

DWA_09 Discussion Questions

In this challenge, you will continue with your “Book Connect” codebase and further iterate on your abstractions.

Previously, you worked on adding abstraction around the book preview functionality of the project. Next, you must turn the book preview abstraction into a fully-working web component. Then, apply the techniques you’ve learned about this module to the book preview.

1. What problems did you encounter converting the book preview to a component?

Setting Up Properties:

- I was not entirely sure about how to properly set up properties within the class to store the bookData and authors information. I know we need to store these properties so that they're accessible throughout the class, also I'm not entirely confident about the correct syntax and structure. I've read about the importance of using the 'this' keyword to reference these properties within the class.

Method Flow:

- Moving Logic to Methods: The original code contains a function called createPreview that handles the creation of the book preview element. I understand that in the class-based approach, we need to encapsulate this logic within a method. However, I'm finding it a bit challenging to understand how to move the existing code into the new method. The sequence of steps involved in creating the preview element is quite complex, and I want to make sure I get it right when transitioning to a method inside the class.

Method Usage:

- Also, I'm getting a hang of the method usage. For example, understanding when and how to use methods like createPreviewElement() and getPreviewElement() is something that might take me a bit of time to fully grasp. I'm eager to learn how these methods fit into the bigger picture of our component's functionality.

2. What other elements make sense to convert into web components? Why?

As I was going through the code, I noticed several parts that could benefit from being converted into web components. By encapsulating elements like the theme selector, search form handling, book list rendering, settings button functionality, and others.

Converting these elements into web components allowed me to promote code reusability, encapsulation, and a more modular design. This approach has the potential to simplify the development efforts by reducing duplicated code and making it easier to maintain specific functionalities.

3. Why does keeping your HTML, CSS and JavaScript in a single file sometimes make sense?

From what I gather, having everything in one file can simplify things, especially for smaller projects. It's easier to distribute and share. But I'm also aware of the downsides. As projects get bigger, maintaining everything in a single file might become challenging. Separating concerns between HTML, CSS, and JavaScript is a good practice for readability.
