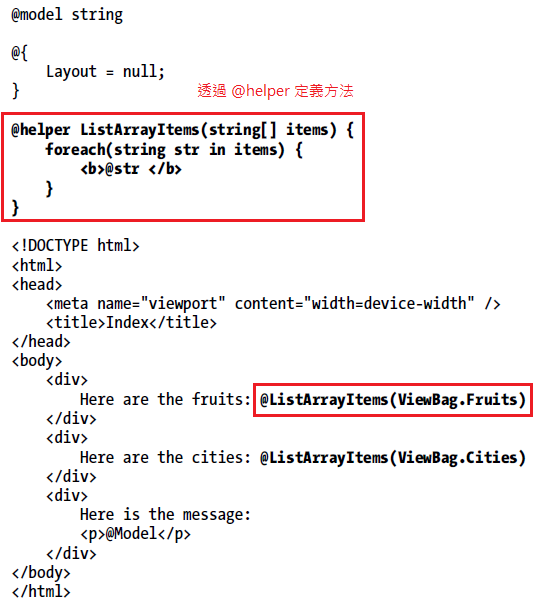
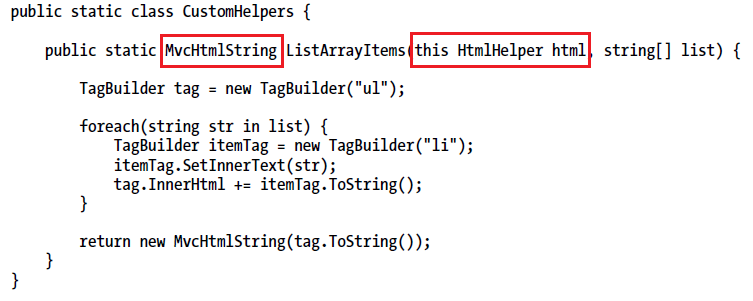
1. **Creatring Custom Helper Methods**
   1. Creating an Inline Helper Method (只是把重複使用的Method寫在單一個View裡面，沒多大用處)



* 1. Creating an External Helper Method (把重複使用的Method寫在class裡面，當擴充方法用，所有View都可以呼叫)

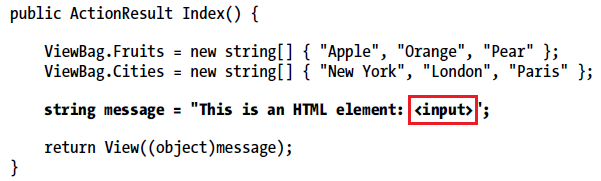
類別跟方法都是宣告為static。注意第一個參數



在View裡面一樣是透過 @Html去呼叫

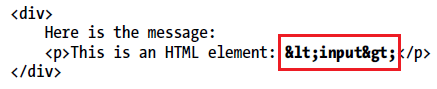
3.PNG

* 1. Managing String Encoding in a Helper Method
     1. Razor的資料編碼
        + 假設傳入View的Model包含此資料 (一個input標籤)

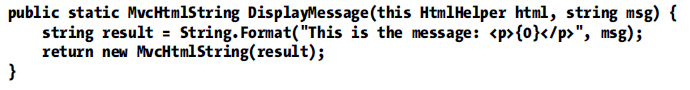


* + - * 但實際上資料被Razor解析後會是輸出下面的結果 (Razor基於安全性，會把資料編碼)。當然，如果你直接在View上面寫<input>，頁面當然是輸出一個可以打字的Text標籤

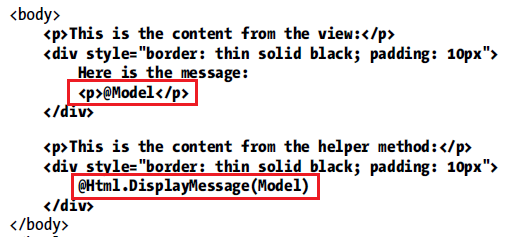
3.PNG



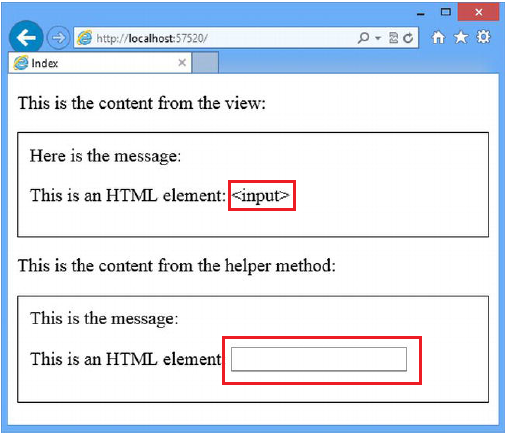
* + 1. Help Method可能造成的問題
       - 假設有一擴充方法如下，回傳的是MvcHtmlString



