
```

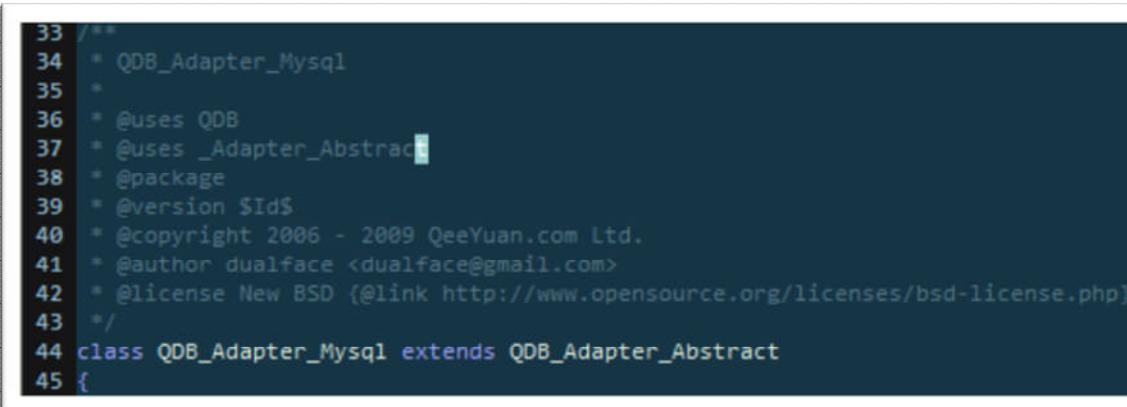
其中注释的作者、版权信息等内容请按照自己的需要修改。现在打开任何一个.php 文件，把光标移动到类定义那一行，按下“ALT+/" 键，可以看到 PDV 插件为我们自动添加了详细的注释信息，如图 7、图 8 所示。



```
33 class QDB_Adapter_Mysql extends QDB_Adapter_Abstract
34 {
35
36     protected $_bind_enabled = false;
37
38     function __construct($dsn, $id)
39     {
40         if (!is_array($dsn)) { $dsn = QDB::parseDSN($dsn); }
41         parent::__construct($dsn, $id);
42         $this->_schema = $dsn['database'];
43     }
44 }
```

图 7 没注释前效果

按下“ALT+/" 后变成，如图 8 所示。



```
33 /**
34 * QDB_Adapter_Mysql
35 *
36 * @uses QDB
37 * @uses _Adapter_Abstract
38 * @package
39 * @version $Id$
40 * @copyright 2006 - 2009 QeeYuan.com Ltd.
41 * @author dualface <dualface@gmail.com>
42 * @license New BSD {@link http://www.opensource.org/licenses/bsd-license.php}
43 */
44 class QDB_Adapter_Mysql extends QDB_Adapter_Abstract
45 {
```

图 8 注释后效果

虽然还有点小问题，但总比自己一行行的敲代码来得快多了。同样，在函数、类成员变量等位置按“CTRL+P”都可以自动添注释，如图 9 所示。

在调试时，难免注释一大段一大段的代码，The NERD Commenter 插件就可以帮上大忙了。NERDCommenter 可以注释单行、多行代码，并且可以在注释和未注释状态间切换。

下载地址：http://www.vim.org/scripts/script.php?script_id=1218

下载文件：NERD_commenter.zip



```

47 /**
48 * _bind_enabled
49 *
50 * @var mixed
51 */
52 protected $_bind_enabled = false;
53
54 /**
55 * __construct
56 *
57 * @param mixed $dsn
58 * @param mixed $id
59 * @return void
60 */
61 function __construct($dsn, $id)
62 {
63     if (!is_array($dsn)) { $dsn = QDB::parseDSN($dsn); }
64     parent::__construct($dsn, $id);
65     $this->schema = $dsn['database'];
66 }

```

图 9 自动注释效果

完整解压缩到\$VIM\vimfiles 目录，并修改_vimrc 文件：

如下所示

```
" NERD Commenter
let NERDShutUp = 1
```

该行设置可以关掉 NERDCommenter 的警告信息。

NERDCommenter 比较常用的操作键有“**,cl**”（注释）、“**,cu**”（取消注释）、“**,cs**”（使用整齐的注释风格）、“**,c<空格>**”（切换注释状态）。多多实验就能体会到这个插件的妙处了 ;-)

自动补全

自动补全是一项非常实用的功能，可以有效减少击键次数。VIM 实际上自带了丰富的自动补全功能，最常用的有：

“**CTRL+P**” 或 “**CTRL+N**” 补全单词

假设正在编辑的文件内容中包括“hello”这个词，那么我在文件的其他位置输入“he”后再按“**CTRL+P**”就可以补全为“hello”。如果有多个补全结果，可以用“**CTRL+N**”和“**CTRL+P**”来选择。



The screenshot shows a Vim session where the user has typed 'he'. A completion menu is displayed below the cursor, listing 'help' (line 6), 'hello' (line 7, highlighted in blue), and 'help' again (line 8, highlighted in green). The menu is labeled '自动补全' (Automatic Completion) at the bottom.

图 10 补全单词效果

CTRL+X, CTRL+L 补全一行

与 CTRL+P(N)不同,该项补全会补全整个一行,如图 11 所示,

```
3 hello, world
4 help me
5
6 help me
7 hello, world
8 help me
9
```

未命名 [+]
正在扫描: d:\www\qeephp\library\q.php
- 行补全 (^INAPY) 匹配 1 / 2

图 11 补全整行句子效果

CTRL+X, CTRL+O 补全关键词

打开一个.php 文件,输入 mysql_c,然后按 CTRL+X, CTRL+O, VIM 会显示所有以 mysql_c 开头的函数。并且选择一个函数时,还会显示该函数的参考信息,如图 12 所示。

```
1 mysql_close([resource link identifier] | bool)
2
3 Sc mysql_change_user( f
332 mysql_client_encoding( f
333 mysql_close( f
334 mysql_connect( f
335 mysql_create_db( f
336 mysql_close()
37
```

J:\www\qeephp\library\q.php [+]
- 全能补全 (^ON^P) 匹配 3 / 5

图 12 补全关键词效果

不过这个功能的缺点就是如果匹配项太多,速度会比较慢。

VIM 除了上述三个常用的补全命令,还有许多其他补全目录,可以用 :help ins-completion 命令来查看帮助文档。

代码模板

Ruby on Rails 宣传视频中出现的 TextMate 编辑器让很多人惊艳。TextMate 编辑器的 Snippets 让开发者可以快速输入大段代码。而 VIM 也有类似的 SnippetsEmu 插件。不过 SnippetsEmu 必须用 vimball 这个插件进行安装。所以我们还得先装上 vimball 插件。

下载地址: http://www.vim.org/scripts/script.php?script_id=1502

下载文件: vimball.tar.gz

将 vimball.tar.gz 完整解压缩到\$VIM\vimfiles 目录后重启 VIM。

SnippetsEmu 插件由两个文件组成,分别是 snippy_plugin.vba 和 snippy_bundles.vba。前者是插件代码,后者是代码模板库。

下载地址: http://www.vim.org/scripts/script.php?script_id=1318

下载文件: snippy_plugin.vba, snippy_bundles.vba

用 VIM 打开下载下来的.vba 文件, 然后输入 “**:so %**” 命令即可完成安装(很无聊的安装方式, 直接解压缩岂不是更方便)。

OK, 现在我们新建一个文件, 并保存为 test.php (只有保存后 VIM 才能根据扩展名识别文件类型)。

在第一行输入 “**phpp<TAB>**” (表示输入 phpp 后再按 Tab 键), 文件内容变成如图 13 显示。

```
1 <?php
2
3 |
4
5 ?>
```

图 13 代码模板效果 1

输入 **foreach<TAB>**, 内容变成图 14 所示。

```
1 <?php
2
3 foreach( ${variable} as ${key} => ${value} )
4 {
5     <{}>
6 }
7 <{}>
8
9 ?>
```

图 14 代码模板效果 2

输入 **rowset<TAB>offset<TAB>row<TAB>**, 内容变成图 15 所示。

