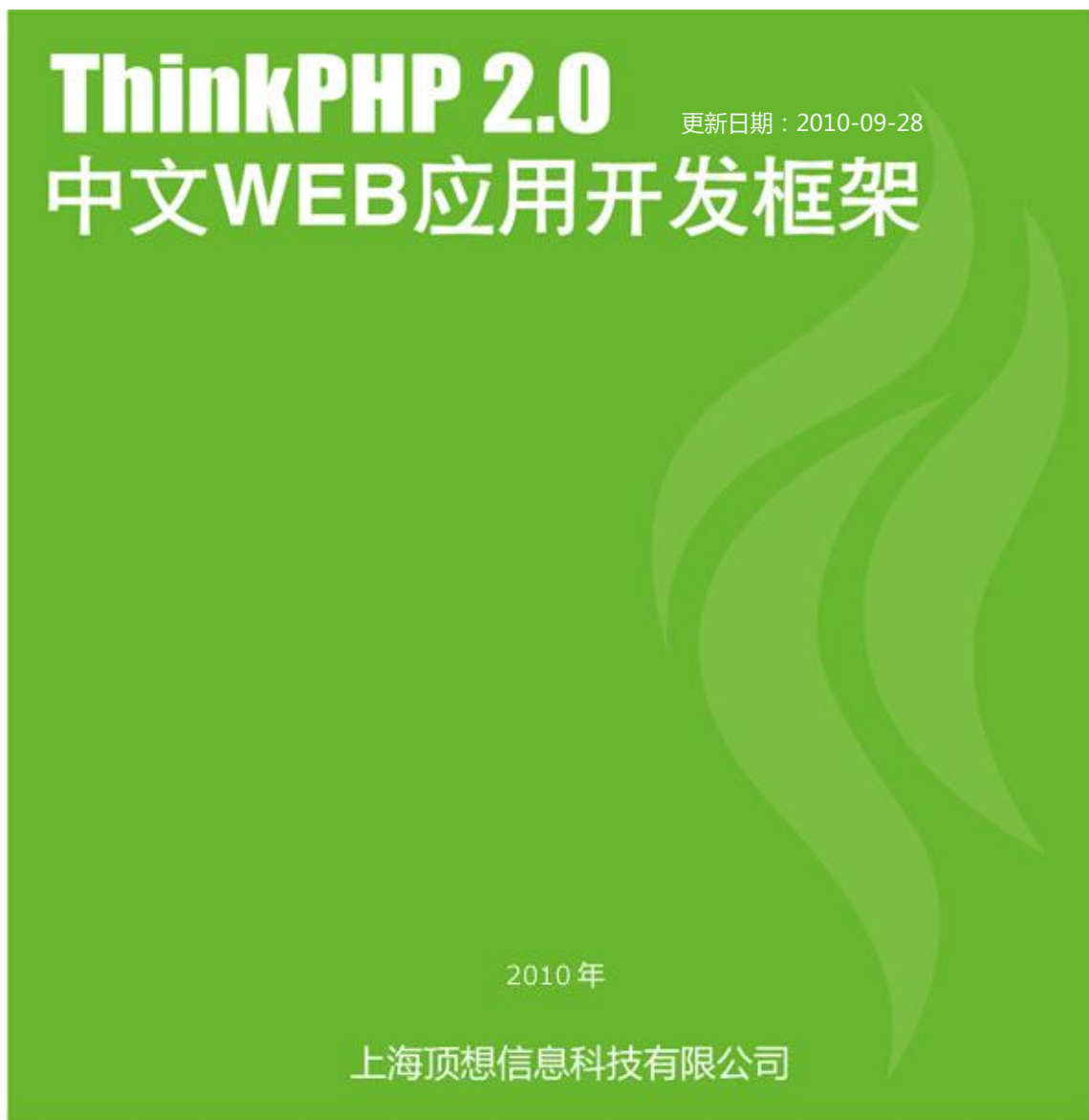




# 常见问题



## 目录

1 . 什么是 MVC ?	4
2 . 如何快速架构项目 ?	5
3 . M 和 D 有什么区别 ?	5
4 . 程序修改了 , 怎么运行的时候没有任何变化 ?	6
5 . 为何新增字段插入数据数据库中显示为空 ?	6
6 . 如何开启调试模式 ?	6
7 . 如何导入自己的函数库 ?	6
8 . 如何更新同字段名的多条记录 ?	6
9 . 为何 RBAC 改了路径就没有权限了 ?	6
10. 为什么 <code>\$this-&gt;error()</code> 和 <code>\$this-&gt;success()</code> 跳转同一个模板文件 ?	7
11. 如何获得上一步插入记录 id ?	7
12. 使用 <code>create()</code> 方法如何隐藏数据库中的字段名 ?	7
13. 模型不需要数据库怎么定义 ?	8
14. 不创建模型类如何自动验证 ?	8
15. 空间不支持 <code>PATHINFO</code> 怎么办 ?	9
16. 如何设置类库的自动加载 ?	11
17. 如何设置字段别名 ?	12
18. 如何去掉 URL 中的入口文件名 ?	12

19.如何减少 URL 目录？	13
20.数据库中表名无规则模型里面怎么使用？	13
21.如何获取当前 Action 和 Model 的名称？	14
22.如何获取表中单个字段值？	14
23.如何避免 JS 代码被模板解析？	15
24.如何添加目录安全文件？	16
25.如何增加模板替换字符串？	17

## 1. 什么是 MVC ?

MVC 是一个设计模式，它强制性的使应用程序的输入、处理和输出分开。使用 MVC 应用程序被分成三个核心部件：模型（M）、视图（V）、控制器（C），它们各自处理自己的任务。

**视图**：视图是用户看到并与之交互的界面。对老式的 Web 应用程序来说，视图就是由 HTML 元素组成的界面，在新式的 Web 应用程序中，HTML 依旧在视图中扮演着重要的角色，但一些新的技术已层出不穷，它们包括 Adobe Flash 和象 XHTML，XML/XSL，WML 等一些标识语言和 Web services。如何处理应用程序的界面变得越来越有挑战性。MVC 一个大的好处是它能为你的应用程序处理很多不同的视图。在视图中其实没有真正的处理发生，不管这些数据是联机存储的还是一个雇员列表，作为视图来讲，它只是作为一种输出数据并允许用户操纵的方式。

