

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?

Boosts software performance:

Direct DOM mutations make your software work faster because you are making targeted changes and not replacing entire HTML sections which requires less data use or computation.

The following are also related to performance: Low bandwidth usage and optimal user experience. Direct DOM Mutation can reduce the size of data transferred over the network. Direct DOM mutations require less rendering and can make software feel more responsive with visual transitions when interacting with it providing smoother user experience.

Control over changes:

Using direct DOM mutations prevents changing how things were looking or working before when you update specific elements, attributes, or properties.

State preservation:

DOM mutations can ensure that states and bindings are preserved, which can help maintain the expected behavior of your software.

Integration with Frameworks:

Modern JS frameworks and libraries use virtual DOM and encourage using direct DOM mutations to optimise performance and maintainable code.

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

JavaScript frameworks abstract away a variety of low-level noise and complexities that developers would otherwise have to manage manually. Some of the key areas where JavaScript frameworks provide abstraction include but not limited to:

DOM manipulation:

Frameworks abstract manual and direct DOM manipulation to reflect changes in the application state and use virtual DOM or reactive data binding to handle updates

Event/form-handling: Frameworks simplify event handling using mechanisms to attach event listeners and manage event propagation and allows for working with a consistent event API.

Global state/State management:

Complex state management becomes manageable with the management solutions that frameworks offer - handling state changes, ensuring consistent data flow and reducing the need to manage state manually. Handling form validations, submissions, is often abstracted using frameworks.

Web Components:

Frameworks encourage the use of components or modular code that can be reused across the application. This abstraction promotes better code organisation and reusability.

3. What essence do JavaScript frameworks elevate?

JavaScript frameworks elevate the development process by providing structure, consistency, and powerful tools, enabling developers to create robust and feature-rich web applications and provide a better user experience.

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

JavaScript frameworks enable developers to focus on creating solutions/products and building user interfaces without worrying about the underlying complexities of the browser and JavaScript ecosystem. This leads to more maintainable and scalable codebases. They hide away the imperative DOM mutations (changes that occur in the DOM of a web page).

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

Understanding the fundamental principles and concepts of a JS Framework to work with the framework effectively. The architecture of the framework in terms of how components interact, how data flows and how it handles rendering and updates. Learning the state management solution like Redux for React framework.