```
1 <?php
2
3 foreach( $rowset as $offset => $row )
4 {
5     |
6 }
7 <{}>
8
9 ?>
```

图 15 代码模板效果 3

现在输入 **print_r(\$row);<TAB>**, 内容变成图 16 所示。

The screenshot shows a Vim window with the following PHP code:

```

1 <?php
2
3 foreach( $rowset as $offset => $row )
4 {
5     print_r($row);
6 }
7
8
9 ?>

```

The cursor is positioned after the opening brace of the foreach loop. A tooltip-like box labeled "插入" (Insert) is visible at the bottom left of the screen, indicating that the user has triggered a completion or insertion mode.

图 16 代码模板效果 4

这比起我们一个字母一个字母的输入可快多了。

SnippetsEmu 自带的 PHP 模板不是很多，但我们可以非常方便的扩充。

打开\$VIM\vimfiles\after\ftplugin\php_snippets.vim 即可看到已有的模板定义。要添加一个新模板，按照下列格式输入即可：

如下所示

```
exec "Snippet getini Q::getIni(\".st.\"option\".et.\")".st.et
```

每一个“st”和“et”之间的内容都是按“Tab”键后要替换的内容。例如 **st."option".et**，当输入内容后，option 就会被输入的内容所替代。最后的 st.et 则是光标所处位置。编写代码模板的最好教材就是\$VIM\vimfiles\after\ftplugin\目录中的*snippets.vim 文件。

PHP 文件语法检查

能够一字不错写完整个代码的人还不存在。所以随时对代码进行语法检查是非常必要的。

虽然 VIM 没有现成的插件完成这个功能，但我们只要简单修改一下_vimrc 文件，语法检查器就可以工作了。

如下所示

```
" PHP syntax check
function! PHP_CheckSyntax()
    setlocal makeprg=d:/apmxe/bin/php526/php.exe\ -l\ -n\ -d\ html_errors=off
    setlocal shellpipe=>

    " Use error format for parsing PHP error output
    setlocal errorformat=%m\ in\ %f\ on\ line\ %l
    make %
endfunction

" Perform :PHP_CheckSyntax()
map <F5> :call PHP_CheckSyntax()<CR>
```

现在打开一个.php 文件, 按 F5, 就可以显示出文件的语法错误了, 如图 17 所示。

The screenshot shows a terminal window with the following content:

```
8 $_REQUEST['point_id']
9     $GLOBALS['variable']=something;
10 require( 'fuck.php' );
11
12 switch (Q::getIni('view_config/flat_dir'))
13 {
14 case '0':
15     echo "fuck";
16     break;
17
18 ?>
19

