

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.

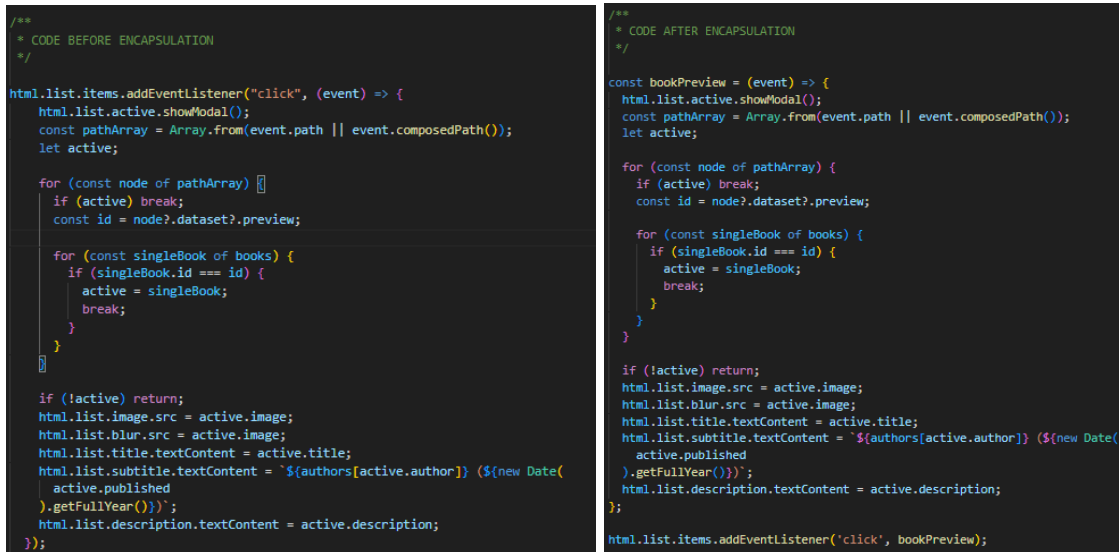
1. What parts of encapsulating your logic were easy?

- Encapsulating the book preview itself was easy.
 - There wasn't much to change besides creating a function and then calling it back.
-

2. What parts of encapsulating your logic were hard?

- In this case it was not hard but I can see the difficulties of encapsulating more complex code on a much larger scale.
-

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



```
/**
 * CODE BEFORE ENCAPSULATION
 */

html.list.items.addEventListener("click", (event) => {
  html.list.active.showModal();
  const pathArray = Array.from(event.path || event.composedPath());
  let active;

  for (const node of pathArray) {
    if (active) break;
    const id = node?.dataset?.preview;

    for (const singleBook of books) {
      if (singleBook.id === id) {
        active = singleBook;
        break;
      }
    }
  }

  if (!active) return;
  html.list.image.src = active.image;
  html.list.blur.src = active.image;
  html.list.title.textContent = active.title;
  html.list.subtitle.textContent = `${authors[active.author]} (${new Date(
    active.published
  ).getFullYear()})`;
  html.list.description.textContent = active.description;
});

/**
 * CODE AFTER ENCAPSULATION
 */

const bookPreview = (event) => {
  html.list.active.showModal();
  const pathArray = Array.from(event.path || event.composedPath());
  let active;

  for (const node of pathArray) {
    if (active) break;
    const id = node?.dataset?.preview;

    for (const singleBook of books) {
      if (singleBook.id === id) {
        active = singleBook;
        break;
      }
    }
  }

  if (!active) return;
  html.list.image.src = active.image;
  html.list.blur.src = active.image;
  html.list.title.textContent = active.title;
  html.list.subtitle.textContent = `${authors[active.author]} (${new Date(
    active.published
  ).getFullYear()})`;
  html.list.description.textContent = active.description;
};

html.list.items.addEventListener('click', bookPreview);
```

- In this case abstracting the book preview function does not matter. The function is called once throughout all the code.
 - Creating an abstraction for the “Show More” feature would make more sense as it’s called more than once throughout the function.
-