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?

Encapsulating the book preview logic into a function or component was a mixed bag for me, with some parts being straightforward and others potentially challenging.

1. Understanding the Need for Encapsulation: The concept of encapsulation is intuitive and aligns with how we naturally organise things in real life. Understanding the need to bundle related data and functionality together is usually straightforward.
2. Basic JavaScript Syntax: The basic syntax of creating functions or objects is simple enough and is often one of the first things learned when starting with the language.

---

2. What parts of encapsulating your logic were hard?

1. Working with factory functions: with factory functions rely heavily on closures, I struggled with understanding how data gets encapsulated within the closure and how to expose the necessary methods.
2. Refactoring the code: Refactoring my existing code to encapsulate into factory functions and closures was a bit of a challenge to identify other aspects of the app to refactor into web components without introducing bugs.
3. Code organisation: Deciding how to organize the code into different files and modules can be a challenge. It’s important to keep the code organized and modular, but understanding how to do this effectively comes with experience.

---

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

Abstracting the book preview into a component was a good idea for the below reasons:

1. Reusability: If the same code needs to be used in multiple places, having it in a separate function (component) allows me to reuse it without duplicating code. This makes my code more concise and easier to maintain.
2. Readability: By encapsulating the code into a component with a descriptive name made the code easier to understand. Other developers can get a good idea of what the code does just by looking at the function (component) name.