!d:/apmxe/bin/php526/php.exe -l -n -d html_errors=off test_.php > D:\users\dualfe\temp\VIeA9.tmp
shell 返回 -1
2 / 3): Parse error: syntax error, unexpected T_VARIABLE
按 ENTER 或其它命令继续■
```

图 17 语法错误显示效果

按回车键回到编辑状态后, 输入目录“`:cw`”就可以打开 quickfix 窗口, 列出所有的语法错误。在 quickfix 窗口中, 选中错误行, 按回车键即可转到源代码的相应位置进行修改。全部问题都修改完毕后, 再次按“F5”进行测试, 通过后输入“`:cccl`”既可关闭“quickfix”窗口。

笔者的 php.exe 文件在 d:\apmxe\bin\php526\目录中, 读者朋友们应该根据自己的情况进行修改。

后记

经过我们一番努力, 一个 PHP IDE 就诞生了。这个 IDE 虽然没有 Zend Studio 那样强大的功能, 但是满足一般的编程是绰绰有余了。而且 VIM 的编辑功能可以说是举世无双, 只要习惯了 VIM 的人就没有更好的选择了。

不过学习 VIM 不是一件轻松的事情, 下面是一些有用的资源, 希望可以帮助大家更好的使用 VIM。

打造 Vim for PHP 的 IDE, 是本文的极好补充

<http://koch.ro/blog/index.php?/archives/63-VIM-an-a-PHP-IDE.html#codesniffer-integration>

用 5 个图表来讲解 VIM 的基本操作, 强烈推荐!

http://www.viemu.com/a_vi_vim_graphical_cheat_sheet_tutorial.html

VIM 使用的系列教程

<http://blog.csdn.net/mylxiaoyi/archive/2005/07.aspx>

一份台湾作者的 VIM 教学，虽然很老，但读起来很轻松，适合入门

<http://edt1023.sayya.org/vim/index.html>

简短的 VIM 速成，可以当作备忘参考

<http://linuxtoy.org/archives/efficient-editing-with-vim.html>

VIM 中文手册在线版

<http://vimcdoc.sourceforge.net/doc/help.html>

VIM 编写 PHP 代码的一些技巧

<http://schlitt.info/applications/blog/index.php?/archives/488-Comfortable-PHP-editing-with-VIM-5.html>

另一份 VIM 技巧文章，可以下载 PDF

<http://www.scribd.com/doc/263139/VIM-for-PHP-Programmers>

300 多个 VIM 配色方案在线预览与下载

<http://www.cs.cmu.edu/~maverick/VimColorSchemeTest/index-c.html>

VIM 的老家，找插件就在这里了

<http://www.vim.org>

最后，送上本文所有插件及配置的压缩包，解压缩到 vim 目录即可使用。

下载地址：http://www.dualface.com/blog/downloads/vimrc_plugins.zip



项目管理

成为团队领导者

Watts S. Humphrey

本文讲的是如何培养领导能力, 以及要成为高效的团队领导者需要做些什么。本文阐述了影响领导者绩效的一些问题, 并对完成高效领导工作所要采取的步骤提出建议。本文每一节讨论的都是领导能力的不同侧面。这些小节如下:

- 什么是领导力?
- 成为领导者和管理者。
- 领导角色。
- 领导的同时进行训练。
- 前路挑战。

1 什么是领导力

领导力被描述为领导的行为。它的意思是超前、领先和引导。它需要你站在团队的前面, 了解团队的动向, 引导团队。作为领导者, 你不能强迫手下按照自己的意志行事, 你必须激励他们。

1.1 让团队有追随你的信心

领导者的最佳定义是: 拥有追随者的人。毕竟, 失去了追随者, 就没办法领导。听上去简单, 但实际情况要比眼前的内容更复杂。首先, 你必须了解自己想要完成的目标。其次, 你需要向手下展示, 为什么这个目标值得奋斗。最后, 你必须实际着手行动, 引导他们来追随自己。

1.2 了解自己想要完成的目标

在软件行业中, 定义目标基本上不是问题。管理层确定好他们期望完成的工作, 他们要求由你来领导团队完成这项工作。因此, 你的目标就是完成工作。虽然你需要更准确地定义管理层的期望, 但整体目标还是清晰的。这也是 TSP (Team Software Process) 启动过程中 1 次会议的内容: 由管理层向团队解释他们的期望。TSP 启动过程可以帮助你完成担任领导者的一步, 至少当管理层可以清晰地描述期望的时候可以做到这一点。

1.3 让团队相信该目标值得奋斗

TSP 启动起到作用的另一处地方是让团队相信目标的重要性。当管理层解释他们对团队的期望的时候, 他们也应该解释原因。如果他们没有给出令人信服的理由, 也许你可以帮助他们来建立信任。目标是向团队传达这项任务对于单位、对于你、对于他们的重要性。

当然了, 这也是启动过程如此开展的原因: 帮助你向团队解释目标, 帮助你激励人员全身心投入到工作当中。

2 次启动会议是为了巩固目标, 让它们对于团队成员而言更具意义。你的目标是让团队相信, 目标非常重要, 值得为之努力奋斗。要做到这一点, 就必须理解成员们的兴趣所在, 这样才能让他们的个人目标与团队和管理层配合起来。如果每名成员都确信团队的目标与自己的目标是一致的, 那样就可以帮助小组凝聚在公共团队目标周围。

1.4 让团队追随

TSP 启动过程可以帮助你定义目标, 让团队成员相信它是物有所值的。让团队追随于你, 最终还是取决于你自己。这个步骤主要是信任和相互支持的问题。要在团队启动之初就建立起信任。如果在制订团队计划的过程中, 你展示出要依赖成员来完成最好、最有积极性的计划, 而他们又认为这些计划可以满足, 这样你就表明了你是信任他们的。之后, 在 9 次启动会议当中, 当你向管理层抗辩计划时, 当你没有获得团队首肯, 不能改变计划的时候, 你就证明了, 自己是可信的。在启动之后, 团队是乐于相信你的。虽然你还没有得到他们的完全信任, 但你已经接近了。

1.5 影响领导力的第一要务

领导力和信任并不是流行竞赛。最棘手的问题之一就是, 平衡团队一致性需求和让团队成员以非常规方式完成任务的临时性需求。虽然你想要团队接受你作为正式成员, 但你还是要像老板一样做决定, 承担责任。这种平衡行为很困难, 它包含着标准和纪律。你有目标要完成, 而且你也了解, 要想成功, 团队就必须以正确的方式完成工作。以前说过的海军上尉 Ben 的例子, 就着重强调了信任的重要性。同时它也展示了, 如果人们对手头的工作不感兴趣, 而只是在为他们要追随领导才这么做, 这个时候纪律性是多么重要。

使用 TSP 的时候, 最有可能影响领导力的问题就是纪律性, 所谓纪律性就是完成这项任务应当具有的准则。虽然与让团队冒着枪林弹雨地冲锋陷阵相比, 这并没有那么困难, 但不遵守纪律是人的天性。遵守纪律让人难受且没有乐趣, 但长远看来, 遵守纪律还是有好处的。要建立和维护纪律性, 就必须真正相信需要纪律, 它可以帮助团队以最实用有效的方式来完成任务。

纪律性工作包括规划和遵循计划, 跟踪和管理个人时间, 要记住质量是最高要务。它需要你对优质工作提出要求, 并且要了解怎样才算是优质工作。你自己也必须追求优质工作。虽然纪律听上去很简单, 但实际上它特别困难。那也是领导力和信任起作用的地方。

1.6 建立信任

在团队启动之后, 必须努力加强你已经建立起的初始信任。基本要求是要诚实, 要处于正常状态, 要一致帮助团队以最佳方式完成工作。只要你坚持说到做到, 不滥竽充数, TSP 过程就可以帮助你做到这一点。

