

DWA_07.4 Knowledge Check_DWA7

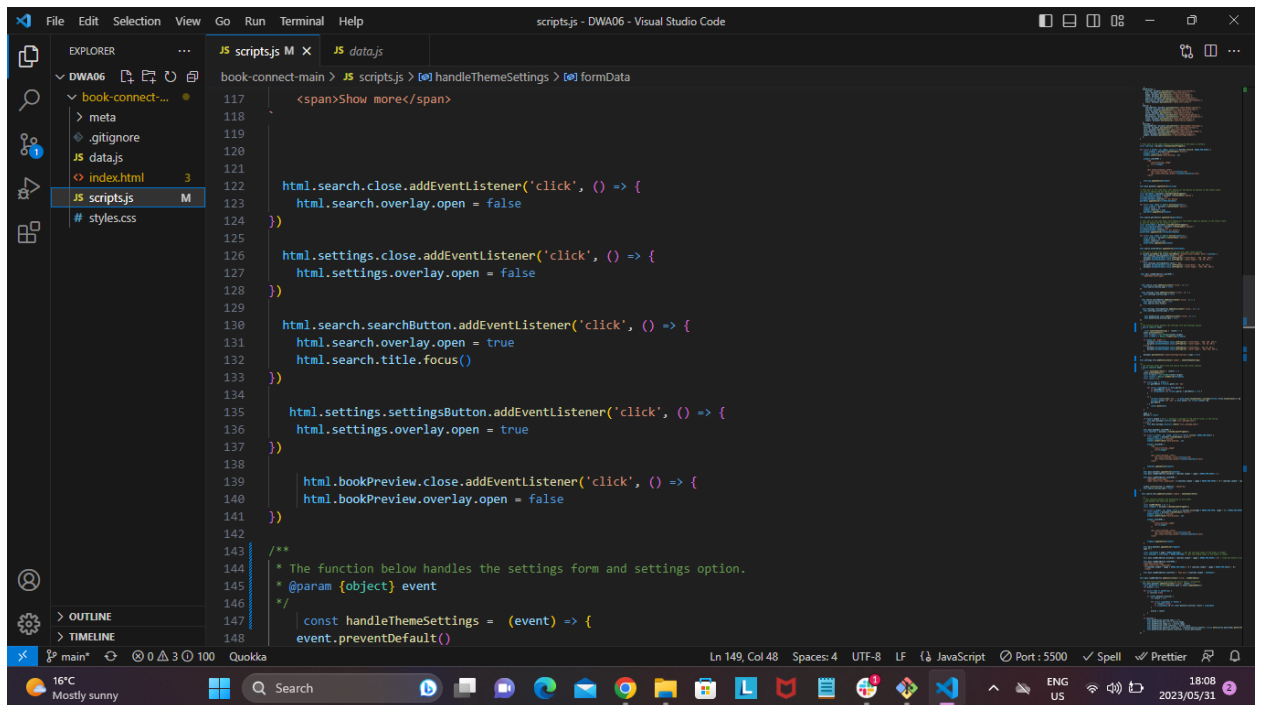
1. Which were the three best abstractions, and why?

- I stored all the elements I retrieved from the DOM in an object, this made it easier to use further down in my code and prevented redundancy.

```
const html = {
  main: {
    booksDiv: document.querySelector('[data-list-items]'),
    loadMoreButton: document.querySelector('[data-list-button]'),
    message: document.querySelector('[data-list-message]')
  },
  bookPreview: {
    overlay: document.querySelector('[ data-list-active]'),
    Blur: document.querySelector('[data-list-blur]'),
    Image: document.querySelector('[ data-list-image]'),
    Title: document.querySelector('[ data-list-title]'),
    Subtitle: document.querySelector('[data-list-subtitle]'),
    description: document.querySelector('[data-list-description]'),
    close: document.querySelector('[data-list-close]')
  },
  search: {
    searchButton: document.querySelector('[data-header-search]'),
    overlay: document.querySelector('[ data-search-overlay]'),
    form: document.querySelector('[data-search-form]'),
    title: document.querySelector('[data-search-title]'),
    genreSelect: document.querySelector('[data-search-genres]'),
    authorSelect: document.querySelector('[ data-search-authors]'),
    close: document.querySelector('[data-search-cancel]'),
    submit: document.querySelector('[data-search-submit]')
  },
  settings: {
    settingsButton: document.querySelector('[data-header-settings]'),
    overlay: document.querySelector('[ data-settings-overlay]'),
    form: document.querySelector('[data-settings-form]'),
    settingsSelect: document.querySelector('[data-settings-theme]'),
    close: document.querySelector('[data-settings-cancel]'),
    submit: document.querySelector('[ data-settings-submit]')
```

```
    },  
  }  
}
```

