

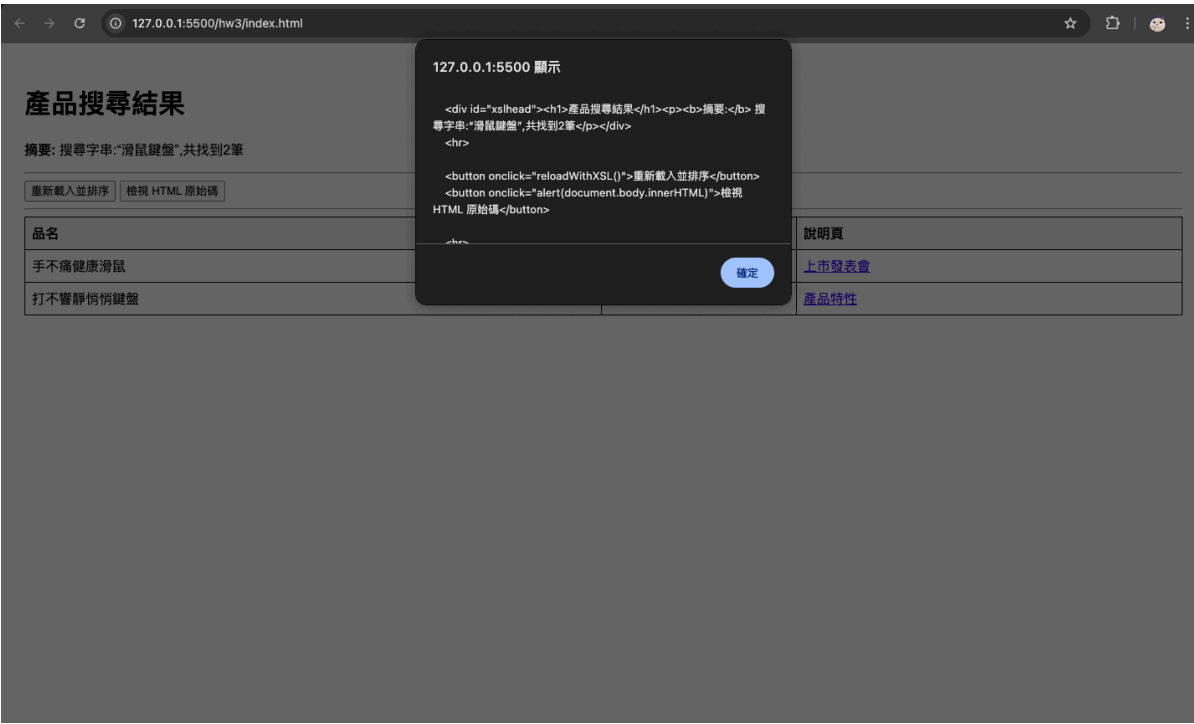
## **XML 動態網頁技術實作：以 Client-side XML Viewer 為例**

### **1. 實作說明：**

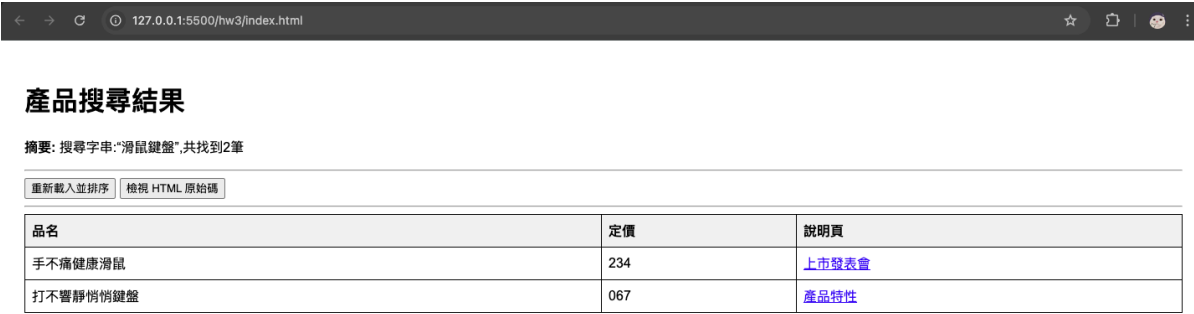
a. 本實作採用 DOM Design Pattern (1): Presentation Transformation 的架構，系統分為三個主要部分：