Keith 是一名新任经理，他按时坐班，规划手下的任务和工作方式。他一个接一个地把手下叫到屋里去，告诉他们要干什么。他的手下把他看成是暴君。他永远不会倾听手下的言语，或者询问他们的想法。他只是发布命令。

结果，Keith 被手下吓了个半死。他认为，作为管理者，他一定要对部门的工作无所不知。他也知道自己并不能做到这一点，因此他很害怕暴露出自己的错误。他生硬的、无感情的管理风格只是隐藏其虚弱内心的一个外表。

这是信任问题。Keith 是很聪明的开发人员，但同时还是一个会犯错误的个体。他不能理解人都会犯错的道理，他也不相信手下如果发现他犯了错还会继续尊重他。他也不理解，人类存活这么长时间的原因是，他们一起工作、相互学习、相互支持。一起工作要有信心，学习需要谦虚，信心则来源于信任。

1.7 维持团队交流

信任是一个相当模糊的概念，看似很难建立。但是，如果你知道从何开始，也许它就不会那么难。事实上，团队交流是最好的开始方式。当开发人员评价上司的时候，他们抱怨最多的都和交流有关 (Raudsepp 1981)。我在 IBM 工作的时候，我曾定期开展过雇员满意度调查。雇员对管理人员最大的抱怨就是他们没有充分的交流。在每一次观点调查当中，这都是事实。开发专家是非常聪明、非常有创造力的人；他们有思想，有观点，他们想要参与并对工作施加影响。交流很关键，不仅是要告诉开发人员做些什么，而且还要显示对他们的尊重，集中他们的想法和能量来达成项目的目标。

2 成为领导者或管理者

我们已经讨论了领导者是什么，但管理者又是什么呢？这里又是循环定义：管理者是进行管理的人。现在我们来看什么是管理。管理则意味控制、指导、发布命令和期望服从。这里的感觉很强。听上去好像管理者留给手下做决定、做判断的空间不多。管理听上去好像是我们虽不介意，但还是不希望施加到我们身上的事情。

总体来讲，管理者可以看成是控制单位或设施的人。典型地，管理者要确定人员如何评价，他们是否应该加薪，应该加多少薪。因此，只要管理者关注着职业管理技巧，并不是尝试管理工作的细节，那么成为管理者并不总是坏处居多。但是，在琐碎的管理工作和领导决策之间保持应有的平衡还是比较困难的。当你的人员明白你是他们的上司、你对他们工作表现的评价和薪水拥有控制权的时候，这就会变得特别困难。因为，他们会区别对待你，不是只把你当成团队的另外一名成员。对于领导主题的两项讨论会澄清这些问题：领导你所管理的团队，以及领导不由你管理的团队。

2.1 领导你所管理的团队

领导由你所管理的人员，其主要问题在于动力。问题的本质是，你做工作是因为他们想要这么做，还是因为他们想要取悦于你。如后面所述，事务型领导发生在人们为了报酬而工作的时候，而变革型领导发生在他们做工作是因为他们相信工作的重要性、他们对参

与和贡献的机会感到兴奋。

虽然事务型领导对于多数工作来讲是标准模式, 但它还有一项本质的缺陷。团队成员的真正目标在于取悦管理者, 而不是完成优质的工作。虽然取悦似乎并不是糟糕的想法, 但这样的动机将会导致糟糕的决策。这并不是因为你必定会支持糟糕的决策, 而是团队成员并不知道你能支持什么。他们最终不会按照你的意愿行事。这样的事后指责将会导致混乱和错误。

当你领导你所管理的团队的时候, 让人们按照你的意愿来行事并不是难事。但是, 让他们采取主动, 做他们认为正确的事情则常常都有问题。你将会面临的主要问题是完成优质技术工作的需求。要鼓励团队成员自己思考, 采取主动, 要激发他们的创造力。你挑选这些开发人员是因为他们有完成工作的能力。你不能单凭自己来完成工作, 你需要的不只是他们的四肢, 你需要的是他们的脑力和思想。

TSP 过程就是要来解决这项问题的。通过把注意力放在目标、计划和绩效上面, 你就可以帮助团队成员关注于工作之上, 而不是思考你对他们的评价。随着你带领团队经过 TSP 启动过程, 随着你引导成员们计划和跟踪自己的工作, 你就可以帮助发展他们的任务和关系成熟度。接下来, 这也将帮助你变成变革型的领导者。

2.2 领导不由你管理的团队

当有一些或全部团队成员都不向你汇报工作的时候, 事务型领导就没有优势了。但是, 因为你还可以提供变革型领导, 因此这不应该是严重的问题。遗憾的是, 又有另外一种“并发症”。这些团队成员向另外一个人报告, 正是这些管理者来控制对他们的评价, 确认他们是否应该加薪。现在, 不存在管理问题, 反而出现了政治问题。我们把政治定义为获得权力的艺术, 而权力是引发行为的能力。

如果你需要那些并不是为你工作的人员对你效忠时, 首先你应该得到他们上司对你的支持。因为上司有权力迫使团队成员按照自己的意愿行事, 你需要让这些团队成员的上司来告诉他们完全参与到你的团队中的重要性, 如果他们确实全身心参与了, 那么他们将会得到积极的评价。虽然这是事务型领导, 但这种事务存在于团队成员和其上司之间。事务很重要, 因为你不会让团队成员的忠诚出现冲突。

一旦你认识到团队成员的忠诚性没有冲突, 那时你就可以把注意力放在变革型领导上面了。你的目标是让他们对工作产生兴奋感, 激励他们完成优质的工作。

2.3 成为变革型领导

要想成为变革型领导, 就必须让团队对工作兴奋起来。有许多种方法可以做到这一点, 而 TSP 通过满足自主型团队的 5 项基本条件就可以帮助你。如第 3 章所述, 5 项基本条件如下所示。

- (1) 成员感和归属感;
- (2) 对团队共同目标的承诺;

- (3) 过程和计划的所有权;
- (4) 制订计划的技能以及遵循计划的纪律性;
- (5) 致力卓越。

TSP 启动过程正是用来建立这些条件的。但是，在所有这些准备过程以后，你如何了解团队对于工作是兴奋的呢？你很幸运地拥有一支开发团队。这些开发人员把整年时间都投入到设计和系统开发工作当中。他们对工作会富有激情，因为那正是他们梦想的工作。在启动结束时，开发人员将会变成一支紧密团结的、充满活力的、富于凝聚力的团队。其挑战在于维护和形成这种活力和热情。本文下一节讨论了在建立和维护自主型团队的这 5 项条件的时候领导者的角色。

3 领导角色

TSP 团队领导者的角色和职责在表 1 中总结。下述小节描述了如何通过合理执行这些职责，来满足自主型团队的 5 项基本条件。那样在成为变革型领导者的道路上你将会非常顺利。

表 1 TSP 团队领导角色和职责

目的	当所有的团队成员都一致地满足他们的角色职责，遵循已定义过程，朝着达成目标和规范的方向工作时，团队就会变得最为高效。
目标	团队领导者的目地是： <ul style="list-style-type: none">• 确保项目成功；• 组建自主而高效的团队；• 充分调动全部团队成员的智力和能力；• 通知管理层。
角色特征	对于团队领导者而言，最有帮助的特征是： <ul style="list-style-type: none">• 你乐于成为领导者，很自然地就承担起领导角色；• 你能够确定关键问题，客观地做出决策；• 你不介意偶尔采取不受欢迎的行动，愿意推动人员完成棘手的任务；• 你尊敬团队成员，愿意倾听他们的观点，乐于帮助他们施展最佳能力，在更高的管理层次支持他们。
团队成员职责	所有的团队成员，包括团队领导者在内，都要以团队成员的身份为达成其职责而负责： <ul style="list-style-type: none">• 满足团队成员的承诺；• 遵循严格的个体过程；• 规划、管理和汇报个人工作；

	<ul style="list-style-type: none">与团队和团队所有成员合作，维持高效多产的工作环境。
领导力	<p>团队领导者领导团队：</p> <ul style="list-style-type: none">维持对团队目标清晰而连贯的关注；确保所有团队成员在工作时有生产力且高效率；确保紧张感，在实用的地方推动加速任务；一致追求每日结果，要认识到每天的进度滑坡；激励和支持团队；挑战团队和团队成员的决策，询问他们曾经考虑过的候选方案；维持对其他相关工作的关注，确保团队利用了以前的可用结果；代表和提供管理层对团队的支持；代表和提供团队对管理层的支持。
人员管理	<p>团队领导者处理团队所有潜在的问题：</p> <ul style="list-style-type: none">管理项目人员分配，招募和培训；对团队交互问题要敏感，当团队自己不能解决问题的时候，要采取步骤解决问题；在安排工作的时候，要考虑团队成员的兴趣和能力；确保任务和工作压力与每名开发者的技能和能力相匹配；保护团队、防止出现偏差。
