



Book-Connect

[IWA19 Presentation](#)

```

pts.js | index.html | data.js
ripts.js > ...
//imports from data.js
import { books,authors,genres } from "./data.js";

let page = 1;
let range = books.length
if (!books || !Array.isArray(books)) {
  throw new Error('Source required');
}
if (!range || range.length < 2) {
  throw new Error('Range must be an array with two numbers');
}

// Define colors for day and night modes
const day = {
  dark: '10, 10, 20',
  light: '255, 255, 255',
};
const night = {
  dark: '255, 255, 255',
  light: '10, 10, 20',
}

//Create document fragment for previews
const fragment = document.createDocumentFragment()
let startIndex = 0;
let endIndex = 36;
const extracted = books.slice(startIndex, endIndex)

//Loop through the extracted previews and append them to the fragment
for (let i = 0; i < extracted.length; i++) {
  const preview = document.createElement('dl')
  preview.className = 'preview'
  preview.dataset.id = books[i].id
  preview.dataset.title = books[i].title
  preview.dataset.image = books[i].image

```

```

JS code.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 data-list-items.appendChild(fragment)
31
32
33 genres = document.createDocumentFragment()
34 element = document.createElement('option')
35 element.value = 'any'
36 element = 'All Genres'
37 genres.appendChild(element)

```

```

107 // create a new option element for authors
108 const allAuthorsOption = document.createElement('option')
109 allAuthorsOption.value = 'any';
110 allAuthorsOption.textContent = 'All authors'; // use textContent instead of innerText
111 const authorSelect = document.querySelector("[data-search-authors]");
112 authorSelect.appendChild(allAuthorsOption); // add the new option element to the select
113 for (const authorId in authors) {
114   const optionElement = document.createElement('option');
115   optionElement.value = authorId;
116   optionElement.textContent = authors[authorId];
117   authorSelect.appendChild(optionElement);
118 }
119
120 /// create a new option element genre
121 const genreSelect = document.querySelector("[data-search-genres]");
122 const allGenresOption = document.createElement('option');
123 allGenresOption.value = 'any';
124 allGenresOption.textContent = 'All Genres';
125 genreSelect.appendChild(allGenresOption);
126 for (const [genreId, genreName] of Object.entries(genres)) {
127   const optionElement = document.createElement('option');
128   optionElement.value = genreId;
129   optionElement.textContent = genreName;
130   genreSelect.appendChild(optionElement)
131 }
132
133
134
135
136 //change themes
137 const settingOverlay = document.querySelector('[data-header-settings]')
138 const themePopUp = document.querySelector('[data-settings-overlay]')
139 settingOverlay.addEventListener('click', ()=>{
140
141
142
143   const dataSettingsTheme = document.querySelector('[data-settings-theme]')
144   const saveButton = document.querySelector("body > dialog:nth-child(5) > div > div > button.overlay__button.overlay__button_primary")

```

```

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
31 data-list-items.appendChild(fragment)
32
33 genres = document.createDocumentFragment()
34 element = document.createElement('option')
35 element.value = 'any'
36 element.textContent = 'All Genres'
37 genres.appendChild(element)
38
39 for ([id, name]; Object.entries(genres); i++) {
40   document.createElement('option')
41   element.value = value
42   element.textContent = text
43   genres.appendChild(element)
44 }
45
46 data-search-genres.appendChild(genres)
47
48 authors = document.createDocumentFragment()
49 element = document.createElement('option')
50 element.value = 'any'
51 element.textContent = 'All Authors'
52 authors.appendChild(element)
53

```



```
JS scripts.js > ...
25 //Create document fragment for previews
26 const fragment = document.createDocumentFragment()
27 let startIndex = 0;
28 let endIndex = 36;
29 const extracted = books.slice(startIndex, endIndex)
30
31 //Loop through the extracted previews and append them to the fragment
32 for (let i = 0; i < extracted.length; i++) {
33     const preview = document.createElement('div')
34     preview.className = 'preview'
35     preview.dataset.id = books[i].id
36     preview.dataset.title = books[i].title
37     preview.dataset.image = books[i].image
38     preview.dataset.subtitle = `${authors[books[i].author]} (${(new Date(books[i].published)).getFullYear())}`
39     preview.dataset.description = books[i].description
40     preview.dataset.genre = books[i].genres
41     preview.innerHTML = `*html*`
42     <div>
43     <image class='preview_image' src='${books[i].image}' alt='book pic' />
44     </div>
45     <div class='preview_info'>
46     <dt class='preview_title'>${books[i].title}<dt>
47     <dt class='preview_author'> By ${authors[books[i].author]}</dt>
48     </div>
49     fragment.appendChild(preview)
50 }
51 const booklist1 = document.querySelector('[data-list-items]')
52 booklist1.appendChild(fragment)
53
54
55
56
57 // create a new option element for authors
58 const allauthorsOption = document.createElement('option')
59 allauthorsOption.value = 'any';
60 allauthorsOption.textContent = 'All authors'; // use textContent instead of innerText
61 const authorSelect = document.querySelector("[data-search-authors]");
```

```
JS CODEJS > ...
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
31 data-list-items.appendChild(fragment)
32
33 genres = document.createDocumentFragment()
34 element = document.createElement('option')
35 element.value = 'any'
36 element = 'All Genres'
37 genres.appendChild(element)
38
39 for ([id, name]; Object.entries(genres); i++) {
40     document.createElement('option')
41     element.value = value
42     element.innerText = text
43     genres.appendChild(element)
44 }
45
46 data-search-genres.appendChild(genres)
47
48 authors = document.createDocumentFragment()
49 element = document.createElement('option')
50 element.value = 'any'
51 element.innerText = 'All Authors'
52 authors.appendChild(element)
```

