DWA_07.4 Knowledge Check_DWA7

---

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

- Modules, the theme.js file.
    - By separating the code into modules, it becomes easier to manage and work with.
    - By creating separate module files and exporting their functionality, we can have an organized and maintainable code.
    -
- Breaking the code into smaller pieces.
    - Smaller code segments are easier to test.

---

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

- Classes
    - Making changes to one part of a class-based system can affect other parts.
    -
- Higher-order functions
    - Using higher-order functions can introduce additional complexity when debugging code. With functions passed as arguments or returned as results, it may be challenging to track the flow of execution and identify issues in the code.
    -

---

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

Single Responsibility Principle (SRP): This principle states that a class should have only one reason to change. By ensuring that each class has a single responsibility, it becomes easier to understand, test, and maintain.

Liskov Substitution Principle (LSP): Higher-order functions should adhere to the LSP by ensuring that functions passed as arguments comply with the expected contract or

interface. This allows for interchangeable function implementations without affecting the behavior of the higher-order function. By following LSP, you can easily substitute

---