

Razor视图引擎

ASP.NET程序设计

主要内容

- ▶ 定义并访问模型类型
- ▶ 减少视图中的重复
- ▶ 指定默认布局
- ▶ 通过控制器将数据值传递给视图
- ▶ 根据数据值生成不同内容
- ▶ 枚举数组或集合
- ▶ 在视图中添加命名空间

示例项目的准备工作

- ▶ 新建空模板的MVC Web工程
- ▶ 定义模型——Product
 - ▶ ProductID (int)
 - ▶ Name (string)
 - ▶ Description (string)
 - ▶ Price (decimal)
 - ▶ Category (string)
- ▶ 定义控制器——HomeController
 - ▶ 创建Product对象myProduct，并赋值
 - ▶ 创建index方法，返回值为view并将myProduct传回。
- ▶ 定义视图——index
 - ▶ 引用Product的model

使用模型对象

- ▶ 在View中输出引用的model的属性信息。

@Model.XXX

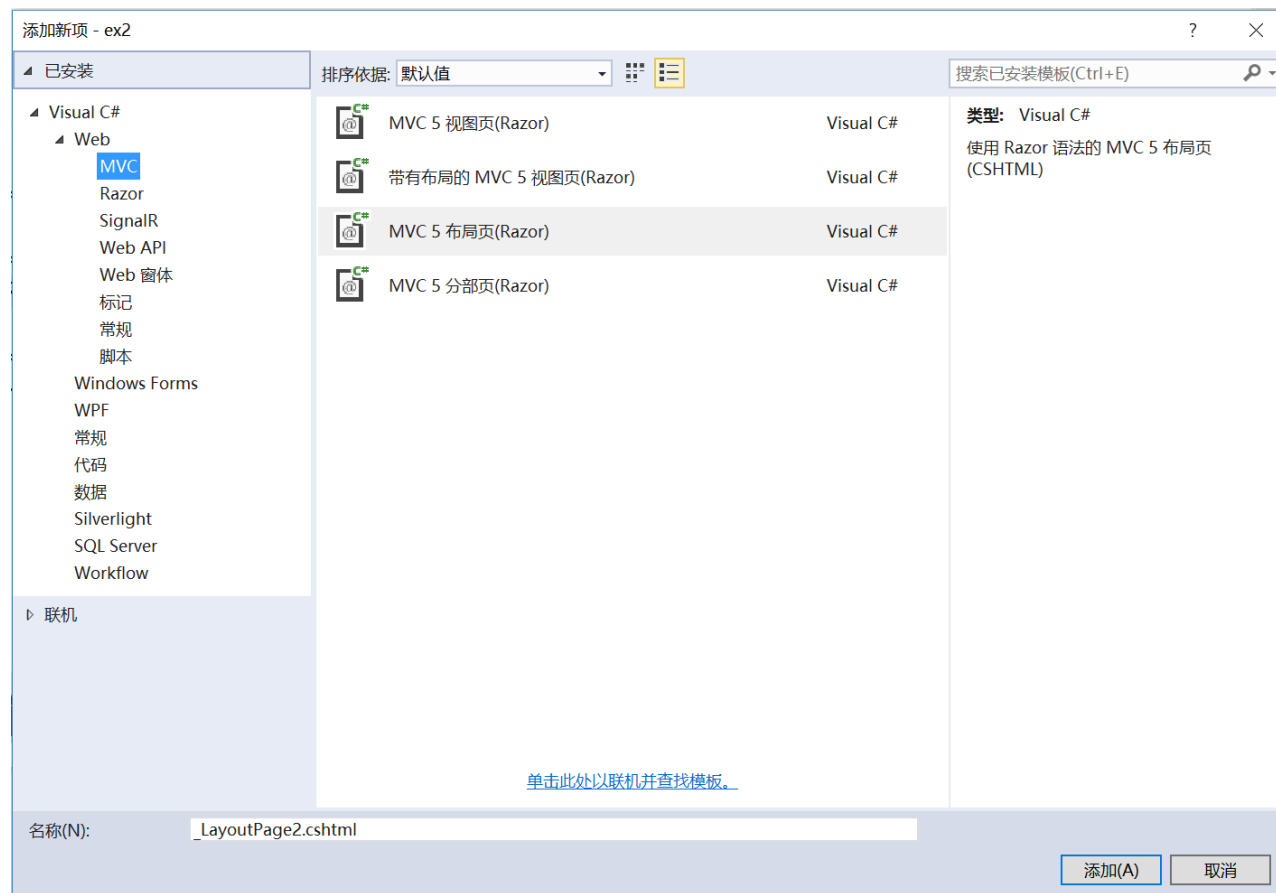
使用布局

```
@{  
    Layout = null;  
}
```

