

大家好，FreeSky 是一个轻量级模板引擎。省去了 Smarty 的繁琐。 尚未涉及太深领域而使用 smarty 这个庞然大物,会觉得拖家带口似的。 故推出 FreeSky1.0 测试版,平时使用的 smarty 时用到的功能在此应有尽有。 标签功能完善，可自定义是否开启缓存，缓存时间，及是否开启调试功能. 针对初学者所写，故大侠还需手下留情。若是能添加更多实用功能，鄙人不胜感激。 一切皆为自由故，想写就写，想做就做，有点想法就开始实践。为了自由！2011/5/16

模板内置了标签如下

**变量:**

{ \$variable }

**条件控制完全可以替代 if 语句 switch:**

{ switch \$variable case value1 }

HTML 代码

{ case value2 }

HTML 代码

{ case value2 }

HTML 代码

{ default }

HTML 代码

{ /switch }

**循环 foreach:**

{ foreach \$variable } { @key } => { @value } { /foreach }

**包含文件 include:** { include filename }

**注释语句** { \*content\* } { \*注释内容\* }

**配置值** { #config\_name# } { #配置名字# }

标签使用注意事项:

## 1>变量标签

>> {\$variable} >> 在\$variable 和 '{' 与 '}' 之间不可有空格。例如{ \$variable}或者{\$variable }都是错误的。

## 2>switch 标签

>> {switch \$variable case a} 大括号内是: switch 空格\$variable 空格 case 空格 a >> {case b} 大括号内是: case 空格 b >> switch 和 '{' 之间不可有空格 值 a 和 '}' 之间不可有空格 >> case 和 '{' 之间不可有空格 b 和 '}' 之间不可有空格 >> 要匹配的值 a 不可添加引号 >> 至少要包含一个 case 例如 {switch \$variable}是不对的, 必须要包含一个 case,即{switch \$variable case a}(如果只包含了这一个 case 语句后面没有{case b}第二个 case 条件则其功能就是 if 语句的功能) >> 不要忘记还有一个{default}哦, 这个是选用的 >> 必须要闭合 switch 标签 {/switch}

## 3>循环 foreach 标签

>> {foreach \$variable}大括号内容是: foreach 空格\$variable >> foreach 和 '{' 之间不可有空格, \$variable 和 '}' 之间不可有空格 >> 不必写 key 和 value, 例如{foreach \$variable as \$key=>\$value}是错误的, 只写{foreach \$variable}即可 >> 在 foreach 循环内使用 key 和 value 值时, 唯一一种方式: {@key} 和 {@value} 即模板默认已分配键和值于 {@key}和{@value}, 记住是 @ 而不是 \$ >> foreach 标签必须关闭 {/foreach}

## 4>包含文件 include

>> {include filename}大括号内是: include 空格 filename >> include 和 '{' 之间不可有空格, filename 和 '}' 之间不可有空格 >> filename 不可加引号 例如{include 'header.html'}是错误的。正确是: {include header.html}

## 注释语句{\*\*}

>> {\*注释内容\*} >> '{' 和 '\*' 之间不可有空格, '\*' 和 '}'之间不可有空格

## 调用配置{#配置名#}

>> {#config\_name#} >> '{' 和 '#' 之间不可有空格, '#' 和 '}'之间不可有空格, 以及 config\_name 和#之间不可有空格 >> 配置名 config\_name 不用加\$不用加引号 >> 配置文件格式使用的 php 返回数组的格式而使效率最佳,具体看后面的范例

## **display 时不用写文件后缀**

例如: 要显示模板内文件 test.html `$fs->display('test.html');`是不对的。正确为: `$fs->display('test');`

**后面有使用范例,可一目了然**

## FreeSky 配置项

```
$fs = new FreeSky();
```

```
$fs->template_dir = 'templates';           //配置模板目录 默认为 templates  
$fs->compile_dir = 'compiles';             //配置编译目录 默认为 compiles  
$fs->config_dir = 'configs';               //配置配置目录 默认为 configs  
$fs->template_postfix = '.html';           //配置模板文件后缀 默认为 html
```

