

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
28   fragment.appendChild(preview)
29 }
30
```

scripts.js / preview

```
1 const books = [];
2 const BOOKS_PER_PAGE = 12;
3 let page = 1;
4
5 if (!books || !Array.isArray(books)) throw new Error('Source required')
6 if (!range || range.length < 2) throw new Error('Range must be an array with two numbers')
7
8 const css = {
9   day : {
10     dark: '10, 10, 20',
11     light: '255, 255, 255',
12   },
13
14   night : {
15     dark: '255, 255, 255',
16     light: '10, 10, 20',
17   },
18 };
19
20
21 for (let i = 0; i < extracted.length; i++) {
22   const { author, image, title, id } = extracted[i];
23   const preview = createPreview({
24     author,
25     id,
26     image,
27     title
28   })
29
30   fragment.appendChild(preview)
31 }
32
```

```

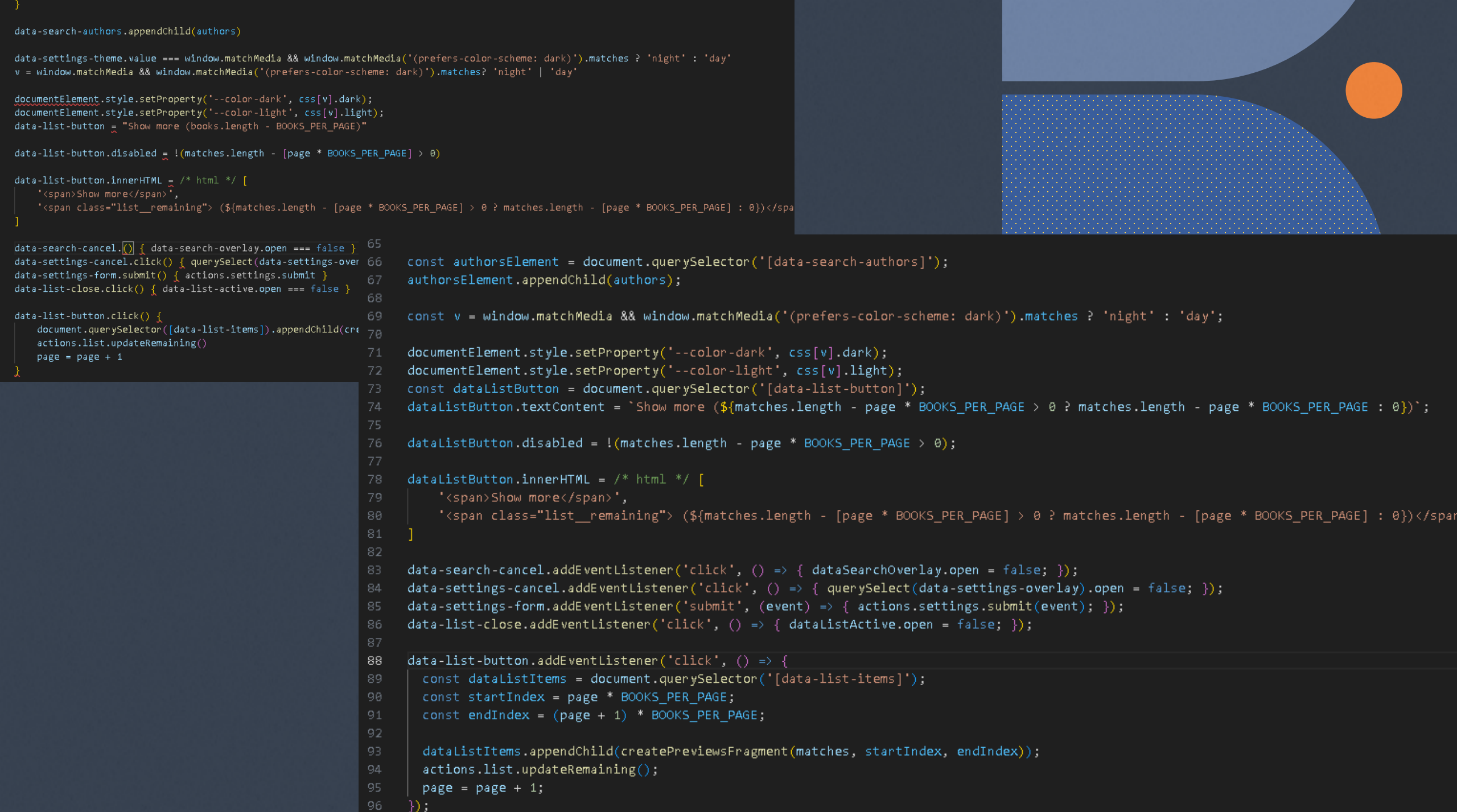
31     any
32 }
33 data-list-items.appendChild(fragment)
34
35 genres = document.createDocumentFragment()
36 element = document.createElement('option')
37 element.value = 'any'
38 element = 'All Genres'
39 genres.appendChild(element)
40
41 for ([id, name] of Object.entries(genres)) {
42     const element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

```

