Smarty模板引擎

[学习目标 2](#_Toc522549518)

[模板引擎的工作原理 2](#_Toc522549519)

[1、如何实现HTML代码和PHP代码分离？ 2](#_Toc522549520)

[2、如何完全去除视图文件中的PHP标记？ 3](#_Toc522549521)

[3、常用PHP模板引擎介绍 3](#_Toc522549522)

[Smarty快速入门 4](#_Toc522549523)

[1、Smarty是什么？ 4](#_Toc522549524)

[2、Smarty下载和目录结构 4](#_Toc522549525)

[3、实例：第1个Smarty案例 5](#_Toc522549526)

[Smarty配置 5](#_Toc522549527)

[1、Smarty左右定界符配置 5](#_Toc522549528)

[2、Smarty常用目录配置 6](#_Toc522549529)

[Smarty模板(视图)中的变量 7](#_Toc522549530)

[1、Smarty普通变量 7](#_Toc522549531)

[2、Smarty保留变量 7](#_Toc522549532)

[（1）Smarty页面请求保留变量 7](#_Toc522549533)

[（2）访问PHP的预定义常量 8](#_Toc522549534)

[（3）Smarty时间戳保留变量 8](#_Toc522549535)

[3、Smarty配置文件变量 9](#_Toc522549536)

[（1）配置文件变量概述 9](#_Toc522549537)

[（2）定义配置文件 9](#_Toc522549538)

[（3）访问Smarty配置文件变量 10](#_Toc522549539)

[（4）配置文件分组 10](#_Toc522549540)

[Smarty循环——foreach 11](#_Toc522549541)

[1、foreach语法格式 11](#_Toc522549542)

[2、实例：输出一维数组 11](#_Toc522549543)

[3、实例：输出二维数组 12](#_Toc522549544)

[4、foreach常用属性应用 12](#_Toc522549545)

[Smarty循环——section循环 13](#_Toc522549546)

[1、section语法格式 13](#_Toc522549547)

[2、实例：输出一维枚举数组 14](#_Toc522549548)

[3、实例：输出二维枚举数组 14](#_Toc522549549)

[4、section控制循环起点、步长值 15](#_Toc522549550)

[Smarty条件判断——if语句 15](#_Toc522549551)

[1、if中运算符 15](#_Toc522549552)

[2、实例：当兵年龄判断 16](#_Toc522549553)

[3、实例：表格隔行变色 16](#_Toc522549554)

[Smarty变量调节器 17](#_Toc522549555)

[1、Smarty变量调节器概述 17](#_Toc522549556)

[2、Smarty中常用变量调节器 17](#_Toc522549557)

[3、date\_format调节器参数及应用 19](#_Toc522549558)

[4、truncate调节器参数及应用 19](#_Toc522549559)

# 学习目标

* 模板引擎的工作原理
* Smarty入门
* Smarty配置
* Smarty中的变量：普通变量、保留变量、配置文件变量
* Smarty中的循环：foreach、section
* Smarty中的if语句
* Smarty变量调节器

# 模板引擎的工作原理

## 1、如何实现HTML代码和PHP代码分离？

实现HTML代码和PHP代码分离的目的是：实现前端设计人员和PHP程序员的分离。

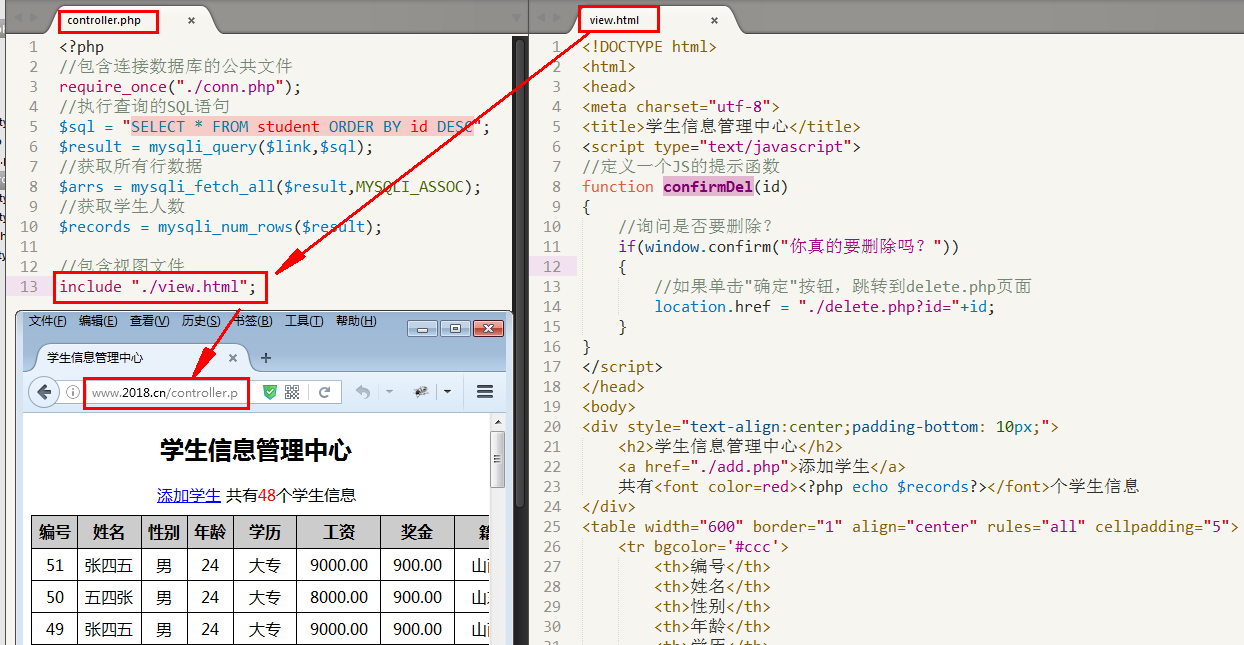
多数Web开发人员要么是精通网页设计，能够设计出漂亮的网页外观，但是编写的PHP代码很糟糕；要么仅熟悉PHP编程，能够写出健壮的PHP代码，但是设计的网页外观很难看。具备两种才能的开发人员很少见。

