

59. 模版引擎驱动和赋值变量

学习要点：

1. 引擎驱动
2. 赋值变量

本节课我们来学习视图中的模板引擎驱动安装，和赋值变量的用法。

一. 引擎驱动

1. MVC 中，M(模型)和 C(控制器)是前面我们所了解的内容，而 V(视图)就是页面；
2. TP6.0 默认不自带 TT 模版引擎，它作为一个可选的扩展给开发人员安装；
3. 也就是说，并不一定非要使用模板引擎的语法规则来开发 V(视图)部分；
4. 如果不用模板引擎，可以在控制器通过 `require()` 方法引入 PHP 文件混编即可；
5. 如果要使用模板引擎，先创建一个用于测试模板引擎的控制器：`Show.php`；
6. 写入模板引擎的调用语法，来判断，是否已经安装了模板引擎扩展；

```
return View::fetch('index'); //出现缺少驱动的错误
```

7. 然后，安装如下驱动后，再刷新页面即可；

```
composer require tophink/think-view
```

二. 赋值变量

1. 在控制器区域，通过 `assign()` 设置一个向模版提供变量的赋值操作；

```
View::assign('name', 'Mr.Lee');
```

2. 而模版区域，只需要通过 `{ $name }` 的语法即可获取到控制器设置的值；

```
{ $name }
```

3. `assign()` 方法，支持通过数组的方式，传递模版变量；

```
View::assign([
    'name' => 'Mr.Lee',
    'age'  => 100
]);
```

4. 也可以直接通过 `fetch()` 方法的第二参数，直接用数组传递模版变量；

```
return View::fetch('index', [
    'name' => 'Mr.Lee',
    'age'  => 100
]);
```

5. 助手函数 `view()`，和 `View::fetch()` 一样；

```
return view('index', [...]);
```

6. 可以使用 `filter()` 方法，对所有模版的变量进行过滤操作；

```
return View::filter(function ($content) {  
    return strtoupper($content);  
})->fetch('index');  
  
return view('index')->filter(function ($content) {  
    return strtoupper($content);  
});
```