=====以下为高级配置，一般情况下不用配置=====

```
$fs-> caching = false;    //关闭缓存，默认为关闭 false 适合于正在开发中，  
                        如果网站已经开发完毕可开启缓存以更
```

快速运行将其设为 true

```
$fs-> cache_time = '3000';           //如果开启了缓存，可以设置缓存时间
```

```
$fs->debugging = true;    //开启调试功能，默认为开启 true 适合于正在开发中(强烈建议不要改动)
```

如果网站已经运行一段时间没有出现问题

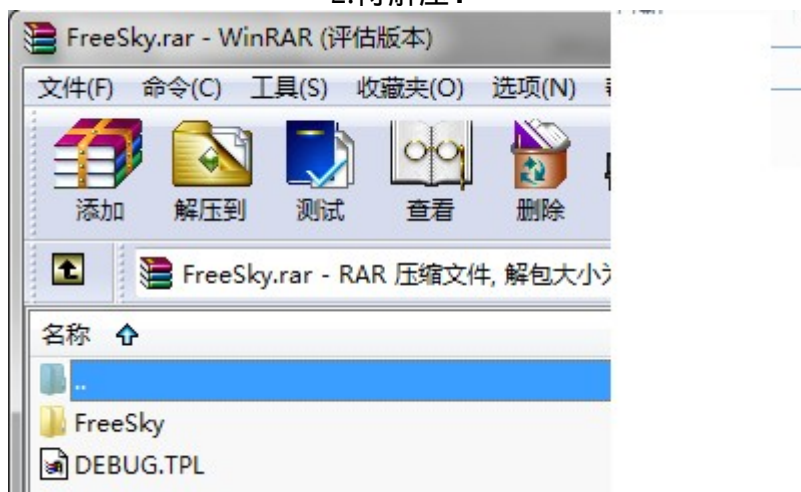
题可将其关闭设为 false

## 第一步解压搭建目录

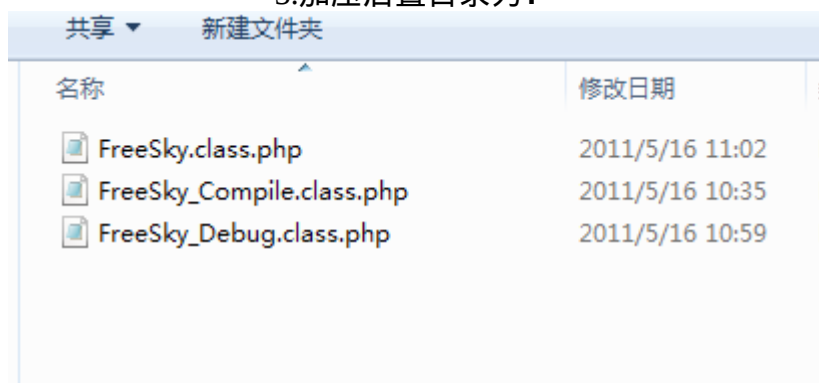
1.解压前：



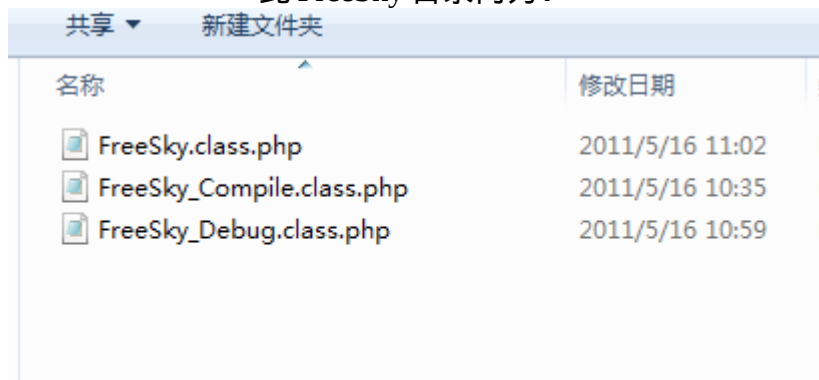
2.待解压：



3.加压后置目录为：



此 FreeSky 目录内为：



4.创建模板目录：templates(可自定义名字) 创建编译目录：compiles(可自定义名字) 创建配置目录：

configs(可自定义名字)