将PHP和HTML混合页面，分成两个独立的页面：

一个是HTML静态页面(视图文件、模板文件)，扩展名是.html，包含HTML、CSS、JS

一个是纯PHP程序页面(控制器文件)，扩展名是.php，主要PHP代码。

运行时，将HTML文件代码，包含到PHP文件中。



## 2、如何完全去除视图文件中的PHP标记？

* 首先，视图中的典型PHP代码是这样的：**<?php echo $name ?>**
* 再者，前端工程师能看得懂的、比较喜欢的HTML标签形式的代码是这样的：**{$name}**
* 最后，我们把 {$name} 替换成 <?php echo $name ?>，就实现了PHP代码和HTML代码的完全分离。
* 模板引擎的原理：就是替换，就是将 {$name} 转换成 <?php echo $name?> 才能被PHP识别并解析。
* 如何替换呢？使用PHP替换函数str\_replace()，将 ”{“ 替换成 ”<?php echo “，将”}”替换成”?>”。



## 3、常用PHP模板引擎介绍

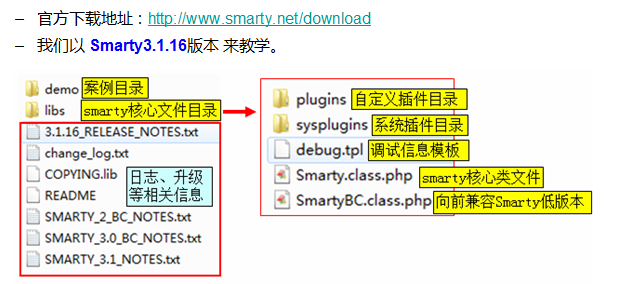
* Smarty，是模板引擎鼻祖。其它的模板引擎都是基于Smarty开发的。
* Template Lite
* TinyButStrong
* XTemplate
* Savant
* phemplate
* Dwoo
* Sugar
* Twig，symfony框架默认的模板引擎；
* FXL Template
* H2o
* ns-template
* Blade：laravel框架默认的模板引擎。

# Smarty快速入门

## 1、Smarty是什么？

Smarty是一个使用PHP写出来的模板引擎，是目前业界最著名的PHP模板引擎之一。它分离了逻辑代码和外在的内容，提供了一种易于管理和使用的方法，用来将原本与HTML代码混杂在一起PHP代码逻辑分离。简单的讲，目的就是要使PHP程序员同前端人员分离，使程序员改变程序的逻辑内容不会影响到前端人员的页面设计，前端人员重新修改页面不会影响到程序的程序逻辑，这在多人合作的项目中显的尤为重要。

