

主題一：CSS 選擇器的命名方式

CSS 選擇器有三個常見的命名指導原則，分別是 OOCSS、SMACSS、以及 BEM。了解和運用這些指導原則，幫助我們撰寫更容易理解和閱讀的 CSS 程式碼。

完成以下學習步驟和實作，並準備報告：

1. 說明 OOCSS、SMACSS、以及 BEM 命名原則的基本想法。
2. 選擇其中一個你最喜歡的命名原則，舉一個實際的版面開發案例來展示給大家看。例如，你可以展示如何運用 OOCSS 命名原則在我們第一週的任務上。

主題二：完善資料驗證程序

在過去的任務中，我們沒有嚴格的要求驗證使用者是否輸入資料，或是資料是否符合特定格式。然而一個良好的系統，無論前端或是後端都必須進行輸入資料的驗證。

完成以下學習步驟和實作，並準備報告：

1. 前端在使用者註冊、登入、留言、查詢、修改會員姓名時，要先驗證資料是否忘記輸入，格式是否正確，並在錯誤時直接提醒使用者。沒有任何錯誤才發送請求到後端。
2. 後端接收到前端的請求時，要驗證資料是否存在，且資料格式是否符合預期，一切都正確才能執行後續的資料庫操作。否則就直接回應錯誤訊息給前端。
3. 請在最後一週的註冊、登入、留言、查詢、修改會員姓名的功能中，建立以上資料驗證程序並展示出來。
4. 在網站系統中，前端驗證和後端驗證的主要用途是什麼？為什麼前端驗證過的資料，後端必須再驗證一次？

主題三：AJAX 與 CORS

AJAX (fetch() 或 XMLHttpRequest) 技術讓我們可以在不換頁的狀況下，從前端發出 HTTP Request 與後端進行互動。然而，使用 AJAX 技術時會碰到跨來源資源分享 (CORS) 的議題。無論是前端或後端，都應該對這個議題有所了解。

完成以下學習步驟和實作，並準備報告：

1. 什麼是 CORS？
2. 我們可以在自己的網頁中，使用 fetch() 或是 XMLHttpRequest 連結到 <https://www.google.com/> 並取得回應嗎？
3. 我們可以在自己的網頁中，使用 fetch() 或是 XMLHttpRequest 連結到 <https://padax.github.io/taipei-day-trip-resources/taipei-attractions-assignment.json> 並取得回應嗎？和上述的狀況，差別在哪裡？
4. 如何開放我們自己開發的 API，讓別的網站透過 fetch() 或是 XMLHttpRequest 連結，達到如同第 3 點的可能性。

主題四：使用主鍵、索引優化資料庫查詢效率

資料表的欄位設計，為了加速查詢效率，會額外設置主鍵 (Primary Key) 和索引 (Index)。是否能進一步優化我們會員系統的資料表呢？

完成以下學習步驟和實作，並準備報告：

1. 了解主鍵 (Primary Key) 和索引 (Index) 的觀念。
2. 請在 member 資料表中加入適當的索引，加快以下 SQL 語句的查詢效率
`SELECT * FROM member WHERE username='test' and password='test'`
3. 如何驗證查詢效率是否真的變更好了？
4. 為什麼索引的設置能有效地改善查詢效率？
5. 使用索引是否能加快 LIKE 模糊查詢的運作效率？為什麼？

主題五：使用 **Connection Pool** 連結資料庫

我們會在程式中建立 **Connection** 來進行和 **MySQL** 的資料庫操作，若進一步使用 **Connection Pool** 就能有效的增加後端伺服器 and 資料庫連線的穩定性以及效能。

完成以下學習步驟和實作，並準備報告：

1. 什麼是 **Connection Pool**？能帶給我們什麼好處？為什麼？
2. 如何使用官方提供的 **mysql-connector-python** 套件，建立 **Connection Pool**。
3. 需要從資料庫取得查詢資料時，如何從 **Connection Pool** 取得 **Connection**，並且在資料操作結束後，歸還 **Connection** 到 **Connection Pool** 中。請展示你完成上述標準操作的程式碼。

主題六：了解並預防 **Cross-Site Scripting (XSS)** 攻擊

跨網站指令碼攻擊 (XSS) 是常見的網站攻擊方式之一，試著研究其原理，複製其攻擊手法，並說明開發者如何預防此攻擊。

完成以下學習步驟和實作，並準備報告：

1. XSS 攻擊的基本介紹。
2. 實際設計一個 XSS 攻擊情境。
3. 根據上述設計的情境，說明開發者該如何預防。