团队培训	<p>团队领导者维持一个严格而高效的工作环境：</p> <ul style="list-style-type: none">促进团队内沟通；让团队获取信息，领导每周项目状态会议；确保团队和团队成员制订了自己的计划；评审个人和团队计划以确保它们进取而实际；确保团队成员遵循了严格的个体实践；确保团队成员坚持约定的团队标准和过程。
质量管理	<p>团队领导者维持对质量的一致关注：</p> <ul style="list-style-type: none">确保定期搜集和分析质量机制；定期激励团队以造成其质量目标；领导团队评审每次合法的集成、系统测试以及用户报告缺陷（DEFECT表和TESTD脚本）。
项目管理	团队领导者管理项目：

	<ul style="list-style-type: none">• 处理资金问题;• 与管理层和其他团队或部门一起解决问题;• 可能的话, 要把所有项目任务委托给团队成员;• 处理所有病态的事件和问题, 至少要到它们定义下来, 分配给团队或团队成员以求注意为止;• 对进度状态以及加速或推延的时机维持关注;• 定期向管理层和客户报告项目状态;• 领导风险评估和跟踪;• 领导问题跟踪和解决;• 确保所有的需求和设计假设以及不确定性得到了立即而详细的验证;• 参与配置控制委员会;• 确保所有的需求和设计时机都及时评估了影响力;• 在变更将会大大影响项目成本或进度的地方, 确保在变更实现之前, 计划事先得到了调整和批准。
团队领导 主要活动	<p>团队领导者激励团队来执行任务并解决问题。</p> <p>每周, 团队领导者都要举行团队会议以便:</p> <ul style="list-style-type: none">• 跟踪所有承诺任务是否已经完成;• 检查所有的团队成员已经提交了所需数据;• 检查 TASK 和 SCHED 模板对于团队和每名团队成员而言是最新的;• 检查所有的 INS、SUMP 和 SUMP 报表已经如期完成;• 检查开发任务的状态;• 检查变更、ITL 和风险活动;• 强迫落后团队成员及时提供所需数据; <p>每周, 团队领导者要:</p> <ul style="list-style-type: none">• 向管理层和客户报告团队状态和进度;• 维护项目备忘录;• 在每个阶段结束后以及在项目结束后、团队领导者领导团队来生成或更新项目最终报表 (SUMMARY 规范)。

3.1 成员感和归属感

成员归属感有两个组成部分: 参与所有重要的团队活动, 在团队当中有一个可接受的角色。几乎毫无例外的是, 领导、管理和培训职责都会包括与团队成员进行个别交流或整体交流。问题是, 要将整个团队视为一个具有凝聚力的团队的同时, 如何通过这些交流手

段去认识和了解每一名团队成员。必须拥有一处公共工作场所，举行定期团队会议，公开解决问题，客观地使用事实和数据。

虽然这些职责相对比较直观，但你能够有效地完成这些职责的方式却不明确。例如，有一位团队成员由于其糟糕的工作表现，已严重危及到整个团队的工作绩效时，很可能会遭到团队其他成员的排斥，你如何避免此类事情发生，而仍然保持团队的团队精神呢？这是态度问题。在我们童年的时候，都曾经玩过“指责”游戏。不管什么东西被泄露了、破坏了、丢失了，第一个问题总会是：“是谁干的？”有罪一方会因其祸害而受到指责，他还要让其他人相信自己永远不会再犯下同样的错误。虽然教育子女这可能是高效的方法，但是对于复杂的开发人员而言，这种方法可能并不会奏效。

一个有指导意义的示例是，核潜艇上的错误处理方式。推进系统内部发生的任何问题都会引起舰长的审问，相干的舰队成员都要集合起来弄清楚哪里出现了错误，以后如何防止此类事件的重复发生。负责人要讲清楚他做了什么，并解释原因。问题预防要包含个体补救措施，每一个人都要关注如何避免未来会危及到潜艇的潜在事故。舰长随后要向舰队司令报告事件和解决方案。

在和团队一起处理这种情形的时候，要私下处理纯粹和人的问题，但对于涉及团队的任何事情，都要和团队一起处理。例如，如果在一次审查当中，一名成员的组件发现存在严重的质量问题，那就要让团队决定要做些什么。因为开发负责人肯定会采取守势，至少在最初是那样的，所以要事先私下讨论问题，解释你的计划和理由。而且最好是指出，任何人都会犯错误。重要的是从错误中学习，让错误对团队的影响保持最小。

3.2 对团队共同目标的承诺

因为目标非常重要。目标是团队动机的重要元素，它们需要你的持续关注。领导能力的首要需求是对团队目标要维持一致而可见的关注。要维护这种关注度，就要把目标拆分成短期里程碑，这样就可以定期跟踪进度。还要确保每名团队成员都了解下一处里程碑，理解其工作如何影响团队达成目标的能力。要做到这一点，一种方式是安排定期管理层评审，然后举行每周团队更新，以检查里程碑状态。

确定瓶颈和问题所在。另一个辅助步骤是确定达到里程碑所涉及的所有任务，将它们列举在有任务分配和日期的图中。随后当团队成员完成任务之后，要让他们核对任务。如果状态图公开发布，每个人都会了解里程碑状态以及为了达成下一次里程碑他们必须要完成的职责。

3.3 过程和计划的所有权

对你所拥有的东西，你都会感觉到有责任，要照料它、改进它，观察它是否被合理地使用了。那也就是你要求团队对项目的感受方式。在自行拥有和团队持有之间有着很大的不同。当管理层确定下来过程和计划的时候，多数成员都会感觉到团队领导者是真正拥有它们的。这很像是拥有汽车和租用汽车之间的差别。租用的汽车你会小心驾驶，但你永远不会考虑洗车或为它服务。

虽然你想要团队成员正确地使用过程，遵循计划，但你想要的却不仅仅是小心驾驶；你想要他们表现得像车主一样：确定问题，提出改进建议，负责完成这些改进。事实上，无需告诉他们要如何往下进行，无需事先求得任何人的同意，这就是你想要他们采取的方式。这是一种所有权的感觉。

那么什么是自主权呢？自主权被定义为拥有的职责、控制和权利。但是，所有权含义更广泛。它还含有唯一的意思：你拥有它，其他人不可以拥有。作为拥有者，你是唯一的拥有者，拥有对这项所有物的最终话语权。

虽然所有权感可以产生自主型团队所需的行为，团队的内聚和协作精神与所有权的单向排他性质之间是存在冲突的。TSP 解决这种二分现象是通过建立团队角色、让团队对角色所有权达成一致、对于界定功能承担唯一性职责来完成的。TSP 角色的意图有三方面内容。

- (1) 定义必须合理解决才能完成成功项目的关键性团队活动；
- (2) 确定处理每项活动的人员；
- (3) 赋予每名团队成员一种所有权感和部分工作职责。

角色管理人员在 TSP 的 2 次启动会议中确定，但团队成员一般都不能立即理解或掌握这些角色的重要性。这需要时间，也需要强化。要提供强化效果，就需要考虑每种团队活动中的角色职责，确保所有的角色管理人员都了解你和团队对他们的期望。如果某名角色管理人员不知道该做什么，那就要回顾你所拥有和期待的角色以及相关事件与问题。随后要指挥角色管理人员事先描述这些领域，确保问题在影响到团队绩效之前就得到解决。

在 TSP 启动之后，你和团队将会相信你的过程和计划。随着工作的完成，你会对任务本身有更多的了解，也会弄明白如何改变计划以更好地反映自己现在的认知。虽然工作和计划之间的渐变分歧非常正常，但你也必须在计划失去实际意义之前先更改它。如果情况确实如此，那么团队就不要再遵循你和团队以及管理层所协定的计划了。更糟糕的是，也不能报告进度，或者让他们警惕一些问题。当管理层发现他们不能再依靠你来遵循承诺计划的时候，他们就会介入指导你的工作。那时候自主型团队就不复存在了。

因为调整团队计划会花费时间，因此你肯定不会过于频繁地执行这项操作。但是，如果等待的时间过长，很快你就不能跟踪项目状态，重新平衡团队的工作负荷，评估变更造成的影响，或者预测任务完成的时间。当团队计划出现了这些问题的时候，要马上改正它们。解决这项问题有许多种方法，典型示例如下：

- 变更的目标。如果项目目标已经发生变化，那就请全部重新启动。加速交付日期、增加功能或者团队资源中的变化都会引发这项问题；
- 人员变更。对于小的变动，一般要举行简短的团队计划会议，来重新评估和重新平衡工作负荷；
- 发现。如果你发现产品比预测的大许多，或者必须开发而不是重新使用一些经过计划的商业现货功能，那就要尽可能地组织一次全面的重新计划会议。如果进度影响