## 2、Smarty下载和目录结构



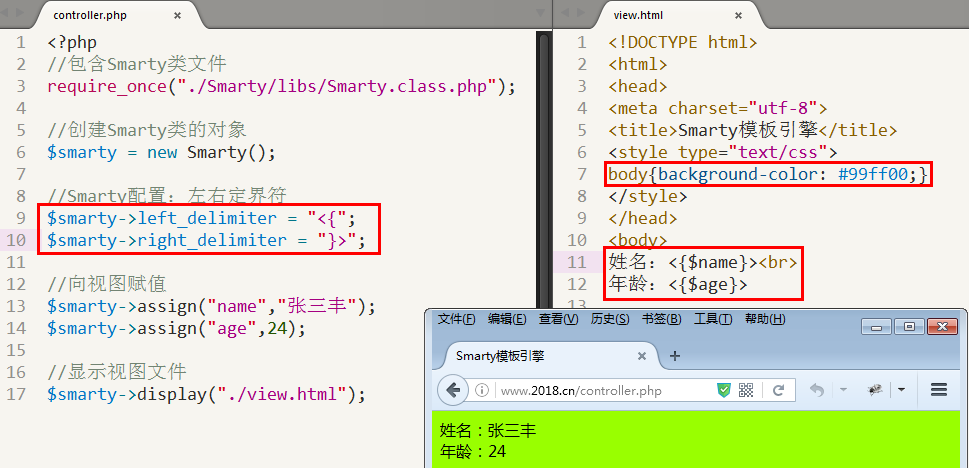
## 3、实例：第1个Smarty案例



# Smarty配置

## 1、Smarty左右定界符配置

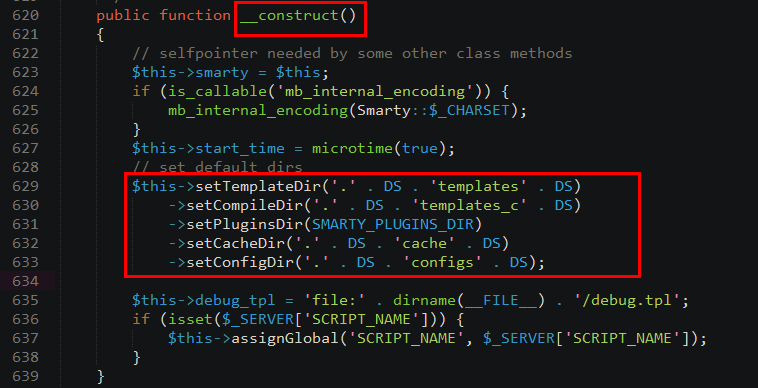
* 在Smarty中，默认使用 "{" 和 "}" 作为变量的定界符。但这样，会与CSS、JS中的大括号冲突。
* 左定界符：**$smarty->left\_delimiter = "string"**
* 右定界符：**$smarty->right\_delimiter = "string"**



## 2、Smarty常用目录配置

|  |  |  |  |
| --- | --- | --- | --- |
| 目录类别 | 默认目录 | 修改方法 | 查看方法 |
| Smarty默认模板目录 | ./templates | $smarty->setTemplateDir() | $smarty->getTemplateDir() |
| Smarty默认配置目录 | ./configs | $smarty->setConfigDir() | $smarty->getConfigDir() |
| Smarty默认编译目录 | ./templates\_c | $smarty->setCompileDir() | $smarty->getCompileDir() |
| Smarty默认缓存目录 | ./cache | $smarty->setCacheDir() | $smarty->getCacheDir() |
| Smarty默认插件目录 | ./libs/plugins | $smarty->setPluginsDir() | $smarty->getPluginsDir() |

查看Smarty常用目录配置：Smarty类文件中的\_\_construct()构造方法

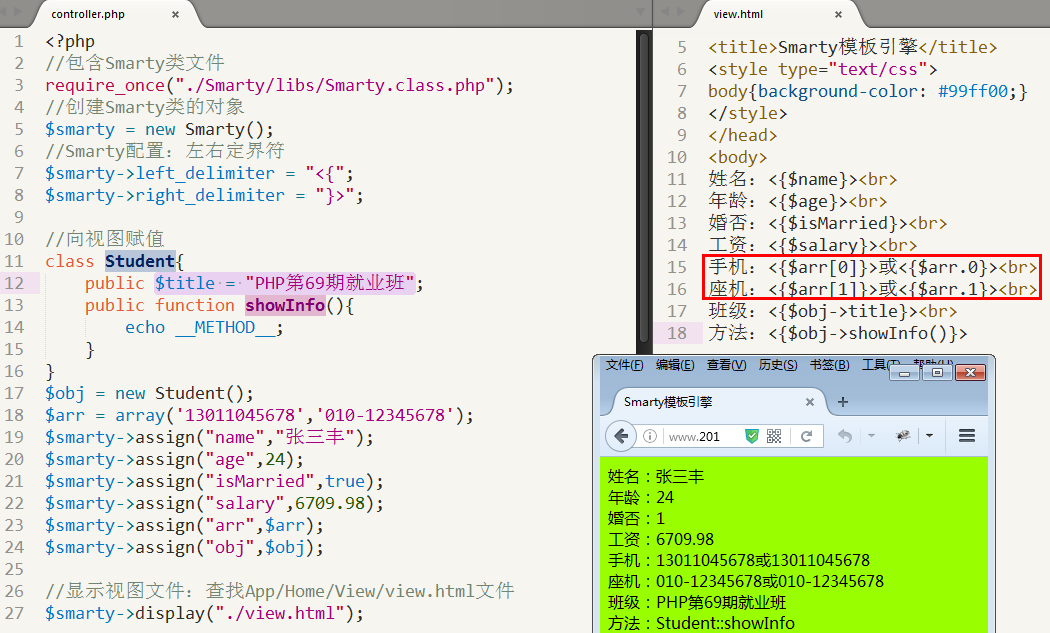




# Smarty模板(视图)中的变量

## 1、Smarty普通变量

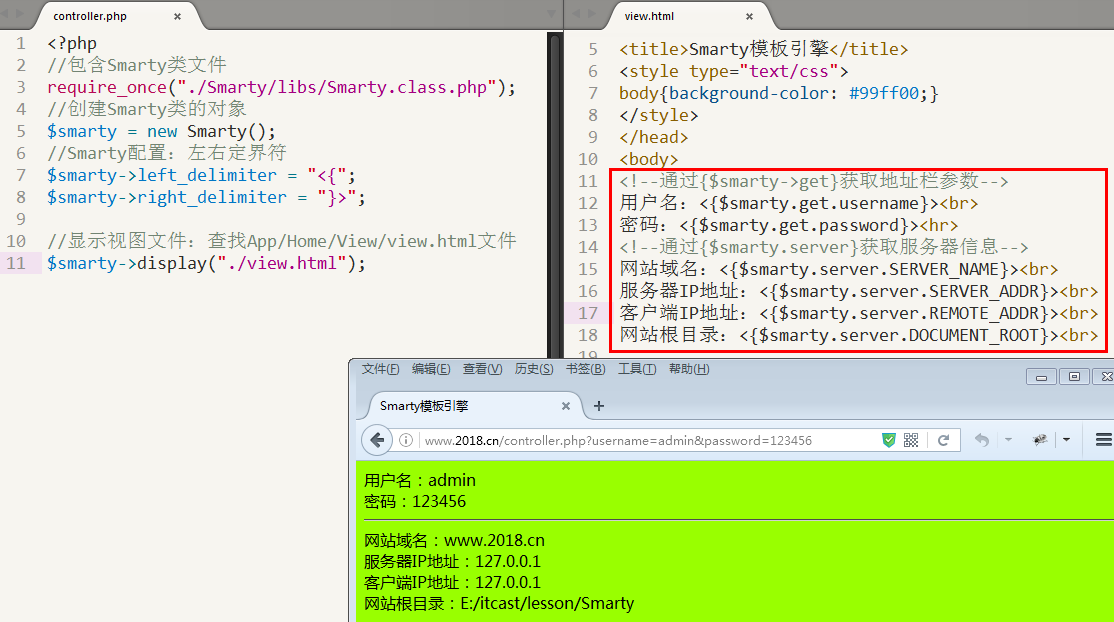
* PHP中的所有变量，都可以在视图文件中使用；
* 使用 $smarty->assign() 向Smarty模板传递普通变量；
* 使用 $smarty->display() 显示指定的视图文件；
* 在Smarty模板中，使用"[ ]"或点"."来访问数组元素；
* 在Smarty模板中，使用"->"来访问对象的属性和方法。



