

FINAL PROJECT CAPSTONE: IWA19

...

BY
KEOTSHEPILE MAJE

Grouping my codes according to functionality

What I realised from the given code is that it consists of three different part.

1. The settings
2. Preview of books
3. The search engine

And everything is all mixed up from the code given. I decided to first group this data as in the order above. Since the project Two JavaScript files, I exported the data.js to scripts.js file and added an additional file to hold my html dataset references.

```
1  import { BOOKS_PER_PAGE, authors, genres, books } from "./data.js";  
2  import { htmlDataset } from "./htmlDataset.js";
```

The settings to change Theme

```
16 data-settings-theme.value === window.matchMedia('
    (prefers-color-scheme: dark)').matches ? 'night' : 'day'
17 v = window.matchMedia '&& window.matchMedia('(prefers-color-scheme: dark)').
    matches? 'night' | 'day'
18
19 documentElement.style.setProperty('--color-dark', css[v].dark);
20 documentElement.style.setProperty('--color-light', css[v].light);
21 data-list-button = "Show more (books.length - BOOKS_PER_PAGE)"
22
23 data-settings-overlay.submit; {
24     preventDefault()
25     const formData = new FormData(event.target)
26     const result = Object.fromEntries(formData)
27     document.documentElement.style.setProperty('--color-dark', css[result.theme].
        dark);
28     document.documentElement.style.setProperty('--color-light', css[result.theme].
        light);
29     data-settings-overlay.open === false
30 }
31
```

There was no 'addEventListener' to call the function. The whole logic is incorrect.

```

4 //Change the theme settings
5 htmlDataset.settings.cancel.addEventListener('click', function(e) {
6     htmlDataset.settings.overlay.close()
7 })
8 htmlDataset.header.settings.addEventListener('click', function(e) {
9     htmlDataset.settings.overlay.show()
10 })
11
12 htmlDataset.settings.overlay.addEventListener('submit', function(e) {
13     e.preventDefault();
14     const theme = {
15         night: {
16             dark: '10, 10, 20',
17             light: '255, 255, 255',
18         },
19         day: {
20             dark: '255, 255, 255',
21             light: '10, 10, 20',
22         },
23     }
24     const themePreference = htmlDataset.settings.theme.value
25     if (themePreference === 'day') {
26         document.documentElement.style.setProperty('--color-dark', theme.day.light)
27         document.documentElement.style.setProperty('--color-light', theme.day.dark )
28     }
29     if (themePreference === 'night') {
30         document.documentElement.style.setProperty('--color-dark', theme.night.light)
31         document.documentElement.style.setProperty('--color-light', theme.night.dark );
32     }
33     htmlDataset.settings.overlay.close()
34 }
35
```

Corrected Function and logic

More Description on the book of choice

```
61 const moreAboutBook = (event) => {
62   htmlDataset.list.active.show()
63   const picture = document.querySelector(`[data-image-${event.target.id}]`).getAttribute('src')
64
65   htmlDataset.list.image.setAttribute('src', picture)
66   htmlDataset.list.blur.setAttribute('src', picture)
67   htmlDataset.list.title.innerHTML = document.querySelector(`[data-title-${event.target.id}]`).innerHTML
68   htmlDataset.list.description.innerHTML = document.querySelector(`[data-description-${event.target.id}]`).innerHTML
69   const year = new Date(document.querySelector(`[data-subtitle-${event.target.id}]`).innerHTML).getFullYear()
70   const name = document.querySelector(`[data-author-${event.target.id}]`).innerHTML
71   htmlDataset.list.subtitle.innerHTML = `${name} (${year})`
72 }
73
74
75 let extracted = books.slice(0, 36)
76 for (const { author, image, title, id, description, published } of extracted) {
77   const preview = createPreview({
78     author,
79     id,
80     image,
81     title,
82     description,
83     published,
84   })
85   preview.addEventListener('click', moreAboutBook)
86   fragment.appendChild(preview)
87 }
88
89 document.querySelector('[data-list-items]').appendChild(fragment)
90
```

This function allows the viewer to view and read the description of the book so that it's easy for them to decide if they want to continue to read the book. This function also includes the Title of the book, year it got published and the name of the author.

The 'for of' loop in the code is responsible for looping over all the books allowing us to see more books and there description.