很大的话，那就要通知管理层，随后举行一次完整的团队重新启动，包含第 1 次和第 9 次会议。

处理这类情形的一般准则如下所示。

(1) 对于影响团队目标和资源的主要改变，要尽可能地举行一次全新的启动或重新启动。如果变更对进度影响过大，以致危及到项目本身，那就要包含第 1 次和第 9 次启动会议。

(2) 对于那些在团队目标或活动上面没有较大影响的变更，要尽可能地举行一次全面的重新计划会议，随后通知管理层任何可能发生的进度调整。还要和团队以及管理层一起评审以前的启动计划，检查它是否应该改变。

(3) 对于影响团队当前工作的变更，要尽快举行一次再平衡和重新计划会议。

(4) 对于只影响到一、两名成员的变更，要让他们调整自己的计划，如果变更还影响到了其他人，那也要让他们和团队一起评审这些计划变更。

3.4 制订计划的技能以及遵循计划的纪律性

要制订出准确、谨慎、完整的计划，所有的团队成员都要在首次团队启动前接受过 PSP 培训。但是，实际遵循详细计划的纪律性只能随着时间发展起来。如果你还没有更新计划，准确地反映工作的话，那团队就不能遵循它。但是，如果你保持计划的新颖性，就能确保只要做到两件事，团队成员就可以遵循计划。

(1) 确保你自己遵循了计划。在评审和团队会议中使用计划，定期依据计划检查项目状态，使用计划来定义工作优先级，为管理层评审做好准备。

(2) 当你和团队成员会谈的时候，要检查其计划是否是最新的，他们是否遵循了计划。如果没有遵循计划，那就找出原因，解决所有的问题。然后还要坚持让团队成员遵循他们的计划。如果你定期、持续地做到了这些事情，那么团队当然就可以遵循计划。

3.5 致力卓越

最终绩效标准是卓越。卓越的工作是苛刻的，苛刻的工作通常都不容易。告诉团队，如果工作很容易的话，那就不需要这么优秀的人员了。不要让他们在看到困难之初就放弃掉。不要接受辩解。要有创造性。要把问题都看成是挑战。正确的方式总会是最好的方式。如果有人很着急，想走捷径，那就要提醒这个人现在还不是做事的时候，如果采取错误的方式，时间肯定会不足。要使用判断和常识，当开发人员定义好了最佳行事方式后，要坚持让他们按照那种方式做事。

卓越的纪律性极难维持。这就是为什么专业运动员要有陪练和培训师、表演艺术家们要有指挥和督导。以优质方式完成困难的工作是不平常的，不能从书本中或者从偶然观察中习得。面对任何苛刻的任务，人们的绩效都会逐渐提升。但“只是通过更加努力”，绩效永远不会得到提升。它要经过培训和实践，需要数据和反馈，要花费时间。可持续改进需要已定义的方法、准确的度量以及专业的指导。

如果一项任务和产品开发一样具有挑战性，持续改进的唯一途径就是接受培训师的指导。培训师必须理解人们正在执行的工作和方法。这表明如果开发者准备改进的话，他们必须使用一套已定义的过程，必须度量并跟踪工作。只有那时，你和团队培训师才会有支持他们的数据。

纪律性还涉及信任。一方面，你必须相信开发者可以做出优异的工作，另一方面，你也知道他们也只是人。如果你正在指导一名奥林匹克运动员，那么你就不需要怀疑这名运动员的卓越能力；在这里卓越性就是假设。如果不专注，没有个人承诺，没有哪名运动员会抵达奥林匹克赛场。在软件行业里，我们常常能够遇到世界冠军，但我们不知道这一点。他们是有超常能力的一群人。要想达到他们的水平，肯定要花费大量的努力、时间和金钱。他们也都会承诺完成优质工作，但他们也是人。他们也需要指导和支持。你的角色就是额外推动他们完成真正优质的工作。

3.6 紧迫感

除了自主型团队的五项基本特性以外，为了完成优质工作，还需要投入活力和激情。第1章里曾说过，“老板的速度就是团队的速度”。如果你过于随和，做事总是慢悠悠的，那么你的团队也会放松滑坡。如果项目非常重要，那就要以其重要性来对待它。不要接受拖延和辩解。“进度一次落后一日”。活力和激情是可以感染的。要一直保持冲劲。相信自己的人员，他们会追随于你。

关键在于你的行为。你自己真的相信团队的目标吗，你是不是非常进取地工作以达成目标呢？你是在支持和使用团队吗，你挑战开发人员做到最好了吗？产品开发并不容易，但你拥有高素质的人员。你的工作就是激励他们完成优质工作。它必须有最高优先级。有一支团队，为了设计一套软硬件产品，已经工作了好几个月时间。他们执行计划、筹措资金，他们被告知，以其所需的获利价格，根本没有市场。开发人员跑到上司 Judy 那里，告诉她必须要取消项目。Judy 没有接受他们的建议。她告诉他们，你们的任务是开发产品，如果投资抵不上初期计划，他们必须弄清楚原因，制订出更好的计划。经过了几个月的艰苦工作，团队终于制订出了一套能够满足财务标准的计划，团队最终生产出了一套成功的产品。

这些问题在开发工作中很常见。这也正是开发的含义：解决问题。如果这项工作以前曾经做过，那么我们就不再有需要了。因此，当开发人员面对困难、准备退却的时候，要挑战他们超越困境，避开它，或者寻找更好的解决途径。不要过快地气馁，不要让团队退却。

4 在领导的同时进行训练

在领导 TSP 团队的时候，主要挑战是激励人员完成优质的工作。因为 you 和团队成员的许多交互都与你完成工作和更好地完成工作的方式有关，因此你常常会觉得你像是培训师而不是管理者。这就引出了下面的问题，你如何在领导人员的同时来培训他们。显见的答案是同时完成两份工作。然而，这是不可能的，原因有两个。第一，针对团队领导者和

培训师必须要做的工作，你没有那么多时间。第二，作为团队领导者，团队成员必须把你当成是团队领导，而不是培训师。

团队成员可以把团队领导者看成是培训师，把培训师看成是导师和知己。即使你想要这么做，你也不会完全接受培训师的模式，至少一开始做不到。但是，如果你开始像个培训师那样思考和做事的话，那就要表达出对成员们的兴趣十分关心的样子，询问他们的个人目标，那样手下也会慢慢觉得你就是培训师。那么你要如何做到这一点，同时还有时间领导项目呢？回答这个问题要涉及领导任务、培训任务以及整个培训态度。

4.1 领导任务

尽管有许多因素对程序员的行为都会有所贡献，你，即团队领导，对人员的要求以及指导和奖励他们的方式，会比单位里其他人对其行为的影响更甚。作为领导者，你必须满足团队的承诺，组建起团队的能力，有消息及时通知管理层。你必须设定团队的步调，建立并维护高标准，支持和保护团队。要做到这些事情，就必须委托大部分传统意义上的管理层职责。

角色管理人员可以帮助你准备每周团队和管理层会议，评审团队成员的数据，处理团队的控制和支持任务。如果单位里面有支持小组，他们也能帮助你。质保小组可以提醒你质量控制问题或者计划上的疏漏，过程小组可以分析团队的数据，提供过程改进指导。要成为培训师和领导者，就必须花时间和自己的人员待在一起。这意味着要借助任何个体或小组的力量来帮助你运营项目。

4.2 培训任务

培训的主要任务是聚拢人才。培训师不能笼统地看待团队成员；他必须以个体来对待他们。这需要每周在每名开发人员身上都投入时间。对于一支具有相当规模的团队，这需要大量的时间投入。

在我的第一个项目中，我有八名开发人员，要设计一套复杂的密码通信系统。因为我刚到单位里来，以前没有做过这样的系统，有许多内容需要学习。我通过提出问题开始。我每天都会和每一名开发人员交谈，弄清楚他的工作和方法。那样如果有技术问题的话，我立刻就会明白，是因为被他人的工作阻碍了呢，还是有个人问题了。几周以后，我了解到开发人员的强项、弱点和兴趣。现在，我理解了如何完成自己的工作，我也掌握了每名成员需要帮助或支持的地方。

如果你每天都投入时间和每名开发人员在一起，你就会了解谁会有问题，而且还会理解哪名成员分配不当或者可以迎接更大的挑战。只要团队规模不太大，这些日常的讨论就不会花费太多时间。但是，它们会赋予你成为高效培训师的第一手知识。

要帮助手下改进，那就要利用好每名开发人员的技能。要关注开发人员当前的任务，认识到改进是渐进的。期望增量式改进，奖励每一次成绩。要把注意力放在激励上面；它是你所拥有的最有效的改进工具。不要只把注意力放在新加入的或者需要帮助的开发人员

身上。你还要和专家以及那些以未知的规律做事的开发者交谈。询问这些开发人员他们的工作和方式。当他们出现问题的时候，要问清楚替代方案，以及他们需要指导的地方。你从这些人身上可以学到很多内容，通过询问问题，你也可以刺激和激励他们。通过回答问题，他们学到的内容要比你从问问题的过程中学到得多。只要你愿意问问题，乐意学习，那样即使在培训最优秀的开发者的时候，都不会有困难。

