Lesson 4

用 Razor 語法提升 View 的智慧與戰鬥力

學習重點 +

- 十五條 Razor 語法規則精選
- Razor 中判斷式與流程控制的運用
- 用 Razor 提升 UI 介面設計的戰鬥力
- 借助 Razor Helper 簡化 Razor 的複雜度
- 從網頁中提煉出可重複使用的 Partial View

前三章範例中，View 給人的印象是忠實顯示 Controller 傳來的資料，貌似樸拙而無突出才能。但事實上 View 也可以很智慧，因為 View 使用的是 Razor 語法，Razor 語法不但能宣告變數、集合，也能用判斷式、迴圈做出智慧型決策及顯示，遠比你想像中要來得強大。

4-1 Razor 概觀

什麼是 Razor？Razor 又稱 Razor Syntax，是用來將 Server Side 的 C#程式嵌入到 HTML 中的標記語法（Markup Syntax）。Syntax 字面上透露它是語法，而非 Language 語言。

❖ Razor 標記語法

所謂的「將 Server Side 的 C#程式嵌入到 HTML 中的標記語法」是何意？以下是一段 Razor 程式：

```
@{
      var Name = "Kevin";
      var Height = 180;
      var Weight = 75;
  }
<p>我的名字: @Name </p>
<p>我的身高: @Height </p>
<p>我的體重: @Weight </p>
```

1. C#程式區塊
C#變數
HTML 的<p> element
2. Razor Inline 表達式

說明：

1. Razor 中只有 HTML 及 C#兩種元素，二者的結合就形成了 Razor 語法。
2. C#程式區塊（Razor Code Block）是以@{...}包覆，裡面是一般 C#程式。

3. Razor Inline 表達式是指「C#變數穿插在 HTML 中」的式子。而 Razor 中預設是 HTML 語言，若遇到@符號，表示它後面接的是 C#指令。
4. Razor 會依不同的規則或符號在 HTML 和 C#之間做切換。

❖ Razor 是語法而非語言

那既然有 Razor 程式，似乎它就是語言，但又為何說 Razor 不是語言？因為 Razor 中只包含 HTML 和 C#，這些都不是 Razor 自己的，一個沒有變數、判斷式和 element tag 的東西，要稱為語言是有些勉強。同時 Razor 程式最後還會被轉換成 C#類別程式來執行，而不是真的有 Razor Language。

❖ Razor 支援的保留關鍵字

而「Razor 中只包含 HTML 和 C#」這句話是有但書的，雖然 HTML 全數可用，但不是所有 C# 指令或關鍵字都能在 Razor 中使用，下圖是 Razor 支援的保留關鍵字，分為兩大類：

1. Razor 關鍵字：section、model、helper、inherits 和 functions 五個關鍵字是 Razor 創造的，用來支持 Razor 語法所需功能。
2. C#關鍵字：這是源自既有的 C#，不是 Razor 所創造。Razor 支援常用的 C#關鍵字，但有些如 namespace 及 class 關鍵字就不支援。

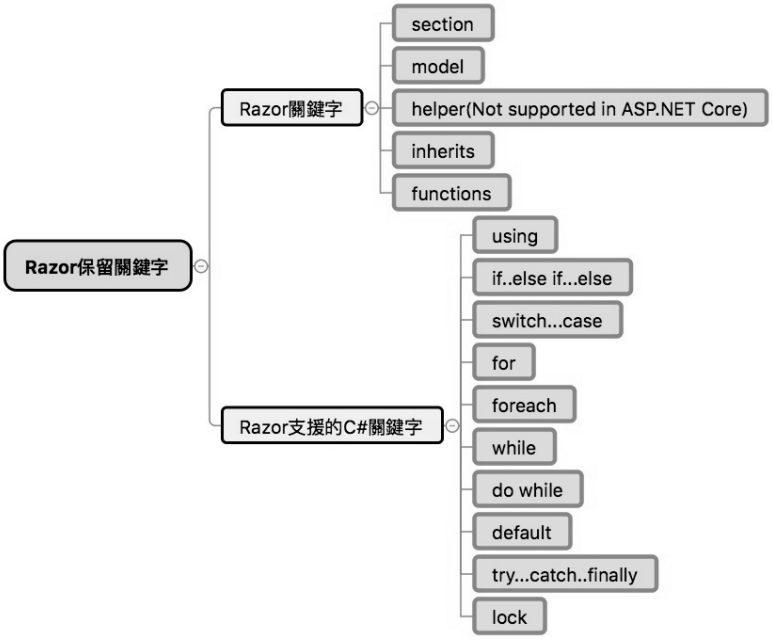

圖 4-1 Razor 保留關鍵字

最後，Razor 語法只能在 View 檢視（.cshtml）中使用，而不能在.html 中使用，所以 View 也稱為 Razor View 或 Razor Page。