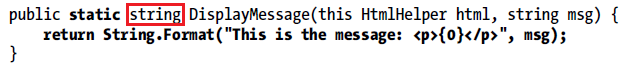
* + - * View的結構，傳入的Model資料同上 (下面那段是呼叫Help Method)



* + - * 頁面輸出 (下面那段變成輸出可打字的Input標籤，此模式可能會造成安全上的疑慮)

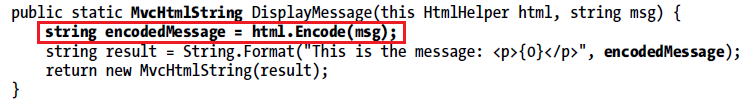


* + 1. 把Help Method輸出的資料也進行編碼
       - 最快的解決方法：把方法的回傳值由MvcHtmlString改為一般的string

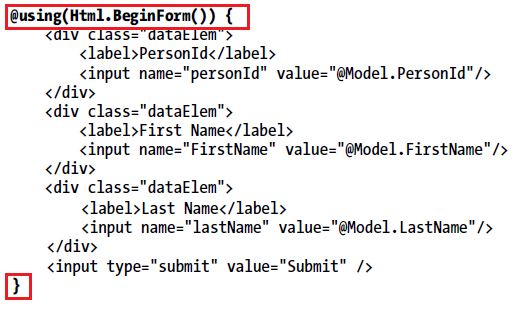


* + - * 注意！上圖寫法的<p>標籤也會被進行編碼，所以可以再改一下寫法

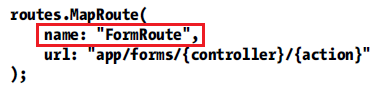
先把傳入的字串編碼過，然後一樣可以回傳MvcHtmlString



1. **Using the Built-In Form Helper Methods**
   1. Creating Form Elements
      1. 最常使用的寫法，可以稍微了解一下



* + 1. Specifyring the Route Used by a Form (讓你可以設定Form Action要對應到哪個Route)
       - 假設定義了以下的一個路由



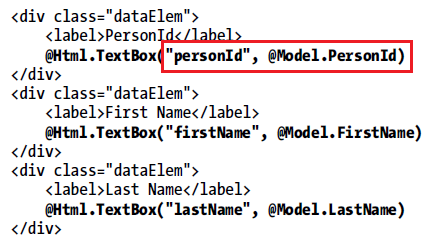
* + - * 則View裡面可以這樣寫 (用BeginRouteForm方法)

3.PNG

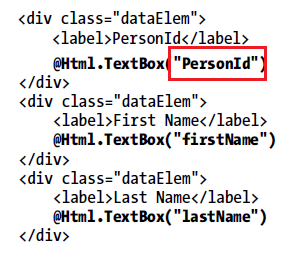
* + - * 則輸出的Form Action如下 (對應到已定義的Route規則)

5.PNG

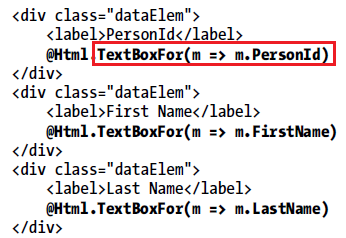
* + 1. Using Input Helpers
       - 第一種：給Html欄位名稱以及對應的值



* + - * 第二種：只給Html欄位名稱 (會以該字串去ViewBag、View Model等位置尋找對應的值)



* + - * 第三種：夠過類似TextBoxFor的方法以及傳入Model的某個屬性 (比較常用)



* + 1. Creating Select Elements
       - 常用的建立方法

