# 一.系统变量

{$Think.server.script\_name } //取得$\_SERVER 变量

{$Think.session.session\_id|md5 } // 获取$\_SESSION 变量

{$Think.get.pageNumber } //获取$\_GET 变量

{$Think.cookie.name } //获取$\_COOKIE 变量

# 二.系统常量

{$Think.const.\_\_FILE\_\_ }

{$Think.const.MODULE\_NAME }

# 三.特殊变量 ，由 ThinkPHP 系统定义的常量

{$Think.version } //版本

{$Think.now } //现在时间

# 四.输出 list 的第 5～15 条记录

<volist name="list" id="vo" offset="5" length='10'>

{$vo.name}

</volist>

# 五.输出偶数记录

<volist name="list" id="vo" mod="2" >

<eq name="mod" value="1">

{$vo.name}

</eq>

</volist>

# 六.Switch 标签

<switch name="name">

<case value="1">value1</case>

<case value="2">value2</case>

<default />default

</switch>

其 中 name 属性可以使用函数以及系统变量，例如：

<switch name="Think.get.userId|abs">

<case value="1">admin</case>

<default />default

</switch>

也 可以对 case 的 value 属性使用变量，例如：

<switch name="userId">

<case value="$adminId">admin</case>

<case value="$memberId">member</case>

<default />default

</switch>

# 七. 比较标签

<eq name="name" value="value">value</eq> // name 变量的值等于 value 就输出

<neq name="name" value="value">value</neq> // name 变量的值不等于 value 就输出

<gt name="name" value="5">value</gt> // name 变量的值大于 5 就输出

<egt name="name" value="5">value</egt> // name 变量的值大于等于 5 就输出

<lt name="name" value="5">value</lt> // name 变量的值小于 5 就输出

<elt name="name" value="5">value</elt> // name 变量的值小于等于 5 就输出

# 八. If标签

<if condition="$name eq 1 "> value1

<elseif condition="$name eq 2" />value2

<else /> value3

</if>

# 九.ThinkPHP系统常量

THINK\_PATH // ThinkPHP 系统目录

APP\_PATH // 当前项目目录

APP\_NAME // 当前项目名称

MODULE\_NAME //当前模块名称

ACTION\_NAME // 当前操作名称

TMPL\_PATH // 项目模版目录

LIB\_PATH // 项目类库目录

CACHE\_PATH // 项目模版缓存目录

CONFIG\_PATH //项目配置文件目录

LOG\_PATH // 项目日志文件目录

LANG\_PATH // 项目语言文件目录

TEMP\_PATH //项目临时文件目录

PLUGIN\_PATH // 项目插件文件目录

VENDOR\_PATH // 第三方类库目录

DATA\_PATH // 项目数据文件目录

IS\_APACHE // 是否属于 Apache

IS\_IIS //是否属于 IIS

IS\_WIN //是否属于Windows 环境

IS\_LINUX //是否属于 Linux 环境

IS\_FREEBSD //是否属于 FreeBsd 环境

NOW\_TIME // 当前时间戳

MEMORY\_LIMIT\_ON // 是否有内存使用限制

MEMORY\_LIMIT\_ON // 是否有内存使用限制

OUTPUT\_GZIP\_ON // 是否开启输出压缩

MAGIC\_QUOTES\_GPC // MAGIC\_QUOTES\_GPC

THINK\_VERSION //ThinkPHP 版本号

LANG\_SET // 浏览器语言

TEMPLATE\_NAME //当前模版名称

TEMPLATE\_PATH //当前模版路径

\_\_ROOT\_\_ // 网站根目录地址

\_\_APP\_\_ // 当前项目（入口文件）地址

\_\_URL\_\_ // 当前模块地址

\_\_ACTION\_\_ // 当前操作地址

\_\_SELF\_\_ // 当前 URL 地址

TMPL\_FILE\_NAME //当前操作的默认模版名（含路径）

WEB\_PUBLIC\_URL //网站公共目录

APP\_PUBLIC\_URL //项目公共模版目录

# 十. 预定义常量

WEB\_LOG\_ERROR=0 // 错误日志类型

WEB\_LOG\_DEBUG=1 // 调试日志类型

SQL\_LOG\_DEBUG=2 // SQL 日志类型

SYSTEM\_LOG=0 // 系统方式记录日志

MAIL\_LOG=1 // 邮件方式记录日志

TCP\_LOG=2 // TCP 方式记录日志

FILE\_LOG=3 // 文件方式记录日志

DATA\_TYPE\_OBJ=1 // 对象方式返回

DATA\_TYPE\_ARRAY=0 // 数组方式返回

URL\_COMMON=0 // 普通模式 URL

URL\_PATHINFO=1 // PATHINFO URL

URL\_REWRITE=2 // REWRITE URL

HAS\_ONE=1 // HAS\_ONE 关联定义

BELONGS\_TO=2 // BELONGS\_TO 关联定义

HAS\_MANY=3 // HAS\_MANY 关联定义

MANY\_TO\_MANY=4 // MANY\_TO\_MANY 关联定义

EXISTS\_VAILIDATE = 0 // 表单存在字段则验证

MUST\_VALIDATE = 1 // 必须验证

VALUE\_VAILIDATE = 2 // 表单值不为空则验证

# 十一.单个字段值的修改

有些时候，我们只需要修改某个字段的值，就可以使用setField方法，而不需要每次都调用save方法。

$Form = M("Form"); // 更改title值

$Form->where('id=5')->setField('title','ThinkPHP');

# 十二.强大的增减字段值运算

ThinkPHP中，可以对某个字段的数据直接执行增减操作。  
对于统计字段，系统还提供了更加方便的setInc和setDec方法。  
$User = M("User"); // 实例化User对象  
$User->where('id=5')->setInc('score',3); // 用户的积分加3  
$User->where('id=5')->setInc('score'); // 用户的积分加1  
$User->where('id=5')->setDec('score',5); // 用户的积分减5  
$User->where('id=5')->setDec('score'); // 用户的积分减1

# 十三.事务

$User = D(user);

$User->startTrans(); // 启动事务

$User->commit(); // 提交事务

$User->rollback(); // 事务回滚

# 十四.正则去掉图片

//过滤掉内容中的图片

foreach($lists as &$list){

//用正则匹配提取图片html标签

$pattern="/<[img|IMG].\*?src=[\'\"](.\*?(?:[\.gif|\.jpg]))[\'|\"].\*?[\/]?>/";

//替换内容设置为空

$replacement = '';

//用preg\_replace()函数将匹配到的内容替换为$replacement.并且输出

$list['content'] = preg\_replace($pattern,$replacement ,$list['content']);

}

# 十五.THINKPHP Trace调试模式配置

//放在配置文件中 默认是关闭的

'SHOW\_PAGE\_TRACE' =>true, // 显示页面Trace信息

# 十六.接口开发跨域问题解决

// 指定允许其他域名访问

header( "Access-Control-Allow-Origin:\*" );

// 响应类型

header( "Access-Control-Allow-Methods:POST,GET" );

// 响应头设置

header('Access-Control-Allow-Headers:x-requested-with,content-type');