```

32
33 data-list-items.appendChild(fragment)
34 const initial = matches.length - page * BOOKS_PER_PAGE;
35 const remaining = hasRemaining ? initial : 0;
36
37
38 const genres = document.createDocumentFragment()
39 const element = document.createElement('option');
40 element.value = 'any';
41 element = 'All Genres';
42 genres.appendChild(element);
43
44 for ([id, name] of Object.entries(genres)) {
45     const element = document.createElement('option')
46     element.value = id;
47     element.innerText = name;
48     genres.appendChild(element);
49 }
50
51 data-search-genres.appendChild(genres)
52
53 const authors = document.createDocumentFragment()
54 element = document.createElement('option')
55 element.value = 'any'
56 element.innerText = 'All Authors'
57 authors.appendChild(element)
58
59 for ([id, name] of Object.entries(authors)) {
60     document.createElement('option')
61     element.value = value
62     element = text
63     authors.appendChild(element)
64 }
65

```



```
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'
```

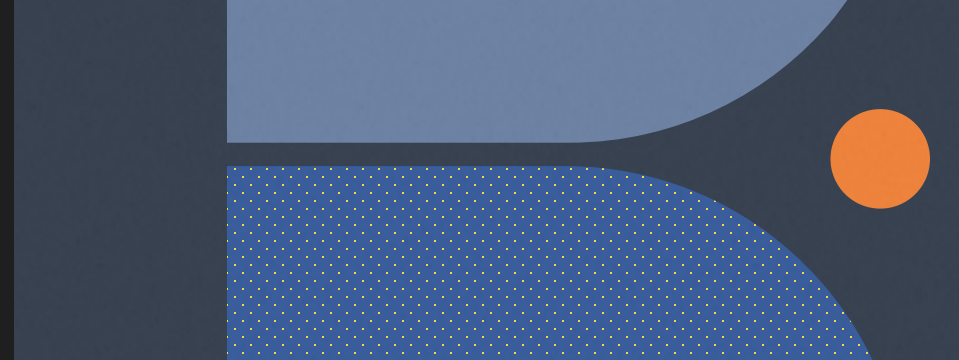
```
documentElement.style.setProperty('--color-dark', css[v].dark);  
documentElement.style.setProperty('--color-light', css[v].light);  
data-list-button = "Show more (books.length - BOOKS_PER_PAGE)"
```

```
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 - [page * BOOKS_PER_PAGE] : 0})</span>'  
]
```

```
data-search-cancel.click() { data-search-overlay.open === false }  
data-settings-cancel.click() { querySelect(data-settings-over  
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, startIndex, endIndex));  
  actions.list.updateRemaining();  
  page = page + 1  
}
```

```
65  
66 const authorsElement = document.querySelector('[data-search-authors]');  
67 authorsElement.appendChild(authors);  
68  
69 const v = window.matchMedia && window.matchMedia('(prefers-color-scheme: dark)').matches ? 'night' : 'day';  
70  
71 documentElement.style.setProperty('--color-dark', css[v].dark);  
72 documentElement.style.setProperty('--color-light', css[v].light);  
73 const dataListButton = document.querySelector('[data-list-button]');  
74 dataListButton.textContent = `Show more (${matches.length - page * BOOKS_PER_PAGE > 0 ? matches.length - page * BOOKS_PER_PAGE : 0})`;  
75  
76 dataListButton.disabled = !(matches.length - page * BOOKS_PER_PAGE > 0);  
77  
78 dataListButton.innerHTML = /* html */ [  
79   '<span>Show more</span>',  
80   '<span class="list__remaining"> (${matches.length - [page * BOOKS_PER_PAGE] > 0 ? matches.length - [page * BOOKS_PER_PAGE] : 0})</span>'  
81 ]  
82  
83 data-search-cancel.addEventListener('click', () => { dataSearchOverlay.open = false; });  
84 data-settings-cancel.addEventListener('click', () => { querySelect(data-settings-overlay).open = false; });  
85 data-settings-form.addEventListener('submit', (event) => { actions.settings.submit(event); });  
86 data-list-close.addEventListener('click', () => { dataListActive.open = false; });  
87  
88 data-list-button.addEventListener('click', () => {  
89   const dataListItems = document.querySelector('[data-list-items]');  
90   const startIndex = page * BOOKS_PER_PAGE;  
91   const endIndex = (page + 1) * BOOKS_PER_PAGE;  
92  
93   dataListItems.appendChild(createPreviewsFragment(matches, startIndex, endIndex));  
94   actions.list.updateRemaining();  
95   page = page + 1;  
96 });
```