4-2 Razor 語法規則

第一次接觸 Razor 的人，對於 Razor 究竟能做什麼常有疑問，簡言之，除了宣告 HTML 外，Razor 能用 C#宣告：變數、陣列、集合、判斷式、迴圈，也能使用 LINQ 語法。

但 Razor 有其語法規則與風格，若想駕馭，就得了解它的語法形式。請參考 MvcRazor 專案的 Views\Razor\RazorRules.cshtml，以下用 15 個規則解說 Razor 語法。

❖ 規則 1：以@符號作為 C#的開頭

Razor 語法包含：❶HTML 語法，❷ C#語法兩部分，遇到 HTML 的 Markup，就解析為 HTML（HTML Parser），遇到單一@符號開頭，就解析為 C#語法（C# Parser）。

❖ 規則 2：以@{...}宣告單行 C#程式

Razor 用@{...}包覆單行 C#程式，@{...}包含的區塊也稱為 Code Block 程式區塊，每行程式的結尾需加上;分號：

Views\Razor\RazorRules.cshtml

```
@{ var City = "Taipei"; }
@{ var PostalCode = 110; }

@*以下用明確型別宣告變數也可以*@
@{ string city = "Taoyuan"; }
@{ int postalCode = 334; }
```

說明：程式區塊中的 C#變數、集合或陳述式僅做設定或運算用途，而不做 HTML 輸出。而變數型別除了用 var 宣告外，亦可使用明確型別。

❖ 規則 3：以@{...}宣告多行 C#程式

多行程式也是用@{...}來包覆，只不過將程式分成多行：

```
@{
    var Name = "Kevin";
    var Height = 180;
    var Weight = 75;
}
```

❖ 規則 4：C#的 Inline 表達式

若 C#變數穿插在 HTML 中則為 Inline 表達式，以下用@Name、@Height 等 Inline 表達式將規則 2 和 3 的變數做顯示：

```
<p>我的名字：@Name </p>
<p>我的身高：@Height </p>
<p>我的體重：@Weight </p>
<p>居住城市：@City  </p>
<p>郵遞區號：@PostalCode </p>
```

HTML 輸出結果為：

```
我的名字：Kevin
我的身高：180
我的體重：75
居住城市：Taipei
郵遞區號：110
```

若以@符號顯示變數值，ASP.NET 一律將變數值做 HTML 編碼，輸出成純文字，例如@("<h1>MVC</h1>")會輸出<h1>MVC</h1>。

❖ 規則 5：C#程式區塊中的 HTML 隱式轉換

@{...}程式區塊中預設語言是 C#，但若夾雜了 HTML 語法，Razor 會自動做隱式轉換，將該部分輸出成 HTML：

```
@{
    var LeapYear = DateTime.IsLeapYear(DateTime.Now.Year);
    <p>今年是否為閏年：@LeapYear </p>
}
```

結果為：

```
今年是否為閏年：False
```

❖ 規則 6：C#關鍵字嚴格區分大小寫（但 VB 不分）

若 C#變數名稱相同，僅大小寫不同，Razor 仍視為兩個獨立變數：

```
@{
    var MyName = "聖殿祭司";
    var myName = "奚江華";
}
<p>
    筆名: @MyName <br />
    姓名: @myName <br />
</p>
```

結果為：

```
筆名: 聖殿祭司
姓名: 奚江華
```

❖ 規則 7：單行註解－@*...*@

單行註解用**@*...*@**表示：

```
@*這是單行註解*@
```

❖ 規則 8：多行註解－@*...*@

多行註解也是用**@*...*@**表示，只不過分成多行：

```
@*多行註解
也有支援*@
```

❖ 規則 9：Razor 隱性表達式－@符號

Razor 隱性表達式是由**@**符號開頭，系統會自動解析為 C#語法：

```
<p>現在的時間是：@DateTime.Now </p>
```

結果為：

```
現在的時間是：2017/11/5 下午 03:21:09
```

❖ 規則 10：Razor 明確表達式－@(...)符號

Razor 明確表達式是由**@(...)**所包覆，明確指出括號內是 C#運算式：

```
<p>兩週前我出國去玩，出發日期是：@((DateTime.Now - TimeSpan.FromDays(14)).
ToShortDateString()) </p>
<p>3+7 的結果是：@(3 + 7) </p>
```

