DWA_07.4 Knowledge Check_DWA7

---

1. Which were the three best abstractions, and why?

- createBookElement(book): This function abstracts the creation of a book element in the DOM (Document Object Model). It takes a book object as input and generates a button element with appropriate attributes and inner HTML. By encapsulating the creation logic within a function, it improves code readability, promotes reusability, and separates the concerns of book element creation from other parts of the code.

- renderBooks(): This function abstracts the process of rendering a list of books on the page. It calculates the visible books based on the current page and the defined number of books per page. It utilizes the createBookElement function to generate book elements and appends them to the DOM. Additionally, it updates the remaining book count and disables the "Show more" button when there are no more books to display. By abstracting the rendering logic, it makes the code more modular and easier to understand and maintain.

- applyTheme(theme): This function abstracts the application of a theme to the page. It takes a theme parameter and adjusts the CSS variables of the document based on the theme value. It provides a convenient way to switch between different color schemes without directly manipulating the CSS properties throughout the codebase. By abstracting the theme application, it enhances code flexibility and reusability, allowing

---

2. Which were the three worst abstractions, and why?

- handleBookItemClick(event): This function handles the click event on a book item in the list. While it attempts to find the corresponding book based on the dataset value, the implementation relies on traversing the event path array, which can be fragile and browser-dependent. It would be better to abstract this logic into a separate function that performs a more reliable and standardized search for the book using the dataset value. This would improve code readability and maintainability by separating the event handling from the book lookup process.

- handleListButtonClick(): This function handles the click event on the "Show more" button to load additional books. Although it updates the page variable and calls the renderBooks function, the name of the function doesn't clearly convey its purpose. It would be beneficial to rename the function to something like loadMoreBooks to make its intention more explicit. Clear and descriptive naming is important for code readability and understanding.

- initialize(): This function serves as the entry point for initializing the script. While it encapsulates the event listener attachments and initial rendering, it also includes some additional responsibilities, such as applying the theme based on user preference and updating the remaining book count. This violates the single responsibility principle, which states that a function or module should have only one reason to change. It would be better to extract the theme application and book count update logic into separate functions to improve the clarity and maintainability of the code.

---

3. How can The three worst abstractions be improved via SOLID principles.

handleBookItemClick(event):

Single Responsibility Principle (SRP): Extract the book lookup logic into a separate function or class responsible for retrieving book details based on the dataset value. This separation of concerns makes the code more maintainable and testable.

B. handleListButtonClick():

Naming: Rename the function to loadMoreBooks() or handleLoadMoreButtonClick() to provide a clear and descriptive name that conveys its purpose.

C. initialize():
Single Responsibility Principle (SRP): Extract the theme-related logic and book count update into separate functions or classes. This separation allows the initialize() function to focus solely on attaching event listeners and performing the initial rendering.
