## Setting the View Model Type（一）

- Razor doesn't know what type will be used by the action method for the view model.
- @model keyword is used to specify the type of the view model.
- @Model expression is used to access the view model object provided by the action method.

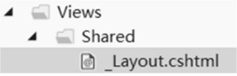
```
@model List<Employee>
……
@foreach (var item in Model)
```

## Using Shared Views

- While most controllers have their own views, views can also be shared so that common functionality doesn't have to be duplicated.
- Layout (布局)
- View start file（视图开始文件）
- View import file（视图导入文件）

## Layout（一）

- The views may contain duplicate elements:
  - setting up the HTML document
  - defining the head section
  - loading the CSS file
- Layout contains the common content that will be used by multiple views.



## Layout（二）

- The content that is unique to each view is inserted into the response by calling the RenderBody method.

```
<!DOCTYPE html>

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

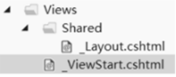
## Layout（三）

- Views that use layouts can focus on just their unique content.

```
@model Product
@{
    Layout = "_Layout";
}
<div class="m-2">
    ……
</div>
```

## View Start File

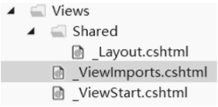
- Instead of setting the Layout property in every view, you can add a view start file to the project that provides a default Layout value.



```
@{
    Layout = "_Layout";
}
```

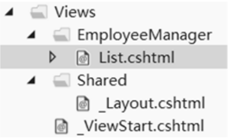
## View Imports File

- By default, all types that are referenced in a Razor view must be qualified with a namespace.
- You can specify a set of namespaces that should be searched for types by adding a view imports file to the project.



```
//导入名称空间
@using SampleApplication.Models
//启用tag helper
@addTagHelper *,
Microsoft.AspNetCore.Mvc.TagHelpers
```

## Project: Implement the List view



```
@model IEnumerable<Employee>
<h2>Employee Info List</h2>
<table>
    <tr>
        <td>Employee ID</td>
        <td>Employee Name</td>
        <td>Employee Title</td>
        <td>Employee Country</td>
    </tr>
    @foreach(Employee e in Model)
    {
        <tr>
            <td>@e.EmployeeId</td>
            <td>@e.EmployeeName</td>
            <td>@e.Title</td>
            <td>@e.Country</td>
        </tr>
    }
</table>
```