## 2、Smarty保留变量

### （1）Smarty页面请求保留变量

* {$smarty.get} 访问$\_GET数组
* {$smarty.post} 访问$\_POST数组
* {$smarty.request} 访问$\_REQUEST数组
* {$smarty.cookie} 访问$\_COOKIE数组
* {$smarty.session} 访问$\_SESSION数组
* {$smarty.server} 访问$\_SERVER数组
* {$smarty.files} 访问$\_FILES数组



### （2）访问PHP的预定义常量

语法：{$smarty.const.预定义常量}



### （3）Smarty时间戳保留变量

语法：{$smarty.now}



## 3、Smarty配置文件变量

### （1）配置文件变量概述

网站上有一些比较简单的变量信息，美工人员可以自行定义并调用。这样可以脱离程序员的依赖，工作比较有主动权。

### （2）定义配置文件

* 配置文件默认目录：./configs
* 设置配置文件目录：$smarty->setConfigDir()
* 读取配置文件目录：$smarty->getConfigDir()
* 配置文件扩展名：.ini或.conf
* 配置文件中注释：#
* 配置文件变量分组：[ ]
* 语法格式：配置名 = 变量值
* 变量不带$符号，变量值不带引号



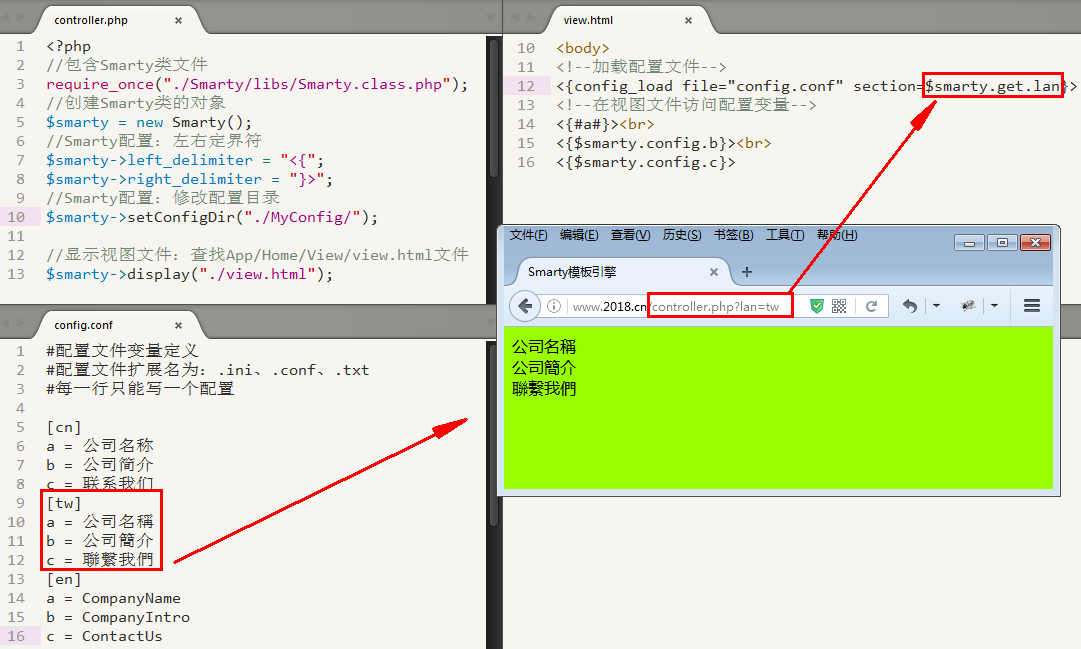
### （3）访问Smarty配置文件变量

* 载入配置文件：**{config\_load file = “配置文件路径”}**
* 语法格式一：**{#配置文件变量名#}**
* 语法格式二：**{$smarty.config.配置变量}**