scripts.js > ...

```

4 //change themes
5 const settingOverlay = document.querySelector('[data-header-settings]')
6 const themePopup = document.querySelector('[data-settings-overlay]')
7 settingOverlay.addEventListener('click', ()=>{
8   themePopup
9 })
10 const dataSettingsTheme = document.querySelector('[data-settings-theme]')
11 const saveButton = document.querySelector("body > dialog:nth-child(5) > div > div > button.overlay_button.overlay_button_primary")
12 saveButton.addEventListener('click', (event) =>{
13   event.preventDefault()
14   if (dataSettingsTheme.value === 'day') {
15     document.querySelector('body').style.setProperty('--color-dark', day.dark)
16     document.querySelector('body').style.setProperty('--color-light', day.light)
17   }
18   if (dataSettingsTheme.value === 'night') {
19     document.querySelector('body').style.setProperty('--color-dark', night.dark)
20     document.querySelector('body').style.setProperty('--color-light', night.light)
21   }
22 }
23 if(saveButton===dataSettingsTheme){
24   close()
25 }
26 ]]
27 //search button
28
29 //Search button to close and open Search overlay
30 const searchbutton = document.querySelector("[data-header-search]");
31 searchbutton.addEventListener('click', (event) => {
32   document.querySelector("[data-search-overlay]").style.display = "block";
33 })
34 const searchCancel = document.querySelector("[data-search-cancel]");
35 searchCancel.addEventListener('click', (event) => {
36   document.querySelector("[data-search-overlay]").style.display = "none";
37 })
38 // Search function to store more options to chose in genres and authors selects
39 const authorFragment = document.createDocumentFragment();
40 let element = document.createElement('option');

```

JS scripts.js | M X JS CODEJS 9+, U X JS scripts.js C:_VWA_18 index.html M JS data.js

```

JS CODEJS > ...
62
63 data-settings-theme.value === window.matchMedia(window.matchMedia('(prefers-color-scheme: dark)').matches ? 'night' : 'day')
64 v = window.matchMedia(window.matchMedia('(prefers-color-scheme: dark)').matches ? 'night' : 'day')
65
66 documentElement.style.setProperty('--color-dark', css[v].dark);
67 documentElement.style.setProperty('--color-light', css[v].light);
68 data-list-button = "Show more (books.length - BOOKS_PER_PAGE)"
69
70 data-list-button.disabled = !(matches.length - [page * BOOKS_PER_PAGE] > 0)
71
72 data-list-button.innerHTML = /* html */ [
73   '<span>Show more</span>',
74   '<span class="list_remaining"> ${matches.length - [page * BOOKS_PER_PAGE] > 0 ? matches.length - [page * BOOKS_PER_PAGE] : 0}</span>',
75 ]
76
77 data-search-cancel.click() { data-search-overlay.open === false }
78 data-settings-cancel.click() { querySelect(data-settings-overlay).open === false }
79 data-settings-form.submit() { actions.settings.submit }
80 data-list-close.click() { data-list-active.open === false }
81
82 data-list-button.click() {
83   document.querySelector([data-list-items]).appendChild(createPreviewsFragment(matches, page x BOOKS_PER_PAGE, {page + 1} x BOOKS PER PAGE)))
84   actions.list.updateRemaining()
85   page = page + 1
86 }
87
88 data-header-search.click() {
89   data-search-overlay.open === true ;
90   data-search-title.focus();
91 }
92
93 data-search-form.click(filters) {
94   preventDefault()
95   const formData = new FormData(event.target)
96   const filters = Object.fromEntries(formData)
97   result = []
98
99   for (book: booksList: i++) {