Razor代码块

- ▶ Razor代码块中允许在视图中包含C#语句。
- ▶ 在一个MVC应用程序中，Razor视图会被编译成C#类，所用的基类定义了Layout属性。
- ▶ 当Layout为null时，表示该视图是自包含的，并且会渲染客户端所需的全部内容。
- ▶ 自包含视图适合简单示例的应用程序。

添加MVC5布局页



新建的“_BasicLayout.cshtml”文件

```
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>@ViewBag.Title</title>
</head>
<body>
  <div>
    @RenderBody()
  </div>
</body>
</html>
```

该方法的调用会将动作方法所指定的视图内容插入到布局标记中

修改 “_BasicLayout.cshtml” 文件

```
<head>
  <meta name="viewport" content="width=device-width" />
  <title>@ViewBag.Title</title>
  <style>
    .content{
      padding:20px;
      border:solid medium black;
      font-size:20pt;
    }
  </style>
</head>
<body>
  <h1>Product Information</h1>
  <div class="content">
    @RenderBody()
  </div>
  <h2>Visit <a href="http://apress.com">Apress</a></h2>
</body>
```


修改 “index.cshtml” 文件（启动视图）

```
@model ex2.Models.Product
```

```
@{
```

```
    ViewBag.Title = "Product Name";
```

```
    Layout = "~/Views/_BasicLayout.cshtml";
```

```
}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <meta name="viewport" content="width=device-width" />
```

```
    <title>Index</title>
```

```
</head>
```

```
<body>
```

```
    <div>
```

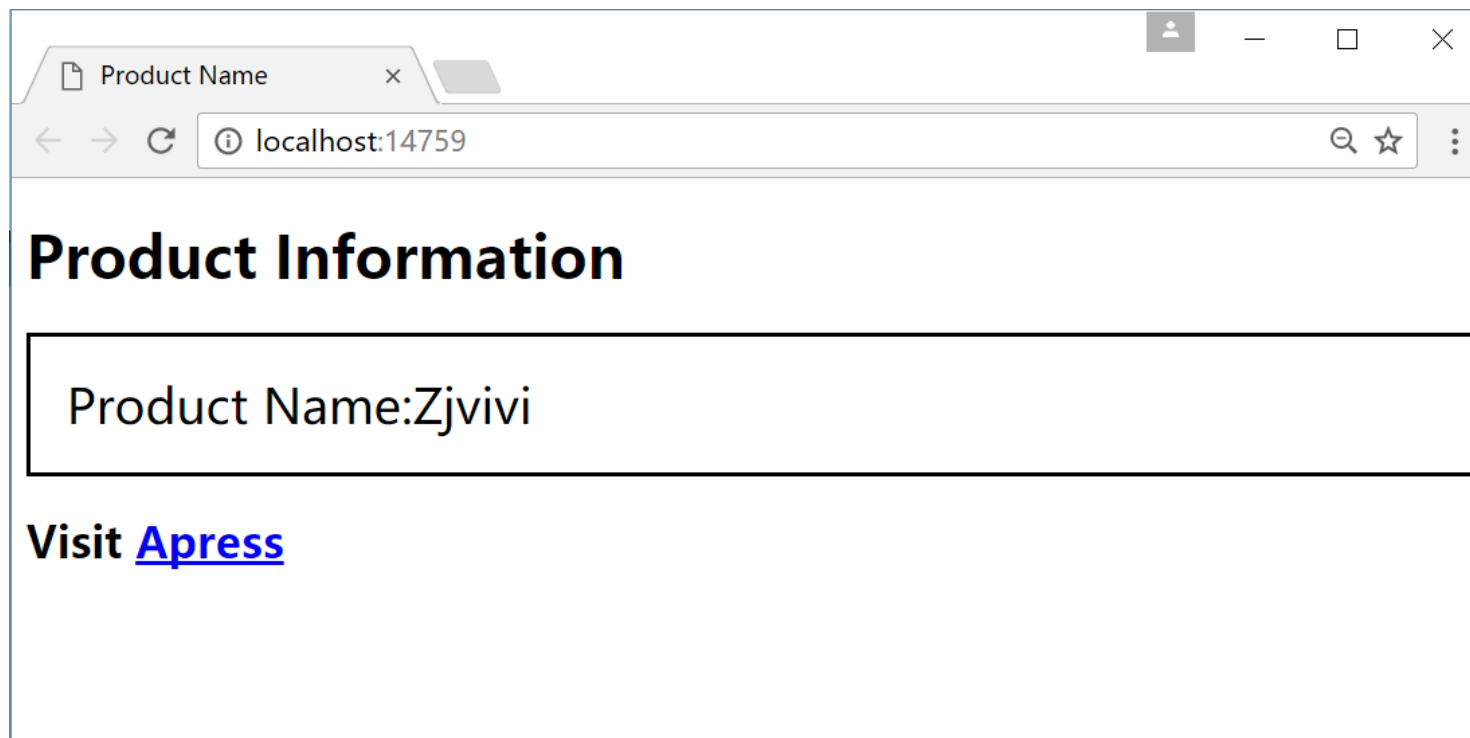
```
        Product Name: @Model.Name
```

```
    </div>
```

```
</body>
```

```
</html>
```

