

26. 视图赋值和过滤

学习要点：

1. 视图赋值
2. 视图过滤

本节课我们来学习视图中怎样给模版页面传递值以及过滤的方法。

一. 视图赋值

1. 在继承控制器基类的情况下，我们可以使用 `assign()` 方法进行赋值：

```
$this->assign('name', 'ThinkPHP');           //{ $name }
```

2. 也可以通过数组的方式，进行多个变量的赋值：

```
$this->assign([
    'username'    =>    '辉夜',           //{ $username }
    'email'       =>    'huiye@163.com'  //{ $email }
]);
```

3. `assign()` 方法和 `fetch()` 方法也可以合二为一进行操作：

```
return $this->fetch('index', [
    'username'    =>    '辉夜',
    'email'       =>    'huiye@163.com'
]);
```

4. 使用 `display()` 方法，可以不通过模版直接解析变量：

```
$content = '{$username}.{$email}';
return $this->display($content, [
    'username'    =>    '辉夜',
    'email'       =>    'huiye@163.com'
]);
```

5. 使用 `view()` 助手函数实现渲染并赋值操作：

```
return view('index', [
    'username'    =>    '辉夜',
    'email'       =>    'huiye@163.com'
]);

return view('index')->assign([
    'username'    =>    '辉夜',
    'email'       =>    'huiye@163.com'
]);
```

6. 使用 `View::share()` 静态方法，可以在系统任意位置做全局变量赋值；
`\think\facade\View::share('key', 'value');` //也支持数组

二. 视图过滤

1. 如果需要对模版页面输出的变量进行过滤，可以使用 `filter()` 方法；

```
$this->assign([
    'username'    =>    '辉 1 夜',
    'email'       =>    'huiye@163.com'
]);

return $this->filter(function($content){
    return str_replace("1", '<br/>', $content);
})->fetch();
```

2. 这里的 `$content` 表示所有的模版变量，找到 1 之后，实现换行操作；
3. 如果控制器有 N 个方法，都需要过滤，可以直接在初始化中全局过滤；

```
public function initialize()
{
    return $this->filter(function($content){
        return str_replace("1", '<br/>', $content);
    });
}
```

4. 也可以使用助手函数实现模版变量的过滤功能；

```
return view()->filter(function($content){
    return str_replace("1", '<br/>', $content);
});
```