



# Capstone Project IWA19

Book Connections

# DocumentQuerySelectors

- Imported data from another file using and imported function
- Created document query selectors to assigned data to a matching variable

```
file Edit Selection View Go Run Terminal Help scriptsjs - YUSNEL019_FTC2301_Gro
index.html JS scriptsjs M JS data.js styles.css
JS scriptsjs > ...
1 import {BOOKS_PER_PAGE, authors, genres, books } from "../data.js"
2
3 const dataHeaderSearch = document.querySelector('[data-header-search]')
4 const dataHeaderSettings = document.querySelector('[data-header-settings]')
5
6 const dataListItems = document.querySelector('[data-list-items]')
7 const dataListMessage = document.querySelector('[data-list-message]')
8 const dataListButton = document.querySelector('[data-list-button]')
9
10 const dataListActive = document.querySelector('[data-list-active]')
11 const dataListBlur = document.querySelector('[data-list-blur]')
12 const dataListImage = document.querySelector('[data-list-image]')
13 const dataListTitle = document.querySelector('[data-list-title]')
14 const dataListSubtitle = document.querySelector('[data-list-subtitle]')
15 const dataListDescription = document.querySelector('[data-list-description]')
16
17 const dataListClose = document.querySelector('[data-list-close]')
18
19 const dataSearchOverlay = document.querySelector('[data-search-overlay]')
20 const dataSearchForm = document.querySelector('[data-search-form]')
21 const dataSearchTitle = document.querySelector('[data-search-title]')
22
23 const dataSearchGenres = document.querySelector('[data-search-genres]')
24 const dataSearchAuthors = document.querySelector('[data-search-authors]')
25
26 const dataSearchCancel = document.querySelector('[data-search-cancel]')
27
28 const dataSettingsOverlay = document.querySelector('[data-settings-overlay]')
29 const dataSettingsForm = document.querySelector('[data-settings-form]')
30
31 const dataSettingsTheme = document.querySelector('[data-settings-theme]')
32 const dataSettingsCancel = document.querySelector('[data-settings-cancel]')
33
```

# Themes

- For Events Listener i used .close() and showModal() to open and close the theme icon.
- Declared css and fixed the object
- Fixed syntax errors in the EventListener function

```
index.html JS scripts.js X JS data.js styles.css
JS scripts.js > ...
38 //themes
39 const css={
40   day : ['255, 255, 255','10, 10, 20'],
41   night : ['10, 10, 20','255, 255, 255']
42 }
43 //themes
44 dataSettingsTheme.value = window.matchMedia && window.matchMedia('(prefers-color-scheme: dark)').matches ? 'night' : 'day'
45 let v = window.matchMedia && window.matchMedia('(prefers-color-scheme: dark)').matches ? 'night' : 'day'
46 //themes
47 dataSettingsForm.addEventListener("submit", (event) => {
48   event.preventDefault()
49   const formSubmit = new FormData(event.target)
50   const option = Object.fromEntries(formSubmit)
51   if (option.theme === 'night') {
52     document.documentElement.style.setProperty('--color-light', css[option.theme][0])
53     document.documentElement.style.setProperty('--color-dark', css[option.theme][1])
54   } else{
55     document.documentElement.style.setProperty('--color-light', css[option.theme][0])
56     document.documentElement.style.setProperty('--color-dark', css[option.theme][1])
57   }
58   dataSettingsOverlay.close()
59 })
60 //themes
61 dataSettingsCancel.addEventListener("click", () => { //cancel clicked closes settingbar
62   dataSettingsOverlay.close()
63 })
64 //themes
65 dataHeaderSettings.addEventListener("click", () => { //opens settings and focuses on themes
66   dataSettingsTheme.focus()
67   dataSettingsOverlay.showModal()
68 })
69
```

# Display list of books

```
/*
/**
 * for loop below creates a list of books
 */
const fragment = document.createDocumentFragment()
const extracted = books.slice(0, 36)

for (let i = 0; i < extracted.length; i++) {
  const { author: authorId, id, image, title } = extracted[i]

  const extractedBooks = document.createElement('button') //creates a button for button effect
  extractedBooks.classList = 'preview'
  extractedBooks.setAttribute('data-preview', id)

  extractedBooks.innerHTML = `/* html */`
    

    <div class="preview_info">
      <h3 class="preview_title">${title}</h3>
      <div class="preview_author">${authors[authorId]}</div>
    </div>

  fragment.appendChild(extractedBooks)
}
dataListItems.appendChild(fragment)
```

- Fixed the for loop function inside
- Defined `document.createElement("button")` to make a button
- The code captures code out of `data.js` to display a list of books from 0 - 36

