

# 第2次隨堂題目-隨堂-QZ2

學號：112111102

姓名：吳映潔

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

- 說明內容
- 執行成果

## 說明程式與內容

1. a.

Ans:

1. b.

Ans:

程式：

```
function getLowStock(products) {  
    return products  
        .filter(item => item.stock < 10)  
        .map(item => item.name);  
}  
  
const lowStockItems = getLowStock(products);  
console.log("庫存少於 10 的項目：" , lowStockItems);
```

說明：

這段程式是為了找出庫存少於10的商品名稱，先用.filter()篩選出stock < 10的商品，然後用.map()取得這些商品的name，最後呼叫函式回傳結果，並用console.log()輸出。

執行結果：

庫存少於 10 的項目： [ 'mouse', 'monitor' ]

上圖為程式執行結果

上圖為程式執行結果

1. c.

Ans:

程式：

程式：

```
function updateStock(products, updates) {  
    const updatesObj = {};  
    updates.split(",").forEach(pair => {  
        const [name, stockStr] = pair.split(":");  
        updatesObj[name.trim()] = Number(stockStr);  
    });  
    for (let item of products) {  
        if (updatesObj.hasOwnProperty(item.name)) {  
            item.stock = updatesObj[item.name];  
        }  
    }  
    for (let item of products) {  
        console.log(`${item.name} 的庫存： ${item.stock}`);  
    }  
    return products;  
}  
  
updateStock(products, "mouse:15,monitor:20");
```

說明：

這段程式是為了更新商品陣列中商品的庫存數量，先將傳入的 `updates` 字串（格式為 "商品名:庫存,商品名:庫存"）用 `split()` 分割後，轉換成一個物件 `updatesObj`，然後用 `:` 分割每組名稱與庫存，建立 `updatesObj` 儲存更新資料，再來針對 `products` 陣列中的每個商品，如果商品名稱存在於 `updatesObj` 中，則將其庫存更新為新的數量，以及使用 `console.log()` 印出更新後每個商品的庫存狀況，最後呼叫函數執行 `updateStock`。

執行結果：

```
Keyboard 的庫存： 25  
mouse 的庫存： 15  
monitor 的庫存： 20  
usb cable 的庫存： 40
```

上圖為程式執行結果

2. a.

Ans:

程式：

程式：

```
switch(url) {  
    case '/':  
        answer = 'index.html 輸出部分';  
        break;  
    case '/calculator':  
        answer = 'index2.html 輸出的部分';  
        break;  
    default:  
        answer = 'error.html 輸出的部分';  
}  
}
```

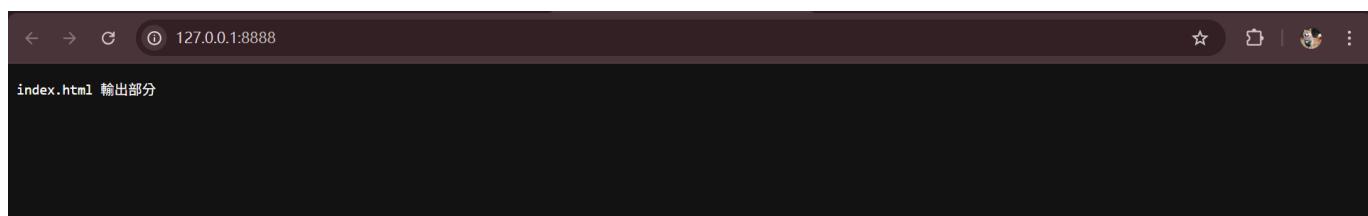
## 說明：

這段程式是為了根據請求的URL，回傳對應的回應內容。

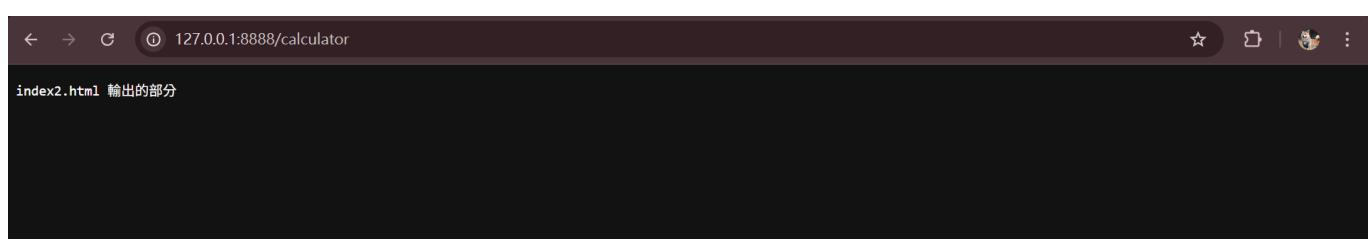
程式中使用switch判斷請求的url。

- 若url為/，回傳"index.html 輸出部分"。
- 若url為/calculator，回傳"index2.html 輸出的部分"。
- 若url為其他(例如/hi)，回傳"error.html 輸出的部分"

