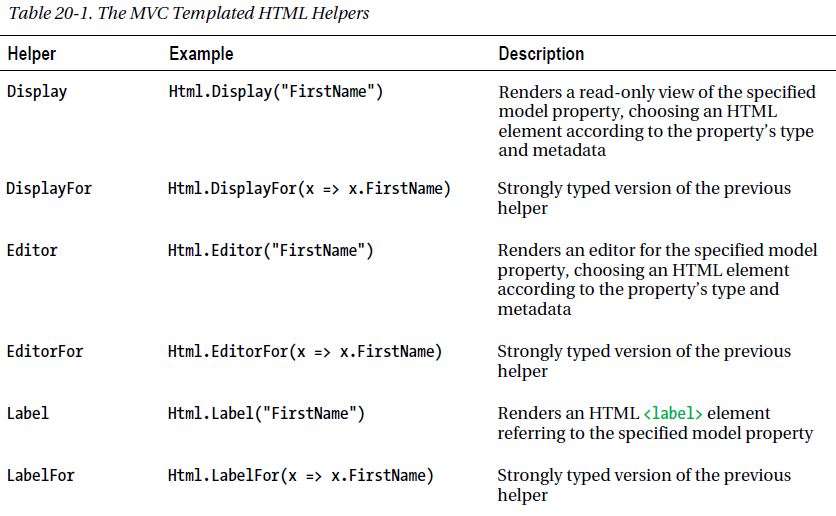
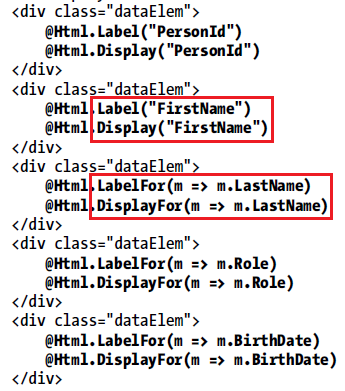
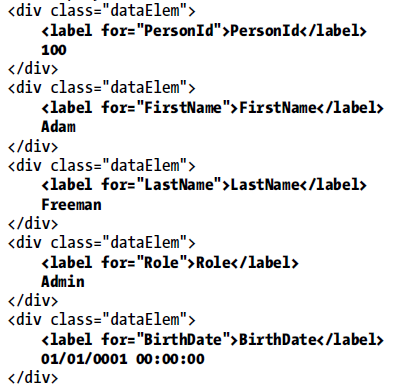
1. **Using Templated Helper Methods**
   1. 內建的Templated Helper Methods



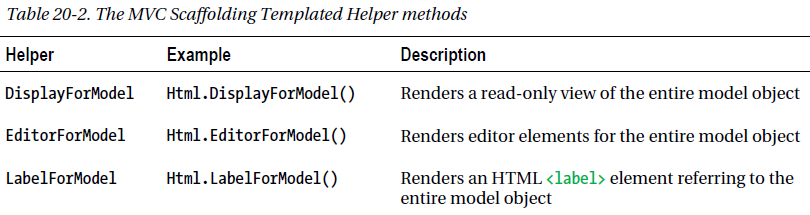
* 1. Generating Label and Display Elements
     1. 假設有以下的View結構

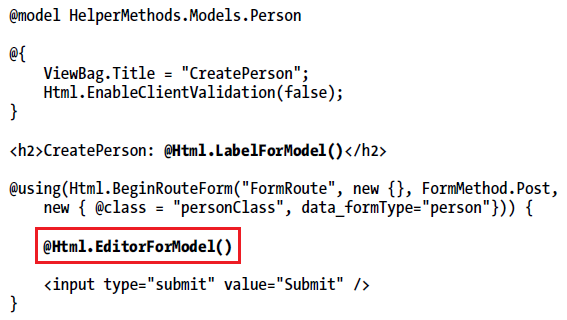


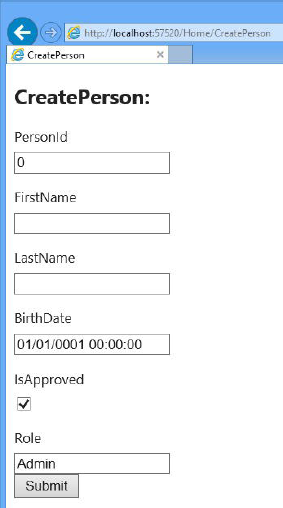
* + 1. 輸出的Html內容會是如下 (注意Display跟DisplayFor，只會輸出Model屬性的值，不會輸出其他的Html內容)



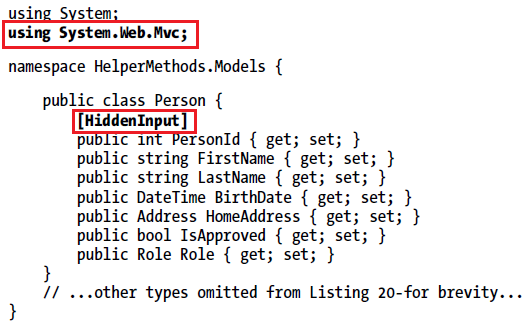
* 1. Using Whole-Model Template Helpers (針對整個View Model去產出Html標籤的一個功能)
     1. 內建的方法