修改后的运行效果（基于基本布局的视图）



设置视图起始文件 (_ViewStart.cshtml)

```
Views
└─ Home
    [ @ ] Index.cshtml
    [ @ ] _BasicLayout.cshtml
    [ @ ] _ViewStart.cshtml
```

```
@{
    Layout = "~/Views/_BasicLayout.cshtml";
}
```

_ViewStart.cshtml

在视图中重新定义了Layout的值，会覆盖之前视图起始文件中所指定的布局

```
@model ex2.Models.Product
@{
    ViewBag.Title = "Product Name";
}
```

Index.cshtml

```
<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
</head>
<body>
    <div>
        Product Name: @Model.Name
    </div>
</body>
</html>
```

演示共享布局

Razor视图

添加NameAndPrice的动作方法

```
public ActionResult NameAndPrice()  
{  
    return View(myProduct);  
}
```

添加NameAndPrice视图

添加视图

×

视图名称(N):

NameAndPrice

模板(T):

Empty

▼

模型类(M):

Product (ex2.Models)

▼

选项:

☐ 创建为分部视图 (C)

☐ 引用脚本库(R)

☐ 使用布局页(U):

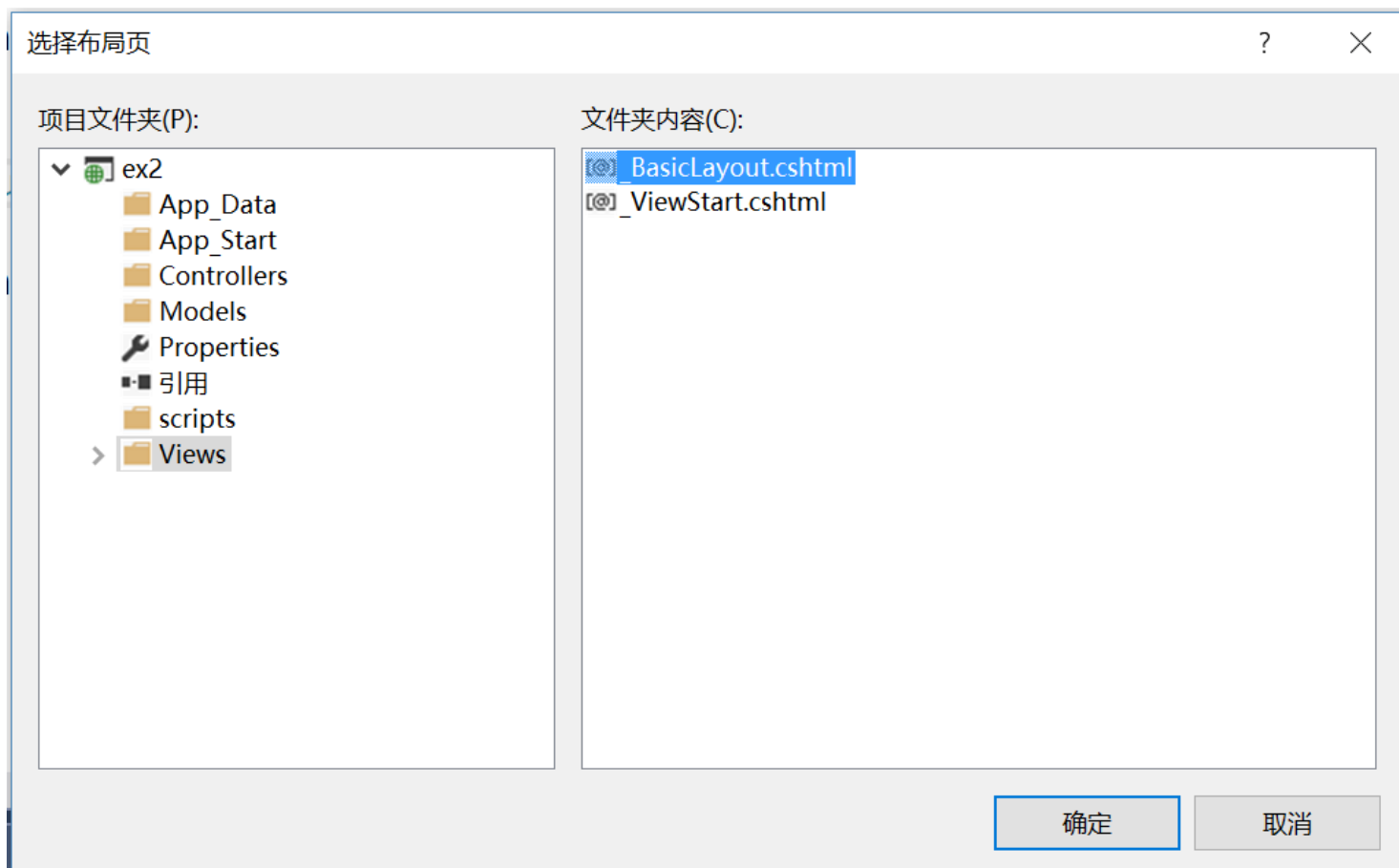
...

(如果在 Razor _viewstart 文件中设置了此选项，则留空)

添加

取消

设置NameAndPrice视图的布局页



修改 NameAndPrice 视图

添加视图

视图名称(N):

模板(T):

模型类(M):

选项:

☐ 创建为分部视图 (C)

☐ 引用脚本库(R)

☒ 使用布局页(U):