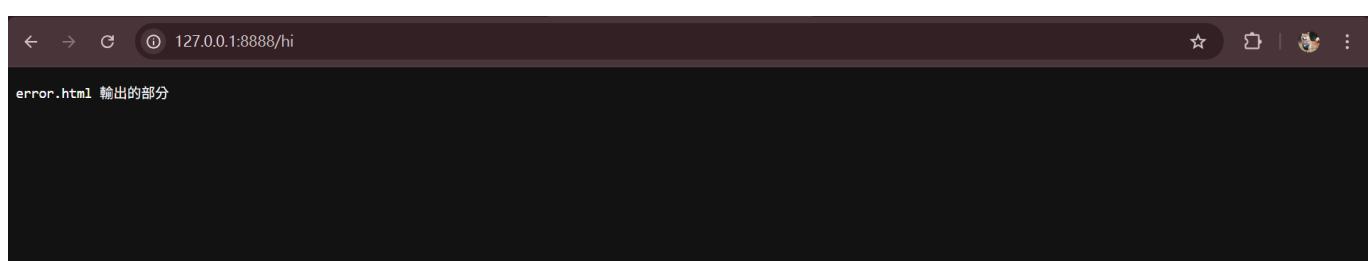
## 執行結果：



當請求url為/時 → 回傳畫面



當請求url為/calculator時 → 回傳畫面



當請求url為其他如/hi時 → 回傳畫面

2. b.

Ans:

程式：

程式：

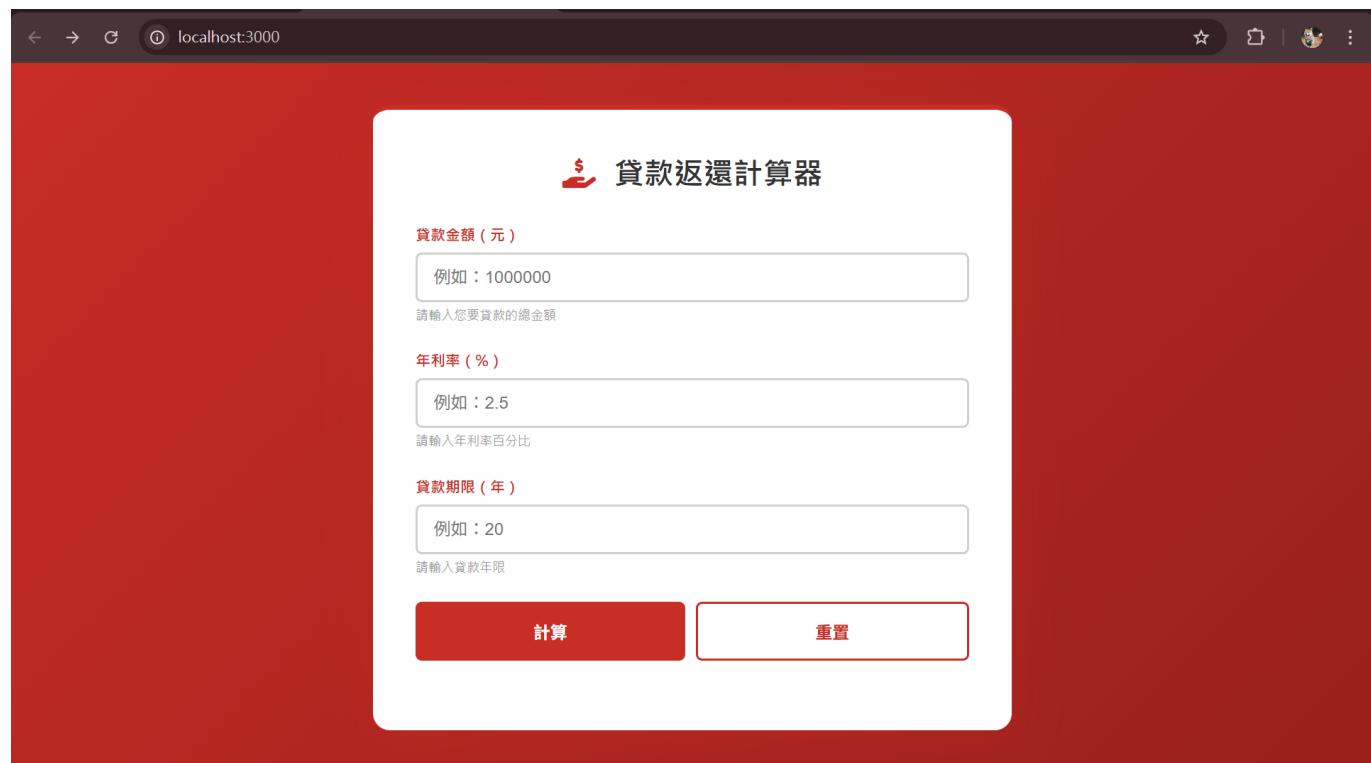
```
switch(req.url) {  
  case '/':  
    filePath = '/index.ejs';  
    break;  
  case '/calculator':  
    filePath = '/index2.ejs';  
    break;  
}  
}
```