- All the code in my files is being used and I didn't include any code that is not being used anywhere. This follows the guidelines provided by the Interface segregation principle.



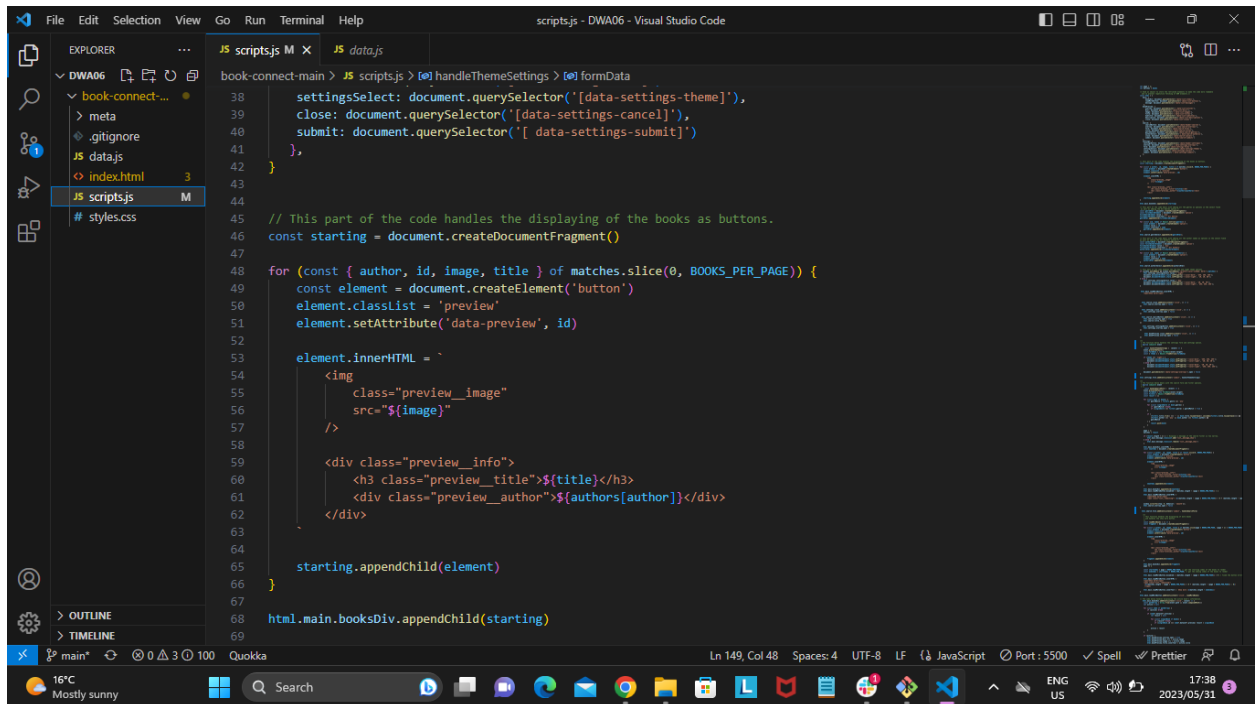
The screenshot shows the Visual Studio Code interface with a project named 'DWA06'. The Explorer sidebar on the left shows the file structure: 'book-connect-main' (containing 'meta', '.gitignore', 'data.js', 'index.html', 'scripts.js', and 'styles.css'). The main editor displays the 'scripts.js' file, which contains JavaScript code for handling theme settings and form data. The code includes event listeners for 'html.search.close', 'html.settings.close', 'html.search.searchButton', 'html.settings.settingsButton', and 'html.bookPreview.close'. A comment at the bottom states: '/* The function below handles the settings form and settings option. */'. The status bar at the bottom indicates 'Ln 149, Col 48', 'Spaces: 4', 'UTF-8', 'LF', 'JavaScript', 'Port: 5500', 'Spell', 'Prettier', and the system clock shows '18:08' on '2023/05/31'.

```
book-connect-main > JS scripts.js > handleThemeSettings > formData  
117     <span>Show more</span>  
118  
119  
120  
121  
122     html.search.close.addEventListener('click', () => {  
123         html.search.overlay.open = false  
124     })  
125  
126     html.settings.close.addEventListener('click', () => {  
127         html.settings.overlay.open = false  
128     })  
129  
130     html.search.searchButton.addEventListener('click', () => {  
131         html.search.overlay.open = true  
132         html.search.title.focus()  
133     })  
134  
135     html.settings.settingsButton.addEventListener('click', () => {  
136         html.settings.overlay.open = true  
137     })  
138  
139     html.bookPreview.close.addEventListener('click', () => {  
140         html.bookPreview.overlay.open = false  
141     })  
142  
143  
144     /**  
145      * The function below handles the settings form and settings option.  
146      * @param {object} event  
147      */  
148     const handleThemeSettings = (event) => {  
        event.preventDefault()  
    }  
}
```

