

28. 模版中函数和运算符

学习要点：

1. 使用函数
2. 运算符

本节课我们来重点学习一下模版中使用函数的方法以及运算符的使用。

一. 使用函数

1. 控制器端先赋值一个密码的变量，模版区设置 md5 加密操作；

```
$this->assign('password', '123456');  
{ $password|md5 }
```
2. 系统默认在编译的会采用 `htmlentities` 过滤函数防止 XSS 跨站脚本攻击；
3. 如果你想更换一个过滤函数，比如 `htmlspecialchars`，可以在配置文件设置；
4. 具体在 `config` 下的 `template.php` 中，增加一条如下配置即可；

```
'default_filter' => 'htmlspecialchars'
```
5. 如果在某个字符，你不需要进行 HTML 实体转义的话，可以单独使用 `raw` 处理；

```
{ $user['email']|raw }
```
6. 系统还提供了一些固定的过滤方法，如下：

函数	说明
<code>date</code>	格式化时间 <code>{ \$time date='Y-m-d' }</code>
<code>format</code>	格式化字符串 <code>{ \$number format='%x' }</code>
<code>upper</code>	转换为大写
<code>lower</code>	转换为小写
<code>first</code>	输出数组的第一个元素
<code>last</code>	输出数组的最后一个元素
<code>default</code>	默认值
<code>raw</code>	不使用转义

```
$this->assign('time', time());  
{ $time|date='Y-m-d' }
```

```
$this->assign('number', '14');  
{ $number|format='%x' }
```

7. 如果函数中，需要多个参数调用，直接用逗号隔开即可；

```
{ $name|substr=0,3 }
```

8. 在模版中也支持多个函数进行操作，用|号隔开即可，函数从左到右依次执行；
`{ $password|md5|upper|substr=0,3 }`
9. 你也可以在模版中直接使用 PHP 的语法模式，该方法不会使用过滤转义：
`{ :substr(strtoupper(md5($password)), 0, 3) }`

二. 运算符

1. 在模版中的运算符有+、-、*、/、%、++、--等；
`{ $number + $number }`
2. 如果模版中有运算符，则函数方法则不再支持；
`{ $number + $number|default='没有值' }`
3. 模版也可以实现三元运算，包括其它写法：

<code>{ \$name ? '正确' : '错误' }</code>	// \$name 为 true 返回正确，否则返回错误
<code>{ \$name ?= '真' }</code>	// \$name 为 true 返回真
<code>{ \$Think.get.name ?? '不存在' }</code>	// ?? 用于系统变量，没有值时输出
<code>{ \$name ?: '不存在' }</code>	// ?: 用于普通变量，没有值时输出
4. 三元运算符也支持运算后返回布尔值判断；
`{ $a == $b ? '真' : '假' }`