

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?

- Consistent Naming and CSS Classes

The use of consistent naming conventions and CSS classes (book-preview, book-preview__image, book-preview__title, etc.) within the BookElement function helps encapsulate the styling and presentation logic. By following a standardized naming approach, it becomes easier to apply styles and maintain the visual consistency of the book preview elements.

- Encapsulating the book preview logic into a separate function,

By doing so, I can easily reuse it in different parts of the codebase. Whether I need to create book previews in the initial page load or dynamically update them based on user interactions, I can invoke the BookElement function and obtain the desired HTML representation. This modularity allows for easier maintenance and modification of the book preview logic without affecting other parts of the code.

2. What parts of encapsulating your logic were hard?

- Handling Dynamic Data

The book preview relies on dynamic data, such as the book's image, title, and author. When encapsulating the logic, it can be challenging to ensure that the preview elements accurately reflect the current data state. Keeping the preview synchronized with the underlying data requires careful handling and updating of the elements when data changes occur. For example, Making sure that the book preview always shows the correct image, title, and author even when the data changes can be tricky.

- Responsiveness and Adaptability

Encapsulating the book preview logic may involve addressing the responsiveness and adaptability of the preview elements. Ensuring that the preview adjusts its layout, size, or content appropriately based on different screen sizes or device orientations can be challenging. Accommodating responsive behavior within the encapsulated logic may require careful planning and consideration of CSS styles, media queries, or frameworks specifically designed for responsive design. For example, Making the book preview look good and work well on different devices, like phones and tablets, can be hard. It should adjust its size and layout to fit different screens.

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

Separation of concerns - which is breaking down a complex task like breaking down a book preview, into smaller, focused parts. By doing this, we can keep things organized and make it easier to work with. When we encapsulate the book preview logic, we put all the code related to it in one place. This makes it simpler to understand how the book preview works, change its behavior if needed, and fix any issues that may arise. It also helps keep the overall codebase tidy and promotes a modular approach, where different parts of the code are independent and can be developed and maintained separately.
