

Capstone Project

IWA_19



JS scripts.js > ...

```
1 matches = books
2 page = 1;
3
4 if (!books && !Array.isArray(books)) throw new Error('Source required')
5 if (!range && range.length < 2) throw new Error('Range must be an array with two numbers')
6
7 day = {
8   dark: '10, 10, 20',
9   light: '255, 255, 255',
10 }
11
12 night = {
13   dark: '255, 255, 255',
14   light: '10, 10, 20',
15 }
16
17 fragment = document.createDocumentFragment()
18 const extracted = books.slice(0, 36)
19
20 for ({ author, image, title, id }; extracted; i++) {
21   const preview = createPreview({
22     author,
23     id,
24     image,
25     title
26   })
27   fragment.appendChild(preview)
28 }
29
30
```

JS scripts.js > ...

```
1 // import from data.js
2
3 import { books, authors, genres, BOOKS_PER_PAGE } from "../data.js";
4
5 let page = 1;
6 let matches = books;
7
8 const starting = document.createDocumentFragment();
9
10 for (const { author, id, image, title } of matches.slice(0, BOOKS_PER_PAGE)) {
11   const element = document.createElement("button");
12   element.classList = "preview";
13   element.setAttribute("data-preview", id);
14
15   element.innerHTML = `
16     
20
21     <div class="preview__info">
22       <h3 class="preview__title">${title}</h3>
23       <div class="preview__author">${authors[author]}</div>
24     </div>
25   `;
26
27   starting.appendChild(element);
28 }
29
30 document.querySelector("[data-list-items]").appendChild(starting);
31
```

```

31     any
32   data-list-items.appendChild(fragment)
33
34   genres = document.createDocumentFragment()
35   element = document.createElement('option')
36   element.value = 'any'
37   element = 'All Genres'
38   genres.appendChild(element)
39
40   for ([id, name] of Object.entries(genres)) {
41     const element = document.createElement('option')
42     element.value = id;
43     element.innerText = name;
44     genres.appendChild(element);
45   }
46
47   data-search-genres.appendChild(genres)
48
49   authors = document.createDocumentFragment()
50   element = document.createElement('option')
51   element.value = 'any'
52   element.innerText = 'All Authors'
53   authors.appendChild(element)
54
55   for ([id, name] of Object.entries(authors)) {
56     document.createElement('option')
57     element.value = value
58     element = text
59     authors.appendChild(element)
60   }
61
62

```

JS scripts.js > ...

```

30   document.querySelector("[data-list-items]").appendChild(starting);
31
32   const genreHtml = document.createDocumentFragment();
33   const firstGenreElement = document.createElement("option");
34   firstGenreElement.value = "any";
35   firstGenreElement.innerText = "All Genres";
36   genreHtml.appendChild(firstGenreElement);
37
38   // Loop through the genres object and create an option element for each genre.
39
40   for (const [id, name] of Object.entries(genres)) {
41     const element = document.createElement("option");
42     element.value = id;
43     element.innerText = name;
44     genreHtml.appendChild(element);
45   }
46
47   document.querySelector("[data-search-genres]").appendChild(genreHtml);
48
49   const authorsHtml = document.createDocumentFragment();
50   const firstAuthorElement = document.createElement("option");
51   firstAuthorElement.value = "any";
52   firstAuthorElement.innerText = "All Authors";
53   authorsHtml.appendChild(firstAuthorElement);
54
55   // Loop through the authors object and create an option element for each author.
56
57   for (const [id, name] of Object.entries(authors)) {
58     const element = document.createElement("option");
59     element.value = id;
60     element.innerText = name;
61     authorsHtml.appendChild(element);
62   }
63

```

```
}  
data-search-authors.appendChild(authors)
```

```
data-settings-theme.value === window.matchMedia && window.matchMedia('(prefers-color-scheme: dark)').matches ? 'night' : 'day'  
v = window.matchMedia && window.matchMedia('(prefers-color-scheme: dark)').matches ? 'night' | 'day'
```

```
documentElement.style.setProperty('--color-dark', css[v].dark);  
documentElement.style.setProperty('--color-light', css[v].light);  
data-list-button = `Show more (books.length - BOOKS_PER_PAGE)`
```

```
data-list-button.disabled = !(matches.length - [page * BOOKS_PER_PAGE] > 0)
```

```
data-list-button.innerHTML = /* html */ [  
  '<span>Show more</span>',  
  '<span class="list__remaining"> (${matches.length - [page * BOOKS_PER_PAGE] > 0 ? matches.length -  
]
```

```
data-search-cancel.click() { data-search-overlay.open === false }  
data-settings-cancel.click() { querySelect(data-settings-overlay).open === false }  
data-settings-form.submit() { actions.settings.submit }  
data-list-close.click() { data-list-active.open === false }
```

```
data-list-button.click() {  
  document.querySelector([data-list-items]).appendChild(createPreviewsFragment(matches, page x BOOKS_PER_PAGE,  
  actions.list.updateRemaining()  
  page = page + 1  
}
```