### （4）配置文件分组

载入分组配置文件变量：**{config\_load file = “配置文件路径” section = “分组名”}**

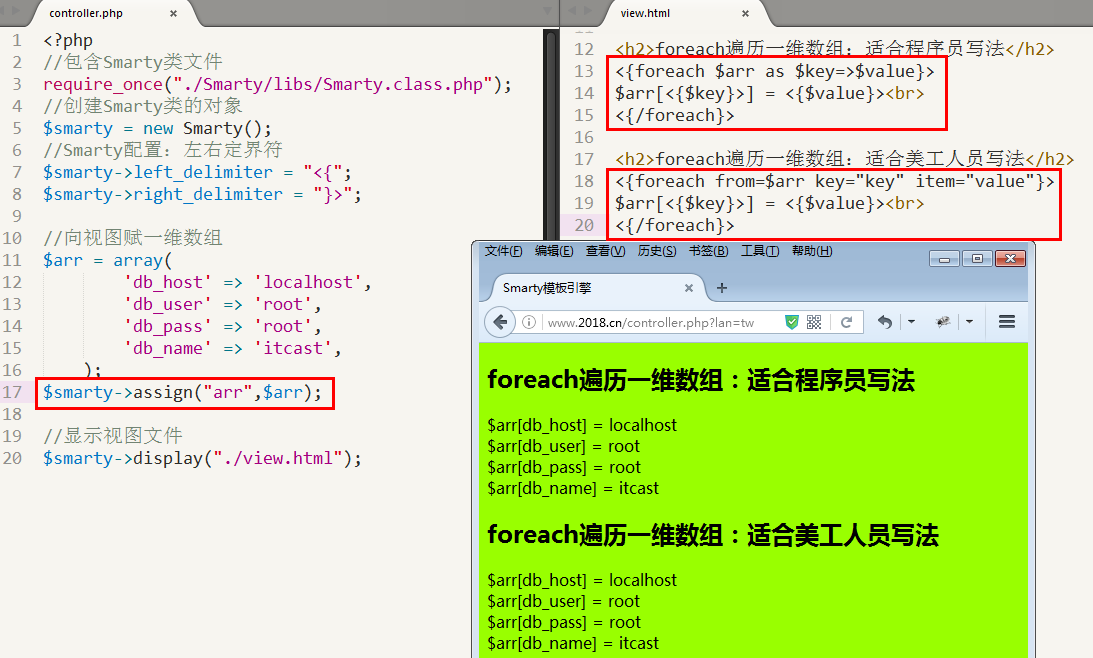


# Smarty循环——foreach

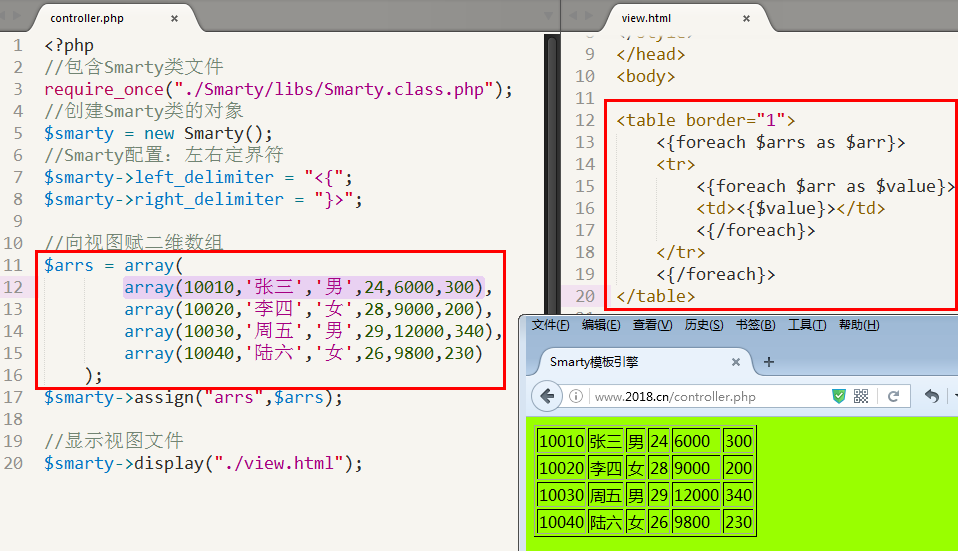
## 1、foreach语法格式

* 语法格式1：**{foreach $arr as $key=>$value} {/foreach}**
* 语法格式2：**{foreach from=$myarr key="mykey" item="myitem"}{/foreach}**
* **提示：foreach可以遍历所有类型的数组，包括：枚举数组、关联数组、混合数组**。

