



E-Commerce Online Shop 网上商店系统 模板制作引导手册

by Lu Hengqi
16-Nov-07

目录

关于 ECSHOP	3
产品特点:	3
1. 前言	5
1.1. 模板系统介绍.....	5
1.2. 涵盖内容	5
1.3. 适用版本	5
1.4. ECSHOP 团队主要成员	5
2. 模板结构说明	6
2.1. 模板目录结构.....	6
2.2. 链接模板文件使用例子.....	6
2.3. 模板文件结构.....	6
3. 模板系统标签说明	9
3.1. 变量调节标签.....	9
3.1.1. escape [编码].....	9
3.1.2. nl2br [换行符替换成]	9
3.1.3. default [默认值]	9
3.1.4. truncate [截取].....	10
3.1.5. strip_tags [去除html标签]	10
3.2. 控制标签	11
3.2.1. if,elseif,else.....	11
3.2.2. foreach,foreachelse	12
3.3. 显示标签	14
3.3.1. cycle	14
3.3.2. html_options	15
3.3.3. html_select_date.....	16
3.3.4. html_radios.....	19
3.3.5. html_select_time.....	21
3.4. 辅助标签	25
3.4.1. assign.....	25
3.4.2. Include	25

关于ECSHOP

ECShop是Comsenz公司推出的一款B2C独立网店系统，适合企业及个人快速构建个性化网上商店。系统是基于PHP语言及MYSQL数据库构架开发的跨平台开源程序。

ECShop悉心听取每一位商家的需求与建议，不仅设计了人性化的网店管理系统帮助商家快速上手，还根据中国人的购物习惯改进了购物流程，实现更好的用户购物体验。

经过近两年的发展，ECShop网店系统无论在产品功能、稳定性、执行效率、负载能力、安全性和SEO支持(搜索引擎优化)等方面都居国内同类产品领先地位，成为国内最流行的购物系统之一。

官方网站:

<http://www.ecshop.com>

官方论坛:

<http://bbs.ecshop.com>

产品特点:

•灵活的模版机制

ECShop开发了独有的高效模板引擎(2.15以前版本使用smarty模板引擎)，并结合了Dreamweaver的模板和库功能，使得编辑制作模板变得更简单。

•开放的插件机制

支付、配送，会员整合都是以插件形式实现。商家可以随时增加或变更的支付方式和配送体系。ECShop支持大部分php开发的论坛系统，包括discuz，phpwind等，只需在后台做简单参数配置，即可完成会员整合。

•功能 AJAX 化

ECShop使用目前流行的AJAX技术，批量数据编辑变得更迅速，方便。

•促销功能

ECShop提供了积分、红包、赠品，夺宝奇兵等多种促销方法。

•高效率的代码和执行性能

通过优化代码与数据库结构，配合ECShop独家设计的缓存机制，在不考虑网速的情况下，网店动态页面与纯静态页面访问速度相当。

•常规功能的更完善实现

针对常规功能尤其是后台管理和购物流程，ECShop进行了更简洁的设计，实现更好的用户体验。

- **搜索引擎优化**

在 SEO（搜索引擎优化）上，ECShop独家支持两种 URL 重写方式，并且是同类软件中第一家支持google / yahoo / microsoft三家共同发布的 sitemaps 0.9 网站索引规范，能够为站点被搜索引擎收录做到最大限度的支持和帮助。

- **内置手机短信网关**

ECShop内置手机短信网关，支持会员信息群发，订单提醒等功能。

- **多语言支持**

支持简体，繁体，英文。

- **源代码开放**

用户可根据自己的需求对ECShop进行定制，扩展。

1. 前言

1.1. 模板系统介绍

ECShop网上商店系统”的模板机制是由ECShop团队自行研发的模板控制系统,和著名的PHP开源模板引擎程序Smarty大部分功能类似。为了方便用户开发模板,并使用了Dreamweaver的模板和库功能,使得编辑模板变得更简单容易。

1.2. 涵盖内容

本文档主要介绍“模板结构说明”、“模板制作范例”和“模板系统标签变量”等。

1.3. 适用版本

本文档由ECShop团队编写整理;文档主要针对ECShop v2.5 + 版本。

如果您发现文档内容有误,或者内容尚未更新到最新版本,又或者您觉得需要添加需要了解的资料,请Email: luhengqi@gmail.com, 并注明哪部分内容错误或者尚未更新,谢谢!

注意:“ecshop v2.5+”表示此文档的适用环境为v2.5或更高版本,低于此版本的可能已经不适用!为了获得更多更强的功能以及更安全的使用环境,请您尽快升级您的ECShop主程序!

