



福州大学至诚学院
FUZHOU UNIVERSITY ZHICHENG COLLEGE

服务端开发技术



6. TP5视图

01

视图实例化方法

02

模板变量赋值方法

03

模板的渲染方法

04

模板内容替换

05

模板布局和模板继承

06

循环标签输出大量数据

Part.1

6.1 视图实例化方法

6.1 视图实例化方法

在控制器中使用视图来调用模板有三种方法：

第一是实例化视图类的规范模式

第二是基类继承的经典模式

第三是助手函数的快捷模式

6.1 视图实例化方法

直接实例化视图类

- 引用视图类

Use `think\View`;

- 实例化视图

`new View()`; // (用new关键字) 直接创建视图对象:

`View::instance()`; //instance调用静态方法

6.1 视图实例化方法

调用视图的方法

(1) **fetch方法**，是用来渲染模板的，**需要一个模板文件。**

`fetch()` //渲染模板

(2) **display方法**，可以**不创建模板文件，直接渲染内容。**

`display()` //渲染内容

(3) **assign方法**，**给当前操作对应的模板文件创建变量，并赋值。**

`assign()` //模板赋值

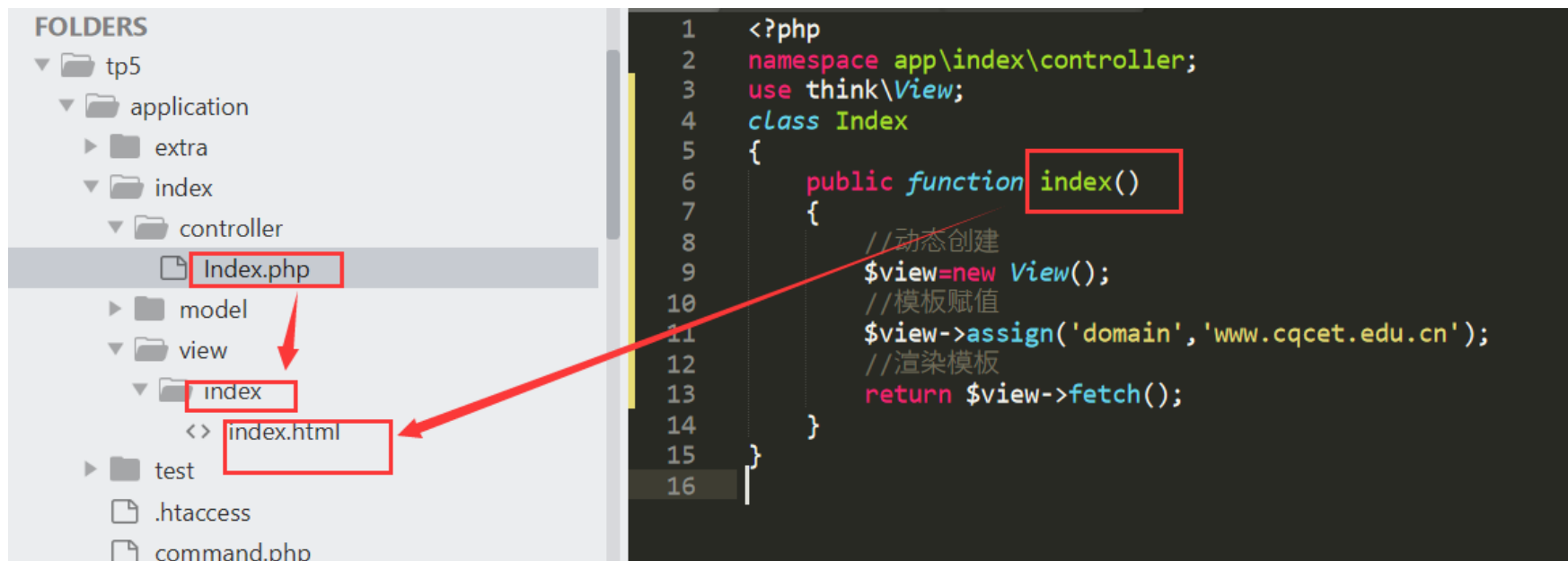
6.1 视图实例化方法

动态创建实例化视图类

application > index > controller > Index.php

```
1  <?php
2  namespace app\index\controller;
3  use think\View;
4
5  class Index
6  {
7      public function index()
8      {
9          //动态创建
10         $view = new View();
11         //模板赋值
12         $view->assign('domain', 'www.fdzcxy.edu.cn');
13         //渲染模板
14         return $view->fetch()
15     }
16 }
```


6.1 视图实例化方法



FOLDERS

- tp5
 - application
 - extra
 - index
 - controller
 - Index.php**
 - model
 - view
 - index**
 - index.html**
 - test
 - .htaccess
 - command.php

```
1 <?php
2 namespace app\index\controller;
3 use think\View;
4 class Index
5 {
6     public function index()
7     {
8         //动态创建
9         $view=new View();
10        //模板赋值
11        $view->assign('domain','www.cqcet.edu.cn');
12        //渲染模板
13        return $view->fetch();
14    }
15 }
16
```

application > index > view > index > <> index.html > ...

```
1 <p>网站域名:<span style="color:rgb(241, 2, 14)">{$domain}</span></p> <!--输出红色字体-->
```

变量名以\$开始，并添加一个开始标记和结束标记（默认是{}）



6.1 视图实例化方法

静态创建实例化视图类

application > index > controller > Index.php

```
1  <?php
2  namespace app\index\controller;
3  use think\View;
4
5  class Index
6  {
7      public function index()
8      {
9          //动态创建
10         $view = View::instance();
11         //模板赋值
12         $view->assign('domain', 'www.fdzcxxy.edu.cn');
13         //渲染模板
14         return $view->fetch();
15     }
16 }
```