# List of genres and authors

- When the search bar is opened . the code allows you to select different genres and authors in the form
- Took content out of for loop and assigned to variable
- Substituted elements with correct names

```
* create List of the genres
*/
const bookGenre = document.createDocumentFragment()
const theGenres = document.createElement('option')
theGenres.value = 'any'
theGenres.textContent = 'All Genres'
bookGenre.appendChild(theGenres)

const genreArray = Object.entries(genres)
for (let i = 0; i < genreArray.length; i++) {
  const [id,name] = genreArray[i]
  const genreOp = document.createElement('option')
  genreOp.value = id
  genreOp.textContent = name
  bookGenre.appendChild(genreOp)
}
dataSearchGenres.appendChild(bookGenre)

/**
* creates list of the author names
*/
const bookAuthors = document.createDocumentFragment()
const theAuthors = document.createElement('option')
theAuthors.value = 'any'
theAuthors.textContent = 'All Authors'
bookAuthors.appendChild(theAuthors)

const authorArray = Object.entries(authors)
for (let i = 0; i < authorArray.length ; i++) {
  const [id, name] = authorArray[i]
  const authOp = document.createElement('option')
  authOp.value = id
  authOp.textContent = name
  bookAuthors.appendChild(authOp)
}
dataSearchAuthors.appendChild(bookAuthors)
```

# List Preview

```
scripts> _
1 //close list items preview
2 dataListClose.addEventListener("click", () => {
3   dataListActive.close()
4 })
5
6
7 //list items preview
8 dataListItems.addEventListener("click", (event) => {
9   const pathArray = Array.from(event.path || event.composedPath())
10  let active = null;
11
12  for (let node of pathArray) {
13    if (active){
14      break;
15    }
16    const previewId = node?.dataset?.preview
17
18    for (const singleBook of books) {
19      if (singleBook.id === previewId){
20        active = singleBook
21        break
22      }
23    }
24  }
25
26  if (!active){
27    return
28  }
29
30  dataListActive.open = true
31  dataListBlur.src, dataListImage.src = active.image
32  dataListTitle.textContent = active.title
33
34  dataListSubtitle.textContent = `${authors[active.author]} (${new Date(active.published).getFullYear()}`
35  dataListDescription.textContent = active.description
36 })
37
```

- Displays books big with all the details of the book when pressed on it
- Fixed the eventsListeners function
- Fixed syntax errors of:
- If statements
- New date()



# Show More button

```
dataListButton.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>`
```

Code displays how much books are covered and “show more” is written on the button.

# Filter books

Although the code here doesn't work . it supposed to filter books then a user filters a book to their liking

```
dataSearchForm.addEventListener("submit", (event) => {  
  event.preventDefault()  
  const formData = new FormData(event.target)  
  const filters = Object.fromEntries(formData)  
  const result = []  
  
  for (let book of booksList.length) {  
    let titleMatch = filters.title.trim() === '' && book.title.toLowerCase().includes(filters.title.toLowerCase())  
    let authorMatch = filters.author === 'any' || book.author === filters.author  
  
    const genreMatch = filters.genre === 'any'  
    for (let genres of book.genres) {  
      if (singleGenre === filters.genre){  
        genreMatch = true  
      }  
    }  
  }  
  
  if (titleMatch && authorMatch && genreMatch){  
    result.push(book)  
  }  
  
  if (display.length < 1){  
    dataListMessage.class.add('list_message_show')  
  } else {  
    dataListMessage.class.remove('list_message_show')  
  }  
}  
  
return result  
}
```

This code supposed to show more books when the showmore button is pressed



# Show more books

This code supposed to show more books when the showmore button is pressed

```
js > ...
dataListItems.innerHTML = ''
const fragment2 = document.createDocumentFragment()
const extracted2 = books.slice(0,36)

for (let i = 0; i < extracted2.length; i++ ) {
  const { author: authorId, id, image, title } = extracted2[i]

  const morebooks = document.createElement('button')
  morebooks.classList = 'preview'
  morebooks.setAttribute('data-preview', id)

  morebooks.innerHTML = /* html */ `
    
    <div class="preview_info">
      <h3 class="preview_title">${title}</h3>
      <div class="preview_author">${authors[authorId]}</div>
    </div>
  `

  fragment2.appendChild(morebooks)
}
dataListItems.appendChild(fragment2)

const hasRemaining = true
const fragment3 = document.createDocumentFragment()
dataListItems.appendChild(fragment3)

const initial = matches.length - (page * BOOKS_PER_PAGE)
const remaining = (initial >= 0 && hasRemaining) ? initial : 0

dataListButton.disabled = (remaining <= 0)

dataListButton.innerHTML = /* html */ `
  <span>Show more</span>
  <span class="list_remaining"> (${remaining})</span>
`

window.scrollTo({ top: 0, behavior: 'smooth' });
dataSearchOverlay.open = false
```