

Github URL

W05-P1: use JavaScript module to import and export

The screenshot shows a Visual Studio Code editor with three files open: `index.html`, `app_11.js`, and `data.js`. The Explorer sidebar on the left shows a project structure with a `demo` folder containing `w05-menu`, `w01`, `w02`, `w03`, `w04`, `w05`, `images`, `app_11.js`, and `data.js`. The `index.html` file (line 57) includes a script tag: `<script type="module" src="app_11.js"></script>`. The `app_11.js` file (line 1) imports a `menu` module: `import { menu } from './data.js'`. The `data.js` file (line 1) exports a `menu` array: `export const menu = [`. The `app_11.js` file also contains logic to render the menu items into the DOM.

```
index.html - 1112-1N-js-demo-211411011 - Visual Studio Code

EXPLORER
1112-1N-JS-DEMO-211411011
├── demo
│   ├── w05-menu
│   ├── w01
│   ├── w02
│   ├── w03
│   ├── w04
│   ├── w05
│   ├── images
│   ├── app_11.js
│   └── data.js
├── index.html
├── logo.svg
├── main.png
├── styles_11.css
├── notes.txt
└── README.md

index.html
44 <h4>buttermilk pancakes</h4>
45 <h4 class="price">$15.99</h4>
46 </header>
47 <p class="item-text">
48 I'm baby woke mlkshk wolf bitters live-edge blue b
49 freegan copper mug whatever cold-pressed
50 </p>
51 </div>
52 </article> -->
53 </div>
54 </section>
55 <!-- javascript -->
56 <!-- <script src="data.js"></script> -->
57 <script type="module" src="app_11.js"></script>
58 </body>
59 </html>
60

JS app_11.js
1 import { menu } from './data.js'
2
3 const sectionCenter = document.querySelector('.section-center');
4 const btnContainer = document.querySelector('.btn-container');
5
6 const displayMenuItems = (menu) => {
7   let displayMenu = menu.map( (item) => {
8     return `
9     <article class="menu-item">
10       <img src=${item.img} alt=${item.title} class="photo" /
11       <div class="item-info">
12         <header>
13           <h4>${item.title}</h4>
14           <h4 class="price">${item.price}</h4>
15         </header>
16         <p class="item-text">
17           ${item.desc}
18         </p>
19       </div>
20     </article>
21   `);
22   console.log('displayMenu before join', displayMenu);
23   const Menu = displayMenu.join('');
24   console.log('displayMenu after join', Menu);
25   sectionCenter.innerHTML = Menu;
26 }
27
28
29
30 const buttons = ['all', ...new Set(menu.map((item) => item.categ
31 console.log(buttons);
32
33 const displayMenuButtons = (buttons) => {
34   let displayButtons = buttons.map( (button) => {
35     return `
36     <button type="button" class="filter-btn" data-id=${button
37   `);
38   console.log(displayButtons);
39   btnContainer.innerHTML = displayButtons;
40 }
```

dc91386 tutelary1105 Thu Mar 16 18:43:01 2023 +0800 W05-P1: use JavaScript module to import and export

W05-P2: implement all filter buttons with all tests in console.io

The screenshot displays a development environment with three main components:

- Code Editor (Left):** Shows the `app_11.js` file with JavaScript code for filtering menu items. A red box highlights the `filterBtns` logic, which uses `document.querySelectorAll` to find filter buttons and `filterBtns.forEach` to add click event listeners. The code logs the category and updates the menu based on the selected filter.
- Browser Preview (Right):** Shows the rendered web page titled "Our Menu - 211411011". It features a header with filter buttons: "All", "Breakfast", "Lunch", "Dinner", "Dessert", and "Shakes". Below the header is a list of menu items, each with an image, name, and price. The "All" filter is currently selected.
- Console (Bottom):** Shows the output of the application. It includes the `filterBtns` array, the `filterBtns.forEach` loop, and the `filterMenu` function. The console also shows the `filterBtns` array and the `filterMenu` function. The `filterMenu` function is defined as:

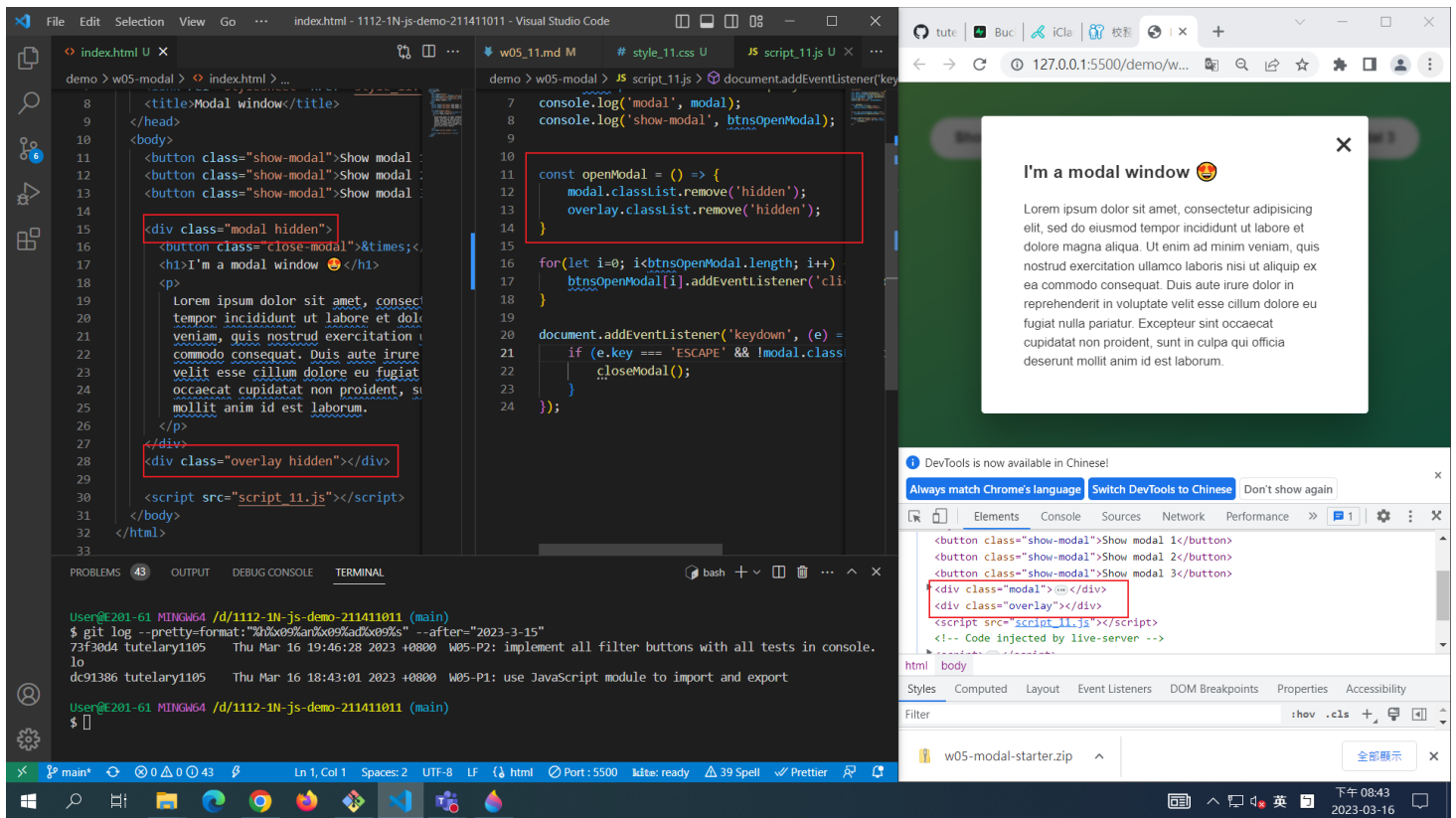
```
const filterMenu = menu.filter((item) => item.category === category);
```

The console log shows the following output:

```
filterBtns
  (6) ["all", "breakfast", "lunch", "dinner", "dessert", "shakes"]
  filterBtns.forEach
    (6) [{"data-id": "all", "category": "all"}, {"data-id": "breakfast", "category": "breakfast"}, {"data-id": "lunch", "category": "lunch"}, {"data-id": "dinner", "category": "dinner"}, {"data-id": "dessert", "category": "dessert"}, {"data-id": "shakes", "category": "shakes"}]
  filterMenu
    (6) [{"data-id": "all", "category": "all"}, {"data-id": "breakfast", "category": "breakfast"}, {"data-id": "lunch", "category": "lunch"}, {"data-id": "dinner", "category": "dinner"}, {"data-id": "dessert", "category": "dessert"}, {"data-id": "shakes", "category": "shakes"}]
```