說明：這段程式是為了根據請求的網址（URL），回傳對應的回應內容，

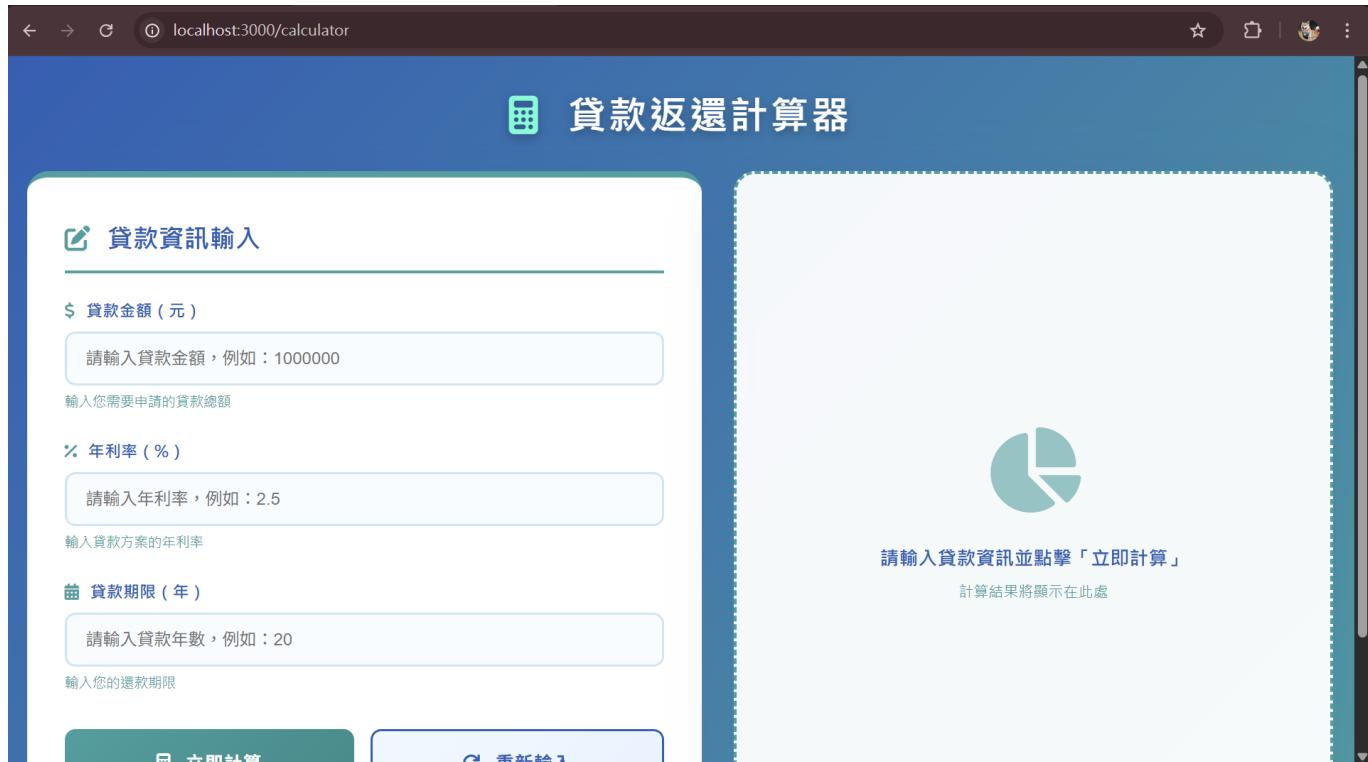
程式中使用switch 判斷請求的url。

- 若url為/，回傳index.ejs畫面。
- 若url為/calculator，回傳index2.ejs畫面。

執行結果：



當請求url為/時 → 回傳畫面



當請求url為/calculator時 → 回傳畫面

2. c.