1.4. ECShop 团队主要成员

•项目策划

Paul Gao, Weber Liu

•程序开发

Weber Liu, ScottYe, WenjinZhang, YanXuan, PaulGao, Stevenliu, Hackfan, Edisonrao

•界面设计

Weber Liu, HengqiLu, Yecior

•文档整理

Lily Chai, Bei Wang

2. 模板结构说明

2.1. 模板目录结构

名称	备注（作用）
css	存放模板.css “样式表” 文件目录
images	存放模板“图片” 文件目录
js	存放模板.js “javascript” 文件目录
library	存放模板.lbi “库文件” 目录

注意：以上目录名全部都不可以更改，因为需要和程序做匹配，使用Unix系列操作系统的用户请注意目录名称大小写。

2.2. 链接模板文件使用例子

文件格式	使用例子
.css	<code><linkhref="style.css" rel="stylesheet" type="text/css" /></code>
备注：注意默认的style.css文件不可以更改文件名或移动到css目录，额外添加的.css文件需要存放到css目录路径，不可以使用其他目录名称，留意红色代码部分的使用。 例子： <code><linkhref="css/style.css" rel="stylesheet" type="text/css" /></code>	
.js	<code><script type="text/javascript" src="js/ajax.js"></script></code>
备注：.js文件必须存放在js目录，不可以使用其他目录名称，注意红色代码部分。	
.gif/.jpg/.png	.dwt文件代码下面路径使用例子： <code><imgsrc="images/a.gif" alt="" /></code> .lbi文件代码下面路径使用例子： <code><imgsrc="../images/a.gif" alt="" /></code>
.lbi	<code><!-- #BeginLibraryItem "/library/page_header.lbi" --><!-- #EndLibraryItem --></code>

注意：链接代码的href、rel、type的次序需要遵行上面例子，否则程序可能无法使用。

2.3. 模板文件结构

一套完整模板应该包含以下文件：

- 29个 .dwt文件（模板框架文件）
- 50个 .lbi文件（模板库文件）
- 1个style.css文件（里面包含了模板的配置代码。）
- 1个screenshot.png文件（用于后台管理显示安装模板。）

名称	文件类型	说明
activity.dwt	模板文件	活动列表页
article_cat.dwt	模板文件	文章列表页
article_pro.dwt	模板文件	系统文章页（如：“公司简介”、“版权信息”。）
article.dwt	模板文件	文章内容页
auction_list.dwt	模板文件	拍卖商品列表页

auction.dwt	模板文件	拍卖商品详情页
brand.dwt	模板文件	品牌商品列表页
catalog.dwt	模板文件	所有分类、品牌页
category.dwt	模板文件	商品列表页
compare.dwt	模板文件	商品比较页
flow.dwt	模板文件	购物车和购物流程页
gallery.dwt	模板文件	商品相册页
goods.dwt	模板文件	商品详情页
group_buy_goods.dwt	模板文件	团购商品详情页
group_buy_list.dwt	模板文件	团购商品列表页
index.dwt	模板文件	首页
message.dwt	模板文件	信息提示页（如：警告信息，登陆成功。）
myship.dwt	模板文件	配送方式查询页
pick_out.dwt	模板文件	选购中心页
receive.dwt	模板文件	收货确认信息页
respond.dwt	模板文件	在线支付结果提示信息页
search.dwt	模板文件	商品高级搜索、搜索结果页
snatch.dwt	模板文件	夺宝奇兵页
tag_cloud.dwt	模板文件	标签云页
topic.dwt	模板文件	专题活动页
user_clips.dwt	模板文件	用户中心页（包含：欢迎页、我的留言、我的评论、我的标签、收藏商品、缺货登记、添加缺货登记、我的推荐、单个商品推荐。）
user_passport.dwt	模板文件	用户安全页（包含：会员登录、会员注册、找回密码。）
user_transaction.dwt	模板文件	用户中心页（包含：个人资料、我的红包、添加红包、我的订单、订单详情、合并订单、订单状态、商品列表、费用总计、收货人信息、支付方式、其他信息、会员余额、积分兑换。）
wholesale_list.dwt	模板文件	批发商品页
ad_position.lbi	库文件	广告位
article_category_tree.lbi	库文件	文章列表页“文章分类树”
auction.lbi	库文件	首页拍卖商品
brand_goods.lbi	库文件	首页品牌商品
brands.lbi	库文件	品牌列表和商品列表页“品牌筛选”
cart.lbi	库文件	购物车
cat_articles.lbi	库文件	首页分类文章
cat_goods.lbi	库文件	首页分类商品
category_tree.lbi	库文件	商品分类树
categorys.lbi	库文件	选购中心“分类列表”
comments_list.lbi	库文件	用户评论表单、内容列表
comments.lbi	库文件	用户评论（ajax载入comments_list.lbi库文件。）
consignee.lbi	库文件	收货地址表单
email_list.lbi	库文件	邮件订阅
filter_attr.lbi	库文件	商品列表页“属性筛选”
goods_article.lbi	库文件	相关文章
goods_attrlinked.lbi	库文件	属性关联的商品
goods_fittings.lbi	库文件	相关配件

