

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?

The preview function: This function takes in parameters (author, id, image, title) and encapsulates the logic to create a preview button element. It sets the class, attributes, and inner HTML of the button element based on the provided data. The function hides the implementation details of creating the button element and returns the final element.

The fragment variable: This variable is used to create a DocumentFragment, which is an in-memory container for holding a group of DOM elements. By appending elements to the fragment, the code encapsulates the logic of building a set of elements before appending them to the actual DOM. This can provide performance benefits as the elements are added to the DOM only once when the fragment is appended.

These parts of the code encapsulate specific functionality and hide the underlying implementation details. They provide an interface for creating a preview button element (preview function) and a container for efficient element creation (fragment variable).

2. What parts of encapsulating your logic were hard?

Error handling: the logic should handle potential errors or unexpected scenarios gracefully. This includes handling invalid or missing inputs, error conditions within the logic, or any exceptional cases that might occur . Identifying potential error scenarios and implementing appropriate error handling can be a challenging task.

Testing and debugging: When encapsulating logic, it is important to thoroughly test and debug the encapsulated code to ensure it functions as expected. This involves writing test cases, providing various inputs, and verifying the output. Debugging can be challenging if issues arise within the encapsulated logic, as it may require isolating the problematic code and understanding the flow of execution.

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

YES

It improves the readability of the code.

It helps simplify and maintains the code

Reusability-the code reuse and reduces duplication which leads to cleaner and more maintainable .