## 2、实例：输出一维数组

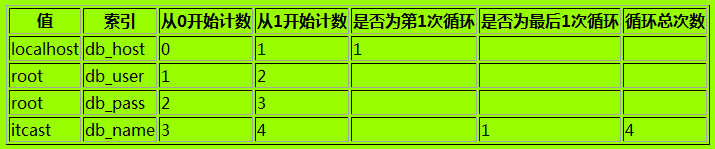
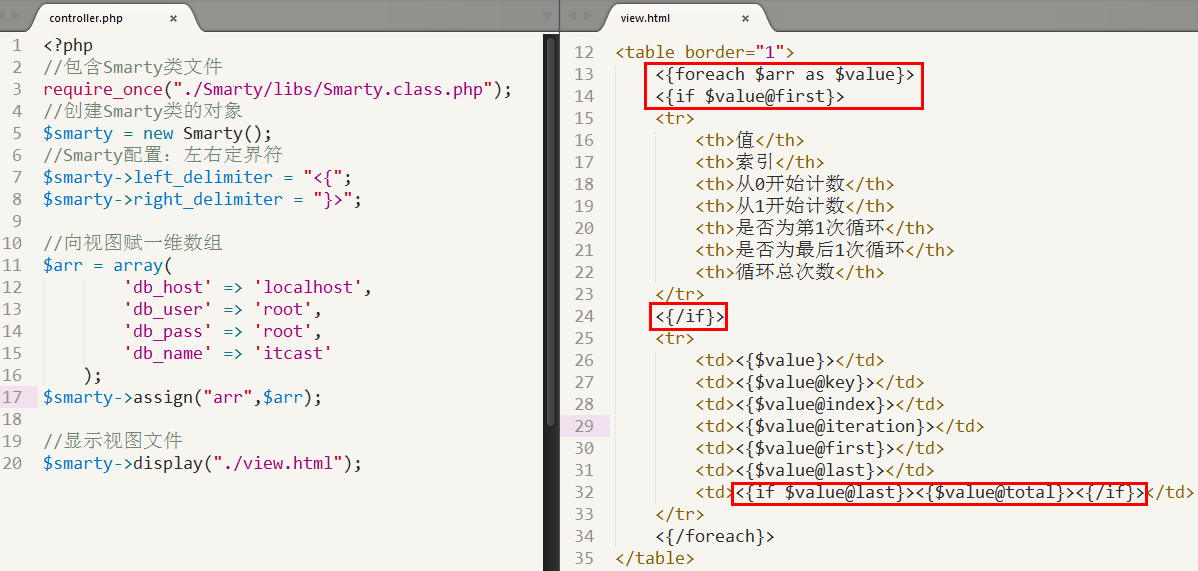


## 3、实例：输出二维数组



## 4、foreach常用属性应用

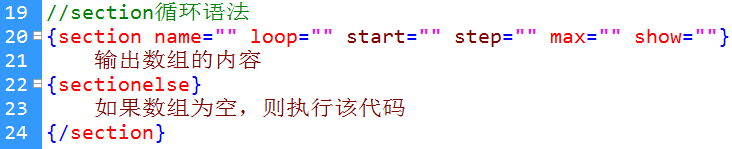
* @key：输出当前值的索引，可能是整型索引，也可能是字符索引；
* @index：当前数组索引，从0开始计算；
* @iteration，当前循环的次数，从1开始计算；
* @first：当首次循环时，值为true；
* @last：当最后一次循环时，值为true；
* @total：是整个循环的次数，可以在foreach内部或外部使用；



# Smarty循环——section循环

## 1、section语法格式

* Section循环就是PHP中的for循环。
* PHP中的for循环对于foreach来说，有什么特点？
* **for循环可以控制循环起点**，而foreach不可以；如：$i = 1 或 $i = 3
* **for循环可以指定步长值**，而foreach不可以；如：$i += 2 或 $i += 3
* **for循环可以控制循环次数**，而foreach不可以；如：$count = 10 或 $count = 5
* **注意：section循环只能遍历枚举数组。**



