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 logic using factory functions was relatively easy because of the abstractions I made previously, it allowed for a clear separation of concerns. Creating factory functions to encapsulate creating book previews helped organise the codebase and make it more modular. It also helps to control what can be changed.

---

2. What parts of encapsulating your logic were hard?
Determining the appropriate level of abstraction. Deciding which functionalities should be encapsulated into functions requires careful consideration of the application's structure, dependencies, and future scalability. Initially, I found it challenging to grasp the concept, particularly because creating a single large function seemed counterintuitive to the idea of abstraction. However, I recognised the value in encapsulating functionalities into smaller, more focused functions as it grants greater control.

---

3. Is abstracting the book preview a good or bad idea? Why?
Abstracting the book preview into its own encapsulated function is a good idea. Encapsulation helps in achieving modularity, reusability, and maintainability. By abstracting the book preview functionality, you isolate it from other parts of the application, making it easier to understand, test, and modify independently. It also allows for easier integration with other parts of the application and promotes code reuse, which can lead to more efficient development and easier maintenance in the long run.