Ans:

程式：

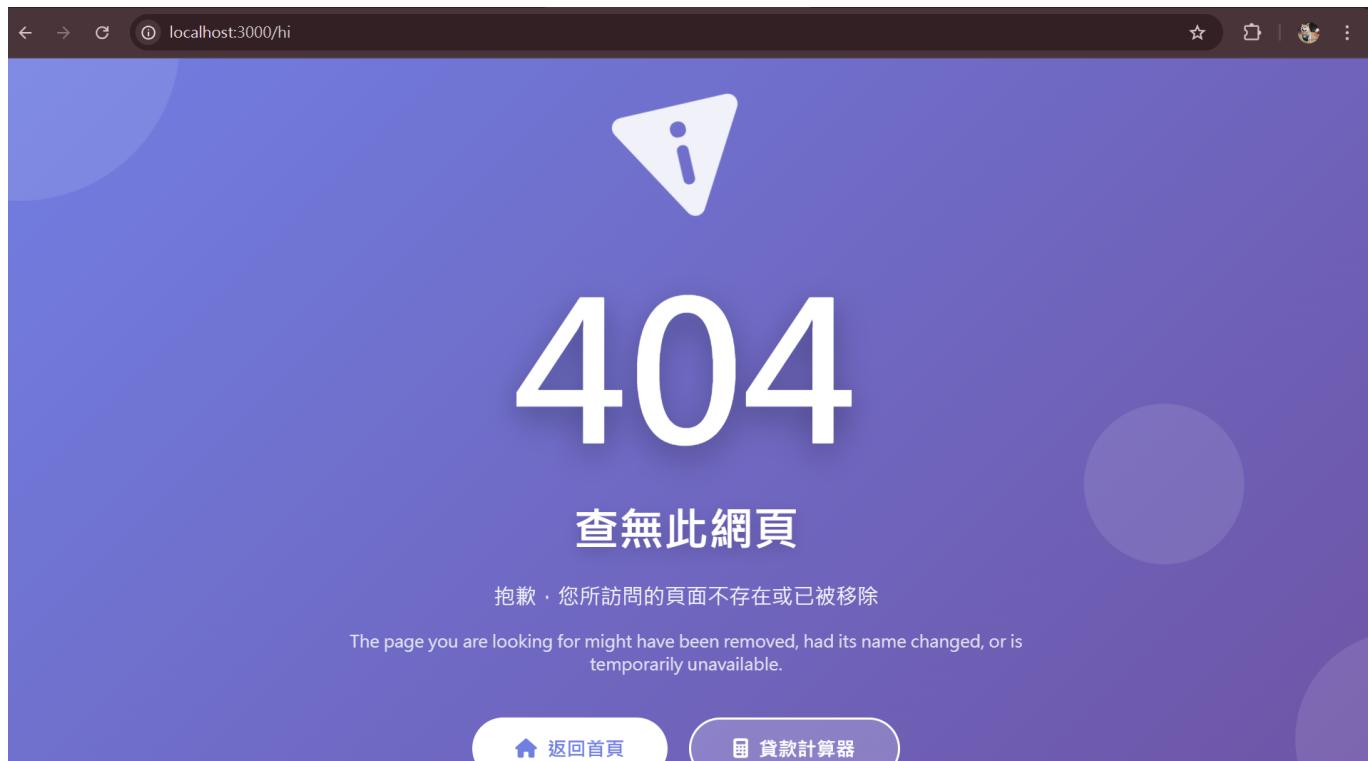
程式：

```
fs.readFile(staticFilePath, (err, content) => {
  if (err) {
    fs.readFile('./index3.ejs', 'utf8', (err2, template) => {
      if (err2) {
        res.writeHead(500, { 'Content-Type': 'text/html; charset=utf-8' });
        res.end('錯誤：無法讀取模板文件 - ' + err2.message);
        return;
      }
      const html = ejs.render(template);
      res.writeHead(404, { 'Content-Type': 'text/html; charset=utf-8' });
      res.end(html);
    });
  } else {
    res.writeHead(200, { 'Content-Type': contentType });
    res.end(content);
  }
});
```

說明：這段程式為嘗試讀取靜態資源並處理錯誤請求，先嘗試讀取靜態資源，如果讀取失敗，代表使用者請求的路徑不存在，則讀取 index3.ejs 作為 404 畫面，再將index3.ejs渲染成HTML後回傳給使用者，HTTP狀態碼

設定為 404，如果靜態資源讀取成功，直接回傳內容給使用者，HTTP 狀態碼為 200。

執行結果：



當請求url為其他如/**hi**時 → 回傳畫面

當請求url為其他如/**hi**時 → 回傳畫面

2. d.

Ans: