

Code Review 紀錄內容			更新日期：2024/08/21
日期	CR 人員	題號	備註
2024/08/21	陳韻安	1. 模擬封裝與私有	HashMap.js
1. 變數命名有意義	<p>修改前：</p> <pre>var HashMap = function () { let obj = {}; };</pre> <p>修改後：</p> <pre>var HashMap = function () { let map = {}; };</pre>		
2. keys 善用原生方法	<p>利用 Object.keys(map) 查看所有 key</p> <p>修改前：</p> <pre>keys: function () { const keySet = []; for (let k in obj) { keySet.push(k); } return keySet; },</pre> <p>修改後：</p> <pre>keys: function () { Object.keys(map); },</pre>		
日期	CR 人員	題號	備註
2024/08/21	陳韻安	1. 模擬封裝與私有	HashMap.html
3. 更改多字串串連方式	<p>多字串串連可使用反引號``將所有 HTML 包起來，再利用 \${} 插入變數</p> <p>修改前：</p> <pre>function printResult(key, value) { result.innerHTML += key + ' : ' + map.get(key) + '
'; }</pre> <p>修改後：</p> <pre>function printResult(key, value) { result.innerHTML += `\${key} : \${map.get(key)}
`; }</pre>		

4. 判斷可以提早結束	<div>提早 return 不需再寫 else 判斷</div> <div>修改前：</div> <pre>if (!newKey) { alert('請輸入key'); return; }; if (!map.contains(newKey)) { map.put(newKey,newValue); printResult(newKey, newValue); } else { alert('Key 已存在'); }</pre> <div>修改後：</div> <pre>if (!newKey) { alert('請輸入key'); return; }; if (map.contains(newKey)) { alert('Key 已存在'); return; } map.put(newKey,newValue); printResult(newKey, newValue);</pre>		
日期	CR 人員	題號	備註
2024/08/21	陳韻安	2. 操作 DOM 物件	DomTable.html
5. 使用 trim() 檢核空白格輸入	<div>修改前：</div> <pre>if (!manufacturer.value) { alert('請填寫製造商'); } else if (!type.value) { alert('請填寫類別');</pre> <div>修改後：</div> <pre>if (!manufacturer.value.trim()) { alert('請填寫製造商'); } else if (!type.value.trim()) { alert('請填寫類別');</pre>		
6. 判斷可以提早結束	<div>提早 return 不需再寫 else 判斷</div> <div>修改前：</div> <pre>if (!manufacturer.value.trim()) { alert('請填寫製造商'); } else if (!type.value.trim()) { alert('請填寫類別'); } else { addData(manufacturer.value, type.value, minPrice.value, price.value); }</pre> <div>修改後：</div> <pre>if (!manufacturer.value.trim()) { alert('請填寫製造商'); return; } if (!type.value.trim()) { alert('請填寫類別'); return; } addData(manufacturer.value, type.value, minPrice.value, price.value);</pre>		

7. 改 寫 radio 按 鈕	<p>新增按鈕中 radio 可改寫：document.createElement 搭配 setAttribute 增加屬性</p> <p>修改前：</p> <pre>const radio = newRow.insertCell(0); radio.innerHTML = '<input type="radio" name="index">';</pre> <p>修改後：</p> <pre>const newRow = data.insertRow(data.rows.length); //tr const radio = newRow.insertCell(0); const radioBtn = document.createElement('input'); radioBtn.setAttribute('type', 'radio'); radioBtn.setAttribute('name', 'index'); radio.appendChild(radioBtn);</pre>
8. 改 寫 刪 除按鈕	<p>新增按鈕中 刪除 可改寫：document.createElement 搭配 setAttribute 增加屬性</p> <p>修改前：</p> <pre>newRow.insertCell(6).innerHTML = '<button onclick="deleteRow(this)" id="delete">刪除</button>'; }; deleteRow = function (e) { e.parentNode.parentNode.remove(); //tr reTable(data); }</pre> <p>修改後：</p> <pre>const deleteData = newRow.insertCell(6); const deleteBtn = document.createElement('button'); deleteBtn.innerHTML = '刪除'; deleteBtn.setAttribute('onclick', 'deleteRow(this)'); deleteData.appendChild(deleteBtn);</pre>
9. 修 正 bug	<p>點選 radio button 沒有帶出最新資料，只帶出輸入資料時存入的值，需改選成 radio button checked 的值</p> <p>修改前：</p> <pre>radio.querySelector('input[type="radio"]').onclick = function () { manufacturer.value = manufacturerData; type.value = typeData; minPrice.value = minPriceData; price.value = priceData; }</pre> <p>修改後：</p> <pre>radio.querySelector('input[type="radio"]').onclick = function () { const radioIndex = radioBtn.parentNode.parentNode.rowIndex; manufacturer.value = data.rows[radioIndex].cells[2].innerHTML; type.value = data.rows[radioIndex].cells[3].innerHTML; minPrice.value = data.rows[radioIndex].cells[4].innerHTML; price.value = data.rows[radioIndex].cells[5].innerHTML; }</pre>

10. 優化：數字檢核

檢核底價與售價是否為數字

修改後：

```
if (isNaN(minPrice.value)) {  
    alert('底價請輸入數字');  
    return;  
}  
if (isNaN(price.value)) {  
    alert('售價請輸入數字');  
    return;  
}
```