goods_gallery.lbi	库文件	商品相册
goods_list.lbi	库文件	商品列表
goods_tags.lbi	库文件	商品标记
goods_relative.lbi	库文件	相关商品
group_buy.lbi	库文件	首页团购商品
help.lbi	库文件	网店帮助
history.lbi	库文件	商品浏览历史
invoice_query.lbi	库文件	首页发货单列表
member_info.lbi	库文件	会员登录表单和登录成功以后的会员信息
member.lbi	库文件	会员登录 (ajax载入member_info.lbi库文件。)
myship.lbi	库文件	配送查询
new_articles.lbi	库文件	最新文章、网店公告
order_query.lbi	库文件	订单查询
order_total.lbi	库文件	订单费用总计
page_footer.lbi	库文件	页面脚部 (包含: 版权信息、联系电话、地址、IM软件和Ajax需要调用的javascript代码。)
page_header.lbi	库文件	页面头部 (包含: 网店LOGO、顶部菜单、主导航)
pages.lbi	库文件	列表分页
price_grade.lbi	库文件	商品列表页“价格筛选”
recommend_best.lbi	库文件	精品推荐
recommend_hot.lbi	库文件	热卖商品
recommend_new.lbi	库文件	新品推荐
recommend_promotion	库文件	促销商品
search_form.lbi	库文件	搜索表单
snatch_price.lbi	库文件	夺宝奇兵最新出价列表 (必须被id="ECS_PRICE_LIST"包含实现ajax刷新。)
snatch.lbi	库文件	夺宝奇兵出价表单 (必须被id="ECS_SNATCH"包含实现ajax刷新。)
top10.lbi	库文件	销售排行
ur_here.lbi	库文件	当前位置
user_menu.lbi	库文件	用户中心菜单
vote_list.lbi	库文件	在线调查
vote.lbi	库文件	在线调查内容
screenshot.png	图片文件	用于后台管理显示安装模板,必须使用.png格式。
style.css	样式表文件	模板默认样式表文件, 里面包含了模板的配置代码。

3. 模板系统标签说明

3.1. 变量调节标签

3.1.1. escape [编码]

参数位置	参数类型	必需	可用参数	默认	描述
1	string	No	html,url,quotes	html	使用何种编码格式。

描述:

用于html转码,url转码,在没有转码的变量上转换单引号,十六进制转码,十六进制美化转码。默认是html转码。

例子:

```
{ $articleTitle }  
{ $articleTitle | escape }  
{ $articleTitle | escape:"html" }    { * escapes & " ' < > * }  
{ $articleTitle | escape:"url" }  
{ $articleTitle | escape:"quotes" }
```

3.1.2. nl2br [换行符替换成
]

描述:

所有的换行符将被替换成
.功能同PHP中的nl2br()函数一样.

例子:

```
{ $articleTitle | nl2br }
```

输出:

Sun or rain expected
today, dark tonight

3.1.3. default [默认值]

参数位置	参数类型	必需	默认	描述
1	string	No	empty	这是变量为空的时候的默认输出。

描述:

为空变量设置一个默认值。

当变量为空或者未分配的时候,将由给定的默认值替代输出。

例子:

```
{ $articleTitle | default:"no title" }  
{ $myTitle | default:"no title" }
```

输出:

Dealers Will Hear Car Talk at Noon.

no title

3.1.4. truncate [截取]

参数位置	参数类型	是否必须	默认	描述
1	integer	No	80	截取字符的数量
2	string	No	...	截取后追加在截取词后面的字符串
3	boolean	No	false	是截取到词的边界（假）还是精确到字符（真）

描述:

从字符串开始处截取某长度的字符.默认是80个.

