Smarty 模版引擎

.....

> 作者:王猛
> QQ:672725440
> 指导:燕十八
> 出自:布尔教育

学习方法

HTML, PHP基础巩固,面向对象,MySQL
以上为理论课程,很难从0摸索出来.
后面的课程多为工具性质的产品

1.1 学习心态:
不要怕，你肯定可以掌握
要敢于摸索,不要被动等老师讲
最快的办法就是试用

1.2 具体步骤:
1. 下载解压
2. 看官方手册/demo/INSTALL/README ,快速搭建一个例子
3. 手册增加例子的复杂度

1.3 错误方式:
先baidu,google一堆网页,或者下载视频.
因为

- 网页上的东西,绝大部分是抄的手册,而且相互转载
- 网页相对于手册是落后的.
- 资料找的越多,越学不进去.

** 前后端 **
PHP和HTML都由一个人来写的时候,可能是这样的

```
<?php
$sql = 'select ....'; //查询等等操作
$title = '今天非常凉爽';
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
    <body>
        <h1><?php echo $title;?></h1>
    </body>
</html>
```

PHP和HTML都由两个人来写时,可能是这样的,因为前端妹子并不懂PHP代码
1.php

```
$title = '这是标题';
$content = 这是内容';
//引入模版
include(./1.html);
```

1.html

```
<h1><?php echo $title; ?></h1>
<p><?php echo $content; ?></p>
```

但是,全段人员,不想看到PHP的标签,怎么办?
我们的代码编程这样:
$title = '这是标题';
$content = '这是内容';
//引入模版
include(./1.html);

1.html

```
<h1>{$title}</h1>
<p>{$content}</p>
```

输出效果，很显然不是我们要的样子
因为前端想要的是{$title}
PHP只能写成<?php echo $title; ?>才能正常显示

怎么办?

解决办法:用程序员把前端想要的标签,转换为PHP需要的标签,这样2者都可以满足了

思路:
前端是1.html 她只会写html,并且要求{$title}显示
后端是1.php 你的需要<?php echo $title ?>显示
因为相同的是$title,不相同的只有 <?php echo 和 ;?> ,是否可以找来第三个文件,把 {? 替换为 <?php ,把 } 替换为 ;?> 呢?就是:先读取1.html内容,找到{$title},因为你内容是{$title}的,无法显示到浏览器,然后我把{$title}给替换成<?php echo $title;?>,输出到第三方文件,第三方文件就成了可以输出的<?php ?>类型的文件.这个时候我再引入这个第三方文件.不就能显示了嘛.

```
function comp($file) {
    $html = file_get_contents($file);
    $html = str_replace('{$', '<?php echo $', $html);
    $html = str_replace('}', ';?>', $html);
    $compfile = $temp.'.php';
    file_put_contents($compfile, $html);
    return $compfile;
}

$sql = 'select ....'; //查询等等
$title = '这是标题';
include( comp('./xx.html') );
```

$file

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
    <h1>{$title}</h1>
</body>
</html>
```

mini模板类

最直接的修改:

```
class Mini{
    function comp($file) {
        $h = file_get_contents($file);
```

```
        $h = str_replace('{$', '<?php echo $', $h);
        $h = str_replace('}', ';?>', $h);
        $compfile = $temp.'.php';
        file_put_contents($tmp, $h);
        return $tmp;
    }
}
$sql = 'select ....'; //查询等等
$title = '这是标题';
$mini = new Mini();
include( comp('./xx.html') );
```

但这后面是面向过程的,我需要修改成比较纯的面向对象的风格
我希望引入这个文件也用调用方法的方式

```
class Mini{
    public function comp($file) {
        $h = file_get_contents($file);
        $h = str_replace('{$', '<?php echo $', $h);
        $h = str_replace('}', ';?>', $h);
        $compfile = $temp.'.php';
        file_put_contents($tmp, $h);
        return $tmp;
    }
    public function display($file){
        include($file);
    }
}
$sql = 'select ....'; //查询等等
$title = '这是标题';
$mini = new Mini();
$mini->display('1.html');
//include( comp('./xx.html') );
```

发现不行,因为你没调用用上面的方法进行写入