6.1 视图实例化方法

继承控制器创建视图对象

```
Use think\Controller;  
class Index extends Controller
```

application > index > controller > Index.php

```
1  <?php  
2  namespace app\index\controller;  
3  use think\Controller;  
4  
5  class Index extends Controller  
6  {  
7      public function index()  
8      {  
9          $this->view->assign('siteNickname', '福州大学至诚学院');  
10         $this->view->assign('domain', 'www.fdzcxxy.edu.cn');  
11         return $this->view->fetch();  
12     }  
13 }
```

6.1 视图实例化方法

<p>网站名称:{\$siteNickname}</p>

<p>网站域名:{\$domain}</p>

← → ↻ ⚠ 不安全 | tp5.com/index.php/index/index/index

网站名称:福州大学至诚学院

网站域名:www.fdzcxy.edu.cn

6.1 视图实例化方法

简化代码

去掉view

application > index > controller > Index.php

```
1  <?php
2  namespace app\index\controller;
3  use think\Controller;
4
5  class Index extends Controller
6  {
7      public function index()
8      {
9          $this->assign('siteNickname', '福州大学至诚学院');
10         $this->assign('domain', 'www.fdzcxy.edu.cn');
11         return $this->fetch();
12     }
13 }
```

Part.2

6.2 模板变量赋值方法

6.2 模板变量赋值方法

assign方法

```
application > index > controller > Index.php
1  <?php
2  namespace app\index\controller;
3  use think\Controller;
4
5  class Index extends Controller
6  {
7      public function index()
8      {
9          //1. 单独赋值
10         $this->assign('siteNickname', '福州大学至诚学院');
11         //2. 批量赋值
12         $this->assign([
13             'nickname'=>'计算机工程系',
14             'lesson'=>'php'
15         ]);
16         //3. 渲染模板
17         return $this->fetch();
18     }
19 }
```

6.2 模板变量赋值方法

```
<p>网站名称:<span style="color:red">{$siteNickname}</span></p>
```

```
<p>我的名字:<span style="color:red">{$nickname}</span></p>
```

```
<p>我学习的课程:<span style="color:red">{$lesson}</span></p>
```

← → ↻ ⚠ 不安全 | tp5.com/index.php/index/index/index

网站名称:福州大学至诚学院

我的名字:计算机工程系

我学习的课程:php

6.2 模板变量赋值方法

通过fetch()或者display()方法传参给模板赋值

fetch()和display()的区别：

- **fetch()方法需要一个模板文件， display()是不需要的。**
- **fetch()方法的第一个参数是模板文件。 display()的第一个参数是要输出的内容， 第二个参数就是模板变量数组。**

6.2 模板变量赋值方法

application > index > controller > Index.php

```
1  <?php
2  namespace app\index\controller;
3  use think\Controller;
4
5  class Index extends Controller
6  {
7      public function index()
8      {
9          //单独赋值
10         $this->assign('siteNickname', '福州大学至诚学院');
11
12         //渲染模板
13         return $this->fetch('index', [
14             'nickname'=>'计算机工程系',
15             'lesson'=>'php'
16         ]);
17     }
18 }
```

← → ↻ ⚠ 不安全 | tp5.com/index.php/index/index/index

网站名称:福州大学至诚学院

我的名字:计算机工程系

我学习的课程:php

6.2 模板变量赋值方法

display方法的第一个参数不是模板，而是要渲染的内容，

application > index > controller > Index.php

```
1  <?php
2  namespace app\index\controller;
3  use think\Controller;
4
5  class Index extends Controller
6  {
7      public function index()
8      {
9          //单独赋值
10         $this->assign('siteNickname', '福州大学至诚学院');
11
12         return $this->display('姓名:{$nickname}, 学习的课程:{$lesson}', [
13             'nickname'=>'计算机工程系',
14             'lesson'=>'php'
15         ]);
16     }
17 }
```

← → ↻ ⚠ 不安全 | tp5.com/index.php/index/index/index

姓名:计算机工程系, 学习的课程:php

6.2 模板变量赋值方法

用助手函数view给模板赋值

application > index > controller > Index.php

```
1  <?php
2  namespace app\index\controller;
3
4  class Index
5  {
6      public function index()
7      {
8          return view('index', [
9              'nickname'=>'计算机工程系',
10             'lesson'=>'php',
11             'siteNickname'=>'福州大学至诚学院'
12         ]);
13     }
14 }
```

← → ↻ ⚠ 不安全 | tp5.com/index.php/index/index/index

网站名称:福州大学至诚学院

我的名字:计算机工程系

我学习的课程:php

Part.3

6.3 模板的渲染方法

6.3 模板的渲染方法

模板文件的对应关系

1. 不带参数调用

The diagram illustrates the relationship between a PHP controller file and a template file in a ThinkPHP 5 project. On the left, a file explorer shows the project structure. On the right, a code editor shows the contents of the `Index.php` file. Red arrows indicate the mapping between the code and the file structure.

FOLDERS

- tp5
 - application
 - extra
 - index
 - controller
 - Index.php** (linked to line 6 of the code)
 - model
 - view
 - index
 - index-1.html (linked to line 14 of the code)
 - index.html (linked to line 12 of the code)
 - test
 - .htaccess

```
1 <?php
2 namespace app\index\controller;
3 //class Index extends \think\Controller
4
5 //直接使用视图类
6 class Index
7 {
8     public function index()
9     {
10         return $this->fetch();
11     }
12 }
13
14
```

6.3 模板的渲染方法

2.跨控制器调用

The image shows a code editor with a file explorer on the left and a PHP code snippet on the right. Red arrows point from the code to the file explorer, illustrating the cross-controller call.

