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 functions (abstractions) one by one into factory functions was the easy part because the logic of the actual function remained the exact same. When creating these factory functions, my main goal was to encapsulate and abstract the existing logic into a reusable form, in the sense that I had to modify the functions to accept parameters and potentially return a value or an object, and that was the “easy” part for this application as compared to integrating the factory functions (I will dwell more on that in the following question.)

---

2. What parts of encapsulating your logic were hard?

The challenging part about encapsulating my logic (converting the functions from just abstraction into encapsulated abstractions) was integrating the factory functions into the application’s architecture . Ensuring that the factory functions worked smoothly with other parts of the codebase required a lot of planning and coordination.
Another challenging part for me was making sure that certain factory functions can handle errors gracefully, as a result I had to validate input parameters and handle cases where required parameters were missing or invalid.

---

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

Abstracting the book preview is a good idea, because the preview logic is used in many parts of the application, for example, when a user wants to render more books or when

a user wants to search (or filter) for certain books, the same logic for “book preview” is required, so instead of writing the same logic over and over again, we can reuse the factory function to create book previews throughout the codebase.
Also abstracting the book preview makes the code more readable and maintainable, as it abstract away the low-level details of creating DOM elements, settings attributes and populating content.