```

JS scripts.js > ...

```
151 // Show more button
152 const showMoreButton = document.querySelector('[data-list-button]')
153
154 // Update the text of the "Show More" button to display how many more items will be displayed
155 const numItemsToShow = Math.min(books.length - endIndex, )
156 const showMoreButtonText = `Show More <span style="opacity: 0.5">${numItemsToShow}</span>`
157 showMoreButton.innerHTML = showMoreButtonText
158 showMoreButton.addEventListener('click', () => {
159
160
161 // Increase the start and end indexes by 36 to get the next batch of books to display
162 startIndex += 36;
163 endIndex += 36;
164
165 // Create new variables with the same values as the updated start and end indexes
166 const startIndex1 = startIndex
167 const endIndex1 = endIndex
168
169 // Use the slice method to extract a subset of the books array based on the updated start and end indexes
170 const extracted = books.slice(startIndex1, endIndex1)
171
172 // Loop through the extracted books and create a preview element for each one
173 for (const {author ,image, title, id , description, published} of extracted) {
174
175 // Create a preview element and set its class and dataset attributes
176 const preview = document.createElement('div')
177 preview.className = 'preview'
178 preview.dataset.id = id
179 preview.dataset.title = title
180 preview.dataset.image = image
181 preview.dataset.subtitle = `${authors[author]} (${(new Date(published)).getFullYear()})`
182 preview.dataset.description = description
183
184 // Set the innerHTML of the preview element to display the book image, title, and author
185 preview.innerHTML = `/*html*/`
186 <div>
187 <img class='preview image' src='${image}' alt='book pic'>/>
```

JS CODE.JS > ...

```
63 data-settings-theme.value === window.matchMedia( && window.matchMedia('(prefers-color-scheme: dark)').matches ? 'night' : 'day'
64 v = window.matchMedia( && window.matchMedia('(prefers-color-scheme: dark)').matches ? 'night' | 'day'
65
66 documentElement.style.setProperty('--color-dark', css[v].dark);
67 documentElement.style.setProperty('--color-light', css[v].light);
68 data-list-button = "Show more (books.length - BOOKS_PER_PAGE)"
69
70 data-list-button.disabled = !(matches.length - [page * BOOKS_PER_PAGE] > 0)
71
72 data-list-button.innerHTML = `/* html */ [
73   '<span>Show more</span>',
74   '<span class="list_remaining"> ${matches.length - [page * BOOKS_PER_PAGE] > 0 ? matches.length - [page * BOOKS_PER_PAGE] : 0}</span>',
75 ]
76
77 data-search-cancel.click() { data-search-overlay.open === false }
78 data-settings-cancel.click() { querySelect(data-settings-overlay).open === false }
79 data-settings-form.submit() { actions.settings.submit }
80 data-list-close.click() { data-list-active.open === false }
81
82 data-list-button.click() {
83   document.querySelector([data-list-items]).appendChild(createPreviewsFragment(matches, page x BOOKS_PER_PAGE, {page + 1} x BOOKS_PER_PAGE)))
84   actions.list.updateRemaining()
85   page = page + 1
86 }
87
88 data-header-search.click() {
89   data-search-overlay.open === true ;
90   data-search-title.focus();
91 }
92
93 data-search-form.click(filters) {
94   preventDefault()
95   const formData = new FormData(event.target)
96   const filters = Object.fromEntries(formData)
97   result = []
98
99   for (book: booksList: i++) {
```



```

scripts.js \ M X JS CODE.JS 9+, U JS scripts.js C:\_VWA_18 < index.html M JS data.js
JS scripts.js > ...
151 // Show more button
152 const showMoreButton = document.querySelector('[data-list-button]')
153
154 // Update the text of the "Show More" button to display how many more items will be displayed
155 const numItemsToShow = Math.min(books.length - endIndex, )
156 const showMoreButtonText = `Show More <span style="opacity: 0.5">${numItemsToShow}</span>`
157 showMoreButton.innerHTML = showMoreButtonText
158 showMoreButton.addEventListener('click', () => {
159
160
161 // Increase the start and end indexes by 36 to get the next batch of books to display
162 startIndex += 36;
163 endIndex += 36;
164
165 // Create new variables with the same values as the updated start and end indexes
166 const startIndex1 = startIndex
167 const endIndex1 = endIndex
168
169 // Use the slice method to extract a subset of the books array based on the updated start and end indexes
170 const extracted = books.slice(startIndex1, endIndex1)
171
172 // Loop through the extracted books and create a preview element for each one
173 for (const {author, image, title, id, description, published} of extracted) {
174
175 // Create a preview element and set its class and dataset attributes
176 const preview = document.createElement('div')
177 preview.className = 'preview'
178 preview.dataset.id = id
179 preview.dataset.title = title
180 preview.dataset.image = image
181 preview.dataset.subtitle = `${authors[author]} (${(new Date(published)).getFullYear()})`
182 preview.dataset.description = description
183
184 // Set the innerHTML of the preview element to display the book image, title, and author
185 preview.innerHTML = `/*html*/
186 <div>
187 
188

