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?

Direct DOM mutations can be more efficient and faster compared to completely replacing HTML elements and updating specific parts of the DOM is usually faster than re-rendering the entire HTML.

Direct DOM mutations allow us to selectively update specific elements or properties, which can be useful for optimizing performance or implementing complex interactions.

Direct DOM mutations can preserve the existing state of elements and their associated event listeners or data bindings, avoiding unnecessary reinitialization.

---

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

Frameworks handle DOM manipulation internally, allowing us to focus on application logic rather than manually updating and synchronizing the DOM.

Frameworks provide abstractions that smooth out differences and inconsistencies between different browsers, making it easier to write cross-browser compatible code.

Frameworks often provide simplified and unified event handling mechanisms, abstracting away the complexities of handling various browser events and event listeners.

---

3. What essence do JavaScript frameworks elevate?

Providing a structured and organized approach, frameworks offer a clear structure and predefined patterns for organizing code, separating concerns, and managing application state.

Promoting reusability and modularity, frameworks encourage the creation of reusable components and modules, allowing developers to build complex applications by composing smaller, reusable pieces.

---

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

Introducing a component-based architecture: Frameworks divide the application into reusable components with well-defined responsibilities, allowing developers to work with higher-level abstractions rather than dealing with low-level details.

Providing declarative syntax: Frameworks often use declarative syntax or configuration, where developers describe what they want the application to look like or how it should behave, and the framework handles the underlying implementation details.

Abstracting away DOM manipulation: Frameworks handle DOM updates and synchronization internally, abstracting away the need for manual DOM manipulation code.

Offering abstractions for common tasks: Frameworks provide abstractions and APIs for common tasks such as routing, data management, and UI rendering, reducing the amount of boilerplate code developers need to write.

---

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

Understanding the framework's architecture and how it manages state, handles data flow, and updates the UI.

Learning the framework's component model and how to create, compose, and reuse components effectively.

Becoming familiar with the framework's API and key features, such as routing, data binding, or state management.

Practicing and gaining experience by building small projects or examples using the framework.

Keeping up with the framework's documentation, updates, and community resources to stay informed about best practices and new features.