FOLDERS

- tp5
 - application
 - extra
 - index
 - controller
 - Index.php
 - User.php
 - model
 - view
 - index
 - index-1.html
 - index.html
 - User
 - index.html
 - test

Code Snippet:

```
1 <?php
2 namespace app\index\controller;
3 //class Index extends \think\Controller
4
5 //直接使用视图类
6 class Index
7 {
8
9     public function index()
10    {
11
12        //1. 不带参数的调用
13        // return $this->fetch();
14        // 2. 跨控制器调用
15        return $this->fetch('user/index');
16
17    }
18
19 }
20
21
```

Red arrows indicate the following relationships:

- From line 6 (`class Index`) to `Index.php` in the `controller` folder.
- From line 15 (`return $this->fetch('user/index');`) to `index.html` in the `view/index/User` folder.
- From line 10 (`public function index()`) to `User.php` in the `controller` folder.

6.3 模板的渲染方法

3.跨模块调用

```
$this->fetch('admin@user/index');
```

//模块名和控制名之间通过@符号进行分隔

4.完整的模板文件名

```
$this->fetch('/public/tpl/user/index.index.html');
```

//绝对地址访问

注意：只要访问的模板不是在视图(view)目录下，就必须采用完整的模板文件名来访问。

Part.4

6.4 模板内容替换

6.4 模板内容替换

替换内容

在框架中预定义五个可以替换的常量：

- **__ROOT__**：对应目录是/。
- **__URL__**：对应目录是module/controller。
- **__STATIC__**：对应目录是__ROOT__ /static/。
- **__CSS__**：对应目录是__ROOT__ /static/css。
- **__JS__**：对应目录是__ROOT__ /static/js。

6.4 模板内容替换

替换方式

(1) 单独替换

通过`fetch()`方法或者助手函数`view()`对当前模板中的字符串进行替换。

```
return $this->fetch('index',[],['__PUBLIC'=>'/public/']);
```

```
return view('index',[],['__PUBLIC'=>'/public/']);
```

6.4 模板内容替换

(2) 全局替换

在应用配置文件config.php中添加一个`view_replace_str`配置项来进行全局替换，这样所有的模板就都可以用了。

```
'view_replace_str'=>[  
    '__PUBLIC__'=>'/public/',  
    '__ROOT__'=> '/',  
],
```

6.4 模板内容替换

简单替换

application > index > controller > Index.php

```
1  <?php
2  namespace app\index\controller;
3  use think\Controller;
4  class Index extends Controller
5  {
6      public function index()
7      {
8          //使用$this->fetch()替换
9          //第一个参数: 模板文件
10         //第二个参数: 模板变量(不能不写, 没有就为空数组)
11         //第三个参数: 替换的规则
12         return $this->fetch('', [], [
13             'www.fdzcxy.edu.cn'=>'福州大学至诚学院'
14         ]);
15     }
16 }
```

6.4 模板内容替换

```
application > index > view > index > <> index.html > ...
```

```
1 <p>我是Index控制器index操作的模板文件</p>  
2 <p>www.fdzcxy.edu.cn</p>
```

← → ↻ ⚠ 不安全 | tp5.com/index.php/index/index/index

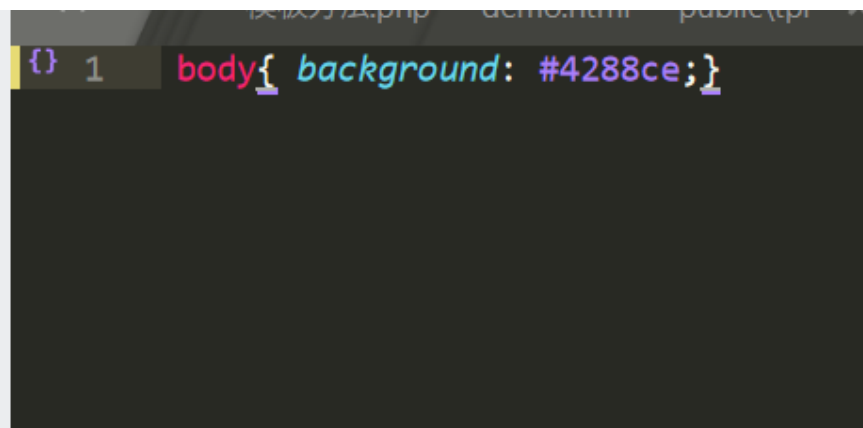
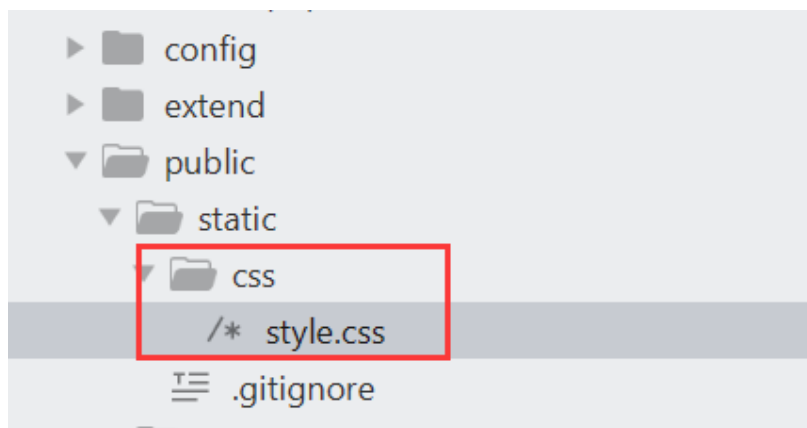
我是Index控制器index操作的模板文件