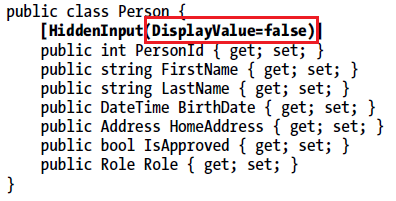




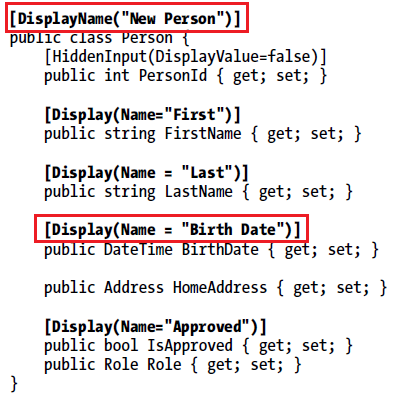
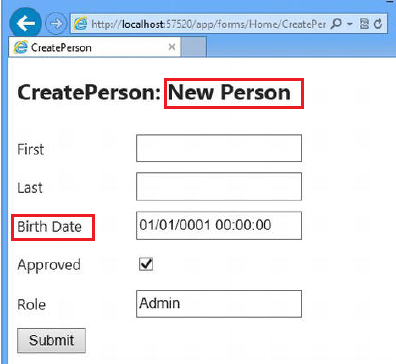


1. **Using Model Metadata**
   1. Using Metadata to Control Editing and Visibility (控制Html Tag的可編輯性與是否隱藏)

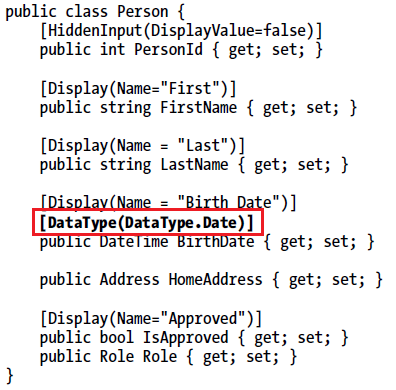


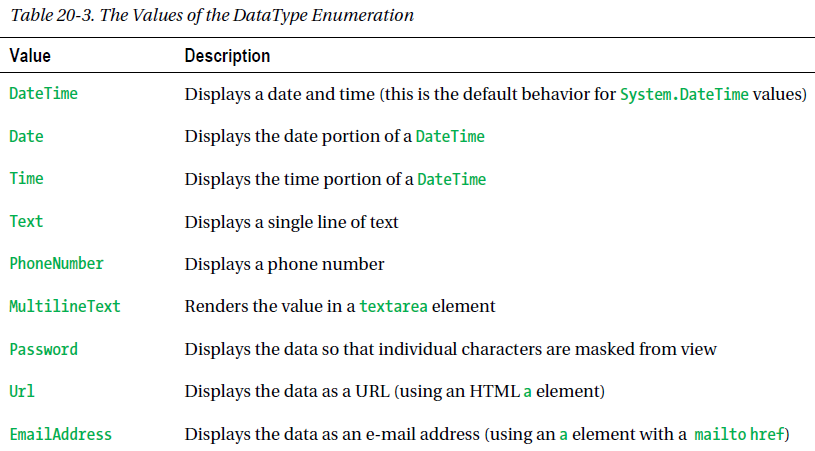


* 1. Using Metadata for Labels (改變Model屬性在顯示所出現的欄位名稱)

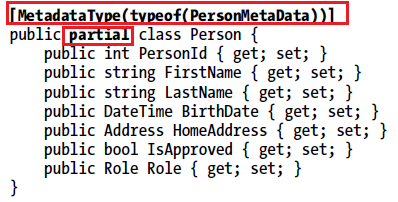
  


* 1. Using Metadata for a Data Values (針對Model屬性做資料種類上的設定，然後在顯示或編輯時，會根據資料種類去呈現不同的結果)

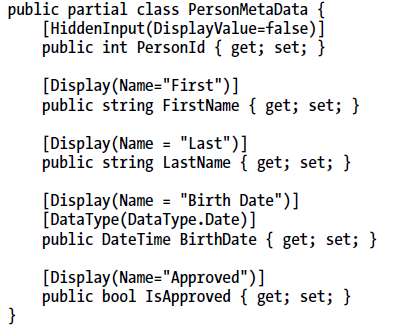




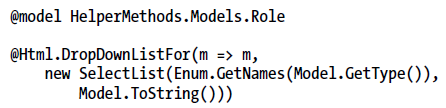
* 1. Applying Metadata to a Buddy Class (如果Model Class是動態產生而導致無法設定Metadata的話，可以透過此方式去設定Metadata)
     1. 使用MetadataType以及partial class



* + 1. 然後在另一個class定義需要的Metadata (不必針對每個屬性都定義，列出需要的屬性就可以)



1. **Customizing the Templated View Helper System**
   1. Creatring a Custom Editor Template (客製化一個針對編輯用的Template，針對Display也可以)
      1. 先確認該Model屬性的型別是什麼 (底下範例的是Role型別)
      2. 在對應路徑底下建立View (用來呈現Template的實際內容)
         * /Views/控制器/Template種類/型別.cshtml
         * /Views/Shared/Template種類/型別.cshtml
           1. Template種類可以是DisplayTemplates或是EditorTemplates
      3. 底下建立的是Role.cshtml (如果你針對的屬性是string，直接建立String.cshtml就可以)



* + 1. 尋找Template的順序