4.3 培训态度

对于培训态度而言，决定性测验是你处理升职的方式。假定首席设计师告诉你，他愿意担任团队领导者。那么，如果你听说团队领导者要开辟另一个部门，假定你认为他是优秀候选人，那么你会推荐你的首席设计师吗？如果没有，那么你的行为就是团队领导者的行为了，而不是培训师的行为。既然你不想失去这名开发人员，如果你提升他的话，很快大家都会知道你提升了自己最优秀的手下。那样你就可以毫不费力地得到更多更优秀的人才。

成为高效培训师的关键点在于要同对待任务一样地对待手下。虽然看似不太现实，但它实际上是可以赢得开发人员忠诚信任的唯一方式。他们都是聪明人，很快就能弄明白你的兴趣和优先考虑的事项。如果你的首要优先事务是任务，他们会清楚，不能信任由你来照顾到他们的兴趣。但是，如果你真正关心他们的发展，他们就不太可能会担心自己的事业，会更好地将精力投入到工作当中。这正是产生优质工作的源泉所在。

5 前路挑战

现代技术发展趋势需要领导和培训师角色的融合。其原因与工作中的趋势以及这些趋势如何影响工程任务的本质相关。工程中的挑战，反过来也会影响工程力量和领导角色的本质。最后，这些变化倡导以人为本的领导风格，或者更具体地说，领导者行事要更像是培训师。

5.1 集成化工程

计算机革命开始的时候系统开发涉及了许多技术和业务规范。项目需要逻辑电路设计师、硬件封装设计师、指令集架构师、规划详细系统逻辑的逻辑设计师。程序员开发系统程序，维护人员进行诊断，制造人员定义加工、组装和测试过程。项目还会涉及装配、应用和市场人员，以及人员、物料、式样、广告和金融组织等因素。系统设计包含了许多活动，尽管这些活动全都相关，它们也可以看成是大量接口和相互依赖性受限的独立小组。

现如今，个体纪律性的进步已经到达极限，开发人员必须越过纪律性而看得更远，进一步寻求机会。可以把系统管理看成是一种多维问题，其中每种新技术都会增加一个新的维度。在一两个维度上看似不能解决的问题在三维、四维上经常会是小事一桩。例如，早期集成电路都是二维布图结构。通过剪切导线长度来提升性能的做法经常是不现实的。但是，出现多层芯片以后，许多诸如此类的二维问题在三维环境下都变得容易解决了。

类似地，从设计进度中排除一个月基本上是不可能的。但是，如果把设计和制造看成网站：<http://www.phpchina.com> 投稿：phper@phpchina.com 《PHPer》 74/81

是一项集成活动，通过早期加工、阶段性设计发布、制造设计，往往可以节省几个月的时间。通过交叠任意规范集，可以执行许多类似的节约操作。如果机构看不出节约的潜在趋势，那只能表明他们没有形成多维的思考习惯。

随着竞争的进一步加强，熟练开发人才的广泛供应，提升绩效、降低成本、缩短工期的压力持续增加。很难发现有哪部分开发可以被当成是孤岛一座。现代化工程的集成程度几乎在各个方面都越来越高。当我们寻求成本、进度、绩效或质量提升时，集成化往往就是答案所在。在未来，我们可以料想到对于集成化的迫切需求。

5.2 集成化领导

如果你和大多数团队领导者一样，把自己的大部分精力投入到了某项事业中。虽然以前你可以从某一维看待这个世界，但现在你必须站在更广阔的环境中看待它。你必须关注完整的产品生命周期。例如，需求和设计工作可以交织在一起吗？系统、硬件、软件、测试、发行、制造或服务小组情况怎样？任意相关分支集合可以更好地关联起来以提升整体绩效吗？

要想集成多种专业，就必须消除小组之间的接口。但是，没有了接口，管理者之间的交流就会变慢，带宽会变窄。没有什么时间可以借助中间人来传达信息，要传达的信息太多了。因为不能再依赖管理者在开发人员中间中继指令，专家们必须直接交流，他们必须要被赋予权力，可以在解决问题的时候排除管理层的干预。领导集成化项目团队的挑战是多方面的，它们需要一种以劳动力为中心的领导观点。这些挑战包括：

- 在多维环境中领导；
- 发展多维开发人员；
- 委托和包含；
- 加强劳动力。

5.3 在多维环境中领导

在多维环境中领导团队的时候，很可能要管理几种不同的分支，几乎可以肯定你要处理更多的分支。在同一支团队、相关团队或独立小组中，要操作这些分支，就必须协调好所有的工作。如果有的团队成员来自于其他分支，他们很可能要接受你与其专职上司的矩阵式管理。在矩阵结构中，开发人员一般要由可以理解分支细节、可以保证专家们经过了适当培训与评估的技术经理来管理。随后这些开发人员会分配到由团队领导者指导和管理开发工作的项目团队中去。随着使用越来越复杂、集成度越来越高的技术，这类矩阵结构将会变得更加普遍，在许多分支当中你需要依赖专职经理来提供指导。

5.4 发展多维开发人员

在领导集成化项目的过程中，努力像集成技术的计划一样详细地整合团队。要让所有的开发人员都忠诚，都做出贡献，那就要关注于目标，要调动整个团队参与工作计划和团

队运营。让每个人都制订个人计划，参与团队会议，参与决策会议。除非开发人员表现得都像是一支集成化和相互依赖的团队，否则他们就不可能会创造出一种集成化和相互依赖的产品。

5.5 委托和包含

如果你所领导的工作用到的不止是自己的专长，那你就必须要像执行人员一样行事。现在你所管理的工作是你所没有完全理解的。在这种情况下有些人会感到不舒服，要努力变成这些新领导的掌握者。随着技术的进步，交叉学科集成程度的提升，这种方法是不切合实际的。更重要的是，这种方法会错失良机。开发人员都应该运用自己的专长。你要把精力放在工作的质量和纪律性、项目目标、项目的许多专业和技术之间的接口上。这些接口很重要，因为其他没有什么人可以管理它们。只有团队领导者和程序经理才可以负责技术之间的接口。

5.6 加强劳动力

与以前相比，现代开发劳动力必须要更加灵活、异构和互相依赖。有才能的开发人员需求巨大，在可预见的未来，这种需求可能会持续下去。要管理劳动力资产，就必须获得有才能的人员，加强和合理应用他们的能力，要设法留住他们。但是，人员越有能力，他们对竞争者的吸引力就越大，要留住他们就越困难。在许多开发单位里，人员调整都是大问题，必须得到足够的关注。研究表明，开发者离开单位并不是因为他们想走，而是因为他们被错误对待，才开始谋求其他职位（Shellenbarger 2000; McLean 1996）。

经验表明，如果雇主提供给开发人员竞争性薪水和津贴，尊重他们，大力控制其工作方式，他们就会忠诚。对待人员的方式在很大程度上决定了人员的忠诚度，雇员忠诚在很大程度上决定了人员调整。作为团队领导者，对手下来说，你就是单位，你对待他们的态度会被他们当成是单位对待他们的态度。这也正是培训和领导必须融合的原因。领导关注

于项目目标，而培训关注于人员。要高效地完成工作，就必须成为高效的领导者，但也必须要以培训师的角度来思考和行事。

6 小结

本文讨论了团队领导者的角色以及高效领导的关键性元素。领导力被定义为领导的行为。领导意味着超前、领先和引导。领导者不要求或强迫其人员来追随；他们会使得手下乐于追随。领导力还涉及纪律以及领导者使得人员追随自己的能力，即使当团队成员以异于往常的方式行事的时候也是如此。团队领导者必须赋予他想要团队成员效仿的行为以具体形式。TSP 启动过程可以帮助发展这类领导者。

当你领导由你来管理的团队时，在要求人员按照你的思维行事的时候，很少会遇到问题。但是，在要求人员采取主动，按照自己认为正确的方式做事的时候经常会遇到问题。这需要变革型管理。TSP 启动过程可以建立起变革型领导的条件，对于人员并不全部为你工作的团队，它甚至也会有效。到了启动结束的时候，开发人员将会开始变成一支充满活
网站: <http://www.phpchina.com> 投稿: phper@phpchina.com 《PHPer》 76/81

力而内聚的团队。你的挑战是维持和建立这种活力与热情。

要想成为变革型领导者，必须满足五项条件：成员归属感、承诺于公共团队目标、过程和计划的所有权、制订计划的技能以及遵循计划的纪律性、致力卓越。TSP 启动过程为建立这些条件提供了一个基础，但你必须继续那个过程。本文提供了指导，并对如何做到这一点提出了建议。

你对团队的职责是设定步调，建立并维护高标准，维护过程纪律性，并且支持团队。活力和激情是可以感染别人的，纪律性极难维护。要做到高效，就必须相信团队，并且要驱动成员们完成优质工作。最佳领导者能够挖掘成员们的能力，甚于成员们的想像。关键在于要相信团队，要激励成员们努力，直到最终成功为止。