```

Ln 132, C

```

scripts.js \ M JS CODE.JS 9+, U X JS scripts.js C:\_VWA_18 < index.html M JS data.js
JS CODE.JS > ...
138
139     fragment.appendChild(element)
140 }
141
142 data-list-items.appendChild(fragments)
143 initial === matches.length - [page * BOOKS_PER_PAGE]
144 remaining === hasRemaining ? initial : 0
145 data-list-button.disabled = initial > 0
146
147 data-list-button.innerHTML = `/* html */`
148     <span>Show more</span>
149     <span class="list_remaining"> ${remaining}</span>
150
151
152 window.scrollTo({ top: 0, behavior: 'smooth' });
153 data-search-overlay.open = false
154 }
155
156 data-settings-overlay.submit; {
157     preventDefault()
158     const formData = new FormData(event.target)
159     const result = Object.fromEntries(formData)
160     document.documentElement.style.setProperty('--color-dark', css[result.theme].dark);
161     document.documentElement.style.setProperty('--color-light', css[result.theme].light);
162     data-settings-overlay.open === false
163 }
164
165 data-list-items.click() {
166     pathArray = Array.from(event.path || event.composedPath())
167     active;
168
169     for (node; pathArray; i++) {
170         if active break;
171         const previewId = node?.dataset?.preview
172
173         for (const singleBook of books) {
174             if (singleBook.id === id) active = singleBook
175

```

```

scripts.js > ...
21 //Settings Cancel button
22 const settingCancel = document.querySelector('[data-settings-cancel]')
23 settingCancel.addEventListener('click', (event) => {
24 document.querySelector("[data-settings-overlay]").style.display = "none";
25 })
26
27
28 //code to display book details
29 const detailsToggle = (event) => {
30     const overlay1 = document.querySelector('[data-list-active]');
31     const title = document.querySelector('[data-list-title]')
32     const subtitle = document.querySelector('[data-list-subtitle]')
33     const description = document.querySelector('[data-list-description]')
34     const image1 = document.querySelector('[data-list-image]')
35     const imageblur = document.querySelector('[data-list-blur]')
36     event.target.dataset.id ? overlay1.style.display = "block" : undefined;
37     event.target.dataset.description ? description.innerHTML = event.target.dataset.description : undefined;
38     event.target.dataset.subtitle ? subtitle.innerHTML = event.target.dataset.subtitle : undefined;
39     event.target.dataset.title ? title.innerHTML = event.target.dataset.title : undefined;
40     event.target.dataset.image ? image1.setAttribute('src', event.target.dataset.image) : undefined;
41     event.target.dataset.image ? imageblur.setAttribute('src', event.target.dataset.image) : undefined;
42 };
43 const detailsClose = document.querySelector('[data-list-close]')
44 detailsClose.addEventListener('click', (event) => {
45 document.querySelector("[data-list-active]").style.display = "none";
46 })
47
48 //Define click to display book information
49
50 const bookclick = document.querySelector('[data-list-items]');
51 bookclick.addEventListener('click', detailsToggle);
52
53
54
55 // Mode Cancel button
56 const cancelModeBtn = document.querySelector('[data-settings-cancel]')
57 cancelModeBtn.addEventListener('click', () =>{

```

Ln 251, Col 52

JS scripts.js \ M	JS CODE.JS 9+, U X	JS scripts.js C:\...JWA_18	<> index.html M	JS data.js
-------------------	--------------------	----------------------------	-----------------	------------

```

JS CODE.JS > ...
155
156 data-settings-overlay.submit; {
157     preventDefault()
158     const formData = new FormData(event.target)
159     const result = Object.fromEntries(formData)
160     document.documentElement.style.setProperty('--color-dark', css[result.theme].dark);
161     document.documentElement.style.setProperty('--color-light', css[result.theme].light);
162     data-settings-overlay.open === false
163 }
164
165 data-list-items.click() {
166     pathArray = Array.from(event.path || event.composedPath())
167     active;
168
169     for (node; pathArray; i++) {
170         if active break;
171         const previewId = node?.dataset?.preview
172
173         for (const singleBook: any {
174             if (singleBook.id === id) active = singleBook
175         })
176     }
177
178     if !active return
179     data-list-active.open === true
180     data-list-blur + data-list-image === active.image
181     data-list-title === active.title
182
183     data-list-subtitle === `${authors[active.author]} (${Date(active.published).year})`
184     data-list-description === active.description
185 }

```