**模型**：模型表示企业数据和业务规则。在 MVC 的三个部件中，模型拥有最多的处理任务。例如它可能用象 EJBs 和 ColdFusion Components 这样的构件对象来处理数据库。被模型返回的数据是中立的，就是说模型与数据格式无关，这样一个模型能为多个视图提供数据。由于应用于模型的代码只需写一次就可以被多个视图重用，所以减少了代码的重复性。

**控制器**：控制器接受用户的输入并调用模型和视图去完成用户的需求。所以当单击 Web 页面中的超链接和发送 HTML 表单时，控制器本身不输出任何东西和做任何处理。它只是接收请求并决定调用哪个模型构件去处理请求，然后确定用哪个视图来显示模型处理返回的数据。

现在我们总结 MVC 的处理过程，首先控制器接收用户的请求，并决定应该调用哪个模型来进行处理，然后模型用业务逻辑来处理用户的请求并返回数据，最后控制器用相应的视图格式化模型返回的数据，并通过表示层呈现给用户。

## 2 . 如何快速架构项目？

首先建立一个文件夹假设取名为 MyTp,然后将下载的 ThinkPHP 放在 MyTp 里面,接着在 MyTp 里面建立一个 index.php(不一定非要是 index.php 也可以是别的),文件写入如下代码

```
define('THINK_PATH','./ThinkPHP/'); //框架路径

define('APP_NAME','MyApp'); // 项目名字

define('APP_PATH','./ MyApp'); //项目路径

require(THINK_PATH."/ThinkPHP.php"); //加载框架

App::run();
```

其中 “.” 代表 index.php 文件所在的路径，然后浏览器输入

**http://127.0.0.1/MyTp/index.php**

运行后刷新 MyTp 文件夹将会发现系统自动创建了 MyApp 文件夹，这只是一个例子，大家可以根据自己的需要对路径设置稍作修改就可以了。

## 3 . M 和 D 有什么区别？

简单说使用 M Model 里面对应文件可有可无，使用 D 必需有，另外在 Mode 对应文件里面有自动验证或者函数之类的业务逻辑也必需用 D.

再打个比方说 M 是刚安装好的操作系统，只有系统自带的应用 还没有自己安装的应用（所以只能调用内置的 Model 提供的属性和方法）D 是已经安装了很多的第三方的应用程序 使用起来更丰富一些（可以调用模型类自己定义的属性和方法）但是很明显，安装了很多的第三方应用后系统性能降低了,执行变慢了,但是功能显然强大了.