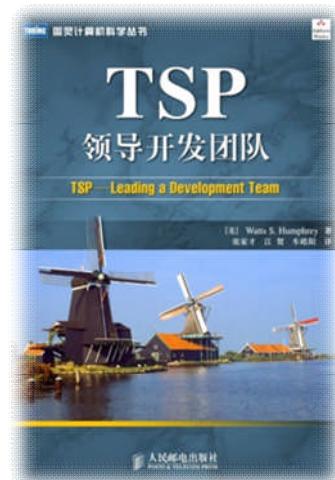
你还必须了解如何表现得像是一名培训师。持续完成严格的智力工作是很困难的，没有动力的人员是做不到的。如果行事像培训师一样，并以个体来对待每名成员，就可以提供此类培训指导，对优质工作提供支持。

本文结束时讨论了专业工作的变化本质，以及对于团队领导者的要求。现如今，提升绩效、降低成本、缩短工期的压力持续增加。集成化是对这些压力的自然解答，可以料想到未来对于集成化的迫切需求。但集成化工程需要集成式领导以及持续增加的多种劳动力。

集成式领导策略关注于工作的质量和纪律性、项目的目标、专家与技术之间的接口。最后，它还关注加强劳动力。为了做到这一点，必须吸引有能力的人员，培养他们的技能，让他们控制自己的工作。对于团队来说，你就是单位，你的行为将会直接影响到他们对单位和团队的忠诚度。要使团队完成最优秀的工作，必须成为高效的领导者，同时还要以培训师的思路来思考和行事。

TSP 是由“软件质量之父”Watts S.Humphrey 提出的一种采用广泛的团队过程。在本书中，作者以深厚的专业功底，准确地向团队领导人员展示了如何领导接受过个体软件过程（PSP）培训的软件工程师团队。书中探索了高效领导和团队协作的各个方面，包括构建合适的团队、TSP 启动过程、制造高质量产品的过程、项目评审、有效使用领导者和团队的职能等，书中还讲述了项目负责人的职责范围、领导能力的基本要素以及可能会面对的许多问题。

网上购买地址：<http://www.turingbook.com/Books>ShowBook.aspx?BookID=126>

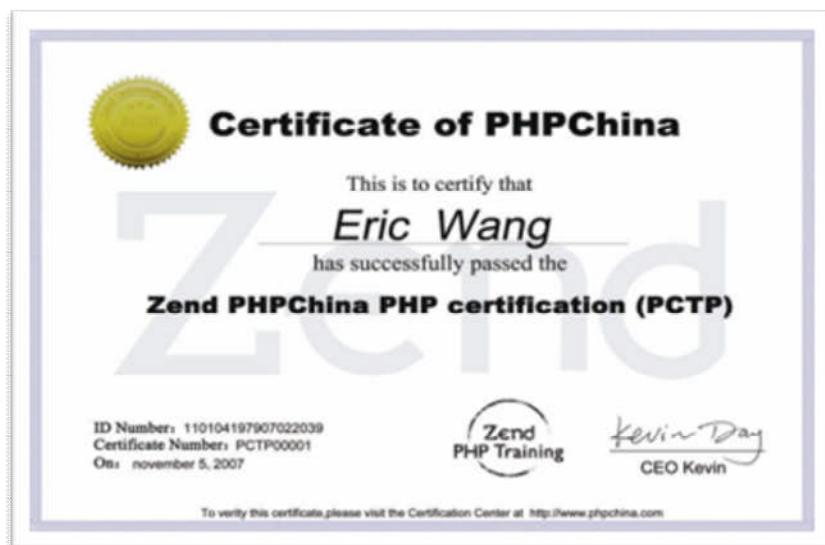


学专业课程 获权威认证

——PHPChina PCTP 培训认证介绍



随着 PHP 技术的不断发展，PHP 技术的应用越来越广泛也更加深入，对编程人员的要求越来越高，PCTI 入门课程已经不能满足企业的需要，应广大 PHP 爱好者的要求和国内互联网企业对 PHP 高级编程人员的急需，PHPChina 推出了 PCTP (PHPChina Training—Professional) 高级认证课程，本课程适合对象为对 PHP 有一定基础或者从事过 PHP 开发相关工作想继续深入学习的编程人员。



PCTP 认证证书样本

PCTP 课程内容简介

首先，我们纵向来看整个 PCTP 的内容安排，PCTP 课程包含 4 大块技术内容，即：PHP5 在 Web 开发中高级技术、数据库技术、Zend 产品和软件工程以及项目管理内容。

1. PHP5 的应用随着时间的推移已经取代 PHP4，PHP5 中相对比较完善的面向对象技术也不断应用在 Web 开发中；数据抽象层介绍了 ADODB 和 PDO，这是数据库与 PHP 之间的完美结合，可以大大提高开发效率，两种方式各有优点；Mail 和模板同样是 PHP 中的热点，特别是模板的应用，模板的种类，在此，我们主要以比较完善和应用比较广泛的 Smarty 为例讲解；Web 服务算是 PHP 的扩展功能，这也体现了 PHP 趋于强大的特点，在 PHP5 中改进了 XML、SAX、DOM、XSLT、SimpleXML 等扩展。
2. 数据库技术是应用开发的基础技术，80% 以上的应用系统需要使用数据库来存放业务数据，当然数据库技术也是个非常复杂的技术，这里主要是针对数据库在 Web 开发中的应用，内容包括数据库优化、规范化、字符集、触发器和数据库设计技巧等知识。
3. Zend Platform 和 Zend Guard 是 Zend 公司的完美产品同时也是 PHP 代码运行优化

的最佳平台和加密工具，它们被广泛的应用于大型服务器上，为 PHP 保驾护航。

4. 软件工程与项目管理一些纯理论概念，来源于实际，同时要应用于实际，是程序员走向更高一层次的基础知识。

PCTP 高级课程是 PCTI 初级课程的进阶，在学习 PCTP 前一定要有 PHP 基础、数据库基础和项目经历，在 PCTP 课程中的各章节之间的联系比较小，只有在项目前加入了 SVN 内容，这部分内容也是希望在后面项目中应用的，综合前面所学知识完成项目。

PCTP 课程特点汇总

1、以就业为导向 —— PCTP 是什么样的课程？

以就业为导向是 PCTP 课程最根本、最核心的特征。PCTP 是在 PCTI 初级认证的基础上的更高一级的认证，是品牌认证，如果说 PCTI 认证是面向 PHP 初级程序员的，那么 PCTP 认证是培养高级程序员，高级软件工程师，PHP 项目经理的认证，获得这个认证有助于学员更好的就业和发展。

通过 PCTI 认证的推广，目前已经培养出众多的 PHP 程序员，但是随着 PHP 技术的完善和被众多企业的应用，暴露出 PHP 程序员急缺的特点，特别是 PHP 高级程序员的缺少，为此推出了 PHPChina PCTP 高级认证课程。

2、以实用技能为核心 —— PCTP 如何选择课程内容？

软件开发领域内所涉及到的技术内容是非常多的，不同的行业会使用不同的技术，不同的项目也会使用不同的技术，在技术选择上我们遵循了实用原则和 80/20 原则：

实用原则：所选择的技术一定是能够解决实际工作中的实际问题的技术。打个比方，假设我们是在学习修车，关于汽车原理、发动机原理、轮胎橡胶特征我们就不会花费精力去学习，因为这些内容在修车过程中很少会使用到，而如何使用各种检测设备和螺丝刀等工具、以及不同品牌汽车修理和保养的经验，则是汽车修理人员最迫切需要掌握的技术。因此 PCTP 课程主要内容也集中在如何解决软件开发项目中所涉及到的技术工具、技术框架、开发流程和编码调试经验等方面。“不要去背诵对象有哪些方法、属性，而是要去使用这个对象去解决实际问题”。

以用为本、学以致用、用时才学、学了就用、综合运用是 PCTP 选择课程内容的最终标准。

3、以案例为主线 —— PCTP 如何组织课程内容？

延续了 PCTI 课程的经验，PCTP 内容的安排是以案例为中心来开展的，并适当地把多个案例联结成一个工程项目。在每一门课程中，都采用了一个实际的开发案例来组织技术内容，理论课上老师会通过一个案例、小项目的讲解来引出技术内容，而上机课要求学员自己逐步完成一个类似的项目。

在案例的选择上，我们在考虑到案例的实用性的同时，也尽可能地提高案例的趣味性、并加强与日常生活中遇到的问题和现象的联系，从而帮助你理解案例内容。

通过这种案例教学的方式，学员不会迷失在浩如烟海的知识中，同时会具备更多的行业知识和项目经验。

4、以动手能力为突破点，以项目经验为学习目标 —— 怎么样才能学好 PCTP 课程？

PCTP 课程最终的学习目标是：“能够运用 PHP 以及相关技术，在团队合作的基础上，参与到整个软件项目开发过程，熟练、规范地完成企业级 B/S 系统模块的编码、调试和文档编写，成为企业紧缺的技能型应用人才”。学习完 PCTP，相当于具备两年或三年的实际软件项目开发经验，这也是目前绝大多数软件企业招聘时对高级程序员的最低要求。

把发动机的各项参数背得再熟练也是不会修车的，而快速判断问题发生的原因，使用扳手、螺丝刀快速搞定问题才是正道。修车高手只能通过大量的实践、总结、再实践才能够达到快刀斩乱麻解决问题的境界。同样，PCTP 的学习也要求学员“动手、动手、再动手”，要敢于编码、乐于编码、大量编码才能够达到熟练的程度。

现代企业要求合作，代码规范是团队开发的基础，没有人能看得懂的代码，不管自己觉得写得如何漂亮，终将被称之为“垃圾代码”，在全部学习过程中，我们要求学员：“代码运行通过了不算代码写完了，代码规范了才算”

我们认为，只有通过大量的、反复的动手实践，才能具备熟练、规范的编码和调试能力，具备真正实用的技能，才有可能被称之为应聘者中“有项目开发经验的人”，成为一名企业真正需要的“人才”，在日益激烈的职业竞争中拔得头筹，PCTP 向这个目标迈进了一大步。

想了解更多信息，参加 PCTP 培训认证，请访问 PHPChina 培训频道：

http://www.phpchina.com/action_train.html



PHPChina | PHP中国开源之路

PHPer

增刊

PHPChina.com 电子期刊