- The data is stored in a separate module (data.js) that only handles the data and the code that handles the actual functionality is stored in a separate file which would be the main js file (scripts.js), this follows the guidelines provided by the

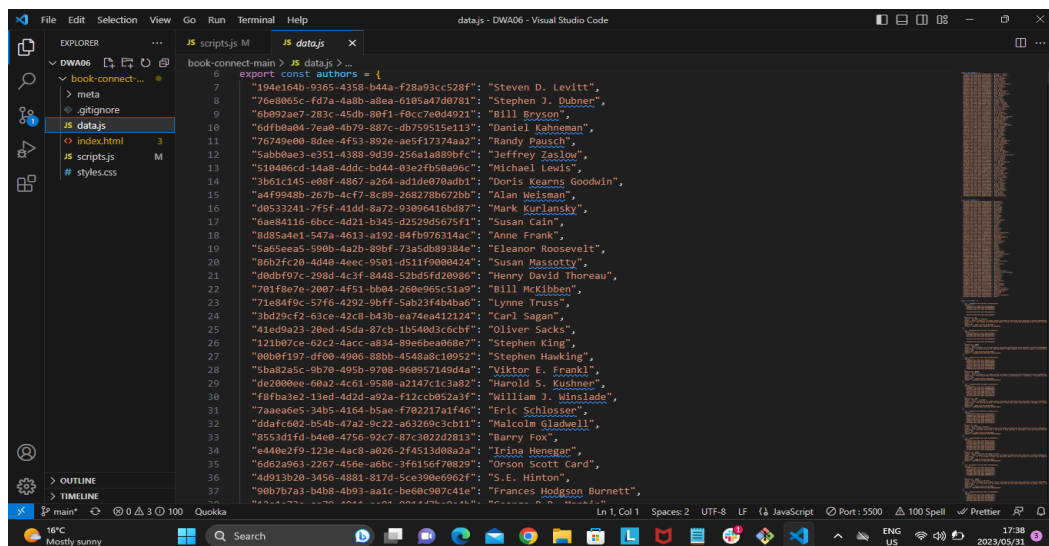
Single responsibility principle (solid).

- Image of the scripts.js file that handles the actual functionality.



```
book-connect-main > JS scripts.js > handleThemeSettings > formData
38
39 settingsSelect: document.querySelector('[data-settings-theme]'),
40 close: document.querySelector('[data-settings-cancel]'),
41 submit: document.querySelector('[data-settings-submit]')
42 },
43 },
44
45 // This part of the code handles the displaying of the books as buttons.
46 const starting = document.createDocumentFragment()
47
48 for (const { author, id, image, title } of matches.slice(0, BOOKS_PER_PAGE)) {
49   const element = document.createElement('button')
50   element.classList = 'preview'
51   element.setAttribute('data-preview', id)
52
53   element.innerHTML = `
54     
58
59     <div class="preview_info">
60       <h3 class="preview_title">${title}</h3>
61       <div class="preview_author">${authors[author]}</div>
62     </div>
63
64   `
65   starting.appendChild(element)
66
67   html.main.booksDiv.appendChild(starting)
68
69 }
```