## 4 . 程序修改了，怎么运行的时候没有任何变化？

注意删除 Runtime 下面的缓存文件

## 5 . 为何新增字段插入数据数据库中显示为空？

删除 Runtime/data/\_fields/ 里面对应的文件

## 6 . 如何开启调试模式？

在配置里面添加

```
'APP_DEBUG' => true
```

## 7 . 如何导入自己的函数库？

可以放在 common/common.php 系统会自动加载该函数

## 8 . 如何更新同字段名的多条记录？

用 \$\_post[ '字段名' ] 将得到一个数组，然后循环更新，参照代码

```
$M = M("Config");  
for($i = 0;$i < count($_POST["id"]); $i++) {  
    $data["id"] = $_POST["id"][$i];  
    $data["body"] = $_POST["body"][$i];  
    $M->save($data);  
}
```

## 9 . 为何 RBAC 改了路径就没有权限了？

**RBAC::AccessDecision()** 这个方法是权限判断的，默认是读取当前项目名称，项目名称又是你自己初始开设的项目目录，当提交 RBAC 时候，在数据表中有录入的初始的项目名称，现在又改项目目录名称了，查询对比不符，所以无权限。

参照代码：**RBAC::AccessDecision('现在的项目名称');** 或者在数据表 Node 里找到你之前的项目目录名称，改成现在的项目名称

## 10.为什么\$this->error() 和\$this->success()跳转同一个模板文件？

Thinkphp 的默认配置错误和成功是一个模板，可以在配置里面添加

```
'TMPL_ACTION_ERROR' => 'Public:error' // 默认错误跳转对应的模板文件
'TMPL_ACTION_SUCCESS' => 'Public:success' //默认成功跳转对应的模板文件
```

## 11.如何获得上一步插入记录 id?

Thinkphp 的 add()方法返回值就是上一步插入数据的 id

## 12.使用 create()方法如何隐藏数据库中的字段名？

Thinkphp 的字段映射功能可以让你在表单中隐藏真正的数据表字段，而不用担心放弃 TP 的自动创建表单对象的功能，假设我们的 User 表里面有 username 和 email 字段，我们需要映射成另外的字段，定义方式如下：

```
class UserModel extends Model{
    protected $_map = array(
        'name' => 'username',
        'mail' => 'email',
    );
}
```

这样，在表单里面就可以直接使用 name 和 mail 名称作为表单数据提交了。在保存的时候会字段转换成定义的字段映射。

## 13.模型不需要数据库怎么定义？

TP2.0 开始，只要定义了模型就会连接数据库，1.5 版本有过虚拟模型的概念已经不存在了，那么，如果我们某些模型根本没有数据库操作，但是又想把一些业务逻辑封装到 model 类里面 怎么办呢？其实，很简单，只要定义的 model 类不继承 Model 类即可，呵呵~例如：

```
class UserModel extends Think{

    // 添加自己的业务逻辑

}
```

类库命名还是保持 UserModel.class.php 不变，这样可以保证自动导入和 import 机制不变，另外，由于没有继承 Model 类，很多 Model 内置的方法和属性肯定不能再使用了。由于大多数方法都是和数据库操作相关的，所以也就无所谓了。

## 14.不创建模型类如何自动验证？

我们知道，ThinkPHP 的模型有自动验证和自动完成功能，但是通常我们需要在模型类里面定义验证因子和完成因子。这样的话，我们使用 M 方法实例化模型的时候就不能使用内置的自动完成和自动验证功能了，其实仍然有办法的，因为 TP 提供了一个强大的属性动态更改的方法 setProperty。利用该方法就完全可以 M 方法实现自动验证功能了，例如：

```
$User = M('User');
$auto = array (

    array('status','1'), // 新增的时候把 status 字段设置为 1

    array('password','md5',1,'function')
```



```

// 对 password 字段在新增的时候使用 md5 函数处理
array('name','getName',1,'callback')

// 对 name 字段在新增的时候回调 getName 方法
array('create_time','time',2,'function'),

// 对 create_time 字段在更新的时候写入当前时间戳
);
$validate = array(

    array('verify','require','验证码必须!'), //默认情况下用正则进行验证

    array('repassword','password','确认密码不正确',0,'confirm' ),

    // 验证确认密码是否和密码一致

    array('password','checkPwd','密码格式不正确',0,'function' ),

    // 自定义函数验证密码格式

);
$User->setProperty('_auto',$auto);
$User->setProperty('_validate',$validate);
if($User->create()){
    $User->add();
}else{
    $this->error($User->getError());
}

```

完成自动验证和自动完成只是 setProperty 方法的一个小技巧而已，更强大的功能还需要你去发挥了。

## 15.空间不支持 PATHINFO 怎么办？

经常遇到的一个问题就是，在本地测试环境没有任何问题，但是部署到客户的正式环境后，发现不管输入什么 URL 地址，访问的永远都是首页（也就是默认模块的默认操作），这个时候，第一感觉就是要查

看空间是否支持 PATHINFO。由于开发工作基本完成，这个时候再去改变 URL 地址的话，模板工作量会比较大。例如，可能要把所有的类似

```
http://serverName/index.php/User/add
```

这样的 URL 地址改成

```
http://serverName/index.php?m=User&a=add
```

如果你的模板里面没有使用 U 方法统一生成 URL 的话，这个工作量随着模板文件的多少会有成倍的增长。

在这样的情况下面，最安全的方式，其实是调整 URL 模式，只需要做两步操作即可：

- 1、在项目配置文件里面设置：**'URL\_MODEL'=>3**
- 2、清空模板缓存目录（通常默认是项目的 **Runtime/Cache/**）

这样设置后，系统的模板文件无需做任何更改，例如原来的模板里面的连接地址是

\_\_URL\_\_/add 或者 \_\_APP\_\_/User/add 这样的话，系统生成的链接会自动变成

http://serverName/index.php?s=/User/add 经过这样的更改之后，大部分主机环境，包括国外的主机均

可支持。如果主机空间支持 REWRITE 和 .htaccess 文件，还可以进一步处理 URL 修改你的 .htaccess 文件

为：

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ /index.php?s=$1 [QSA,PT,L]
</IfModule>
```

经过这一步的处理后，那么你的 URL 仍然可以变回：

```
http://serverName/index.php/User/add
```

或者是

```
http://serverName/User/add/
```

关于在 IIS 下面使用 Rewrite 可以参考：Apache 与 IIS,URL 重写机制的不同及 REQUEST\_URI

## 16.如何设置类库的自动加载？

自动加载类库，是指在无需通过 require 和 TP 内置的 import 方法加载类库文件即可在需要的时候自动加载，自动加载机制可以让代码更简洁，并且利用得当的话，效率反而比手动加载有提升。

自动加载机制有四个方式：

- 1、系统的列入核心编译类别的类库都无需加载即可使用
- 2、定义了别名的类库会自动加载
- 3、当前项目的模型和 Action 类都会自动加载
- 4、自动搜索路径下面的类库可以自动加载

自动加载的类库文件命名必须是以 class.php 为后缀的。

从加载的效率上来看，依次递增，由此可见，给项目的类库定义别名是一个高效的方法，为项目中经常使用的类库定义别名是一个好的习惯，而且定义别名的副作用就是可以改变系统内置的自动定位机制。

添加别名定义使用 alias\_import 方法

```
alias_import(array(  
    'myClass' => LIB_ATH.'/Common/myClass.class.php',  
    'myUtil' => LIB_ATH.'/Common/myUtil.class.php',  
  
    // ... 定义更多的别名  
  
))  
);
```

我们可以在项目的公共文件 common.php 的最后添加这段代码。然后，我们在其他地方直接使用类库，

例如：

```
$class = new myClass();
```

这个时候系统会自动根据 myClass 定义类库路径 自动加载找到 myClass 类。

如果你有很多类库 不想——定义别名的话，可以使用定义自动搜索路径的方法：

方法是定义 **APP\_AUTOLOAD\_PATH** 配置参数，该参数惯例配置的值是 'Think.Util.'，也就是说所有位于基类库 Think/Util/ 目录下面的类库都可以自动加载，但是我们还可以增加更多的搜索路径，例如：

'APP\_AUTOLOAD\_PATH'=>'Think.Util.,@.Common.',这样的话，所有位于项目应用类库 Lib/Common/ 下面的类库也会自动加载。

## 17.如何设置字段别名？

连贯操作的 field 方法可以用于设置查询的返回字段，根据数据库的查询优化建议，无论要返回多少字段，

都尽量显示指定要查询的字段名。`$User->field('id,nickname as name,status')->select();`这里把

nickname 设置成 name 别名后，查询结果里面就存在 name 字段而不存在 nickname 字段了。

利用这个技巧，我们可以实现一些实际并不存在的动态字段，例如，返回

