DWA_08 Discussion Questions

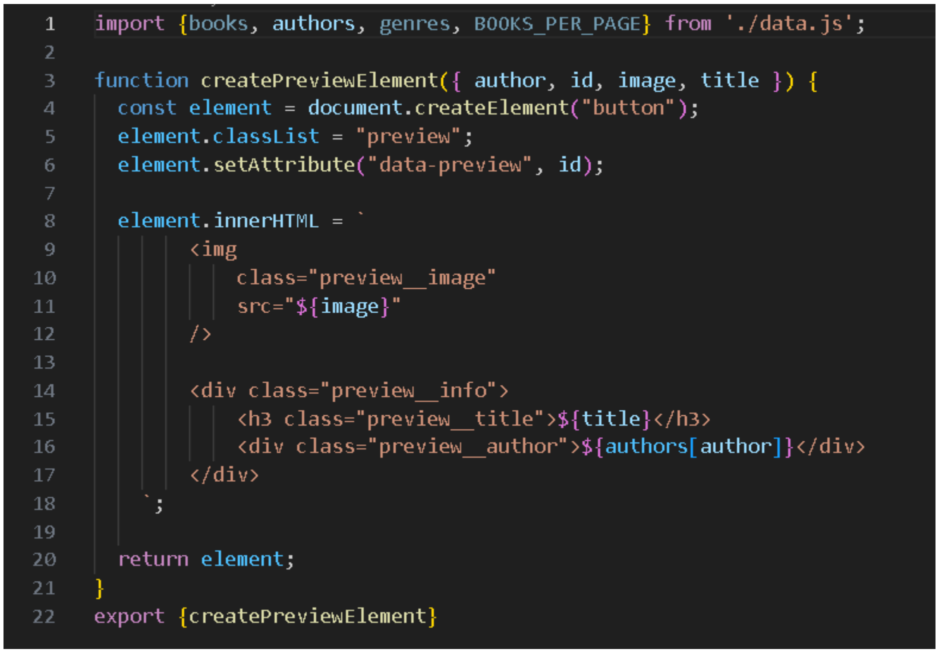
In this module you will continue with your “Book Connect” codebase, and further iterate on your abstractions. You will be required to create an encapsulated abstraction of the book preview by means of a single factory function. If you are up for it you can also encapsulate other aspects of the app into their own abstractions.

To prepare for your session with your coach, please answer the following questions. Then download this document as a PDF and include it in the repository with your code.

---

1. What parts of encapsulating your logic were easy?

Non was easy but i think this was the easiest part

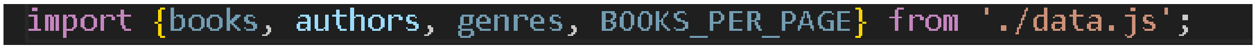


```
import {books, authors, genres, BOOKS_PER_PAGE} from './data.js';

function createPreviewElement({ author, id, image, title }) {
  const element = document.createElement("button");
  element.classList = "preview";
  element.setAttribute("data-preview", id);

  element.innerHTML = `
        <img
          class="preview__image"
          src="${image}"
        />

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

  return element;
}
export {createPreviewElement}
```

2. What parts of encapsulating your logic were hard?

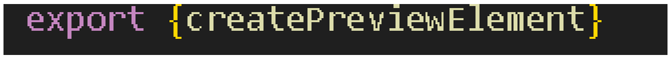
The linking the information between the 2 files:

This is the importing of the information from the data.js file too my encapsulated file (book.js file) =

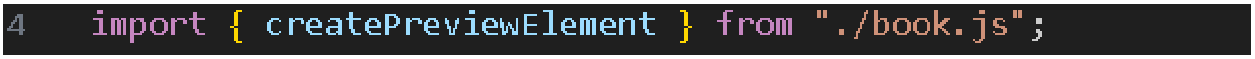


```
import {books, authors, genres, BOOKS_PER_PAGE} from './data.js';
```

This was the linking in my encapsulated (data.js file) file too the calls file (scripts.js file)



```
export {createPreviewElement}
```



```
import { createPreviewElement } from "./book.js";
```

3. Is abstracting the book preview a good or bad idea? Why?

Yes because It hides the inner details and only exposes what's necessary, you can use what's outside without knowing what's inside, keeping things safe and organized.