你也可以指定第二个参数作为追加在截取字符串后面的文本字符串.该追加字符串被计算在截取长度中。

默认情况下,smarty会截取到一个词的末尾。

如果你想要精确的截取多少个字符,把第三个参数改为"true"

例子:

```
{ $articleTitle }
{ $articleTitle | truncate }
{ $articleTitle | truncate:30 }
{ $articleTitle | truncate:30:"" }
{ $articleTitle | truncate:30:"---" }
{ $articleTitle | truncate:30:"":true }
{ $articleTitle | truncate:30:"...":true }
```

输出:

Two Sisters Reunite after Eighteen Years at Checkout Counter.

Two Sisters Reunite after Eighteen Years at Checkout Counter.

Two Sisters Reunite after...

Two Sisters Reunite after

Two Sisters Reunite after---

Two Sisters Reunite after Eigh

Two Sisters Reunite after E...

3.1.5. strip_tags [去除html标签]

描述:

去除<和>标签,包括在<和>之间的任何内容.

例子:

```
{ $articleTitle }
{ $articleTitle | strip_tags }
```

输出:

Blind Woman Gets New Kidney from Dad she Hasn't Seen in years.

Blind Woman Gets New Kidney from Dad she Hasn't Seen in years.

3.2. 控制标签

3.2.1. if,elseif,else

描述:

Smarty 中的 if 语句和 php 中的 if 语句一样灵活易用, 并增加了几个特性以适宜模板引擎. if 必须于 /if 成对出现. 可以使用 else 和 elseif 子句. 可以使用以下条件修饰词: eq、ne、neq、gt、lt、lte、le、gte、ge、is even、is odd、is not even、is not odd、not、mod、div by、even by、odd by、==、!=、>、<、<=、>=. 使用这些修饰词时必须和变量或常量用空格隔开.

例子:

```
{if $name eq "Fred"}
    Welcome Sir.
{elseif $name eq "Wilma"}
    Welcome Ma'am.
{else}
    Welcome, whatever you are.
{/if}
```

```
{* an example with "or" logic *}
{if $name eq "Fred" or $name eq "Wilma"}
    ...
{/if}
```

```
{* same as above *}
{if $name == "Fred" || $name == "Wilma"}
    ...
{/if}
```

```
{* the following syntax will NOT work, conditional qualifiers
must be separated from surrounding elements by spaces *}
{if $name=="Fred" || $name=="Wilma"}
    ...
{/if}
```

```
{* parenthesis are allowed *}
{if ( $amount < 0 or $amount > 1000 ) and $volume >= #minVolAmt#}
    ...
{/if}
```

```
{* you can also embed php function calls *}
{if count($var) gt 0}
    ...
{/if}
```

```
{* test if values are even or odd *}
{if $var is even}
    ...
{/if}
```

```
{if $var is odd}
...
{/if}
{if $var is not odd}
...
{/if}

{* test if var is divisible by 4 *}
{if $var is div by 4}
...
{/if}

{* test if var is even, grouped by two. i.e.,
0=even, 1=even, 2=odd, 3=odd, 4=even, 5=even, etc. *}
{if $var is even by 2}
...
{/if}

{* 0=even, 1=even, 2=even, 3=odd, 4=odd, 5=odd, etc. *}
{if $var is even by 3}
...
{/if}
```

3.2.2. foreach,foreachelse

iteration:

iteration 用于显示当前循环的执行次数[待考]

iteration 总是从 1 开始，每执行一次增加 1.[待考]

first:

当前 foreach 循环第一次执行时 first 被设置成 true.

last:

当前 foreach 循环执行到最后一遍时 last 被设置成 true.

show:

show 是 foreach 的一个参数. 取值为布尔值 true 或 false. 如果指定为 false 该循环不显示, 如果循环指定了 foreachelse 子句, 该子句显示与否也取决于 show 的取值.

total:

total 用于显示循环执行的次数, 可以在循环中或循环执行后调用.

属性	类型	是否必须	缺省值	描述
from	string	Yes	n/a	待循环数组的名称
item	string	Yes	n/a	当前处理元素的变量名称
key	string	No	n/a	当前处理元素的键名
name	string	No	n/a	该循环的名称, 用于访问该循环

描述:

foreach 是除 section 之外处理循环的另一种方案(根据不同需要选择不同的方案).

foreach 用于处理简单数组(数组中的元素的类型一致), 它的格式比 section 简单许多, 缺点是只能处理简单数组.

foreach 必须和 /foreach 成对使用, 且必须指定 from 和 item 属性.

name 属性可以任意指定(字母、数字和下划线的组合).

foreach 可以嵌套, 但必须保证嵌套中的 foreach 名称唯一.

from 属性(通常是数组)决定循环的次数.

foreachelse 语句在 from 变量没有值的时候被执行.