```
$User->join('think_card card on think_user.id=card.user_id')->field('id,count(card.id) as card_count')->select();
```

## 18.如何去掉 URL 中的入口文件名？

通常的 URL 里面含有 index.php，为了达到更好的 SEO 效果可能需要去掉 URL 里面的 index.php，通过

URL 重写的方式可以达到这种效果，通常需要服务器开启 URL\_REWRITE 模块才能支持。

下面是 Apache 的配置过程，可以参考下：

- 1、httpd.conf 配置文件中加载了 mod\_rewrite.so 模块
- 2、AllowOverride None 将 None 改为 All
- 3、确保 URL\_MODEL 设置为 2

4、把.htaccess 文件放到入口文件的同级目录下，其代码如下：

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ index.php/$1 [QSA,PT,L]
</IfModule>
```

## 19.如何减少 URL 目录？

按照 TP 的默认 URL 模式，通常是：

```
http://serverName/模块名/操作名/变量 1/值 1/变量 2/值 2...
```

很多人担心这样的 URL 会导致目录层次过深，而且由于这样的 URL 改变了当前的相对路径，所以如果不注意写法，经常会导致 JS 和 CSS 加载不到。问题就在于这个"/"，这两个问题都可以通过一个小技巧解决，

而且不影响你的开发，只需要在项目配置文件中设置

```
'URL_PATHINFO_DEPR'=>'-' ,
```

这个配置默认值是"/" 我们更改为"-"

配置修改以后，上面的 URL 地址就可以变成：

```
http://serverName/模块名-操作名-变量 1-值 1-变量 2-值 2...
```

不过要注意的是，模板里面的链接地址最好是用 U 方法动态生成的，而不是固定写死的，否则模板会有一些的修改工作。

## 20.数据库中表名无规则模型里面怎么使用？

只需要在模型类里面添加

```
protected $trueTableName = '你的真实表名';
```

定义 `trueTableName` 属性后 会忽略当前的数据表前缀和 `tableName` 定义。而无论当前的模型名称是  
否和表名一致，如果你的数据表还存在跨库的情况，可以加上数据库的定义

```
protected $dbName = '数据库名称';
```

## 21.如何获取当前 Action 和 Model 的名称？

由于某些原因，我们经常会在项目中定义一个公共的 Action，例如 `CommonAction`，然后在里面添加一些公共的操作方法，在这些公共方法里面，我们常常需要获取当前的 Action 名称，我们可以调用  
Action 的 `getActionName` 方法，使用如下：

```
$name = $this->getActionName();
```

这样就能确保正确获取继承的 Action 类的 Action 名称。注意必须在 Action 类里面使用早期 1.5 版本里面  
可以使用

```
$name = $this->name;
```

来完成同样的功能，但是由于和模板赋值机制有冲突，因此 2.0 版本开始就取消了，这点需要注意。

获取当前 Model 的名称道理跟上面一样只不过获取当前 Model 用 `getModelName()`方法。

## 22.如何获取表中单个字段值？

在连贯操作中 我们可以使用 `field` 方法来定义要返回的字段

```
$list = $User->field('id,name')->select();  
dump($list);
```

会输出：

```
array(3) {  
    [0] => array(2) {  
        ["id"] => string(1) "1"  
        ["name"] => string(5) "admin"  
    }  
    [1] => array(2) {
```

```

        ["id"] => string(1) "2"
        ["name"] => string(8) "thinkphp"
    }
    [2] => array(2) {
        ["id"] => string(1) "3"
        ["name"] => string(4) "test"
    }

```

如果不想返回数据集，而只是想返回一个以 id 为索引的包含 name 的数组，那么可以简单使用

```

$data = $User->getField('id,name');
dump($data);

```

会输出：

```

array(3) {
    [1] => string(5) "admin"
    [2] => string(8) "thinkphp"
    [3] => string(4) "test"
}

```

注意 `$User->getField('name')` 和 `$User->getField('id,name')` 的返回值类型是完全不同的，前者只是返回 name 的值，并且始终只有一个。

