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?

1. Factory Functions: Creating factory functions for book previews and the book list was relatively straightforward. Factory functions help encapsulate related properties and methods into separate objects, providing clear boundaries and abstraction.

2. JSDoc Comments: Adding JSDoc comments to document the functions and their parameters is straightforward. JSDoc comments enhance code documentation and make it more understandable and maintainable.

3. Initialization and Validations: Encapsulating the initialization logic for the book list and performing data validations (checking for valid data) was also relatively easy. This ensures that the code starts with valid data.

---

2. What parts of encapsulating your logic were hard?

1. DOM Manipulation: The code involves extensive DOM manipulation and interaction. Encapsulating DOM-related logic can be challenging because the DOM is inherently mutable and not structured as objects with well-defined interfaces. Keeping the DOM manipulation encapsulated and organized can be complex, especially when working with large and complex user interfaces.

2. Event Listeners: Managing event listeners and handling events in a way that maintains encapsulation can be challenging. It's important to ensure that event listeners are appropriately scoped and do not lead to memory leaks or unintended interactions with other parts of the code.

3. Search and Filtering: Encapsulating search and filtering logic can be complex, particularly when dealing with various filter criteria and dynamically updating the displayed results. Ensuring that filters and search operations are encapsulated and reusable can be a challenging task.

---

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

It is a good idea based on the following:

1. Modularity: Encapsulation allows you to encapsulate the internal details and implementation of the book preview within the factory function. This modular approach makes it easier to understand, maintain, and update the code in the future.
2. Reusability: By encapsulating the book preview logic, you can easily reuse the factory function to create multiple instances of book previews throughout your application. This can be particularly useful if you have a dynamic list of books.
3. Separation of Concerns: Encapsulation helps in separating the concerns of creating and interacting with book previews from the rest of your application logic. This separation makes the codebase more organized and easier to manage.