福州大学至诚学院

这样的简单文字替换并不是它的主要用途，主要用途还是对一些静态资源的替换。

6.4 模板内容替换

用fetch实现静态的替换



6.4 模板内容替换

```
<link rel="stylesheet" href="/static/css/style.css">
```

```
<p>我是Index控制器index操作的模板文件</p>
```

```
<p>www.fdzcxu.edu.cn</p>
```

← → ↻ ⚠ 不安全 | tp5.com/index.php/index/index/index

我是Index控制器index操作的模板文件

福州大学至诚学院

6.4 模板内容替换

application > index > controller > Index.php

```
1  <?php
2  namespace app\index\controller;
3  use think\Controller;
4  class Index extends Controller
5  {
6      public function index()
7      {
8          //使用$this->fetch()替换
9          //第一个参数: 模板文件
10         //第二个参数: 模板变量(不能不写, 没有就为空数组)
11         //第三个参数: 替换的规则
12         return $this->fetch('', [], [
13             'www.fdzcxxy.edu.cn'=>'福州大学至诚学院',
14             '__CSS__'=>'/static/css'
15         ]);
16     }
17 }
```

6.4 模板内容替换

```
<link rel="stylesheet" href="__CSS__/style.css">  
<p>我是Index控制器index操作的模板文件</p>  
<p>www.fdzcxxy.edu.cn</p>
```

← → ↻ ⚠ 不安全 | tp5.com/index.php/index/index/index

我是Index控制器index操作的模板文件

福州大学至诚学院

6.4 模板内容替换

使用助手函数替换

使用助手函数将前面的\$this->fetch()改成view()

```
application > index > controller > Index.php
1  <?php
2  namespace app\index\controller;
3  use think\Controller;
4  class Index extends Controller
5  {
6      public function index()
7      {
8          //使用view()替换
9          //第一个参数: 模板文件
10         //第二个参数: 模板变量(不能不写, 没有就为空数组)
11         //第三个参数: 替换的规则
12         return view('', [], [
13             'www.fdzcxy.edu.cn'=>'福州大学至诚学院',
14             '__CSS__'=>'/static/css'
15         ]);
16     }
17 }
```

Part.5

6.5 模板布局和模板继承

6.5 模板布局和模板继承

公共模板

在很多页面中都有一些共同的部分，**比如头部、尾部和侧边栏等**。可以将这些剥离出来，制作成公共文件，在需要的时候直接导入，实现代码的共享，无论是创建新页面还是更新，都非常方便，这样的公共文件叫作公共模板文件，在ThinkPHP 5中是用模板布局来解决的。

6.5 模板布局和模板继承

建立index.html模板文件

```
<h3 style="background: lightskyblue">我是页面的页头header</h3>
```

```
<p>我是Index控制器index操作的模板文件</p>
```

```
<h3 style="background: lightskyblue">我是页面的尾部footer</h3>
```

6.5 模板布局和模板继承

控制器代码

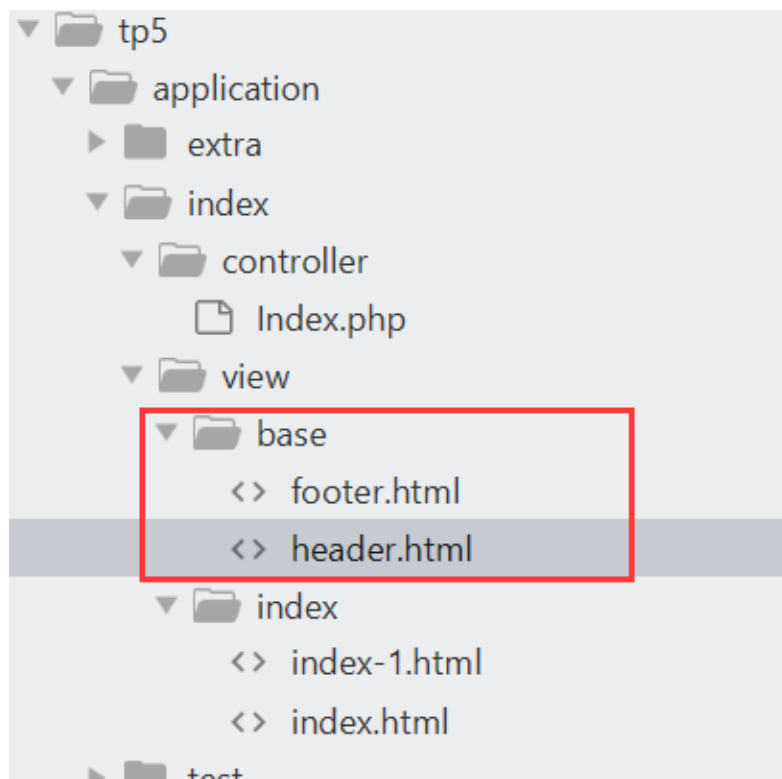
```
<?php
namespace app\index\controller;
class Index extends \think\Controller
{
    public function index()
    {
        return $this->fetch();
    }
}
```