name= 随便命名

loop= 最后一个循还号

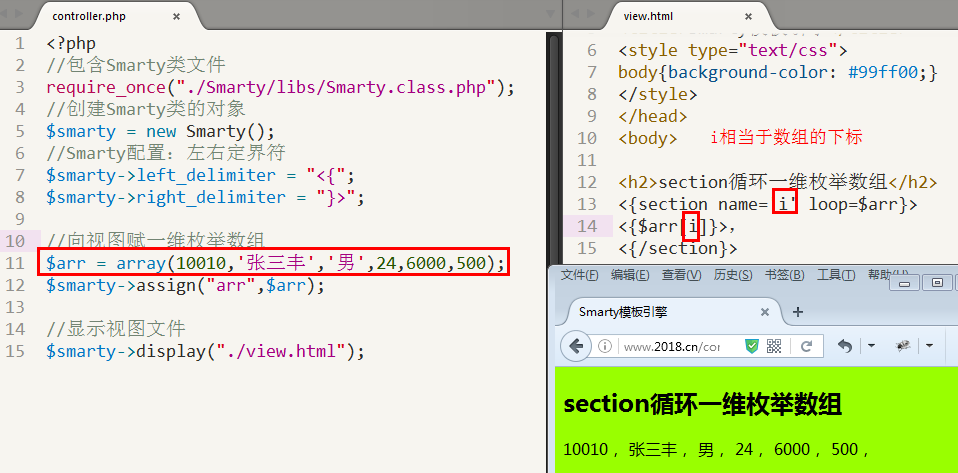
start= 起点

step= 跳的值的个数

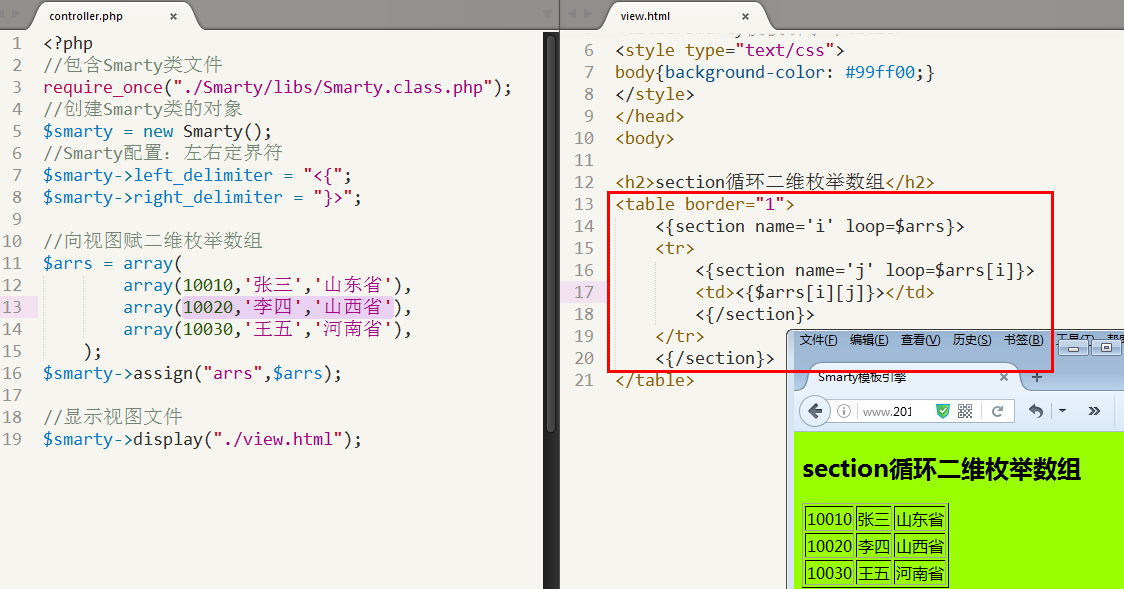
max= 最大输出几个值

show=

## 2、实例：输出一维枚举数组



## 3、实例：输出二维枚举数组



## 4、section控制循环起点、步长值



# Smarty条件判断——if语句

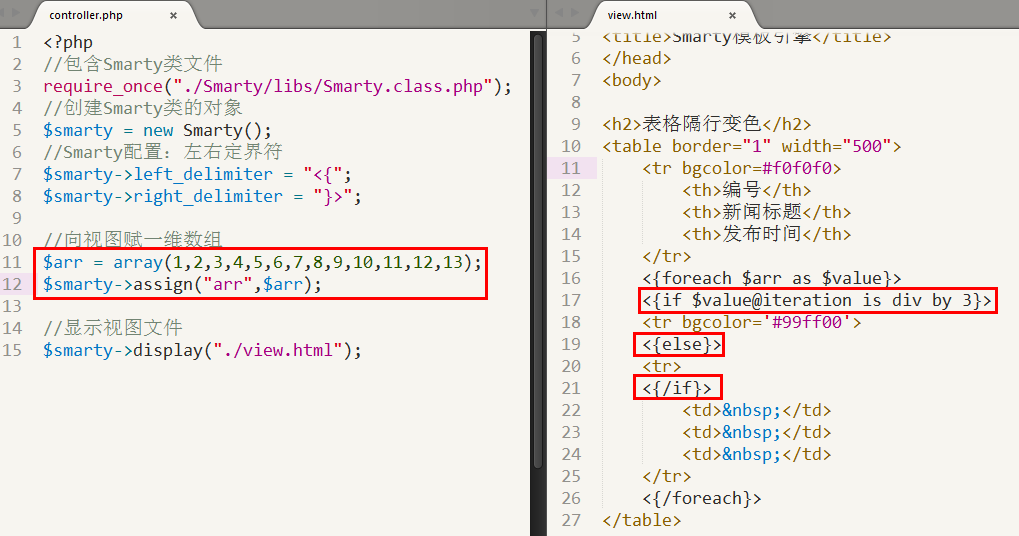
## 1、if中运算符



## 2、实例：当兵年龄判断



## 3、实例：表格隔行变色

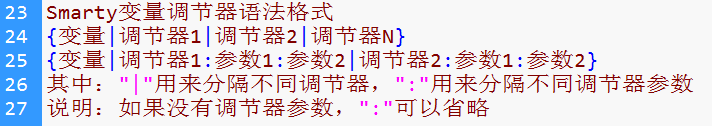




# Smarty变量调节器

## 1、Smarty变量调节器概述

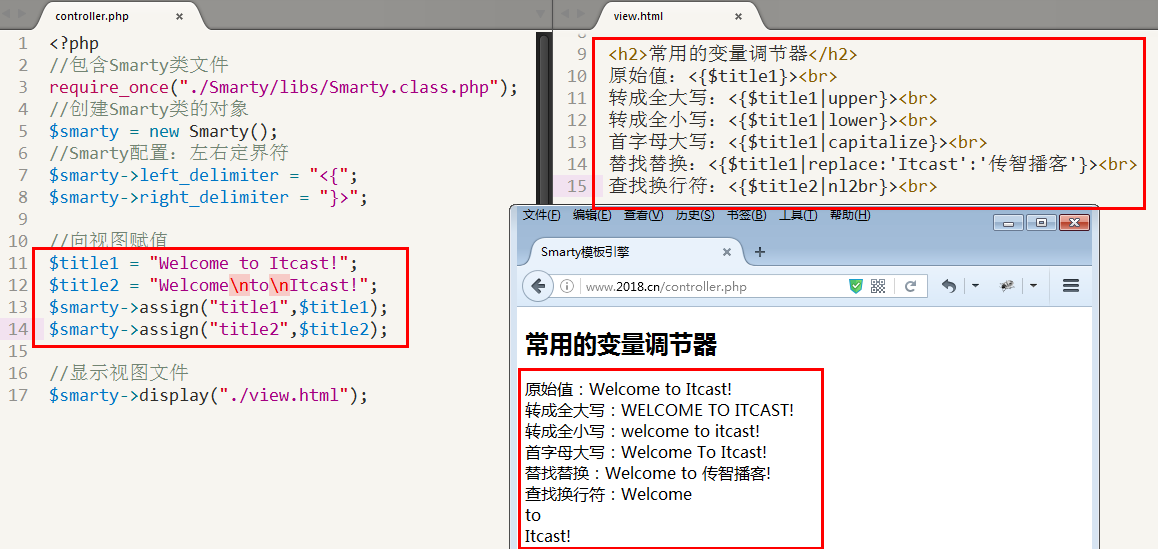
* 变量修饰器可以用于格式化变量；
* 使用修饰器，需要在变量的后面加上|(竖线)并且跟着修饰器名称。
* 修饰器可能还会有附加的参数以便达到效果。
* 参数会跟着修饰器名称，用:(冒号)分开。
* 同时，默认全部PHP函数都可以作为修饰器来使用 (不止下面的)，而且修饰器可以被联合使用。
* 修饰器可以作用于任何类型的变量，数组或者对象。