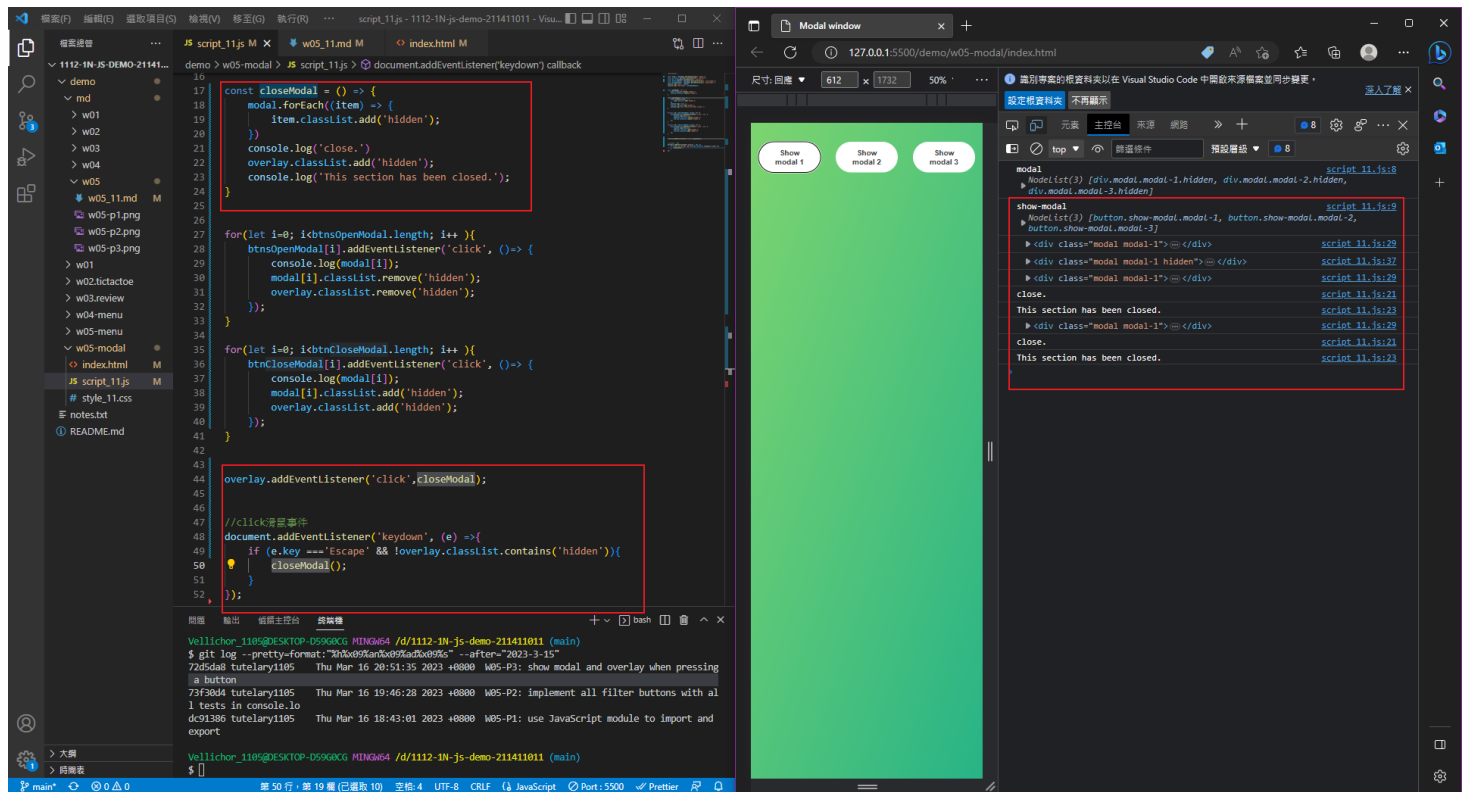
73f30d4 tutelary1105 Thu Mar 16 19:46:28 2023 +0800 W05-P2: implement all filter buttons with all tests in cc

