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

The main problems were the syntax., it took a while to get used to. There was also bit of confusion regarding the understanding of the difference between creating an instance of the component and rendering it to the DOM. The render method is explicitly called after creating an instance to generate the DOM elements, which must then be manually added to the document.

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

Here are some functions that would make sense to convert into components, because it can further enhance modularity, reusability, and maintainability of the codebase:

Dropdown Component (populateDropdown**)**: This function creates and populates dropdown elements for genres and authors. Turning it into a component would make the dropdown creation process reusable for any other kind of list you might need in the application, such as a list of publishing years, languages, etc. A Dropdown Component could take parameters for the target list, data to populate with, and an initial "all" text, making it versatile for various dropdowns.

Theme Switcher Component (handleSettingsOverlay**)**: This function manages theme switching in the application. Making it a component could encapsulate all theme-related logic, including applying a theme, saving user preferences, and initializing the theme based on user preferences or system settings. This encapsulation would make the theme-switching functionality easy to manage and possibly reuse in different parts of the application or in other projects.

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

Keeping HTML, CSS, and JavaScript in a single file can reduce the amount of files you have in a project and it can group things better in terms of what HTML, CSS, and JavaScript belongs to what specific component you are creating.