- i. **Input Data (ex.xml)**: 模擬後端資料庫輸出的 XML 格式資料，包含產品貨號、品名與價格等資訊。
- ii. **Presentation (ex1.xsl)**: 定義資料的呈現樣式 (Table 結構) 與排序邏輯，將內容與外觀分離。
- iii. **Main Program (index.html)**: 作為 Client-side 的控制器。
  1. 使用 DOM API (DOMParser) 解析 XML 文件樹。
  2. 透過 `getElementsByTagName` 隨機存取 XML 中的「摘要」節點並顯示。
  3. 使用 XSLT 引擎將 `ex.xml` 與 `ex1.xsl` 動態結合成 HTML 表格呈現給使用者。

### **2. 實作畫面：**



a.



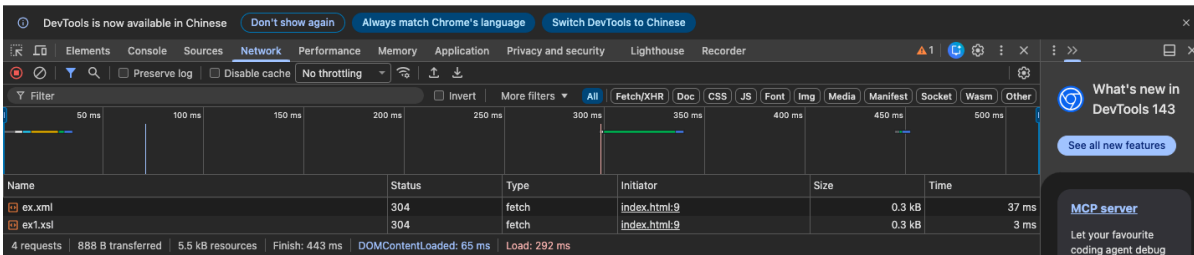
b.

產品搜尋結果

摘要: 搜尋字串:"滑鼠鍵盤",共找到2筆

重新載入並排序 檢視 HTML 原始碼

品名	定價	說明頁
手不痛健康滑鼠	234	<a href="#">上市發表會</a>
打不響靜悄悄鍵盤	067	<a href="#">產品特性</a>



C.

3. 討論：

- a. **1. DOM Tree 的運作優勢** 在實作中，我們將 XML 文件載入記憶體後形成 DOM Tree 。這允許程式對文件進行「隨機存取」(Random Access)，例如在程式碼中我們直接抓取 <摘要> 標籤的內容顯示，而不需像 SAX Parser 那樣從頭循序讀取，這種特性非常適合用於瀏覽器端的動態網頁互動。
- b. **資料與呈現分離 (Separation of Data and Presentation)** 本實作最核心的價值在於將資料 (.xml) 與樣式 (.xsl) 完全切開。若未來需要改變網頁外觀(例如將表格改為清單)，或改變排序方式(依價格排序)，只需替換 XSL 檔案或修改 XSL 邏輯即可，完全不需要更動後端輸出的 XML 資料結構。
- c. **技術演進：從 ActiveX 到現代標準 (重要加分題)** 投影片原始範例使用

ActiveXObject("Microsoft.XMLDOM"), 這是早期 IE 瀏覽器的專有技術。在本次實作中，為了符合現代 Web 標準(Chrome/Edge/Firefox)，我將

程式碼改寫為使用 W3C 標準的 DOMParser 與 XSLTProcessor。這不僅解決了跨瀏覽器相容性問題, 也反映了 XML 技術從 IE 獨大時期到現在標準化時期的演進。