```
class Mini{
    public function comp($file) {
        $h = file_get_contents($file);
        $h = str_replace('{$', '<?php echo $', $h);
        $h = str_replace('}', ';?>', $h);
        $compfile = $temp.'.php';
        file_put_contents($tmp, $h);
        return $tmp;
    }
    public function display($file){
        $file = $this->comp($file);
        include($file);
    }
}
$sql = 'select ....'; //查询等等
$title = '这是标题';
$mini = new Mini();
$mini->display('1.html');
//include( comp('./xx.html') );
```

发现没有找到$title
因为include相当于复制,并没有$title
想想有没有方法----->把标题和内容想办法写到属性里面,这样,方法内部就可以使用这个属性了

因为你标题和内容都需要使用,并且是对应的,标题=>内容.我需要弄一个方法

$mini->assign('title',内容);

```
class Mini{
```

```
    public function comp($file) {
        $h = file_get_contents($file);
        $h = str_replace('{$', '<?php echo $', $h);
        $h = str_replace('}', ';?>', $h);
        $compfile = $temp.'.php';
        file_put_contents($tmp, $h);
        return $tmp;
    }
    public function display($file){
        $file = $this->comp($file);
        include($file);
    }
}
$sql = 'select ....'; //查询等等
$title = '这是标题';
$mini = new Mini();
$mini->assign('title',$title);
$mini->display('1.html');
//include( comp('./xx.html') );
```

```
class Mini{
    public $data = array();
    function tit($file){
        $h = file_get_contents($file);
        $h = str_replace('{$','<?php  echo $this->data[\'',$h);
        $h = str_replace('}','\'];?>',$h);
        $tmp = $file.'.php';
        file_put_contents($tmp, $h);
        return $tmp;
    }
    public function assign($k,$v){
        $this->data[$k] = $v;
    }
    public function display($file){
        require($this->tit($file));
    }
}
$sql = "select....";
$title = '你是谁?';

$mini = new Mini();
//require './1.html';
//require($mini->tit('1.html'));
$mini->assign('title',$title);
$mini->display('./1.html');
```

学习smarty

基本使用

复制核心文件到项目内

Smarty 的引入

Smarty 是一个类, 引入过程与普通的类没有区别

1. include,require包含此文件

2. 实例化

基础配置（缓存目录，边界符……）

```
require './libs/Smarty.class.php';
$smarty = new Smarty;
$smarty->template_dir = './dir';//设置模版路径(当前如果有模版,优先找当前目录,没有再找定义的目录)
$smarty->left_delimiter = '<{'; //设置左边界符
$smarty->right_delimiter = '}>';//设置右边界符
$smarty->caching = true;//开启缓存
$smarty->cache_lifetime = 120; //缓存生命周期(秒)
$t = '北京真TM好';
```

```
$smarty->assign('t',$t);
$smarty->display('2.html');
```

基础用法

Smarty 的赋值

```
1: $smarty->assign('key',value);
```

发生的变化 $smarty->_tpl_vars[key] = value;
和我们自己写的类的处理方式类似

2: 如果 assign($arr) 中的第一个参数是数组
则会循环数组,把数组的每一个单元追加到_tpl_vars属性上(后面会讲)

模板中如何引用变量

1.字符串型,数值型, 直接通过 `{标签名}`来引用 例: {$title}

2.数组变量,用 `$标签名.键或$标签名[键]`来引用数组单元
例: `{$stu.name} {$stu[age]}`
`建议使用后者,即中括号语法,兼容性更强`

```
//2.php
$arr = array('a'=>'班长','b'=>'女班长');
$sm->assign('title',$arr);
$sm->display('2.html');
```

```
//2.html
<body>
    <h1><{$title['a']}></h1>
    <h1><{$title.b}></h1>
</body>
```

3.对象
用 `$标签名->属性名` 来引用对象的属性值
用 `$标签名->方法()` 来调用对象的方法的返回值