6.5 模板布局和模板继承

分离公共的头部与尾部文件

在视图文件夹view下面创建公共模板目录base，然后建立两个文件，并在两个文件中写代码



6.5 模板布局和模板继承

header.html

```
<h3 style="background: lightskyblue">我是页面的页头header</h3>
```

footer.html

```
<h3 style="background: lightskyblue">我是页面的尾部footer</h3>
```

6.5 模板布局和模板继承

index.html

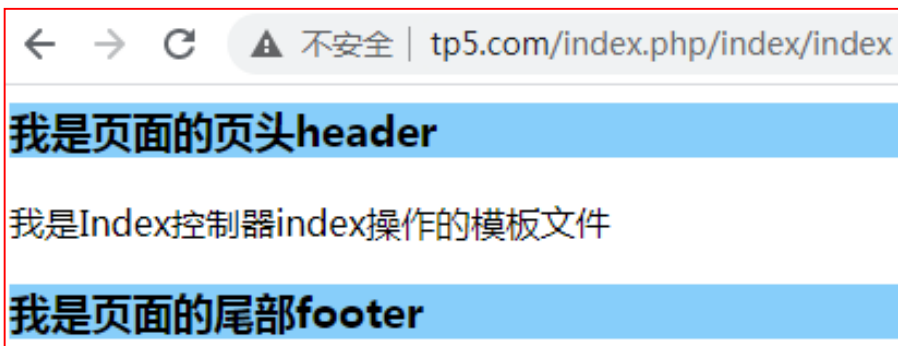
```
{include file="base/header" /}
```

<!--引用头部文件-->

```
<p>我是Index控制器index操作的模板文件</p>
```

```
{include file="base/footer" /}
```

<!--引用尾部文件-->



6.5 模板布局和模板继承

全局配置

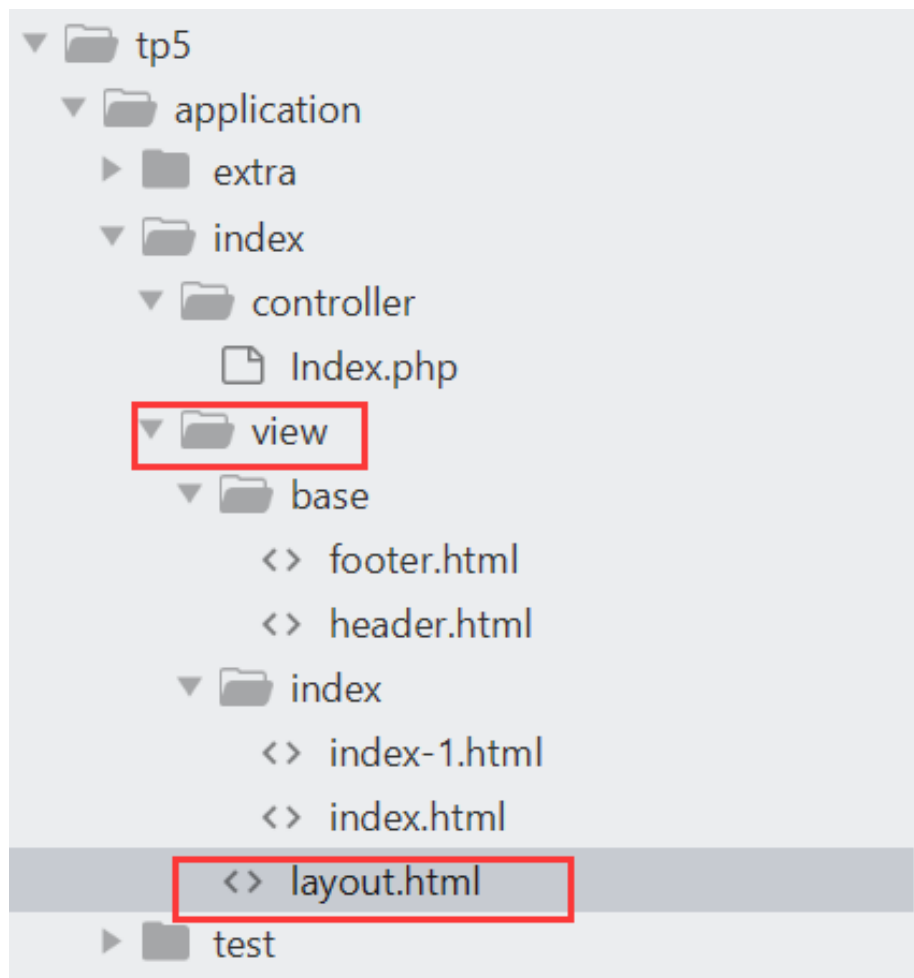
全局配置是通过在配置文件中添加一个配置项`template`来实现的。

`config.php`:

```
config > config.php
1  <?php
2  return [
3      'template'=>[
4          'layout_on'=>true,
5          'layout_name'=>'layout'
6      ],
7  ];
```