Add more books button

```
105 const bookNotSeen = () => {  
106   const showMore = document.querySelector('[data-list-button]')  
107   const divOFShowMore = document.createElement('div')  
108  
109   const showMoreText = /*html*/  
110     `  
111       <span>Show more</span>  
112       <span class="list__remaining"> (${booksRemaining})</span>  
113     `;  
114   divOFShowMore.innerHTML = showMoreText;  
115   showMore.appendChild(divOFShowMore);  
116   return booksRemaining  
117 }  
118  
119 bookNotSeen() // Show more(1320)  
120
```

This function is the one responsible of adding more books to the page to give the reader/ viewer more options to choose from. Data-list-button is selected and a div element is created to add the span element of show more to. The button will give information on how many books are still left to see and explore.

Add more button

```
120
121 //Function to add more books
122 htmlDataset.list.button.addEventListener('click', function (event) {
123     event.preventDefault();
124     page += 1;
125     booksRemaining = matches.length - [page * BOOKS_PER_PAGE]
126     let rangeLast = page * BOOKS_PER_PAGE
127     let rangeFirst = rangeLast - 36
128     extracted = books.slice(rangeFirst, rangeLast)
129
130     for (const { author, image, title, id, description, published} of extracted) {
131         const preview = createPreview({
132             author,
133             id,
134             image,
135             title,
136             description,
137             published,
138         })
139         preview.addEventListener('click', moreAboutBoook)
140         fragment.appendChild(preview)
141     }
142     document.querySelector('[data-list-items]').appendChild(fragment)
143     bookNotSeen()
144 }
145 )
146
```

Each time the button is pressed the number of books left to explore will decrease by 36 while the number of pages increase by 1.

The search menu form

```
157  /*
158  * Inside the search form.
159  * The option to search for genre.
160  * The option to search for Authors.
161  */
162
163  const genresFragment = document.createDocumentFragment();
164  let genresList = document.createElement('option');
165  genresList.value = 'any';
166  genresList.innerText = 'All genre';
167  genresFragment.appendChild(genresList)
168
169  for (const [id, nameOfGenre] of Object.entries(genres)) {
170    const genresList = document.createElement('option');
171    genresList.value = id;
172    genresList.innerText = nameOfGenre;
173    genresFragment.appendChild(genresList)
174  }
175  htmlDataset.search.genres.appendChild(genresFragment);
176
177  const authorsFragment = document.createDocumentFragment()
178  let authorsList = document.createElement('option')
179  authorsList.value = 'any'
180  authorsList.innerText = 'All Authors'
181  authorsFragment.appendChild(authorsList)
182
183  for (const [id, nameOfAuthor] of Object.entries(authors)) {
184    authorsList = document.createElement('option')
185    authorsList.value = id
186    authorsList.innerText = nameOfAuthor
187    authorsFragment.appendChild(authorsList)
188  }
189  htmlDataset.search.authors.appendChild(authorsFragment);
190
```

The search menu form has three option. To type the title of the book, to choose the book based on the name of the author or the genre.

The option element is created and the author's name or the type of genre is append to the form for the read to choose and select.

Search Functionality

```
194 htmlDataset.search.form.addEventListener('submit', (event)=>{
195     event.preventDefault();
196     htmlDataset.search.overlay.style.display = 'none'
197     document.querySelector('[data-list-items]').innerHTML = ''
198     const formData = new FormData(event.target)
199     const titleSearch = formData.get('title').toLowerCase();
200     const genreSearch = formData.get('genre').toLowerCase();
201     const authorSearch = formData.get('author').toLowerCase();
202     const filteredBooks = [];
203     for (let i = 0; i < books.length; i++) {
204         const book = books[i];
205         if (genreSearch === 'any' && authorSearch === 'any') {
206             if (book.title.toLowerCase().includes(titleSearch)){
207                 filteredBooks.push(book);
208             }
209         }
210         if (genreSearch === 'any') {
211             if (book.title.toLowerCase().includes(titleSearch) && book.author ===
212                 authorSearch){
213                 filteredBooks.push(book);
214             }
215         }
216         if (titleSearch === '') {
217             if (book.author === authorSearch && book.genres.includes(genreSearch))
218                 filteredBooks.push(book);
219             }
220         }
221         if (titleSearch === '' && authorSearch === 'any' ) {
222             if (book.genres.includes(genreSearch)){
223                 filteredBooks.push(book);
224             }
225         }
226     }
227 }
```

The search function will select the books with the characters related to the words typed or the author's name or the genre of the book. And if there is match then those book are pushed to the document to be viewed.

THANK YOU FOR YOU TIME