```
class A{
    public $b='啥叫土豪';
}

$arr = new A();
$sm->assign('title',$arr);
$sm->display('2.html');
```

```
<body>
    <h1><{$title->b}></h1>
    <h1></h1>
</body>
```

4.smarty中的系统变量
有些系统变量,不需要assign(), 即可使用, 引用时以 $smarty 开头
例:
`$smarty.now ,被解析成 time();`
`$smarty.get.key ---> $_GET[key]`
`$smarty.const.常量---> echo 常量名`

```
<h1><{$smarty.get.id}></h1>//记得地址栏传值
<h1><{$smarty.now}></h1>//时间戳,格式化后面讲
```

还有很多,勤查手册

5.从配置文件得到的变量 配置文件可以用来存储常用且很少变的数据,比如网站名,备案号 通过配置文件得到这些信息,不必去读数据库,可以省一些数据库的开销. 配置文件的写法

例:xxx.conf
配置项1=值1
配置项2=值2
配置文件的载入
{Config_load file="xxx.conf"}
引用配置选项的值
建议用后者,即中括号语法,兼容性更强
`{$smarty.config.配置项}或者{#配置项#}`

```
xxx.conf //不用管后缀,就当是配置文件
aa=123
bb=456
```

```
<body>
    <{Config_load file="xxx.conf"}>
    <h1><{$smarty.config.aaa}></h1>
</body>
```

总结:模板中的变量有3种来源
1: assign赋值得到的变量 , 存储在_tpl_vars属性中
2: $smarty系统变量, 对于cookie,session,get,post,$_SERVER等信息,存储在_smarty属性中
Smarty会自动捕捉,并保存起来,形成系统变量,可以直接用标签来引用.
3: 从配置文件读取的变量 , 存储在_config属性中;

定界符冲突的问题
如果smarty用定界符,比如 {,},
此定界在js,css里都有很可能碰到,如果碰到,会当成smarty标签来解析,进而引发错误发
生.
1:换定界符,如 {> <}
2:用literal标签,"原义","字符意义的","无夸张的"
例: {literal}h1{background:gray}{/literal} ---> 一般不用

判断、循环、运算

- 提问: *模板的职责,在于输出数据. 基于此,模板中不应该有if/else,等逻辑相关的东西. 为什么 还要在模板中加逻辑控制的功能?

场景:
某商城,针对vip会员 和 普通会员, 显示不同的效果
如:
VIP: 1.87元
普通: 399.87元
如果模板上没有任何逻辑控制,那么只能在PHP上做逻辑判断.
只能如下方式来完成:

```
// xx.php
if($lev == 'vip') {
    $smarty->display('vip.html');
}else {
    $smarty->display('comm.html');
}
```

判断

```
{if $ss === xxx}
    xxxx
```

```
{else if $xx ===xx}
    xxxxx
{else}
    xxxx
{/if}
```

```
//3.php
require './smarty/Smarty.class.php';
$sm = new Smarty;
$n = mt_rand(4,6);

$sm->assign('n',$n);
$sm->display('3.html');
```

```
<body>
    <h1>
        {if $n == 4 }法律是什么?
        {else if $n == 5}法律就是两边打架不公平
        {else}法律跑来指手画脚
        {/if}

        if(){

        }elseif(){

        }else{

        }
    </h1>
</body>
```

循环

```
//4.php
require './smarty/Smarty.class.php';
$news = array(
    array('id'=>1,'title'=>'计算机','content'=>'根据人体模仿出来的');
);
$sm = new Smarty;
$smarty->assign('news' , $news);
$sm->display('4.html');
```

```
//4.html
{foreach $new as $k=>$v}
    {$v.title}
    {$v.id}
{/foreach}
```

注意:不要用模版里面的foreach,因为性能非常低,而且这不是模版的职责,它只是有这些功能而已

模板中的运算符

