



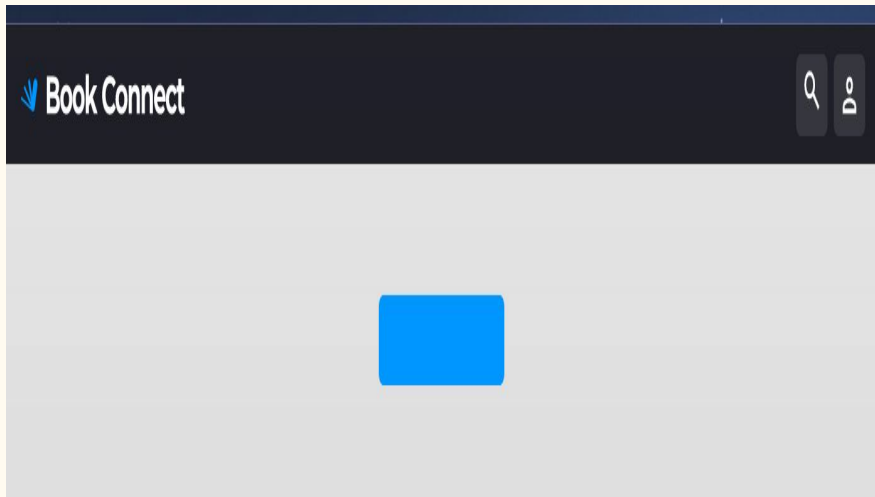
**BOOK CONNECT**

# Steps for debugging a javascript code.

1. Get rid of syntax errors or any red lined code.
2. Always console log especially any code you unfamiliar with before deleting or changing.
3. Read your code to understand the functionality to develop your code  

---

structure



```
import { authors, genres, books, BOOKS_PER_PAGE } from './data.js'
```

```
let matches = books;
```

```
let page = 1;
```

```
let range = [0, 36];
```

```
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')
```

```
/**
```

```
 * THE AUTHOR PARAMETER VARIABLE WILL BE ACCESSING AUTHORS OBJECT VALUES
```

```
 * ID PARAMETER WILL BE USED AS AN OBJECT KEY TO ACCESS ARRAY BOOKS WHICH HAS NESTED ARRAY OBJECTS.
```

```
 * TITLE PARAMETER THAT WILL ACCESS ARRAY BOOKS BY BOOK NAME
```

```
 */
```

```
const listItems = document.querySelector('[data-list-items]')
```

```
function createPreview({ author, id, image, title }) {
```

```
    const preview = document.createElement('button');
```

```
    preview.classList.add('preview');
```

```
    preview.setAttribute('data-preview', id)
```

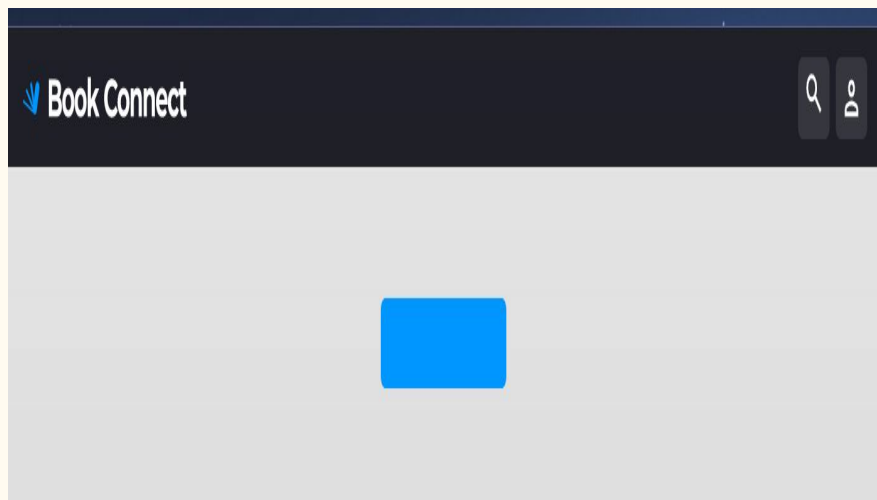
```
    preview.innerHTML = `![${title}>](${image})
```

```
    return preview
```

```
  }
```

```
let extracted = books.slice(0, 36)
```

```
let fragment = document.createDocumentFragment();
```



```
let extracted = books.slice(0, 36)
let fragment = document.createDocumentFragment();
for (const { author, image, title, id } of extracted) {
  const fragment = document.createDocumentFragment()

  let preview = createPreview({
    author,
    id,
    image,
    title
  })
  fragment.appendChild(preview)
}
```



Journeys in English

Bill Bryson



Allegiant

Veronica Roth



Life Without Principle

Henry David Thoreau



Inferno

Dan Brown



Lettre à un otage

Antoine de Saint-Exupéry



Harry Potter and the Goblet of Fire

Mary GrandPré

```
fragment = document.createDocumentFragment()
extracted = books.slice(0, 36)
for (const { author, id, image, title } of extracted) {

  const element = document.createElement('button')
  element.classList = 'preview'
  element.setAttribute('data-preview', id)
  element.innerHTML = /* html */ `

    <div class="preview__info">
      |   <h3 class="preview__title">${title}</h3>
      |   <div class="preview__author">${authors[author]}</div>
    | </div>

  `

  fragment.appendChild(element)
}
listItems.appendChild(fragment)
```



