



Использование Smarty в системе шаблонов Elite-Board 2



Оглавление

[Оглавление](#)

[Предисловие](#)

[Базовый синтаксис](#)

[Типы данных](#)

[Основные переменные](#)

[Модификаторы переменных](#)

[Структура системы шаблонов](#)

[Общая информация по php-модулям](#)

[Динамические блоки](#)



Предисловие

Настоящая инструкция представляет из себя вольный пересказ основных принципов, изложенных на сайте smarty.net. Это всего лишь попытка простыми словами описать систему шаблонов Smarty с учётом специфики её работы в Elite-Board 2.

Крайне рекомендую обратиться к первоисточнику и использовать именно его в работе над сайтом:

<http://www.smarty.net/manual/ru/>



Базовый синтаксис

Базовый синтаксис определяется собственно синтаксисом системы Смарты, все типы данных и операции в которой выделяются особыми тегами:

{% - открывающий тег

%} – закрывающий тег

Использование символа «*» после открывающего и перед закрывающим тегом используется для комментариев.

Пример

```
{%if $place == 'h' or $place == 'f'%}
{%*обычное условие, почти как в PHP, дальше идёт тело
условия*%}
<!--//растяжки вверху-внизу -> //-->
<table border='0' width='100%' align='center'>
<tr><td width='100%' align='center'>{%html%}</td></tr>
</table>
<!--// <- растяжки вверху-внизу //-->
{%тело условия продолжается вплоть до
закрывающего тега if %}
{%/if%}
```

Путь к файлам шаблонов задаётся системой автоматически. Если возникла необходимость подключить дополнительный файл, используется следующая конструкция:

```
{%include file='header.html'%}
{%*подключаемый файл лежит в той же папке, что и все
остальные файлы шаблона*%}
```



Типы данных

В системе используются следующие типы данных:

- **переменные**, например

```
{% $html %}  
{% *не требуется операторов echo или print, просто  
ставим переменную в нужное место* %}
```

- **массивы**, например `{% $locale %}`

```
{% *Возможно несколько вариантов вывода элемента  
массива, в Системе. Как правило используются  
такие: * %}  
{% $locale.32 %} {% *вывод 32-го элемента языкового массива * %}  
{% $locale[32] %} {% *можно записать и так * %}  
{% $blocks.h %} {% *вывод верхнего блока, для PHP аналогично  
записи  
echo $blocks['h']; * %}
```

- **константы**, доступ к ним осуществляется из зарезервированной переменной `$smarty`, вот пример из `header_page.html`:

```
<div style='position:absolute; z-index:99; top:0px; left:500px; width:300px;  
height:100px; background-image:url({% $smarty.const.THEME %}images/region.png);  
text-align: center;'>  
{% *для вывода абсолютного пути к картинке  
использовали константу THEME, содержащую этот путь * %}  
[...]  
    <table border='0' width='100%'>  
        <td width='50%' align='left'  
class='title'>{% $smarty.const.TITLE %}</td>  
        <td width='50%' align='center'>{% $blocks.h %}</td>  
    </table>  
{% *А здесь вывод заголовка TITLE * %}
```



Основные переменные

\$settings['path'], может принимать значения '/' или '/path/to/script/' в зависимости от того в какой директории лежит скрипт - можно использовать в шаблонах, чтобы сослаться на корень сайта:

```
[<a href="javascript:popup('{%$settings.path%}region_changer/', 500, 400);">{%$locale.7%}</a>]
```

\$settings['path_region'], принимает значения '/' или '/Russia/Moscow/', в зависимости от параметров текущего региона пользователя - можно использовать в шаблонах, чтобы сослаться на корень сайта с учётом текущего региона

```
<a href='{%$settings.path_region%}messages/{%$message.link%}'>{%$message.title%}</a>
```

\$smarty.const.THEME, принимает значение '/templates/ThemeName/' – абсолютный путь к файлам темы

```
<link rel="stylesheet" type="text/css" ref="{%$smarty.const.THEME%}style.css" />
```

\$smarty.TITLE – основной заголовок

\$settings.user – массив с параметрами зарегистрированного и авторизованного пользователя; false, если пользователь не авторизован. Пример из block_user.html:

```
{%if $settings.user%}
{%$locale.18%}, {%$settings.user.name%!}
[...]
{%else%}
[...]
{%/if%}
```

Прочие элементы этого массива:

- \$settings.user.id – идентификатор юзера
- \$settings.user.name – его имя
- \$settings.user.status - статус: 0-неодобренный пользователь, 1-одобренный пользователь, 2 модератор
- \$settings.user.balance – средства на счету



Основные массивы данных

\$title_normal - основной тайтл, содержит все названия и идентификаторы текущей ветви дерева рубрик:

```
(  
  [0] => Array  
  (  
    [name] => Основная категория  
    [link] => main  
  )  
  [1] => Array  
  (  
    [name] => Подкатегория  
    [link] => sub  
  )  
  ...  
)
```