```
//4.html
{foreach $new as $k=>$v}
    {$v.title}
    {$v.id *4}
{/foreach}
```

变量调节器及模板编译的特点

可以把标签变量的值,做一些修改.
例: `{$intro|upper}` ,会把 `$info` 的内容转换为大写
原理 : 把 `$intro` 作为参数,传给upper调节器对应的函数,

并显示该函数值,而不是`$intro`;
$intro 标签变量本身会当成调节器的第1个参数自动传入,
如果需要传更多参数,在调节器后面,用' :'隔开更多参数.
例 `{$new s| truncate:7:"..."}`

变量调节器

date_format [格式化日期]
default [默认值]
escape [编码]
indent [缩进]
lower [小写]
nl2br [换行符替换成 <\br />]
replace [替换]
strip [去除(多余空格)]
strip_tags [去除html标签]
upper [大写]
示例:
{$smarty.now|date_format:'%Y-%m-%d %H:%M:%S'}

```
$sm->assign('u','jiJIUHJopoi');
$sm->display('5.html');


{$k|upper} //全部转为大写
{$smarty.now|date_format:'Y-m-d'}
{$smarty.now|date_format:'%Y-%m-%d'}
```

模板编译的特点:
Smarty的一个特点: 先编译成.php文件,再执行该 PHP
Smarty在第一次运行的时候,稍慢,包含了编译模板+包含执行
后面的运行中速度会快,因为直接包含PHP文件.

问: smarty什么时间重新编译模板?
答: 根据模板修改的时时间.

如果想强制编译: 比如在开发过程中
可以force_compile=true 强制编译

display和fetch的区别

display 调用的ftech方法
即: display() = echo fetch();
推测:

```
直接看源码, 找到元凶:
$smarty->display('xx.html'); //输出
$smarty->fetch('xx.html'); //不输出
echo $smarty->fetch('xx.html'); //输出
```

```
// file Smarty.class.php 831行左右
public function display($template = null, $cache_id = null, $compile_id
= null, $parent = null) {
// display template
$this->fetch($template, $cache_id, $compile_id, $parent, true);
}

// 806行左右
public function fetch() {
// ....
return $_template->render(true, false, $display);//在这里只是返回
}
```

```
// file sysplugins/smarty_internal_template.php 302行左右
// display or fetch
if ($display) {
// ...
echo $content;//display在这个地方echo了
return '';
} else {
```

```
// ...
return $content;
}
```

包含子模板

头部尾部的文件引用等，很实用；
{include file='./xxx.html'}

缓存

开启缓存

```
//6.php
$sm->caching = true;//开启缓存
$sm->cache_lifetime = 120; //缓存生命周期(秒)
```

缓存时间是最重要的,因为缓存就是临时存放,没有生命周期不叫缓存,叫长眠

1:首选打开缓存配置项 $smarty->caching = true;
2:缓存生命周期的配置选项: $smarty->cache_lifetime = ๓๓>
3: $smarty->isCached() 判断是否有缓存,如果有缓存,则用缓存，避免数据交互操作

```
$a = 'aaa';
if(!$smarty->isCached('xx.html') ){
    $sm->assign('a',$a);
    echo 1111;
}
$sm->display('x.html');
```

```
if( !$smarty->isCached('xx.html') ) {
    $rs = mysql_query('select * from art limit 3' , $conn);
    $art = mysql_fetch_assoc($rs);
    $smarty->assign('art' , $art);
    echo 'from mysql';
}
$smarty->display('07.html');
```

缓存存放的目录在cache里面

观察两个文件存放的有什么区别.
1个是存放的php代码,因为这个文件是专门用于编译的php文件
另一个存放的是直接的结果.
就是一个负责处理代码,另一个直接存放处理完的结果.
所以要快,就是静态页面,不做处理,直接读

单模板多缓存

1个模板还可以根据其他参数,缓存多个结果,比如商品页面根据商品id来为每个商品缓存一个页面.

布局可能一样的,可能就是商品某个小地方不一样,没必要经过全部编译再输出,而是就编译某个地方. 根据ID只缓存一小部分,或者每一个ID建不同的缓存

原则: 凡是能够影响到页面内容的参数,就要根据此参数影响cached_id
这样,不同参数才能对应不同的缓存内容