Journeys in English

Bill Bryson



Allegiant

Veronica Roth



Life Without Principle

Henry David Thoreau



Inferno

Dan Brown



Lettre à un otage

Antoine de Saint-Exupéry



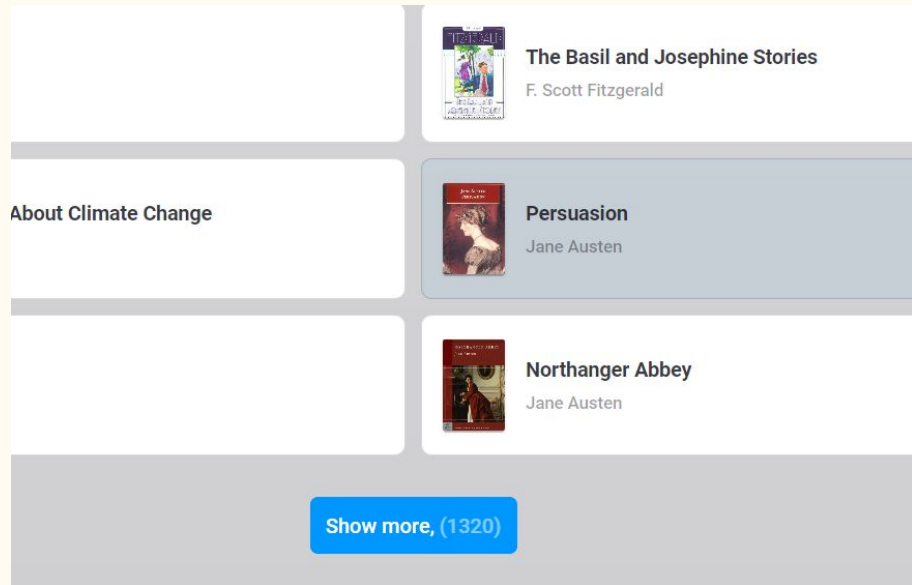
Harry Potter and the Goblet of Fire

Mary GrandPré

```
/**
 * CREATING FRAGMENTS THAT COTAIN DOCUMENT OBJECT ELEMENTS FOR OBJECT GENRE AND OBJECT AUTHOR
 */
const genresFrag = document.createDocumentFragment();
let element = document.createElement('option');
element.value = 'any';
element.innerText = 'All Genres';
genresFrag.appendChild(element);
for (const [id, name] of Object.entries(genres)) {
  element = document.createElement('option');
  element.value = id;
  element.innerText = name;
  genresFrag.appendChild(element);
}
const searchGenre = document.querySelector('[data-search-genres]');
searchGenre.appendChild(genresFrag);

const searchGenre = document.querySelector('[data-search-genres]');
searchGenre.appendChild(genresFrag);

const authorsFrag = document.createDocumentFragment();
let elementA = document.createElement('option');
elementA.value = 'any';
elementA.innerText = 'All Authors';
authorsFrag.appendChild(elementA);
for (const [id, name] of Object.entries(authors)) {
  elementA = document.createElement('option');
  elementA.value = id;
  elementA.innerText = name;
  authorsFrag.appendChild(elementA);
}
const searchAuthors = document.querySelector('[data-search-authors]');
searchAuthors.appendChild(authorsFrag);
```



```
/*
 * LISTENER FUNCTION FOR THAT INCREASES NUMBER OF BOOKS TO BE SCROLLED BASED ON WHAT THE USER WANTS VIEW
 */

listBtn.addEventListener('click', () => {
  const startIndex = page * BOOKS_PER_PAGE;
  const endIndex = (page + 1) * BOOKS_PER_PAGE;
  const fragment = document.createDocumentFragment();

  for (let i = startIndex; i < endIndex && i < matches.length; i++) {
    const book = matches[i];
    const { author, image, title, id } = book;

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

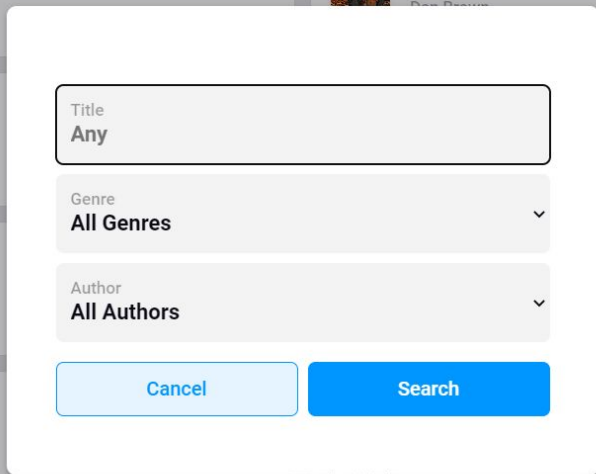
    element.innerHTML = `
    
    <div class="preview_info">
      <h3 class="preview_title">${title}</h3>
      <div class="preview_author">${authors[author]}</div>
    </div>
    `;

    fragment.appendChild(element);
  }

  listItems.appendChild(fragment);

  page += 1;

  const remaining = matches.length - endIndex;
  listBtn.disabled = endIndex >= matches.length;
  listBtn.textContent = `Show more (${remaining})`;
});
```