例子1:

```
{* 该例将输出数组 $custid 中的所有元素的值 *
```

```
{foreach from=$custid item=curr_id}
```

```
    id: {$curr_id}<br>
```

```
{/foreach}
```

输出:

```
id: 1000<br>
```

```
id: 1001<br>
```

```
id: 1002<br>
```

例子2:

```
{* The key contains the key for each looped value
```

assignment looks like this:

```
$smarty->assign("contacts", array(array("phone" => "1", "fax" => "2", "cell" => "3"),
```

```
array("phone" => "555-4444", "fax" => "555-3333", "cell" => "760-1234")));
```

```
*)
```

```
{* 键就是数组的下标, 请参看关于数组的解释 *
```

```
{foreach name=outer item=contact from=$contacts}
```

```
    {foreach key=key item=item from=$contact}
```

```
        {$key}: {$item}<br>
```

```
    {/foreach}
```

```
{/foreach}
```

输出:

```
phone: 1<br>
```

```
fax: 2<br>
```

```
cell: 3<br>
```

```
phone: 555-4444<br>
```

```
fax: 555-3333<br>
```

```
cell: 760-1234<br>
```

foreach 循环有自己的变量名, 使用该变量名可以访问该循环. 使用方法为

{`$smarty.foreach.foreachname.varname`}, 其中 foreachname 即在 foreach 中指定的 name 属性.

3.3. 显示标签

3.3.1. cycle

属性	类型	是否必须	缺省值	描述
name	string	No	default	轮转的名称
values	mixed	Yes	N/A	待轮转的值, 可以用逗号分隔的列表(请查看 delimiter 属性)或一个包含多值的数组.
print	boolean	No	true	是否输出值
advance	boolean	No	true	是否使用下一个值(为 false 时使用当前值)
delimiter	string	No	,	指出 values 属性中使用的分隔符, 默认是逗号.
assign	string	No	n/a	输出值将被赋给模板变量的名称

描述:

Cycle 用于轮转使用一组值. 该特性使得在表格中交替输出颜色或轮转使用数组中的值变得很容易.

如果需要在模板中使用多个轮转, 需要给出唯一的 name 属性.

用户可以设置 print 属性为 false 强制不输出当前值. 该特性可以很方便地略过某个值.

advance 属性用于重复使用某个值. 当该属性设置为 false 时, 下次调用该轮转时将输出同样的值.

如果指定了 "assign" 这个特殊属性, 该轮转的输出值将被赋给由 assign 指定的模板变量, 而不是直接输出.

例子:

```
{section name=rows loop=$data}
<tr bgcolor="{cycle values="#eeeeee,#d0d0d0"}">
<td>{$data[rows]}</td>
</tr>
{/section}
```

输出:

```
<tr bgcolor="#eeeeee">
<td>1</td>
</tr>
<tr bgcolor="#d0d0d0">
<td>2</td>
</tr>
<tr bgcolor="#eeeeee">
<td>3</td>
</tr>
```

3.3.2. html_options

属性	类型	是否必须	缺省值	描述
values	array	Yes, unless using options attribute	n/a	包含下拉列表各元素值的数组
output	array	Yes, unless using options attribute	n/a	包含下拉列表各元素显示值的数组
selected	string/array	No	empty	已选定的元素或元素数组
options	associative array	Yes, unless using values and output	n/a	包含值和显示的关联数组
name	string	No	empty	下拉菜单的名称

描述:

自定义函数 `html_options` 根据给定的数据创建选项组。该函数可以指定哪些元素被选定。要么必须指定 `values` 和 `output` 属性，要么指定 `options` 替代。

如果给定值是数组，将作为 OPTGROUP 处理，且支持递归。所有的输出与 XHTML 兼容。

如果指定了可选属性 `name`，该选项列表将被置于 `<select name="groupname"></select>` 标签对中。如果没有指定，那么只产生选项列表。

上表未提到的其它参数在 `<select>` 标签中以 "名称/属性" 对的方式显示。如果没有指定可选属性 `name` 这些参数将被忽略。

例子:

`index.php:`

```
require('Smarty.class.php');
$smarty = new Smarty;
$smarty->assign('cust_ids', array(1000,1001,1002,1003));
$smarty->assign('cust_names', array('Joe Schmoe','Jack Smith','Jane Johnson','Carlie Brown'));
$smarty->assign('customer_id', 1001);
$smarty->display('index.tpl');
```