6.5 模板布局和模板继承

layout.html



6.5 模板布局和模板继承

layout.html

```
{include file="base/header" /}
```

```
{__CONTENT__}
```

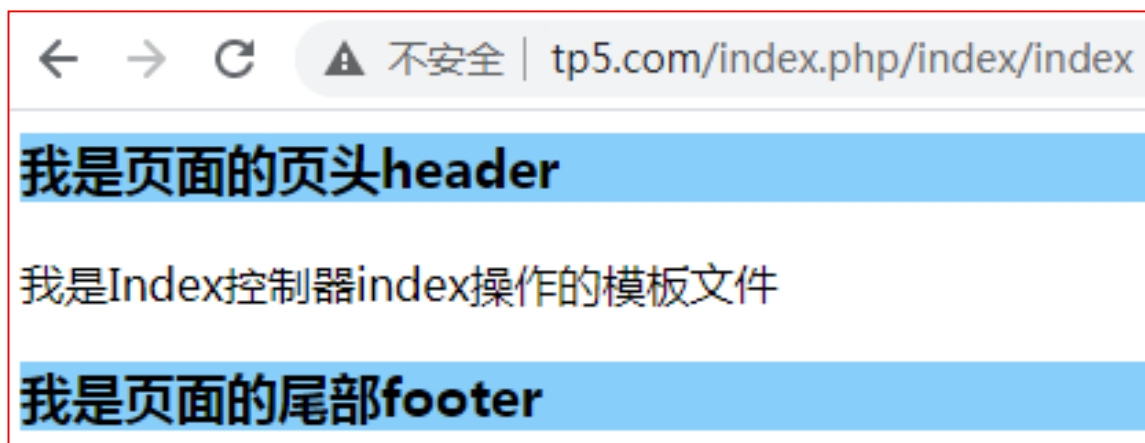
```
{include file="base/footer" /}
```

其中， **{__CONTENT__}** 是要被替换的内容。

6.5 模板布局和模板继承

index.html

```
<p>我是Index控制器index操作的模板文件</p>
```



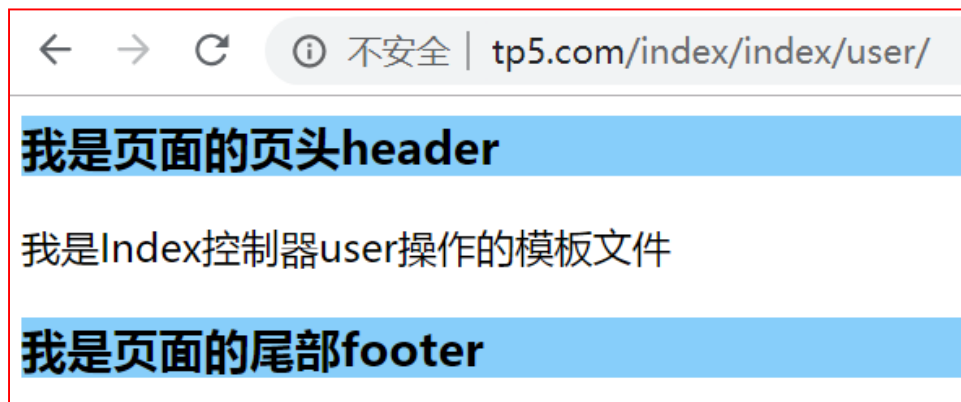
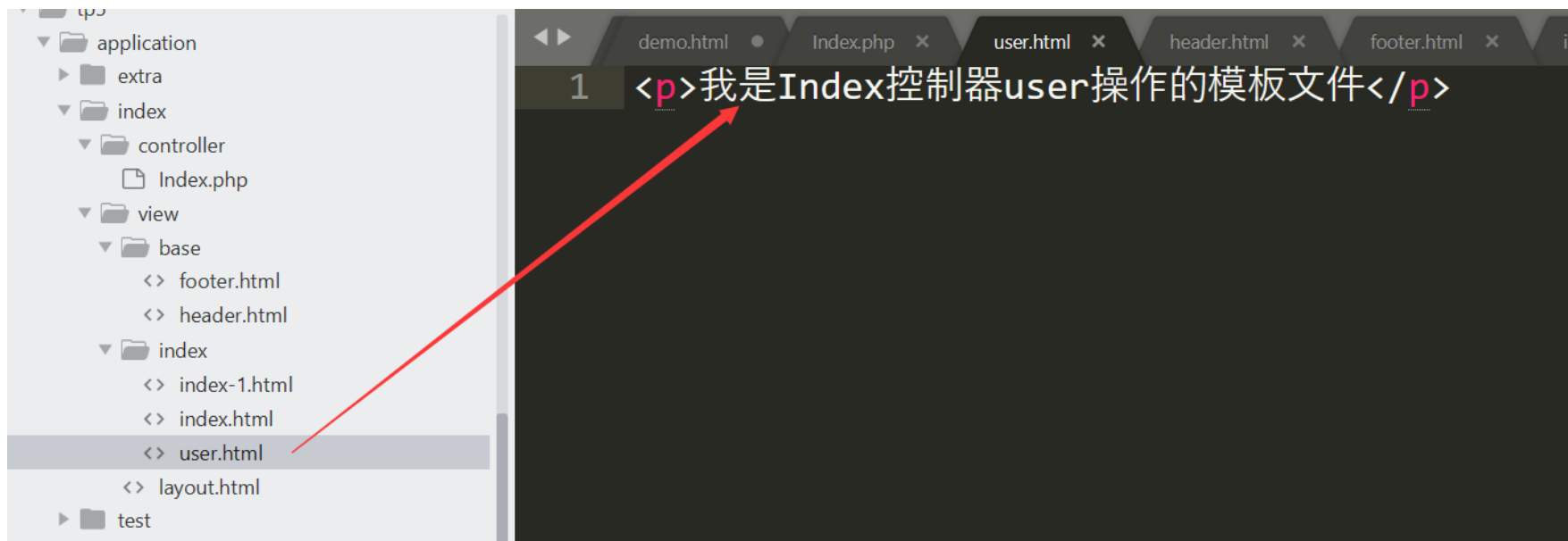
访问流程：从控制器中渲染模板的时候，并不是直接渲染index.html模板文件，而是先渲染布局layout.html文件，将公共文件header和footer导入，然后把index.html里面的内容替换为layout.html文件中的字符串{__CONTENT__}，从而形成一个模板输出。