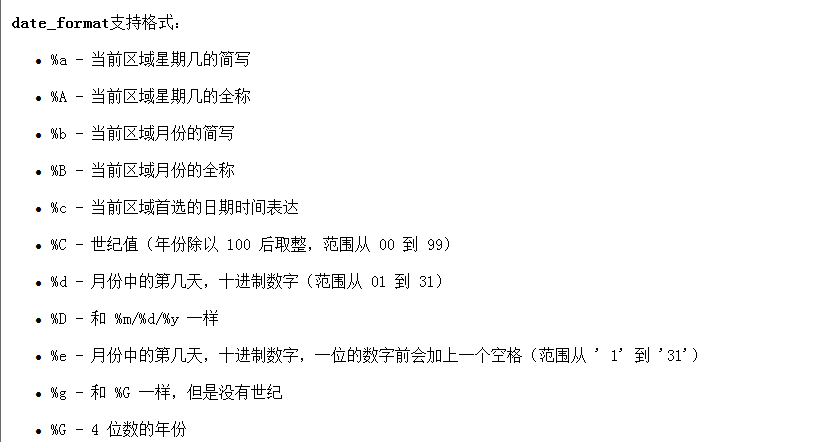
## 2、Smarty中常用变量调节器

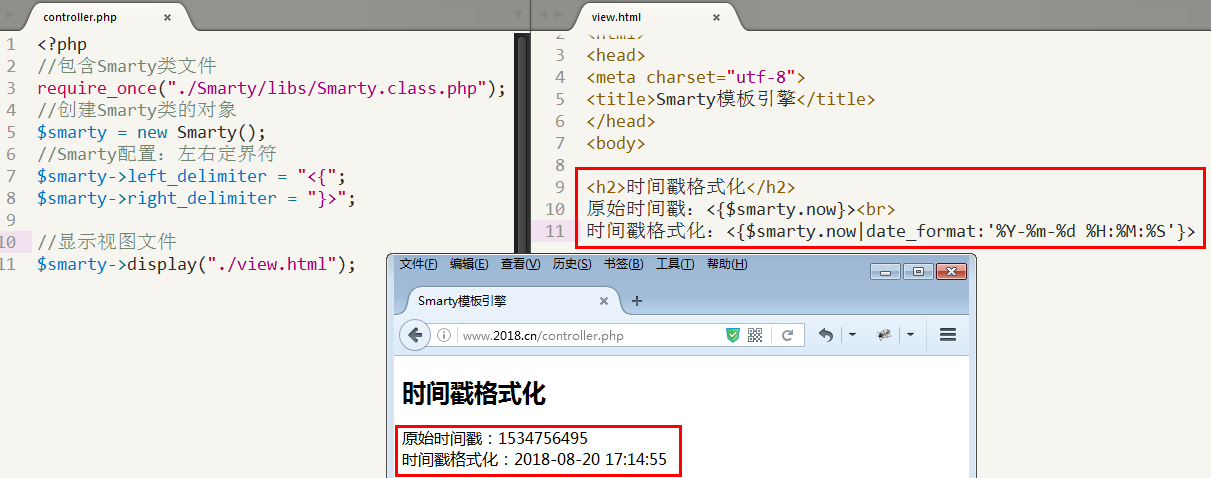
|  |  |  |  |
| --- | --- | --- | --- |
| 调节器 | 含义 | PHP函数 | 示例演示 |
| upper | 将变量值转成大写字母。 | strtoupper() | {$title|upper} |
| lower | 将变量值转成小写字母 | strtolower() | {$title|lower} |
| capitalize | 每个单词的第一个字母大写 | ucwords() | {$title|capitalize} |
| nl2br | 将变量中"\n"回车  全部转换成HTML的 <br /> | nl2br() | {$title|nl2br} |
| replace | 对变量进行简单的查找和替换 | str\_replace() | {$title|replace:’a’:’b’} |
| date\_format | 将日期和时间格式化成strftime()的格式。 | strftime() | {$time|date\_format:%Y-%m-%d} |
| truncate | 截取字符串到指定长度，默认长度是80 | substr() | {$title|truncate:80:’…’} |





## 3、date\_format调节器参数及应用





## 4、truncate调节器参数及应用

* smarty中的truncate调节器，对应PHP的substr()函数或mb\_substr()函数。
* substr()函数是按字节截取字符串；
* mb\_substr()函数是按字符截取字符串，但需要开启PHP的扩展。



