

# DWA\_07.4 Knowledge Check\_DWA7

---

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

- The "html" object stores all HTML elements accessed through the DOM. This makes the code easier to understand.
  - The "createBookItems" function automates the creation of book items. It makes the code reusable and maintainable.
  - The "populateDropdown" function automates the population of the dropdown menus from both the genres and authors objects. It makes the code reusable and maintainable.
- 

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

- The "html" object is useful, but complex because it stores all the elements for each component. It would have been better to separate these into different objects for each component(eg. Search option, settings option, etc.)
  - The "themeOptions" object should have been placed in another module as it is related to a separate component and manipulates CSS elements, not html elements like the rest of the code.
  - The "createBookItems" function is good for automation, but contains inline html, which could possibly cause confusion and make the html harder to maintain in the future.
- 

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

- Apply the Single Responsibility Principle by separating the inline html generating component from the data-fetching component of the "createBookItem" function abstraction.
- Apply the Single Responsibility Principle and create separate modules for different components of the codebase.
- Apply the Dependency Inversion Principle by passing values as parameters of functions where possible instead of hardcoding them (or accessing them globally).