6.5 模板布局和模板继承

创建一个user操作

```
<?php
namespace app\index\controller;
class Index extends \think\Controller
{
    public function index()                //渲染模板
    {
        return $this->fetch();
    }
    public function user()
    {
        return $this->fetch();
    }
}
```

6.5 模板布局和模板继承



6.5 模板布局和模板继承

模板继承

语法如下：

```
{block} 内容 {/block}
```

- (1) 模板继承是在父模板中用block标签来为子模板的内容预留位置
- (2) 子模板中必须将父模板中预留的位置(区块)全部用代码来实现
- (3) 在子模板中将block区块的内容部分留空的话，将会删除对应父模板中的区域
- (4) 区块中可以使用模板变量，也可以导入外部文件
- (5) 子模板中可以用__block__来引用父模板中原区块的内容

6.5 模板布局和模板继承

模板继承的原理

- (1) 制作一个父模板，假定当前父模板的名称是base.html。在父模板中最典型的结构如下：

```
{include file='header' /}  
  
{block name='区块1'} 内容 {/block}  
  
{block name='区块2'} 内容 {/block}  
  
.....  
  
{include file='footer' /}
```

6.5 模板布局和模板继承

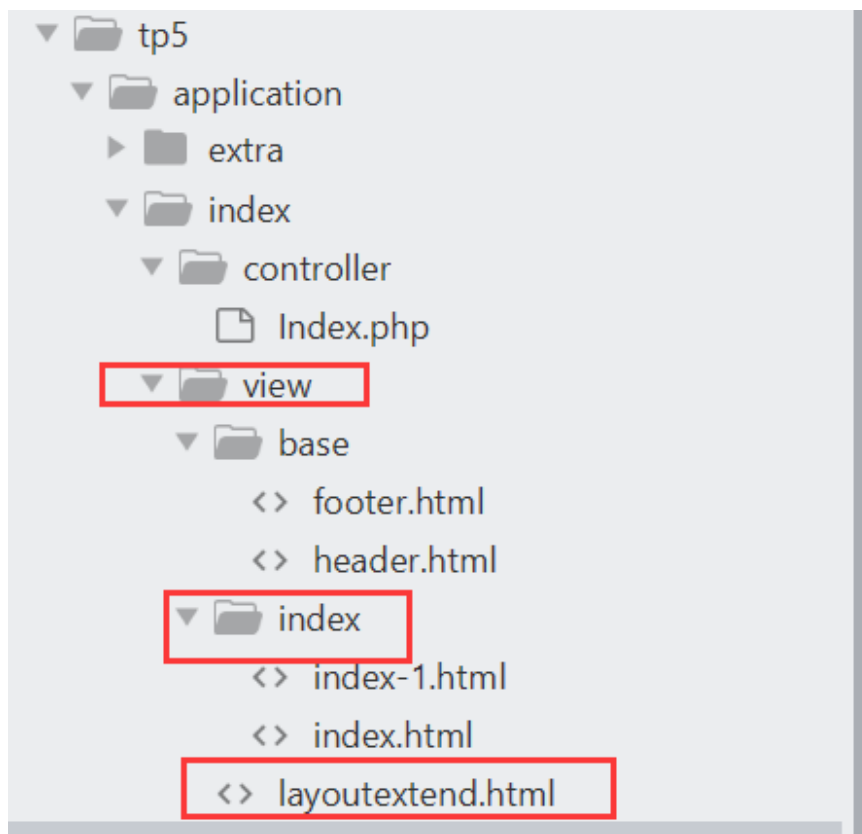
(2) 制作子模板，并在子模板中继承。子模板的结构如下：

```
{extend name=" base" /}  
  
{block name='区块1'} 代码 {/block}  
  
{block name='区块2'} 代码 {/block}  
  
.....
```

在子模板中必须要用`extend`标签来指明当前子模板继承的是哪一个父模板

6.5 模板布局和模板继承

父模板



6.5 模板布局和模板继承

layoutextend.html

```
<!doctype html>
<html lang="zh">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
  {block name="nav"}导航{/block}
  {block name="info"}信息列表{/block}
  {block name="link"}友情链接{/block}
</body>
</html>
```


6.5 模板布局和模板继承

index.html

← → ↻ ⚠ 不安全 | tp5.com/index.php/index/index/index

- [首页](#) [关于](#) [公司新闻](#) [联系](#)
- 欢迎来到福州大学至诚学院
- 欢迎来到福州大学至诚学院
- 欢迎来到福州大学至诚学院
- 欢迎来到福州大学至诚学院

[163官网](#) [thinkphp官网](#) [百度](#)

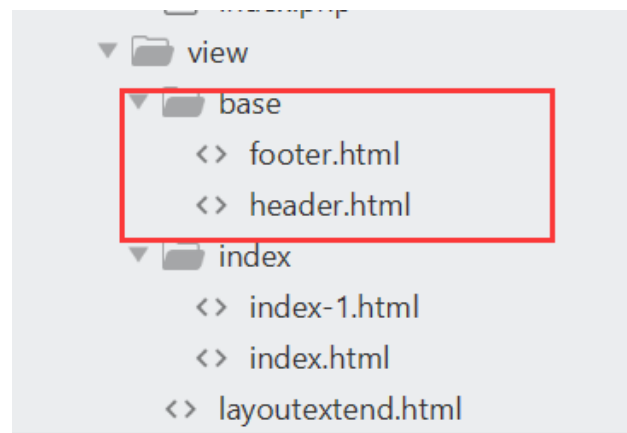
```
1 {extend name="layoutextend"}
2 {block name="nav"}
3 <ul>
4     <li>
5         <a href="">首页</a>
6         <a href="">关于</a>
7         <a href="">公司新闻</a>
8         <a href="">联系</a>
9     </li>
10 </ul>
11 {/block}
12 {block name="info"}
13 <ul>
14     <li>欢迎来到福州大学至诚学院</li>
15     <li>欢迎来到福州大学至诚学院</li>
16     <li>欢迎来到福州大学至诚学院</li>
17     <li>欢迎来到福州大学至诚学院</li>
18 </ul>
19 {/block}
20 {block name="link"}
21     <a href="https://www.163.com">163官网</a>
22     <a href="http://www.thinkphp.cn/">thinkphp官网</a>
23     <a href="https://www.baidu.com">百度</a>
24 {/block}
```

