

本週要完成網站首頁的 RWD 版面，要持續載入新的景點資訊供使用者瀏覽，並提供互動式的關鍵字搜尋的功能。這次專案中，請勿使用任何第三方前端套件，包括 **jQuery**、**Bootstrap** 等等。

為了強調前後端分離架構，請勿更動任何已經寫好的 **# Pages** 四個後端的路由函式。

Part 2 - 1：完成 RWD 靜態頁面

請按照 Figma 設計稿 (<https://www.figma.com/file/MZkYBH31H5gyLoZoZq116j>) 中的首頁設計，完成 RWD 的靜態版面。(不用包括跳出式的會員註冊登入視窗)

Part 2 - 2：串接景點 API，取得並展示第一頁的景點資訊

請透過前端的 `fetch()` 技術，串接上週完成的旅遊景點 API (`/api/attractions`)，在網頁載入時，立刻將第一頁的景點資訊，包含景點名稱、分類和捷運站名，顯示在畫面上。

景點列表中的圖片使用每個景點的第一張圖片即可。

Part 2 - 3：完成自動載入後續頁面的功能

偵測頁面捲動到最下方的時候，自動載入下一個頁面的景點資訊，並繼續顯示在畫面下方，請參考以下步驟：

1. 第一次載入後，取得、記錄 API 回傳的 `nextPage` 資訊，當作下一次載入頁面的依據。
2. 偵測頁面是否已經捲動到最下方：可利用 `IntersectionObserver` 物件；或者利用 `window` 的 `scroll` 事件，搭配 HTML 元件的 `getBoundingClientRect` 等等方法進行偵測。
3. 若頁面捲動到最下方，呼叫 API 載入下一頁的資訊，由於網路連線有時間差，請在前端想辦法阻擋 API 可能被瞬間連續呼叫狀況。
4. 取得下一頁的資訊後，接續把景點顯示在畫面下方，重新紀錄後端回傳的 `nextPage` 資訊，當作下一次要載入頁面的依據。若 `nextPage` 為空值，則停止做下一頁的載入。

Part 2 - 4：完成關鍵字搜尋功能

讓使用者可以在網頁上方的輸入框，輸入關鍵字，運用 `fetch()` 技術在沒有重新載入整個頁面的條件下，串接旅遊景點 API (`/api/attractions`)。

取得搜尋結果後，取代原先的景點列表，把搜尋到的景點呈現出來，並同樣完成載入更多的效果；如果沒有搜尋到任何景點，用文字簡單的表示沒有結果即可。

Part 2 - 5 : 完成景點分類關鍵字填入功能

網頁上方的搜尋框除了可以直接輸入關鍵字外，還必須支援以下互動：

1. 網頁載入時，立刻利用景點分類 API (`/api/categories`) 取得目前所有的景點分類，建立暫時隱藏的分類區塊。
2. 使用者點選搜尋框時，打開跳出式分類區塊，顯示景點分類列表。
3. 點選景點分類，將景點分類名稱填入搜尋框，隱藏跳出式分類區塊。
4. 點選畫面的其他位置，隱藏跳出式分類區塊。

工作任務繳交方式：

請按照 Part 0 的 Git、GitHub 工作流程，平時在 `develop` 分支進行所有開發、測試、專案部屬的動作。確定要繳交時，發出一個 `Pull Request` 準備合併程式到 `main` 分支，並邀請 Chao-Wei Peng (cwpeng) 擔任 Code Reviewer，靜待結果。

請在 **Pull Request** 的訊息中填寫你的名字，並提供已上線頁面實際可測試網址，例如：

`http://140.112.3.5:3000/`