```

98 data-header-search.click() {
99   data-search-overlay.open === true ;
100   data-search-title.focus();
101 }
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
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[book.id]
131
132     data-header-search.addEventListener('click', function() {
133       data-search-overlay.open === true ;
134       data-search-title.focus();
135     });
136
137     data-search-form.addEventListener('submit', (event) => {
138       event.preventDefault();
139       const formData = new FormData(event.target);
140       const filters = Object.fromEntries(formData);
141       let result = [];
142
143       for (const book of booksList) {
144         const titleMatch = filters.title.trim() === '' || book.title.toLowerCase().includes(filters.title.toLowerCase());
145         const authorMatch = filters.author === 'any' || book.author === filters.author;
146         let genreMatch = false
147
148         if (filters.genre === 'any'){
149           for (const genre of book.genres) {
150             if (genre === filters.genre) {
151               genreMatch = true;
152               break; }
153           }
154         }
155
156         if (titleMatch && authorMatch && genreMatch) {
157           result.push(book);
158         }
159       }
160
161       if (result.length < 1) {
162         data-list-message.classList.add('list__message_show');
163       } else {
164         data-list-message.classList.remove('list__message_show');
165       }
166     });
  
```

```

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>

```

```

135 dataListItems.innerHTML = '';
136 const fragment = document.createDocumentFragment();
137 const extracted = source.slice(range[0], range[1]);
138
139 for (const { author, image, title, id } of extracted) {
140
141   const element = document.createElement('button');
142   element.classList = 'preview';
143   element.setAttribute('data-preview', id);
144
145   element.innerHTML = /* html */ `
146     
150
151     <div class="preview__info">
152       |   <h3 class="preview__title">${title}</h3>
153       |   <div class="preview__author">${authors[authorId]}</div>
154     </div>
155   `
156
157   fragment.appendChild(element);
158 }
159
160 dataListItems.appendChild(fragments);
161 const initial = matches.length - [page * BOOKS_PER_PAGE];
162 const remaining = hasRemaining ? initial : 0;
163 dataListButton.disabled = initial > 0;
164
165 dataListButton.innerHTML = /* html */ `
166   <span>Show more</span>
167   <span class="list__remaining"> (${remaining})</span>
168 `

```

```

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('--c 174
179     document.documentElement.style.setProperty('--c 175
180     data-settings-overlay).open === false 176
181 } 177
182
183 data-list-items.click() {
184     pathArray = Array.from(event.path || event.comp 178
185     active; 179
186
187     for (node; pathArray; i++) { 180
188         if active break; 181
189         const previewId = node?.dataset?.preview 182
190
191         for (const singleBook of books) { 183
192             if (singleBook.id === id) active = sing 184
193         } 185
194     } 186
195
196     if !active return 187
197     data-list-active.open === true 188
198     data-list-blur + data-list-image === active.ima 189
199     data-list-title === active.title 190
200
201     data-list-subtitle === '${authors[active.author] 191
202     data-list-description === active.description 192
203 } 193
204 194
195
196
197
198
199
200
201
202
203
204

```

```

dataSettingsOverlay.addEventListener('submit', (event) => {
    event.preventDefault();
    const formData = new FormData(event.target);
    const result = Object.fromEntries(formData);
    document.documentElement.style.setProperty('--color-dark', css[result.theme].dark);
    document.documentElement.style.setProperty('--color-light', css[result.theme].light);
    dataSettingsOverlay.open = false
}),

data-list-items.addEventListener('click' , (event) => {
    let active = null;
    const pathArray = Array.from(event.path || event.composedPath());

    for (const node of pathArray) {
        if (active) break;
        const previewId = node?.dataset?.preview;

        for (const singleBook of books) {
            if (singleBook.id === previewId) {
                active = singleBook;
                break;
            }
        }

    }

    if (!active) return;
    dataListActive.open = true;
    dataListBlur.src = active.image;
    dataListTitle.innerText = active.title;

    dataListSubtitle.innerText = `${authors[active.author]} (${Date(active.published).getFullYear()})`;
    dataListDescription = active.description;
})

```