DWA_07.4 Knowledge Check_DWA7

---

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

1. createElement Function : creates elements by taking object arrays and document fragments as arguments.
2. createSelectOption function : creates select menu from object entries
3. Function that opens and closes overlays

Encapsulation of Complexity: It hides implementation details, allowing users of the abstraction to focus on the high-level concepts and tasks without being burdened by detailed internal workings.

Reusability: reusable in different parts of the codebase or even across projects. It solves a specific problem or provides a useful functionality that can be applied in various contexts. Reusability reduces code duplication, improves maintainability, and promotes code efficiency.

---

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

A handle that shows more books from using the createElement function

1. Inadequate Error Handling: An abstraction that fails to handle errors or exceptions properly can lead to unpredictable behavior or failures. If the abstraction lacks appropriate error reporting, validation, or fallback strategies, it may result in fragile code, unexpected failures, and difficulty in troubleshooting.

---

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

Liskov Substitution Principle (LSP): The LSP states that subtypes should be substitutable for their base types without affecting the correctness of the program. If a bad abstraction violates LSP, it can lead to unexpected behavior and errors. Refactoring the abstraction to ensure that derived types adhere to the contract and behavior defined by the base type improves code reliability and consistency.