`index.tpl:`

```
<select name=customer_id>
    {html_options values=$cust_ids selected=$customer_id output=$cust_names}
</select>
```

`index.php:`

```
require('Smarty.class.php');
$smarty = new Smarty;
$smarty->assign('cust_options', array(
    1001 => 'Joe Schmoe',
    1002 => 'Jack Smith',
    1003 => 'Jane Johnson',
    1004 => 'Charlie Brown'));
$smarty->assign('customer_id', 1001);
$smarty->display('index.tpl');
```

index.tpl:

```
<select name=customer_id>
    {html_options options=$cust_options selected=$customer_id}
</select>
```

输出:

```
<select name=customer_id>
    <option value="1000">Joe Schmoe</option>
    <option value="1001" selected="selected">Jack Smith</option>
    <option value="1002">Jane Johnson</option>
    <option value="1003">Charlie Brown</option>
</select>
```

3.3.3. html_select_date

属性	类型	是否必须	缺省值	描述
prefix	string	No	Date_	变量名称前缀
time	timestamp/ YYYY-MM-DD	No	UNIX时间戳 或年-月-日	使用时间类型 (data/time)
start_year	string	No	年份或与当前 年份的相对值	下拉列表中第一个年份, 或 与当前年份的相对值 (正/负 几年)
end_year	string	No	同start_year	下拉列表中最后一个年份, 或与当前年份的相对值 (正/ 负 几年)
display_days	boolean	No	true	是否显示天
display_months	boolean	No	true	是否显示月
display_years	boolean	No	true	是否显示年
month_format	string	No	%B	月份的表示方法 (strftime)
day_format	string	No	%02d	天显示的格式 (sprintf)
day_value_format	string	No	%d	天的表示方法 (sprintf)
year_as_text	boolean	No	false	是否以文本方式显示年份
reverse_years	boolean	No	false	逆序显示年份
field_array	string	No	null	如果指定了名称, 选定的区 域将以[Day],[Year],[Month] 的形式返回给PHP(待考)
day_size	string	No	null	如果给定, 为标签添加大小 属性
month_size	string	No	null	如果给定, 为标签添加大小 属性
year_size	string	No	null	如果给定, 为标签添加大小 属性
all_extra	string	No	null	如果给定, 为所有标签添加 附加属性
day_extra	string	No	null	如果给定, 为标签添加附加 属性
month_extra	string	No	null	如果给定, 为标签添加附加 属性

year_extra	string	No	null	如果给定, 为标签添加附加属性
field_order	string	No	MDY	显示区域的顺序
field_separator	string	No	\n	各区域间输出的分隔字符串
month_value_format	string	No	%m	月份值的strftime表示方法, 默认为 %m

描述:

自定义函数 `html_select_date` 用于创建日期下拉菜单. 它可以显示任意年月日.

例子:

`{html_select_date}`

输出:

```
<select name="Date_Month">
<option value="1">January</option>
<option value="2">February</option>
<option value="3">March</option>
<option value="4">April</option>
<option value="5">May</option>
<option value="6">June</option>
<option value="7">July</option>
<option value="8">August</option>
<option value="9">September</option>
<option value="10">October</option>
<option value="11">November</option>
<option value="12" selected>December</option>
</select>
<select name="Date_Day">
<option value="1">01</option>
<option value="2">02</option>
<option value="3">03</option>
<option value="4">04</option>
<option value="5">05</option>
<option value="6">06</option>
<option value="7">07</option>
<option value="8">08</option>
<option value="9">09</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
<option value="13" selected>13</option>
<option value="14">14</option>
<option value="15">15</option>
<option value="16">16</option>
<option value="17">17</option>
<option value="18">18</option>
<option value="19">19</option>
<option value="20">20</option>
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
```

```
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
</select>
<select name="Date_Year">
<option value="2001" selected>2001</option>
</select>
```

例子:

```
{* start and end year can be relative to current year *}
{html_select_date prefix="StartDate" time=$time start_year="-5" end_year="+1"
display_days=false}
```

输出:

```
<select name="StartDateMonth">
<option value="1">January</option>
<option value="2">February</option>
<option value="3">March</option>
<option value="4">April</option>
<option value="5">May</option>
<option value="6">June</option>
<option value="7">July</option>
<option value="8">August</option>
<option value="9">September</option>
<option value="10">October</option>
<option value="11">November</option>
<option value="12" selected>December</option>
</select>
<select name="StartDateYear">
<option value="1999">1995</option>
<option value="1999">1996</option>
<option value="1999">1997</option>
<option value="1999">1998</option>
<option value="1999">1999</option>
<option value="2000" selected>2000</option>
<option value="2001">2001</option>
</select>
```