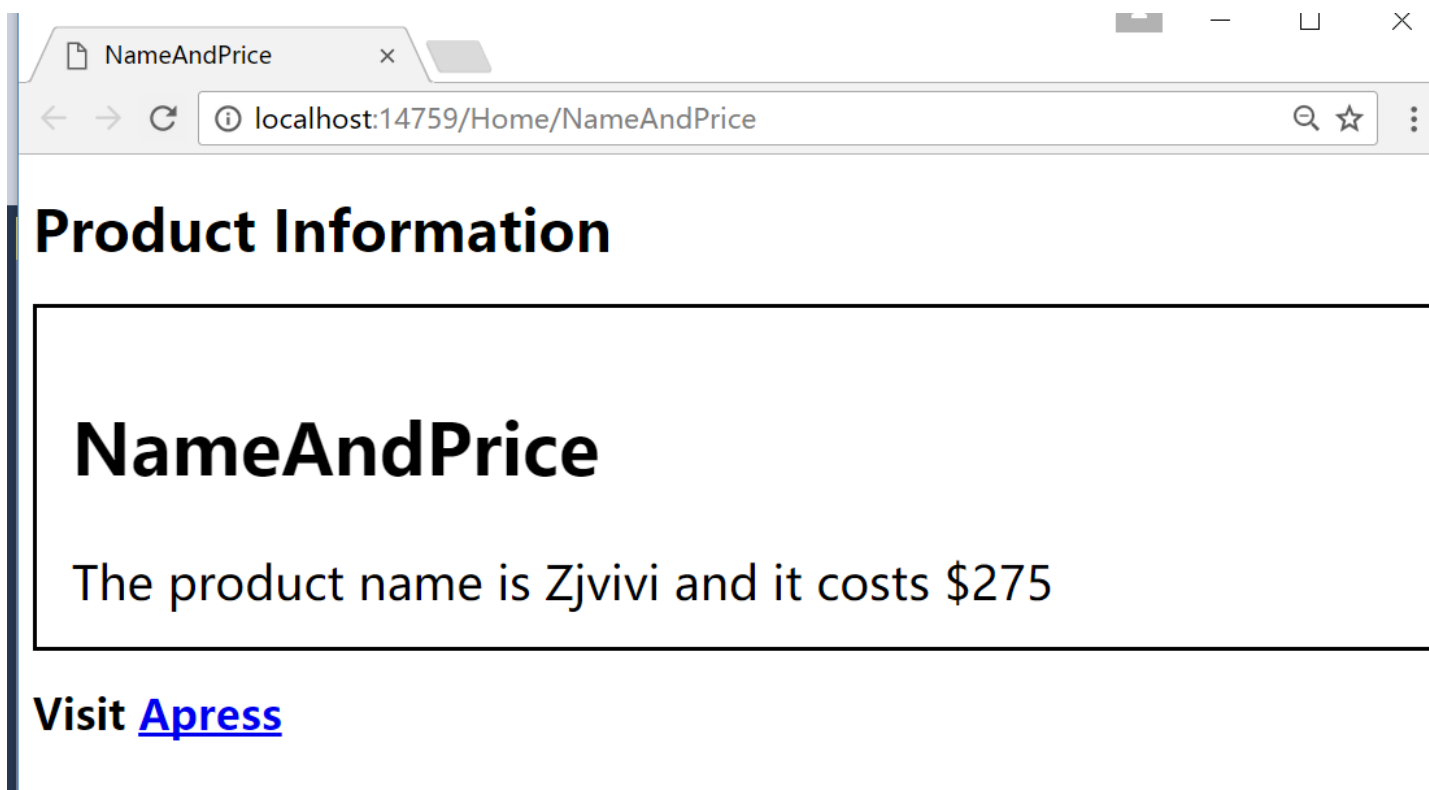
(如果在 Razor_viewstart 文件中设置了此选项, 则留空)

```
@model ex2.Models.Product
```

```
@{  
    ViewBag.Title = "NameAndPrice";  
    Layout = "~/Views/_BasicLayout.cshtml";  
}
```

```
<h2>NameAndPrice</h2>  
The product name is @Model.Name and it costs $@Model.Price
```


运行结果



使用Razor表达式

Razor视图

在HomeController中添加DemoExpression方法

```
public ActionResult DemoExpression()  
{  
    ViewBag.ProductCount = 1;  
    ViewBag.ExpressShip = true;  
    ViewBag.ApplyDiscount = false;  
    ViewBag.Supplier = null;  
    return View(myProduct);  
}
```