\$title_reverse - то же самое, но в реверсивном порядке (для вывода <title>, где перечисление рубрик идёт задом-наперёд)

\$region_array - массив с текущей веткой регионов

```
(  
  [0] => Array  
  (  
    [name] => Россия  
    [uri] => /russia/  
  )  
)
```

\$blocks - блоки для текущей страницы

```
(  
  [t] => <!--//блок вверху -> //-->  
  [b] => <!--//блок внизу -> //-->  
  [l] => <!--//блок слева -> //-->  
  [r] => <!--//блоки справа -> //-->  
  [h] => <!--//растяжки вверху -> //-->  
  [f] => <!--//растяжки внизу -> //-->  
  [m] => Array  
)
```



```
(
    [0] =><!--// блок среди объявлений 1-> //-->
    [1] =><!--// блок среди объявлений 2-> //-->
    ...
)
)
```

\$categories - линейный массив всех категорий

```
Array
(
    [1] => Array
    (
        [id] => 1
        [uri] => Avtotransport
        [name] => А в т о т р а н с п о р т
        [img] => 9.jpg
        [parent] => 0
        [sort] => 0
    )

    [2] => Array
    (
        [id] => 2
        [uri] => sale
        [name] => П р о д а ж а
        [img] =>
        [parent] => 1
        [sort] => 1
    )
)
```

\$allregions - линейный массив всех регионов

```
Array
(
    [1] => Array
    (
        [id] => 1
        [uri] => Russia
        [name] => Р о с с и я
        [parent] => 0
        [ipgeo_field] => s
        [ipgeo_value] => RU
        [sort] => 0
    )

    [2] => Array
    (
```




```
[id] => 2
[uri] => spb
[name] => С а н к т - П е т е р б у р г
[parent] => 1
[ipgeo_field] => c
[ipgeo_value] => С е в е р о - З а п а д н ы й  ф е д е р а л ь н ы й
о к р у г / С а н к т - П е т е р б у р г / С а н к т - П е т е р б у р г
[sort] => 1
)
)
```

\$tree - вложенный массив текущей ветки категорий. Он используется на главной странице и страницах категорий для отображения ссылок на все подкатегории.

Пример использования в tree1.html

```
<ul class="tree_list">
  {%foreach from=$tree item=el%}
    <li class="first_level">
      {%assign var="uri" value=$settings.region_path%}
      {%if $el.img != ""%}
        <img class='png' src='{%'$settings.path%'images/cat/{%'$el.img%}'
alt='{%'$el.name%}' />
      {%/if%}
      {%assign var="count" value=$el.id|count_msg%}
      <a href="{%'$uri%}{%'$el.uri%}'/">{%'$el.name%}</a>&nbsp;<span {%if $count == 0%}
class='empty'{%'$count%}</span>
      {%if $el.cat%}
        {%assign var="uri" value=$uri|cat:$el.uri|cat:"/"%}
        <ul>{%include file="tree1_recursive.html" el=$el.cat%}</ul>
      {%/if%}
    </li>
  {%/foreach%}
</ul>
```

\$full_tree.region, **\$full_tree.cat** - вложенные массивы **всех** рубрик и регионов на сайте, корневые уровни которых начинаются с элемента с id=0, а все дочерние элементы расположены в элементе массива с ключом cat. Таким образом, обход массива нужно начинать с **\$full_tree.region.cat** или **\$full_tree.cat.cat**



Пример использования

```
<p>Д е р е в о  р у б р и к </p>
<ul>
    {%foreach from=$full_tree.cat.cat item=e1%}
    <li>
        {%assign var="uri" value=$settings.path%}
        <a href="{% $uri %}{% $e1.uri %}/">{% $e1.name %}</a>
        {%if $e1.cat%}
        {%assign var="uri" value=$uri|cat:$e1.uri|cat:"/"%}
        <ul>{%include file="tree1_recursive.html" el=$e1.cat%}</ul>
        {%/if%}
    </li>
    {%/foreach%}
</ul>
<p>Д е р е в о  р е г и о н о в </p>
<ul>
    {%foreach from=$full_tree.region.cat item=e1%}
    <li>
        {%assign var="uri" value=$settings.path%}
        <a href="{% $uri %}{% $e1.uri %}/">{% $e1.name %}</a>
        {%if $e1.cat%}
        {%assign var="uri" value=$uri|cat:$e1.uri|cat:"/"%}
        <ul>{%include file="tree1_recursive.html" el=$e1.cat%}</ul>
        {%/if%}
    </li>
    {%/foreach%}
</ul>
```