```
//7.php
require './smarty/Smarty.class.php';
$sm = new Smarty;
$sm->caching = true;
$sm->cache_lifetime = 3;
```

```
$a = 'aaaa';
$id = $_GET['id'];
if(!$sm->isCached('7.html',$id)){
    $sm->assign('a',$id);

}
$sm->display('7.html',$id);
```

删掉cache的缓存
查看缓存文件的变化,发现有单个的缓存文件,读的时候只读你出现更改的地方,其他模版还是原来的样子

局部缓存

不缓存标签
可以以不缓存的方式assign()变量。
适用于单个标签不缓存。

例:

```
 $smarty->assign('foo',$foo,true); //第三个参数 true 不缓存
```

```
//8.html
{$a}
{$b}
```

```
//8.php
require './smarty/Smarty.class.php';
$sm = new Smarty;
$sm->caching = true;
$sm->cache_lifetime = 3;

$sm->assign('a','aaaaa',true);
$sm->assign('b','bbbbb');
$sm->display('8.html');
```

观察缓存文件,发现有一部分出现了不缓存情况
因为有些情况是这个东西经常变,需要经常编译,所以不需要缓存.

nocache 标签 标签
在模板中,不需要缓存的地方,用 {nocached}{/nocached} 包住.
适用于大段的不缓存效果
html模板
例: {nocache}{$smarty.now}{/nocache}

模板调用PHP函数
模板中用{insert name=xx} 直接调用PHP中的insert_xx的函数

```
//8.php
require './smarty/Smarty.class.php';
$sm = new Smarty;
$sm->caching = true;
$sm->cache_lifetime = 3;

//写一个函数,没有调用,直接去模版中使用
function insert_bb(){
    echo 111111;
}

$sm->assign('a','aaaaa',true);
$sm->assign('b','bbbbb');
$sm->display('8.html');
```

```
$smarty->assign('foo',$foo,true); //第三个参数 true 不缓存
```

```
//8.html
{$a}
{$b}

{insert name=bb}
```

模板引擎之殇

某些编程语言，没有模板不行，但是， PHP可以不用

```
print("<html>")
print("<h1>")
print("你好")
print("</h1>")
print("</html>")
```

这种情况下,python与html混杂在一起.
没有模板支持,前端制作人员不懂python很难工作.
所以需要让前端用模板的形式来写页面,

而python再用专门的一个类去处理模板,并输出.
常用的python模板类有
jinja2， cheetah， mako， webpy， bottle， tornado
思考: 相比python,为什么php可以不用模板?

php本身就是一种标签语言

php代码可以嵌套在html中
混杂在H TML代码中smarty标签和PH P代码.
对于web前端开发人员来说,
本质上

```
<p>{$t}</p>
和
<p><?php echo $t;?></p>
```

对于前端人员，没有本质的区别,
但是，性能却降低了；
又但是，缓存还是有价值的
模板的副作用：
解析编译，消耗性能
增多了很多变量 页面内的变量,都要赋值到smarty对象-->_tpl_vars 属性上, 多了一个变

还有什么爱你的理由呢?

- 缓存(但有很多更高效的缓存工具,如memcached)
- 面试会问到,有些古老的项目还会用到.
- 一些框架(TP,laravel),也有自己的模板引擎. 和Smarty的模板语法触类旁通,降低学习成本

MVC和smarty的关系

MVC只是大家写代码习惯了以后的一种分工方式,也可以叫潜规则.
model 数据处理层
controller 业务逻辑层
view 视图展示层

```
function add($a,$b){
    return $a+$b;
}
```

```php
$a = $_GET['a'];
$b = $_GET['b'];
if(!empty($a) && !empty($b)){//C层业务逻辑
    $res = add($a,$b);//M层 数据处理
}else{
    $res = '无法运算';
}

echo '<html><h1>'.$res.'</h1></html>';//v展示层
```

MVC和smarty到底是个什么关系？

雷锋和雷峰塔的关系.只是比较像而已,或者涵盖而已.或者只是用到smarty里面的一些东西而已.