- Image of the data.js file that handles the data.



```
book-connect-main > JS data.js >
6 export const authors = {
7   "194e164b-9365-b44a-f28a93cc528f": "Steven D. Levitt",
8   "76e8065c-f07a-4a8b-a8ea-6185-a1709781": "Stephen J. Dubner",
9   "6b092ae7-203c-45db-80f1-f6cc7e0d4921": "Bill Bryson",
10  "edfb0a04-7ea0-4b79-887c-db759515e113": "Daniel Kahneman",
11  "76749e00-8dee-4f53-892e-ae5f17374aa2": "Randy Pausch",
12  "5abb0ae3-e351-4388-9d39-256a1a889bfc": "Jeffrey Zaslow",
13  "510406cd-14a8-4ddc-bd44-03e2f5b0a96c": "Michael Lewis",
14  "3b61c145-e08f-4807-a264-ad1de07b0d81": "Doris Kearns Goodwin",
15  "a4f9940b-207b-4c47-8c89-268278b072b5": "Alan Weissman",
16  "d0533241-7f5f-41dd-8a72-93806416bd87": "Mark Kurlansky",
17  "6ae84116-bbcc-4d21-b345-d2529d5675f1": "Susan Cain",
18  "8d954e1-547a-4613-a192-84fb97631aac": "Anne Frank",
19  "5a65ee5-590b-4a2b-89bf-73a5db89384e": "Eleanor Roosevelt",
20  "86b2fc2b-4d48-4eec-9501-d511f0000424": "Susan Massotty",
21  "00dbf97e-200d-4c3f-8440-52b05f2d0906": "Henry David Thoreau",
22  "701f8e7e-2007-4f51-bb04-260e965c51a9": "Bill McKibben",
23  "71e84f9c-57f6-4292-9bff-5ab23f4b4ba6": "Lynne Truss",
24  "3bd29cf2-61ce-42c8-b43b-ea74ea412124": "Carl Sagan",
25  "41ed9a23-20ed-45da-87cb-1b540d3c6cbf": "Oliver Sacks",
26  "121b07ce-62c2-4acc-a834-b9e6bea068e7": "Stephen King",
27  "0009f107-df00-4006-80bb-4548a6c10952": "Stephen Hawking",
28  "5ba82a5c-9b70-495b-9788-968957140d4a": "Viktor E. Frankl",
29  "de2000ee-60a2-4c61-9580-a2147c1c3a82": "Harold S. Kushner",
30  "f8fba3e2-13ed-4d2d-a92a-f12ccb052a3f": "William J. Winslade",
31  "7aae6e5-34b5-4164-b5ae-f782217a1f46": "Eric Schlosser",
32  "ddaf6c02-b54b-47a2-9c22-a63269c3cb11": "Malcolm Gladwell",
33  "855d61f4-b40e-4756-92c7-87c302202813": "Barry Fox",
34  "e440e210-129a-4ac0-a036-2f4515d0082a": "Irina Heneghan",
35  "ed62a063-2207-456e-a0bc-3f6156f70829": "Orson Scott Card",
36  "4d913b20-3456-4881-817d-5c390e6962f2": "S.E. Hinton",
37  "90b7b7a3-b4b8-4b93-aa1c-be08c907c41e": "Frances Hodgson Burnett",
38 }
```

2. Which were the three worst abstractions, and why?

- Using loops to loop over data.
 - It overcomplicates the code and makes it longer because many comments should be used to explain its contents and what it does.
- Incorrectly using JSDOC to document code.
 - instead of making the code easier to understand it makes the code more complex.

3. How can The three worst abstractions be improved via SOLID principles.

- After doing research on higher-order functions, I can say that it is better to use Higher order functions instead of loops as loops are an example of procedural programming and higher-order functions are an example of functional programming that are easier to read and don't require many comments to explain.
 - Correctly use JSDOC to document code as this will comply with the guidelines provided by the dependency inversion principle because you are first writing out a guide to what the code will use and then writing the actual code.
-