出發日期是今天日期減 14 天，推算出兩週前的日期，結果為：

```
兩週前我出國去玩，出發日期是：2017/10/22
3+7 的結果是：10
```

❖ 規則 11：以文字顯示@符號，需用@@表示

在 HTML 中顯示@文字，需加上第二個@做跳脫，也就是**@@**。例如在 HTML 顯示頭昏的表情符號@_@，語法為：

```
<p>
    @@_@@ <br />
<p>
<p>
    但 Email 和超連結例外<br />
    我的電子郵件：dotnetcool@gmail.com <br />
    <a href="mailto:service@domain.com">Service@domain.com</a>
</p>
```

結果為：

```
@_@
但 Email 和超連結例外
我的電子郵件：dotnetcool@gmail.com
Service@domain.com
```

❖ 規則 12：字串變數中的雙引號顯示

如果字串變數想顯示雙引號，在最前頭先加上@，字串內再用連續兩個""雙引號表示：

```
@{ var word = @"子曰：""三人行，必有我師焉""...";}
<p>@word</p>
```

結果為：

```
「子曰："三人行，必有我師焉"...」。
```

❖ 規則 13：用@(...)將 HTML 或 JS 編碼成純文字

@(...)內除了做運算外，還可將表達式做 HTML 編碼，例如將 HTML 或 JavaScript 的表達式編碼成 HTML 文字：

```
@{ var msg = @"<button type='button' onclick='alert(""Hi JavaScript"")'> Raw
原始字串,不做 HTML 編碼</button>"; }
<p>@(msg)</p>  ◄── 用@(...)進行 HTML 編碼
@("<span>Hello MVC!</span>") <br />
```

結果為：

```
<button type='button' onclick='alert("Hi JavaScript")'> Raw 原始字串, 不做 HTML 編
碼</button>
<span>Hello MVC!</span>
```

說明：無論 HTML 或 JavaScript 都會被編碼成純文字，目的是增加網頁安全性，不被注入網頁攻擊程式。

❖ 規則 14：用@Html.Raw()顯示原始字串, 不做 HTML 編碼

同一個 msg 字串變數，若想顯示原始值，不讓 HTML 或 JavaScript 被編碼，可用@Html.Raw(...)指令顯示：

```
@{ var msg = @"<button type='button' onclick='alert(""Hi JavaScript"")'> Raw 原
              始字串,不做 HTML 編碼</button>"; }
<p>@Html.Raw(msg)</p>
```

說明：

1. 結果會產生一個 HTML <button>按鈕，而不是文字，按下會有 JavaScript Alert 警告訊息。
2. 但若沒做 HTML 編碼可能會有潛在安全性問題。

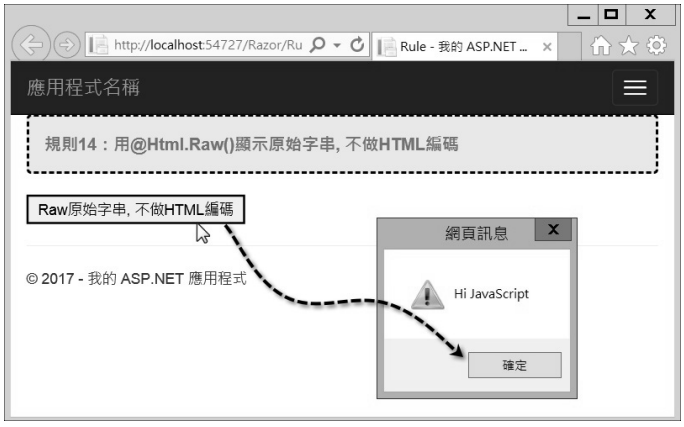

圖 4-2 顯示原始字串不做 HTML 編碼

❖ 規則 15：磁碟路徑表示法

字串變數若包含磁碟路徑如「c:\FileFolder\」，可在最前面加上@符號：

```
@{ var filePath = @"c:\FileFolder\"; }
<p>磁碟路徑: @filePath</p>
```

結果為：

```
磁碟路徑: c:\MvcFolder\
```

若要把檔案虛擬路徑轉換成實際磁碟路徑，可用 Server.MapPath() 指令：

```
@{
    var imageVirtualPath = @"/Assets/images/SteveJobs.jpg";
}
<p>Virtual Path 虛擬路徑: @imageVirtualPath</p>
<p>Physical Path 實際路徑: @Server.MapPath(imageVirtualPath)</p>
```

結果為：

```
Virtual Path 虛擬路徑: /Assets/images/SteveJobs.jpg
Physical Path 實際路徑:
C:\MvcExamples\MvcRazor\MvcRazor\Assets\images\SteveJobs.jpg
```