В этих примерах инклюдится файл tree1_recursive.html, который затем подключает сам себя:

```
{%foreach from=$el item=e1%}
    <li>
        {%assign var="uri2" value=$uri%}
        {%assign var="count" value=$e1.id|count_msg%}
        <a href="{% $uri %}{% $e1.uri %}/">{% $e1.name %}</a>&nbsp;<span{%if $count == 0%}
class='empty'{%/if%}>({% $count %})</span>
        {%if $e1.cat%}
            {%assign var="uri" value=$uri|cat:$e1.uri|cat:"/"%}
            <ul>{%include file="tree1_recursive.html" el=$e1.cat%}</ul>
        {%/if%}
        {%assign var="uri" value=$uri2%}
    </li>
{%/foreach%}
```



Модификаторы переменных

- чаще всего это просто функция, которая применяется в переменной, для обрезания лишних символов или вывода даты в особом формате. После модифицируемого значения указывается символ | (вертикальная черта) и название модификатора

Примеры из list.html:

```
{%assign var='count' value=$blocks.m|@count %}
{%*
здесь использован модификатор @count для подсчета
количества рекламных блоков на данной странице, кроме
того, использована операция assign для создания новой
переменной «count» и присваивания ей значения,
полученного модификатором @count
*%}
[...]
```

{% \$locale.13%}: {% \$message.date_add|date_format:"%d.%m.%Y %H:%M"%}

```
{%*
все временные переменные скрипта представлены в виде
Unix timestamp, и чтобы привести их в читаемый вид используем
такой модификатор:
date_format:"%d.%m.%Y %H:%M"
*%}
[...]
```

{% \$message.title|truncate:400:'..'%}

```
{%*если заголовок оказался длиннее 400 символов,
обрезаем его и дописываем «...»*%}
```



Структура системы шаблонов

Два шаблона обрамляют вывод основного кода скрипта:

- body1.html - верх
- body2.html - низ

Эти модули подключают остальные составляющие системы:

Файл	Описание
add.html	Форма добавления и редактирования объявления
articles.html	Каталог статей в категории
articles_cat.html	Список категорий статей
articles_item.html	Текст отдельной статьи
block_counter.html	Блок статистики
block_promo.html	Рекламные блоки
block_nav.html	Блок навигации
block_tags.html	Блок с облаком тегов
block_user.html	Информационный блок пользователя
blocks.html	Оформление блоков
comment.html	Оформление комментариев
deleted.html	Сообщение при удалении объявления
fields.html	Поля ввода дополнительных полей форм добавления и поиска объявлений.
footer.html	«Подвал» сайта
header.html	Верхняя часть сайта
list.html	Вывод списка объявлений
message.html	Вывод одного объявления



popup_sendmail.html	Всплывающее окно с формой отправки письма
prolong_few.html	Страница продления срока действия объявления
search.html	Страница модуля отбора и сортировки
search_block.html	Блок с параметрами поиска и отбора объявлений
subscribe.html	Html код письма, отправляемого подписчикам (нельзя использовать стили)
tree.html	Вывод дерева рубрик
users_panel.html	Внешний вид панели управления пользователей. Далее идут модули, подключаемые к этой панели
- users_order.html	сделать заказ
- users_orders.html	список заказов
- users_pay.html	оплатить
- users_state.html	выписка по счёту
- users_subscribe.html	подписка на объявления
- users_registr.html	Форма регистрации пользователя
- users_login.html	Авторизация пользователя
- users_password_recovery.html	Напомнить пароль
- users_shop.html	Управление магазином пользователя
shops.html	Отображение публичного списка магазинов и публичной части отдельного магазина



Общая информация по php-модулям

Скрипт может легко расширяться за счет подключаемых модулей.

Подключаемый модуль должен:

- находиться в директории modules
- иметь расширение .inc (например notepad.inc или help.html.inc)
- содержать строки:

```
<?php
if(!$exe){
    // вывод headers (установка cookies, вывод content-type)
}
else{
    // собственно модуль, вывод которого будет
    располагаться
    // между шаблонами body1.html и body2.html
}
?>
```

Модуль будет запускать по адресу example.com/notepad или example.com/help.html по данному примеру.

Для добавления нового раздела админки, создаётся такой же файл с суффиксом *admin* и названия модуля в комментарии первой строки, например help.html_admin.inc:

```
<?php // Название раздела админки
if(!$exe){
    // headers
}
else{
    // модуль
}
?>
```



Динамические блоки

Динамические PHP блоки могут создаваться двумя способами:

- из панели администрирования, создаёте PHP блок и вводите код в соответствующее поле;
- посредством системы динамических блоков в шаблонах:
 - создается шаблон с именем файла `block_ИмяБлока.html`
 - в панели администрирования создаётся новый блок, в теле которого задается переменная `%block_ИмяБлока%`,
 - в админке указываются места отображения этого блока
 - блок подключен и отображается. Примеры: `block_counter.html`, `block_nav.html` и др.