DWA_12 Knowledge Check

To complete this Knowledge Check, ensure you have worked through all the lessons in **Module 12: Declarative 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 are the benefits of direct DOM mutations over replacing HTML?

- Animation and transitions: If you want to apply animations or transitions to the DOM changes, direct DOM mutations provide better support. By modifying the individual elements, you have more flexibility in applying animations and transitions using CSS or JavaScript libraries.
- Efficient memory usage: Replacing HTML may require storing and managing additional copies of the HTML structure in memory, especially when dealing with complex pages or frequent updates. With direct DOM mutations, you work directly with the existing DOM, reducing memory usage and potential memory leaks.
- Preservation of state-When you replace HTML, you typically remove the existing elements and replace them with new ones. This can result in the loss of any associated event handlers, data bindings or other states that were associated with those elements. Direct DOM mutations, on the other hand, preserve the state of the modified elements, ensuring that any existing functionality or data binding remains intact.

---

2. What low-level noise do JavaScript frameworks abstract away?

- DOM Manipulation- JavaScript frameworks abstract away the complexities of directly manipulating the DOM.

- Event Handling-Frameworks simplify event handling by providing abstractions that allow developers to define event listeners and handle events in a more structured way.
- State Management-Managing application state can be complex, especially in large and complex applications. JavaScript frameworks often provide state management solutions, such as centralized state stores (e.g Redux in React) or reactive data systems(eg Vue's reactivity system)
- Virtual DOM-Many JavaScript frameworks use a virtual DOM abstraction to optimize rendering and minimize direct DOM manipulations.
- Cross-browser Compatibility- JavaScript frameworks often handle cross-browser compatibility concerns behind the scenes.
- Routing-Frameworks typically include routing capabilities to handle navigation within a single-page application (SPA).

---

3. What essence do JavaScript frameworks elevate?

- Modularity and Component-Based Development: JavaScript frameworks promote a modular approach to building applications.
- Declarative and Reactive Programming: JavaScript frameworks often introduce declarative and reactive programming paradigms. Instead of imperatively describing each step of how the application should be rendered or updated, developers can specify the desired state or structure of the user interface.
- Tooling and Developer Experience: JavaScript frameworks often come with comprehensive tooling ecosystems, including development servers, build systems, package managers and developers's tools. These tools enhance the developer experience by providing features like hot module replacement, code reloading, debugging utilities and performance analysis.

---

4. Very broadly speaking, how do most JS frameworks achieve abstraction?

- Most JavaScript frameworks achieve abstraction by providing higher-level APIs, abstractions, and patterns that shield developers from underlying complexities of the browser’s native JavaScript APIs and DOM Manipulation.
- It uses the following techniques
    1. Component-Based Architecture
    2. Declarative Syntax
    3. Virtual DOM
    4. Event handling Abstractions
    5. State Management solutions
    6. Cross-Browser Compatibility
    7. Tooling and Abstraction

---

5. What is the most important part of learning a JS framework?

- JavaScript frameworks enable developers to work at a higher level of abstraction, focusing on application logic and user interface design rather than dealing with the intricacies of low-level browser APIs and DOM manipulation.
- This abstraction simplifies development, improves code organization and maintainability and enhances the overall developers experience.