W05-P3: show modal and overlay when pressing a button



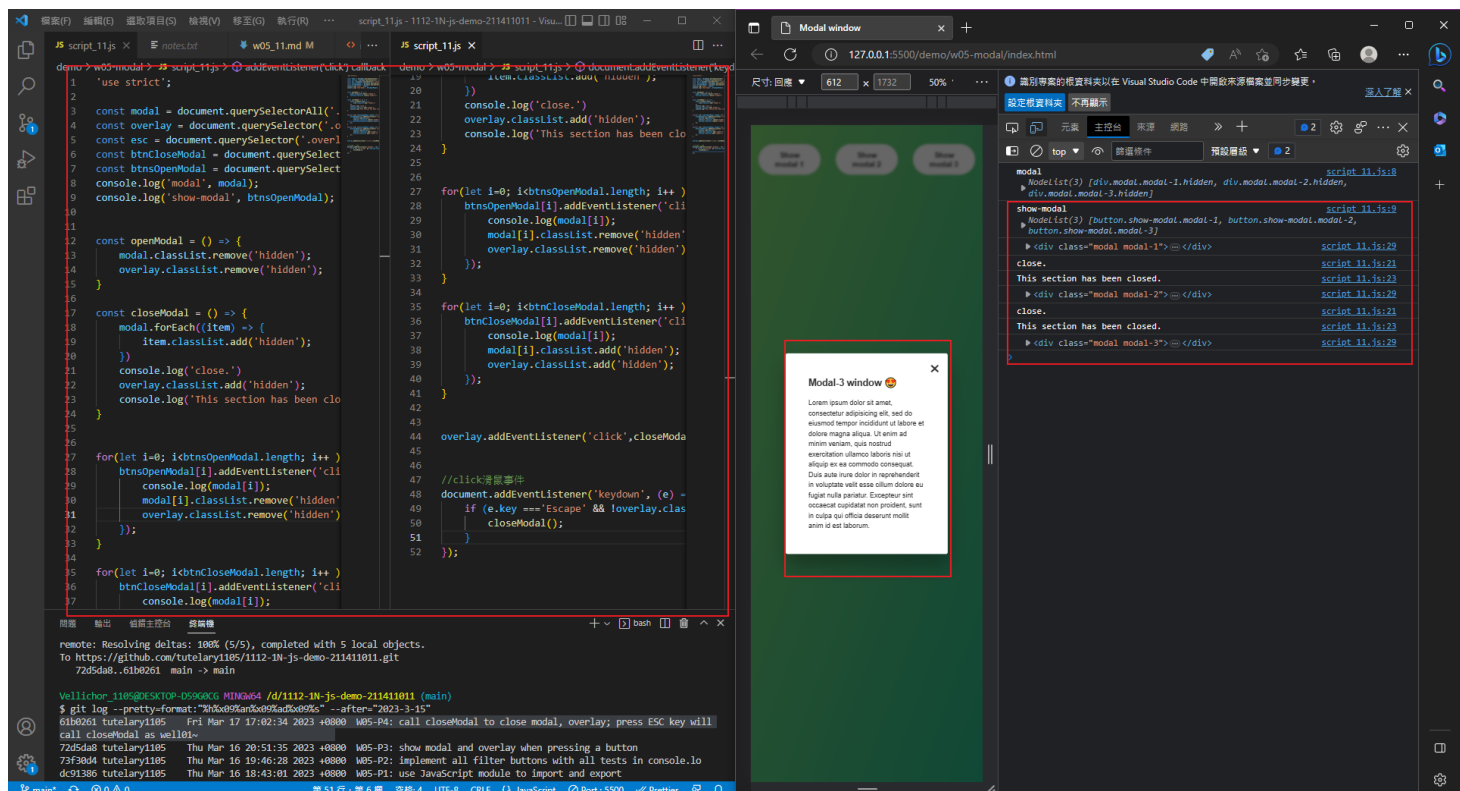
72d5da8 tutelary1105 Thu Mar 16 20:51:35 2023 +0800 W05-P3: show modal and overlay when pressing a button

W05-P4: call closeModal to close modal, overlay; press ESC key will call closeModal as well



61b0261 tutelary1105 Fri Mar 17 17:02:34 2023 +0800 W05-P4: call closeModal to close modal, overlay; press ESC

W05-P5: press 3 different buttons can show 3 different modals



e240adf tutelary1105 Fri Mar 17 17:05:21 2023 +0800 W05-P5: press 3 different buttons can show 3 different modals

W05-logs

```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09s" --after="2023-3-15"
e240adf tutelary1105      Fri Mar 17 17:05:21 2023 +0800    W05-P5:    press 3 different buttons can show 3 different modals
61b0261 tutelary1105      Fri Mar 17 17:02:34 2023 +0800    W05-P4:    call closeModal to close modal, overlay; press ESC key will call closeModal as well01~
72d5da8 tutelary1105      Thu Mar 16 20:51:35 2023 +0800    W05-P3:    show modal and overlay when pressing a button
73f30d4 tutelary1105      Thu Mar 16 19:46:28 2023 +0800    W05-P2:    implement all filter buttons with all tests in console.log
dc91386 tutelary1105      Thu Mar 16 18:43:01 2023 +0800    W05-P1:    use JavaScript module to import and export
```

\$ git log --pretty=format:"%h%x09%an%x09%ad%x09s" --after="2023-3-15"