3.3.4. html_radios

属性	类型	是否必须	缺省值	描述
name	string	No	radio	单选按钮列表的名称
values	array	Yes, 或指定 options 属性	n/a	包含单选按钮值的数组
output	array	Yes, 或指定 options 属性	n/a	包含单选按钮显示值的数组
checked	string	No	empty	已选定的元素
options	associative array	Yes, 或指定 values 属性	n/a	包含值和显示的关联数组
separator	string	No	empty	分隔每个单选按钮的字符串

描述:

自定义函数 `html_radios` 根据给定的数据创建单选按钮组。该函数可以指定哪个元素被选定。要么必须指定 `values` 和 `output` 属性，要么指定 `options` 替代。所有的输出与 XHTML 兼容。上表未提到的其它参数在 `<input>` 标签中以"名称/属性"对的方式显示。

例子:

`index.php:`

```
require('Smarty.class.php');
$smarty = new Smarty;
$smarty->assign('cust_ids', array(1000,1001,1002,1003));
$smarty->assign('cust_names', array('Joe Schmoe','Jack Smith','Jane Johnson','Charlie Brown'));
$smarty->assign('customer_id', 1001);
$smarty->display('index.tpl');
```

`index.tpl:`

```
{html_radios values=$cust_ids checked=$customer_id output=$cust_names
separator="<br />"}
```

`index.php:`

```
require('Smarty.class.php');
$smarty = new Smarty;
$smarty->assign('cust_radios', array(
    1001 => 'Joe Schmoe',
    1002 => 'Jack Smith',
    1003 => 'Jane Johnson',
    1004 => 'Charlie Brown'));
$smarty->assign('customer_id', 1001);
$smarty->display('index.tpl');
```

`index.tpl:`

```
{html_radios name="id" options=$cust_radios checked=$customer_id separator="<br />"}
```

输出:

```
<input type="radio" name="id[]" value="1000">Joe Schmoe<br />
<input type="radio" name="id[]" value="1001" checked="checked"><br />
<input type="radio" name="id[]" value="1002">Jane Johnson<br />
<input type="radio" name="id[]" value="1003">Charlie Brown<br />
```

3.3.5. html_select_time

属性	类型	是否必须	缺省值	描述
prefix	string	No	Time_	变量名称前缀
time	timestamp	No	UNIX时间戳 或年-月-日	使用时间类型 (data/time)
display_hours	boolean	No	true	是否显示小时
display_minutes	boolean	No	true	是否显示分钟
display_seconds	boolean	No	true	是否显示秒
display_meridian	boolean	No	true	是否显示正午界 (上午/下午)
use_24_hours	boolean	No	true	是否使用24小时制
minute_interval	integer	No	1	分钟下拉列表的间隔
second_interval	integer	No	1	秒钟下拉列表的间隔
field_array	string	No	n/a	输出值到该值指定的数组
all_extra	string	No	null	如果给定, 为标签添加附加属性
hour_extra	string	No	null	如果给定, 为标签添加附加属性
minute_extra	string	No	null	如果给定, 为标签添加附加属性
second_extra	string	No	null	如果给定, 为标签添加附加属性
meridian_extra	string	No	null	如果给定, 为标签添加附加属性

描述:

自定义函数 `html_select_time` 用于创建时间下拉菜单. 它可以显示任意时分秒.

例子:

```
{html_select_time use_24_hours=true}
```

输出:

```
<select name="Time_Hour">
<option value="00">00</option>
<option value="01">01</option>
<option value="02">02</option>
<option value="03">03</option>
<option value="04">04</option>
<option value="05">05</option>
<option value="06">06</option>
<option value="07">07</option>
<option value="08">08</option>
<option value="09" selected>09</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
<option value="13">13</option>
<option value="14">14</option>
<option value="15">15</option>
<option value="16">16</option>
<option value="17">17</option>
<option value="18">18</option>
<option value="19">19</option>
<option value="20">20</option>
<option value="21">21</option>
<option value="22">22</option>
```

```

<option value="23">23</option>
</select>
<select name="Time_Minute">
<option value="00">00</option>
<option value="01">01</option>
<option value="02">02</option>
<option value="03">03</option>
<option value="04">04</option>
<option value="05">05</option>
<option value="06">06</option>
<option value="07">07</option>
<option value="08">08</option>
<option value="09">09</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
<option value="13">13</option>
<option value="14">14</option>
<option value="15">15</option>
<option value="16">16</option>
<option value="17">17</option>
<option value="18">18</option>
<option value="19">19</option>
<option value="20" selected>20</option>
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
<option value="32">32</option>
<option value="33">33</option>
<option value="34">34</option>
<option value="35">35</option>
<option value="36">36</option>
<option value="37">37</option>
<option value="38">38</option>
<option value="39">39</option>
<option value="40">40</option>
<option value="41">41</option>
<option value="42">42</option>
<option value="43">43</option>
<option value="44">44</option>
<option value="45">45</option>
<option value="46">46</option>
<option value="47">47</option>
<option value="48">48</option>

