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. **Function Modularity:** The logic for creating the button element and its child elements (image, title, author, and description) is encapsulated within a single function. This promotes modularity by isolating the specific functionality related to creating the button element.

2**. Encapsulating Complexity:** The createButtonElement function abstracts away the complexity of creating the button element and its child elements(image, title, author, description). The inner implementation details are hidden, making it easier to use the function without worrying about the intricacies of the underlying DOM manipulation.

---

2. What parts of encapsulating your logic were hard?

1. **Integration with Existing Code:** Encapsulating the logic in the createButtonElement function may require integrating it with existing code that interacts with the DOM or relies on specific event handlers. Ensuring smooth integration and compatibility with the surrounding code can be a challenge, particularly if there are conflicting approaches or design patterns.
2. **Balancing Generality and Specificity:** Encapsulating the logic requires finding the right balance between generality and specificity. The function should be general enough to be reusable across different contexts and book objects, but also specific enough to fulfill the intended purpose. Striking this balance can be a

challenge, as it often involves making design decisions based on the specific requirements and constraints of your application.

---

3. Is abstracting the book preview a good or bad idea? Why?
It's a good idea because abstracting the book preview separates it from other unrelated code, such as event handling, and this separation of concerns enhances code readability, maintainability, and testability.

But the book preview ain’t cute anymore because I thought it was cool when I had to click the image the preview will just pop out.