Title  
**Any**

Genre  
**All Genres** ▼

Author  
**All Authors** ▼

Cancel Search

```
/**
 * THE MY LISTENER FUNCTIONS FOR DOM ELEMENTS
 */

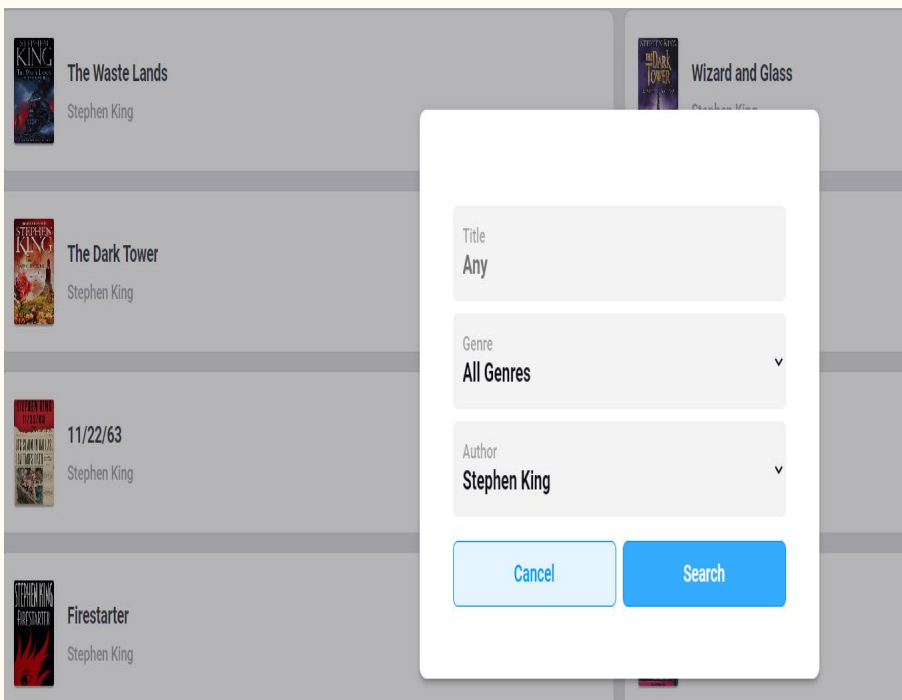
const cancelSearch = document.querySelector('[data-search-cancel]')
cancelSearch.addEventListener('click', () => {

  document.querySelector('[data-search-overlay]').open = false
})

const cancelSett = document.querySelector('[data-settings-cancel]')
cancelSett.addEventListener('click', () => {
  document.querySelector('[data-settings-overlay]').open = false
});

const settForm = document.querySelector('[data-header-settings]')
settForm.addEventListener('click', (e) => {
  e.preventDefault()
  document.querySelector('[data-settings-overlay]').open = true
});
```





```
/**
 * THE SEARCH LISTENER FUNCTION FOR SEARCHING BOOKS BY AUHTOR, GENRE OR ALL SELECTION
 */

const searchForm = document.querySelector('[data-search-form]')
searchForm.addEventListener('submit', (event) => {
  event.preventDefault();

  const formData = new FormData(searchForm);
  const filters = Object.fromEntries(formData);
  const result = [];

  for (let i = 0; i < matches.length; i++) {
    const book = matches[i];

    const titleMatch = filters.title.trim() === '' || book.title.toLowerCase().includes(filters.title.toLowerCase());
    const authorMatch = filters.author === 'any' || book.author === filters.author;
    const genreMatch = filters.genre === 'any' || book.genres.includes(filters.genre);

    if (titleMatch && authorMatch && genreMatch) {
      result.push(book);
    }
  }
}
```

No results found. Your filters might be too narrow.

Show more, (1320)

```
if (result.length < 1) {  
  listMessage.classList.add('list__message_show');  
  listItems.innerHTML = '';  
} else {  
  listMessage.classList.remove('list__message_show');  
  listItems.innerHTML = '';  
  
  const fragment = document.createDocumentFragment();  
  
  for (const book of result) {  
    const { author, image, title, id } = book;  
  
    const element = document.createElement('button');  
    element.classList = 'preview';  
    element.setAttribute('data-preview', id);  
  
    element.innerHTML = `  
      
    <div class="preview__info">  
      <h3 class="preview__title">${title}</h3>  
      <div class="preview__author">${authors[author]}</div>  
    </div>  
  `;  
  fragment.appendChild(element);  
}
```

```
{
```

```
function end
```

```
}
```