

66. 模版的加载包含输出

学习要点：

1. 包含文件
2. 输出替换
3. 文件加载

本节课我们来学习一下模版标签中的文件的包含、输出以及加载。

一. 包含文件

1. 使用`{include}`标签来加载公用重复的文件，比如头部、尾部和导航部分；
2. 在模版 `view` 目录创建一个 `public` 公共目录，分别创建 `header`、`footer` 和 `nav`；
3. 然后创建 `Block` 控制器，引入控制器模版 `index`，这个模版包含三个公用文件；

```
{include file='public/header,public/nav'/{}}
```

4. 也可以包含一个文件的完整路径，包括后缀，如下：

```
{include file="../../../view/public/nav.html"/{}}
```

5. 模版的标题和关键字，可以通过固定的语法进行传递；
6. 对于标题，在控制器先设置一下标题变量，然后设置`{include}`设置属性；

```
{include file='public/header'
        title='$title' keywords='关键字'/{}}
```

7. 切换到 `public/header.html` 模版页面，使用`[xxx]`的方式调用数据；

```
<title>[title]</title>
<meta name="keywords" content="[keywords]" />
```

二. 输出替换

1. 有时，我们需要调用一些静态文件，比如 `css/js` 等；
2. 那么，直接写完整路径，比较繁长，可以把这些路径整理打包；
3. 在目前二级目录下，`view.php` 中，配置新增一个参数；

// 模版替换输出

```
'tpl_replace_string' => [
    '__JS__' => '../static/js',
    '__CSS__' => '../static/css',
]
```

4. `html` 文件调用端，直接通过`__CSS__(__JS__)`配置的魔术方法调用即可；

```
<link rel="stylesheet" type="text/css" href="__CSS__/basic.css">
<script type="text/javascript" src="__JS__/basic.js"></script>
```

5. 在测试的时候，由于是更改的配置文件刷新，每次都要删除编译文件才能生效；

三. 文件加载

1. 传统方式调用 CSS 或 JS 文件时，采用 `link` 和 `script` 标签实现；
2. 系统提供了更加智能的加载方式，方便加载 CSS 和 JS 等文件；
3. 使用 `{load}` 标签和 `href` 属性来链接，不需要设置任何其它参数；

```
{load href='__CSS__/basic.css' /}  
{load href='__JS__/basic.js' /}
```

2. 也支持 `href` 多属性值的写法，如下：

```
{load href='__CSS__/basic.css, __JS__/basic.js'}
```

3. `{load}` 还提供了两个别名 `{js}`、`{css}` 来更好的实现可读性；

```
{js href='__JS__/basic.js'}  
{css href='__CSS__/basic.css'}
```

4. `{js}` 和 `{css}` 只是别名而已，识别 `.js` 还是 `.css` 是根据后缀的；