共享 新建文件夹		
名称	修改日期	类型
FreeSky.class.php	2011/5/16 11:02	PHP 文件
FreeSky_Compile.class.php	2011/5/16 10:35	PHP 文件
FreeSky_Debug.class.php	2011/5/16 10:59	PHP 文件

5.创建测试文件 test.php 搭建成功 整体目录如下:

名称	修改日期	类型
FreeSky	2011/5/16 14:32	文件夹
templates	2011/5/16 14:33	文件夹
compiles	2011/5/16 14:33	文件夹
configs	2011/5/16 14:33	文件夹
DEBUG.TPL	2011/5/16 9:46	TPL 文件
test.php	2011/5/16 14:34	PHP 文件

### test.php 内容如下

```
< ?php
header('Content-type:text/html;charset=utf-8');
include 'FreeSky/FreeSky.class.php';
$fs = new FreeSky();

$fs->template_dir = 'templates'; //配置模板目录, 如果目录为 templates 则此步可省
$fs->compile_dir = 'compiles'; //配置编译目录, 如果目录为 compiles 则此步可省
$fs->config_dir = 'configs'; //配置配置目录, 如果目录为 configs 则此步可省

$variable = '我是变量';
$fs->assign('one', $variable); //用于测试变量, 或者直接写 $x->assign('one', '我是变量');

$fs->assign('two', 'c'); //用于测试 switch

$two = array('a','b','c');
$fs->assign('ar',$two); //用于测试 foreach

//测试包含文件 include, 注释{*注释内容*}, 取配置值{#配置名#}都在模板内容, 下面写出模板内容

$fs->display('test');
?>
```

以下为各个文件内容

在 templates 文件夹内创建文件 test.html 内容如下:

```
=====变量使用=====
    变量: {$one}
=====switch 使用=====
    {switch $two case a}
        变量的值为 a 时显示的 HTML 代码
        {case b}
        变量的值为 b 时显示的 HTML 代码
        {case c}
        变量的值为 c 时显示的 HTML 代码
        {case d}
        变量的值为 d 时显示的 HTML 代码
        {default}
    变量的值不为上面几种情况时默认显示的 HTML 代码
    {/switch}
=====foreach 使用=====
    {foreach $ar}
        键: {@key} 值: {@value}
    {/foreach}
=====在 foreach 内嵌套 switch 语句=====
    {foreach $ar}
        {@value}--
        {switch @value case b}
            当@value 等于 b 时显示的 HTML 代码--
        {/switch}
    {/foreach}
=====include 使用=====
    {include footer.html}
=====注释使用=====
    {*这里面的内容不会显示*}
=====调用配置使用=====
    配置名为 color 的值: {#color#}
```

由于上面模板调用了 include 的语句包含了一个 footer.html 文件, 故在 templates 文件夹内创建文件 footer.html 内容如下:

~~~~~我是页脚内容~~~~~

configs 目录下的创建配置文件 config.php ( 配置文件名字必须为 config.php )

在此声明: 配置文件采用 php 返回数组的方式而达到最佳效率, 内容如下

```
< ?php
return array(
    //配置名=>配置值
    'hello'=>'你好',
    'color'=>'红色', //上面仅是调用了这一个配置值
    'world'=>'世界'
)
?>
```



## 运行 test.php 最终显示结果如下

```
=====变量使用=====
        变量：我是变量
=====switch 使用=====
        变量的值为 c 时显示的 HTML 代码
=====foreach 使用=====
        键：0  值：a
        键：1  值：b
        键：2  值：c
=====在 foreach 内嵌套 switch 语句=====
        a--b--当@value 等于 b 时显示的 HTML 代码--c--
=====include 使用=====
        ~~~~~我是页脚内容~~~~~
=====注释使用=====
=====调用配置使用=====
        配置名为 color 的值：红色
```

作者：活雷锋

E-mail: [txthinking@gmail.com](mailto:txthinking@gmail.com)

PHP 技术交流群：92834040