JS scripts.js > ...

```
--  
64 document.querySelector("[data-search-authors]").appendChild(authorsHtml);  
65  
66 // Change the theme based on the user's choice of dark or light mode.  
67  
68 if (  
69   window.matchMedia &&  
70   window.matchMedia("(prefers-color-scheme: dark)").matches  
71 ) {  
72   document.querySelector("[data-settings-theme]").value = "night";  
73   document.documentElement.style.setProperty("--color-dark", "255, 255, 255");  
74   document.documentElement.style.setProperty("--color-light", "10, 10, 20");  
75 } else {  
76   document.querySelector("[data-settings-theme]").value = "day";  
77   document.documentElement.style.setProperty("--color-dark", "10, 10, 20");  
78   document.documentElement.style.setProperty("--color-light", "255, 255, 255");  
79 }  
80  
81 document.querySelector("[data-list-button]").innerText = `Show more (${  
82   books.length - BOOKS_PER_PAGE  
83 })`;   
84 document.querySelector("[data-list-button]").disabled =  
85   matches.length - page * BOOKS_PER_PAGE > 0;  
86  
87 document.querySelector("[data-list-button]").innerHTML = `  
88   <span>Show more</span>  
89   <span class="list__remaining"> | (${  
90     matches.length - page * BOOKS_PER_PAGE > 0  
91     ? matches.length - page * BOOKS_PER_PAGE  
92     : 0  
93   })</span>  
94 `;
```

```

99 data-header-search.click() {
100   data-search-overlay.open === true ;
101   data-search-title.focus();
102 }
103 data-search-form.click(filters) {
104   preventDefault()
105   const formData = new FormData(event.target)
106   const filters = Object.fromEntries(formData)
107   result = []
108
109   for (book; bookslist; i++) {
110     titleMatch = filters.title.trim() = '' && book.title.toLowerCase().includes(filters.title.toLowerCase())
111     authorMatch = filters.author = 'any' || book.author === filters.author
112
113     {
114       genreMatch = filters.genre = 'any'
115       for (genre; book.genres; i++) { if singleGenre = filters.genre { genreMatch === true
116     }
117
118     if titleMatch && authorMatch && genreMatch => result.push(book)
119   }
120
121   if display.length < 1
122   data-list-message.class.add('list__message_show')
123   else data-list-message.class.remove('list__message_show')
124
125   data-list-items.innerHTML = ''
126   const fragment = document.createDocumentFragment()
127   const extracted = source.slice(range[0], range[1])
128
129   for ({ author, image, title, id }; extracted; i++) {
130     const { author: authorId, id, image, title } = props

```

JS scripts.js > ...

```

96 document.querySelector("[data-search-cancel]").addEventListener("click", () => {
97   document.querySelector("[data-search-overlay]").open = false;
98 });
99
100 document
101   .querySelector("[data-settings-cancel]")
102   .addEventListener("click", () => {
103     document.querySelector("[data-settings-overlay]").open = false;
104   });
105
106 document.querySelector("[data-header-search]").addEventListener("click", () => {
107   document.querySelector("[data-search-overlay]").open = true;
108   document.querySelector("[data-search-title]").focus();
109 });
110
111 document
112   .querySelector("[data-header-settings]")
113   .addEventListener("click", () => {
114     document.querySelector("[data-settings-overlay]").open = true;
115   });
116
117 document.querySelector("[data-list-close]").addEventListener("click", () => {
118   document.querySelector("[data-list-active]").open = false;
119 });
120
121 document
122   .querySelector("[data-settings-form]")
123   .addEventListener("submit", (event) => {
124     event.preventDefault();
125     const formData = new FormData(event.target);
126     const { theme } = Object.fromEntries(formData);
127