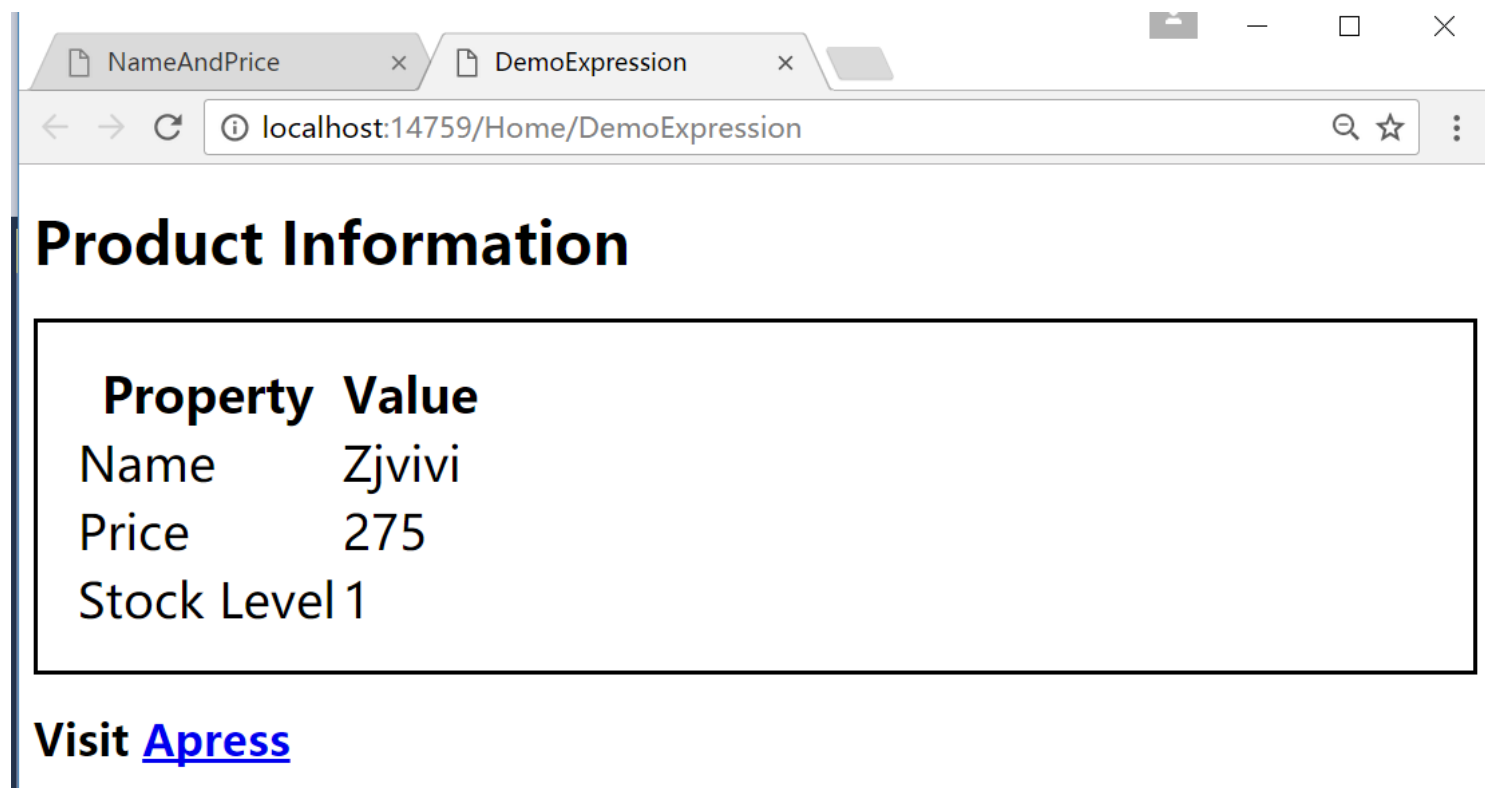
添加 DemoExpression 视图

```
@model ex2.Models.Product
```

```
@{  
    ViewBag.Title = "DemoExpression";  
}
```

```
<table>  
    <thead>  
        <tr>  
            <th>Property</th>  
            <th>Value</th>  
        </tr>  
    </thead>  
    <tbody>  
        <tr>  
            <td>Name</td>  
            <td>@Model.Name</td>  
        </tr>  
        <tr>  
            <td>Price</td>  
            <td>@Model.Price</td>  
        </tr>  
        <tr>  
            <td>Stock Level</td>  
            <td>@ViewBag.ProductCount</td>  
        </tr>  
    </tbody>  
</table>
```

运行结果



The screenshot shows a web browser window with two tabs: 'NameAndPrice' and 'DemoExpression'. The address bar displays 'localhost:14759/Home/DemoExpression'. The main content area features a heading 'Product Information' followed by a table. The table has two columns, 'Property' and 'Value', and contains three rows of data: 'Name' with value 'Zjvivi', 'Price' with value '275', and 'Stock Level' with value '1'. Below the table, there is a text prompt 'Visit' followed by a blue underlined link 'Apress'.

