

# 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?

\*Creating a book preview

```
/**
 * Creates a preview element for a book.
 * @param {Book} book - The book object.
 * @returns {PreviewElement} - The created preview element.
 */
function createBookPreview(book) {
  // ...
}
```

\*Initializing the app.

```
const app = {
  init() {
    // ...
  },
};

// Call the init method to initialize the app
app.init();
```

---

## 2. What parts of encapsulating your logic were hard? Filtering and updating the book list:

\*Filtering and updating the book list:

The logic for filtering the books based on user input and updating the displayed book list involves multiple steps and conditions. It requires iterating over the books, applying filters, and creating a new result array. Additionally, updating the page counter, enabling/disabling the "Show more" button, and updating the remaining count also add complexity. Encapsulating this logic in a clean and efficient manner can be more challenging.

\*Handling the active book details:

The code handles the event when a book preview is clicked, and it retrieves the corresponding book details to display in the active book section. This involves finding the clicked preview element, matching it with the correct book, and updating the active book details. Managing these interactions and encapsulating the logic can be more intricate.

---

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

Abstracting the book preview is a good or bad idea ultimately depends on the specific requirements, complexity, and architecture of the application. In some cases, if the book preview functionality is simple and only used in a single location, abstracting it into a separate component might introduce unnecessary complexity. It's important to consider the trade-offs and evaluate the specific needs of the application before deciding on the level of abstraction.