```

```

135 data-list-items.innerHTML = ``
136 const fragment = document.createDocumentFragment()
137 const extracted = source.slice(range[0], range[1])
138
139 for ({ author, image, title, id }; extracted; i++) {
140   const { author: authorId, id, image, title } = props
141
142   element = document.createElement('button')
143   element.classList = 'preview'
144   element.setAttribute('data-preview', id)
145
146   element.innerHTML = /* html */ `
147     
151
152     <div class="preview__info">
153       <h3 class="preview__title">${title}</h3>
154       <div class="preview__author">${authors[authorId]}</div>
155     </div>
156   `
157
158   fragment.appendChild(element)
159 }
160
161 data-list-items.appendChild(fragments)
162 initial === matches.length - [page * BOOKS_PER_PAGE]
163 remaining === hasRemaining ? initial : 0
164 data-list-button.disabled = initial > 0
165
166 data-list-button.innerHTML = /* html */ `
167   <span>Show more</span>
168   <span class="list__remaining"> (${remaining})</span>

```

JS scripts.js > ...

```

128   if (theme === "night") {
129     document.documentElement.style.setProperty(
130       "--color-dark",
131       "255, 255, 255"
132     );
133     document.documentElement.style.setProperty("--color-light", "10, 10, 20");
134   } else {
135     document.documentElement.style.setProperty("--color-dark", "10, 10, 20");
136     document.documentElement.style.setProperty(
137       "--color-light",
138       "255, 255, 255"
139     );
140   }
141
142   document.querySelector("[data-settings-overlay]").open = false;
143 });
144
145 document
146   .querySelector("[data-search-form]")
147   .addEventListener("submit", (event) => {
148     event.preventDefault();
149     const formData = new FormData(event.target);
150     const filters = Object.fromEntries(formData);
151     const result = [];
152
153     for (const book of books) {
154       let genreMatch = filters.genre === "any";
155
156       for (const singleGenre of book.genres) {
157         if (genreMatch) break;
158         if (singleGenre === filters.genre) {
159           genreMatch = true;
160         }
161       }

```

```

174 data-settings-overlay.submit; {
175     preventDefault()
176     const formData = new FormData(event.target)
177     const result = Object.fromEntries(formData)
178     document.documentElement.style.setProperty('--color-dark', css[result.theme].dark);
179     document.documentElement.style.setProperty('--color-light', css[result.theme].light);
180     data-settings-overlay.open === false
181 }
182
183 data-list-items.click() {
184     pathArray = Array.from(event.path || event.composedPath())
185     active;
186
187     for (node; pathArray; i++) {
188         if active break;
189         const previewId = no
190
191         for (const singleBook of books) {
192             if (singleBook.id === node?.dataset?.preview) {
193                 result = singleBook;
194             }
195         }
196
197         if !active return
198         data-list-active.open ==
199         data-list-blur + data-li
200         data-list-title === acti
201         data-list-subtitle === '
202         data-list-description ==
203     }
204
205     document
206     .querySelector("[data-list-items]")
207     .addEventListener("click", (event) => {
208         const pathArray = Array.from(event.path || event.composedPath());
209         let active = null;
210
211         for (const node of pathArray) {
212             if (active) break;
213
214             if (node?.dataset?.preview) {
215                 let result = null;
216
217                 for (const singleBook of books) {
218                     if (result) break;
219                     if (singleBook.id === node?.dataset?.preview) result = singleBook;
220                 }
221
222                 active = result;
223             }
224         }
225
226         if (active) {
227             document.querySelector("[data-list-active]").open = true;
228             document.querySelector("[data-list-blur]").src = active.image;
229             document.querySelector("[data-list-image]").src = active.image;
230             document.querySelector("[data-list-title]").innerText = active.title;
231             document.querySelector("[data-list-subtitle]").innerText = `${
232                 authors[active.author]
233             } (${new Date(active.published).getFullYear()})`;
234             document.querySelector("[data-list-description]").innerText =
235                 active.description;
236         }
237     });

```

```

229
230 document.querySelector("[data-list-button]").addEventListener("click", () => {
231     const fragment = document.createDocumentFragment();
232
233     for (const { author, id, image, title } of matches.slice(
234         page * BOOKS_PER_PAGE,
235         (page + 1) * BOOKS_PER_PAGE
236     )) {
237         const element = document.createElement("button");
238         element.classList = "preview";
239         element.setAttribute("data-preview", id);
240
241         element.innerHTML = `
242             
246
247             <div class="preview__info">
248                 <h3 class="preview__title">${title}</h3>
249                 <div class="preview__author">${authors[author]}</div>
250             </div>
251         `;
252
253         fragment.appendChild(element);
254     }
255
256     document.querySelector("[data-list-items]").appendChild(fragment);
257     page += 1;
258 });

```