6.5 模板布局和模板继承

分离父模板

父模板中的公共部分可能会在很多地方用到，所以要分离父模板的头部和尾部。

创建公共的模板文件目录base，并在base中创建两个文件，一个是头部文件header.html，一个是尾部文件footer.html



6.5 模板布局和模板继承

header.html

```
<!doctype html>
<html lang="zh">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
```

6.5 模板布局和模板继承

footer.html

```
</body>  
</html>
```

layoutextend.html

```
{include file="base/header" /}  
  {block name="nav"}导航{/block}  
  {block name="info"}信息列表{/block}  
  {block name="link"}友情链接{/block}  
{include file="base/footer" /}
```

← → ↻ ⚠ 不安全 | tp5.com/index.php/index/index/index

- [首页](#) [关于](#) [公司新闻](#) [联系](#)
- 欢迎来到福州大学至诚学院
- 欢迎来到福州大学至诚学院
- 欢迎来到福州大学至诚学院
- 欢迎来到福州大学至诚学院

[163官网](#) [thinkphp官网](#) [百度](#)

Part.6

6.6 循环标签输出大量数据

6.6 循环标签输出大量数据

循环标签

volist标签

语法如下：

```
{volist name="模板变量" id="临时变量" offset="索引"  
length="数量" key="循环变量" empty="提示信息"}
```

6.6 循环标签输出大量数据

foreach标签

语法如下：

```
{foreach name="模板变量" item="临时变量"}
```

for标签

```
{for start="开始值" end="结束值" comparison="比较关系"  
step="步进值" name="循环变量名"}
```

6.6 循环标签输出大量数据

volist循环

```
<?php
namespace app\index\controller;
class Index extends \think\Controller
{
    public function index()
    {
        $user = [];
        for ($key = 0; $key < 10; $key++)
        {
            $user[]=[
                'nickname'=>'陈小明'.$key."号",
                'sex'=>$key ? "男":"女",
                'age'=>rand(15,40),
                'salary'=>rand(3200,6800)
            ];
        }
        //注意：user是二维数组
        return $this->view->fetch('', ['user'=>$user]);
    }
}
```


6.6 循环标签输出大量数据

```
<table border="1" cellspacing="0" cellpadding="2" width="40%" align="center">
  <caption style="font-size: 22px;font-weight: bold">员工信息表</caption>
  <tr style="background: lightskyblue">
    <th>编号</th>
    <th>姓名</th>
    <th>性别</th>
    <th>年龄</th>
    <th>工资</th>
  </tr>
  <!--volist循环-->
  {volist name="user" id="vo"}
  <tr>
    //循环变量的第一个是$key, 第二个是姓名$vo.name, 第三个是性别, 第四个是年龄, 第五个是工资
    <td>{$key}</td>
    <td>{$vo.nickname}</td>
    <td>{$vo.sex}</td>
    <td>{$vo.age}</td>
    <td>{$vo.salary}</td>
  </tr>
{/volist}
</table>
```

6.6 循环标签输出大量数据

ⓘ 不安全 | tp5.com/index/index/index

员工信息表

编号	姓名	性别	年龄	工资
0	陈小明0号	女	27	4734
1	陈小明1号	男	36	4285
2	陈小明2号	男	15	4893
3	陈小明3号	男	29	6147
4	陈小明4号	男	29	3697
5	陈小明5号	男	24	3829
6	陈小明6号	男	16	4909
7	陈小明7号	男	21	5856
8	陈小明8号	男	23	5073
9	陈小明9号	男	20	4723

6.6 循环标签输出大量数据

foreach循环

```
<table border="1" cellspacing="0" cellpadding="2" width="40%" align="center">
  <caption style="font-size: 22px;font-weight: bold">员工信息表</caption>
  <tr style="background: lightskyblue">
    <th>编号</th>
    <th>姓名</th>
    <th>性别</th>
    <th>年龄</th>
    <th>工资</th>
  </tr>
  <!--foreach循环-->
  {foreach name="user" item="value"}           //复制循环体
  <tr>
    <td>{$key}</td>
    <td>{$value.nickname}</td>
    <td>{$value.sex}</td>
    <td>{$value.age}</td>
    <td>{$value.salary}</td>
  </tr>
  {/foreach}
</table>
```

6.6 循环标签输出大量数据

tp5.com/index/index/index

员工信息表

编号	姓名	性别	年龄	工资
0	陈小明0号	女	16	4508
1	陈小明1号	男	18	3617
2	陈小明2号	男	18	3938
3	陈小明3号	男	37	6662
4	陈小明4号	男	29	4005
5	陈小明5号	男	17	5330
6	陈小明6号	男	17	3846
7	陈小明7号	男	23	5634
8	陈小明8号	男	33	4975
9	陈小明9号	男	25	5066