Property	Value
Name	Zjvivi
Price	275
Stock Level	1

Visit [Apress](#)

设置标签属性的值

```
<div data-discount="@ViewBag.ApplyDiscount"  
      data-express="@ViewBag.ExpressShip"  
      data-supplier="@ViewBag.Supplier">
```

The containing element has data attributes

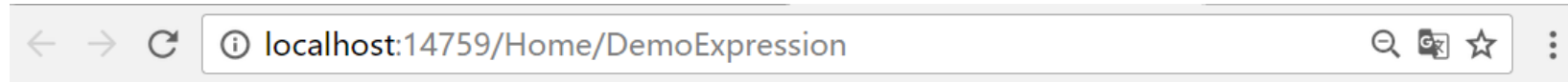
```
</div>
```

Discount: <input type="checkbox" checked="@ViewBag.ApplyDiscount" />

Express: <input type="checkbox" checked="@ViewBag.ExpressShip" />

Supplier: <input type="checkbox" checked="@ViewBag.Supplier" />

运行结果



Product Information

Property	Value
Name	Zjvivi
Price	275
Stock Level	1
The containing element has data attributes	
Discount:	<input type="checkbox"/>
Express:	<input checked="" type="checkbox"/>
Supplier:	<input type="checkbox"/>

Visit [Apress](#)

使用条件语句

```
<tr>
  <td>Stock Level</td>
  <td>
    @switch ((int)@ViewBag.ProductCount)
    {
      case 0:
        @: Out of Stock
        break;
      case 1:
        <b>Low Stock(@ViewBag.ProductCount)</b>
        break;
      default:
        @ViewBag.ProductCount
        break;
    }
  </td>
</tr>
```

C#的条件关键字前放@字符

@:表示阻止该条解释为C#语句

将switch语句改成if语句?

HomeController 中添加 DemoArray 方法

```
public ActionResult DemoArray()  
{  
    Product[] array =  
    {  
        new Product {Name="Kayak",Price=275M },  
        new Product {Name="Lifejacket",Price=48.95M },  
        new Product {Name="Soccer",Price=19.50M },  
        new Product {Name="Corner",Price=34.95M }  
    };  
    return View(array);  
}
```

添加 DemoArray 视图

```
@model ex2.Models.Product[]

@{
    ViewBag.Title = "DemoArray";
    Layout = "~/Views/_BasicLayout.cshtml";
}

@if (Model.Length > 0)
{
    <table>
    <thead>
    <tr>
    <th>Product</th>
    <th>Price</th>
    </tr>
    </thead>
    <tbody>
    @foreach (ex2.Models.Product p in Model)
    {
        <tr>
        <td>@p.Name</td>
        <td>$@p.Price</td>
        </tr>
    }
    </tbody>
    </table>
}
else
{
    <h2> No Product data</h2>
}
```

枚举数组和集合运行结果



Product Information

Product	Price
Kayak	\$275
Lifejacket	\$48.95
Soccer	\$19.50
Corner	\$34.95

Visit [Apress](#)

处理命名空间

```
@using ex2.Models  
@model Product[]
```

```
@{  
    ViewBag.Title = "DemoArray";  
    Layout = "~/Views/_BasicLayout.cshtml";  
}  
@if (Model.Length > 0)  
{  
    <table>  
        <thead>  
            <tr>  
                <th>Product</th>  
                <th>Price</th>  
            </tr>  
        </thead>  
        <tbody>  
            @foreach (Product p in Model)  
            {  
                <tr>  
                    <td>@p.Name</td>  
                    <td>$@p.Price</td>  
                </tr>  
            }  
        </tbody>  
    </table>  
}
```