```

```

<option value="49">49</option>
<option value="50">50</option>
<option value="51">51</option>
<option value="52">52</option>
<option value="53">53</option>
<option value="54">54</option>
<option value="55">55</option>
<option value="56">56</option>
<option value="57">57</option>
<option value="58">58</option>
<option value="59">59</option>
</select>
<select name="Time_Second">
<option value="00">00</option>
<option value="01">01</option>
<option value="02">02</option>
<option value="03">03</option>
<option value="04">04</option>
<option value="05">05</option>
<option value="06">06</option>
<option value="07">07</option>
<option value="08">08</option>
<option value="09">09</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
<option value="13">13</option>
<option value="14">14</option>
<option value="15">15</option>
<option value="16">16</option>
<option value="17">17</option>
<option value="18">18</option>
<option value="19">19</option>
<option value="20">20</option>
<option value="21">21</option>
<option value="22">22</option>
<option value="23" selected>23</option>
<option value="24">24</option>
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
<option value="32">32</option>
<option value="33">33</option>
<option value="34">34</option>
<option value="35">35</option>
<option value="36">36</option>
<option value="37">37</option>
<option value="38">38</option>

```

```
<option value="39">39</option>
<option value="40">40</option>
<option value="41">41</option>
<option value="42">42</option>
<option value="43">43</option>
<option value="44">44</option>
<option value="45">45</option>
<option value="46">46</option>
<option value="47">47</option>
<option value="48">48</option>
<option value="49">49</option>
<option value="50">50</option>
<option value="51">51</option>
<option value="52">52</option>
<option value="53">53</option>
<option value="54">54</option>
<option value="55">55</option>
<option value="56">56</option>
<option value="57">57</option>
<option value="58">58</option>
<option value="59">59</option>
</select>
<select name="Time_Meridian">
<option value="am" selected>AM</option>
<option value="pm">PM</option>
</select>
```


3.4. 辅助标签

3.4.1. assign

描述:

assign 用于在模板被执行时为模板变量赋值。

属性	类型	是否必须	缺省值	描述
var	string	Yes	n/a	被赋值的变量名
value	string	Yes	n/a	赋给变量的值

描述:

assign 用于在模板被执行时为模板变量赋值。

例子:

```
{assign var="name" value="Bob"}
```

输出:

The value of \$name is Bob.

3.4.2. Include

属性	类型	是否必须	缺省值	描述
file	string	Yes	n/a	待包含的模板文件名
assign	string	No	n/a	该属性指定一个变量保存待包含模板的输出
[var ...]	[var type]	No	n/a	传递给待包含模板的本地参数，只在待包含模板中有效

描述:

Include 标签用于在当前模板中包含其它模板。当前模板中的变量在被包含的模板中可用。必须指定 file 属性，该属性指明模板资源的位置。

如果设置了 assign 属性，该属性对应的变量名用于保存待包含模板的输出，这样待包含模板的输出就不会直接显示了。

例子1:

```
{include file="header.tpl"}
{* body of template goes here *}
{include file="footer.tpl"}
```

可以在属性中传递参数给待包含模板。传递给待包含模板的参数只在待包含模板中可见。如果传递的参数在待包含模板中有同名变量，那么该变量被传递的参数替代。

例子2:

```
{include file="header.tpl" title="Main Menu" table_bgcolor="#c0c0c0"}
{* body of template goes here *}
{include file="footer.tpl" logo="http://my.domain.com/logo.gif"}
```

包含 \$template_dir 文件夹之外的模板请使用 模板资源 说明的格式。

例子3:

```
{* absolute filepath *}
{include file="/usr/local/include/templates/header.tpl"}

{* absolute filepath (same thing) *}
{include file="file:/usr/local/include/templates/header.tpl"}

{* windows absolute filepath (MUST use "file:" prefix) *}
{include file="file:C:/www/pub/templates/header.tpl"}

{* include from template resource named "db" *}
{include file="db:header.tpl"}
```