输出的结果为：

```
string(5) "admin"
```

如果你只想取一个字段的值，但是希望返回数组的话，可以使用

```
$User->getField('id,id');
```

总结：模型类的 `getField` 方法是一个双关方法

## 23. 如何避免 JS 代码被模板解析？

如果使用 TP 内置的模板引擎，而且采用默认的标志设置的话，在某些情况下，如果不注意，

`{${'name'}.value}` 这样的标签很容易被内置模板引擎误解析。解决这样的问题有三个方法，现列举如下：

1、`{${'name'}.value}` 改成 `{ ${'name'}.value }`

因为内置模板引擎的解析规则是"{"后面紧跟"\$"符号才会解析变量 因此只要在 "{" 和 "\$"之间添加空格就不会被误解析了

## 2、使用内置的 literal 标签包含 JS 代码

<literal>JS 代码</literal> 包含在 literal 标签中的代码将会直接输出，不进行任何解析

## 3、定制模板引擎标签的定界符

例如：

```
'TMPL_L_DELIM'=>'<{' ,  
'TMPL_R_DELIM'=>'>}'
```

# 24.如何添加目录安全文件？

在有些服务器环境下面，是开启了 apache 的目录浏览权限的，这样就会导致用户可以通过 URL 访问到你的应用目录，查看到你有哪些模块和模板文件，显然对系统的安全性方面造成了一定的影响。

对于这样的情况，TP 提供了一个目录安全文件写入的功能，能够在项目的编译过程自动生成各个目录的安全文件，避免直接访问目录。要开启这个功能，我们只需要在项目的入口文件里面添加下面的定义：

```
define('BUILD_DIR_SECURE', true);
```

然后访问项目（必须在自动生成项目目录之前访问），这样就会自动给项目目录生成目录安全文件（默认会在相关的目录下面生成空白的 index.htm 文件），并且可以自定义安全文件的文件名

DIR\_SECURE\_FILENAME，默认是 index.html，如果你想给你们的安全文件定义为 default.html 可以使用

```
define('DIR_SECURE_FILENAME', 'default.html');
```

还可以支持多个安全文件写入，例如你想同时写入 index.html 和 default.html 两个文件，以满足不同的服务器部署环境，可以这样定义：



```
define('DIR_SECURE_FILENAME', 'index.html,default.html');
```

默认的安全文件只是写入一个空白字符串，如果需要写入其他内容，可以通过

DIR\_SECURE\_CONTENT 参数来指定，例如：

```
define('DIR_SECURE_CONTENT', 'deney Access!');
```

**注意：**

如果在后期设置，需要删除 Runtime 目录 才能重新生成目录安全文，确保相关目录的可写权限

## 25.如何增加模板替换字符串？

TP 提供了模板替换字符串功能，该功能主要可以帮助实现：

- 1、方便模板的本地单独预览；
- 2、方便在模板在不同的环境目录下面动态输出；

这个机制可以使得模板文件的定义更加方便，默认的替换规则有：

../Public：会被替换成当前项目的公共模板目录 通常是 /项目目录/Tpl/default/Public/

\_\_PUBLIC\_\_：会被替换成当前网站的公共目录 通常是 /Public/

\_\_TMPL\_\_：会替换成项目的模板目录 通常是 /项目目录/Tpl/default/

\_\_ROOT\_\_：会替换成当前网站的地址（不含域名）

\_\_APP\_\_：会替换成当前项目的 URL 地址（不含域名）

\_\_URL\_\_：会替换成当前模块的 URL 地址（不含域名）

\_\_ACTION\_\_：会替换成当前操作的 URL 地址（不含域名）

\_\_SELF\_\_：会替换成当前的页面 URL

**注意：**这些特殊的字符串是严格区别大小写的。

现在的问题是，如何定制项目需要的替换规则，比如我想更改某个替换规则并增加新的规则。其实，很简单，我们只需要在项目配置文件中配置 `TMPL_PARSE_STRING` 就可以完成。如果有相同的数组索引，就会更改系统的默认规则。例如：

```
'TMPL_PARSE_STRING' => array(  
    '__PUBLIC__' => '/Common', // 更改默认的__PUBLIC__ 替换规则  
    '__UPLOAD__' => '/Public/Uploads/', // 增加新的上传路径替换规则  
)
```

ThinkPHP 文档小组 2010-9-28

[HTTP://ThinkPHP.CN](http://ThinkPHP.CN) | WEB 应用开发最佳实践框架

大道至简，开发由我 WE CAN DO IT , JUST THINK