

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?

Dynamic Updates: The DOM allows you to update the structure and content of a web page dynamically in response to user actions or events. This capability is essential for creating interactive and responsive web applications.

Separation of Concerns: By separating content from style and behavior, you can achieve a more maintainable and cleaner codebase.

Performance: When you make changes to the DOM, the browser can optimize the rendering process by batching multiple changes together, thereby minimizing reflows and repaints. This optimization can significantly improve performance compared to altering HTML content directly.

Event Handling: The DOM provides mechanisms for handling events such as button clicks, enabling interaction within web applications.

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

DOM Manipulation: Instead of manually selecting elements and updating their properties or content, frameworks abstract this process and directly manipulate the DOM. They provide higher-level APIs that simplify these operations.

Event Handling: Handling events across different browsers can be challenging, which is why frameworks offer normalized event handling mechanisms. These abstractions hide the differences between browsers, making event binding and handling much easier.

State Management: Managing states across components can be complex. Frameworks like React and Vue.js provide state management solutions that handle the intricacies of state updates and ensure reactivity.

Optimizations and Performance: Optimizing applications for performance involves tasks such as minimizing DOM reflows and repaints, efficient memory management,

and improving rendering speed. Frameworks often include optimizations by default or provide guidelines for best practices.

3. What essence do JavaScript frameworks elevate?

Abstraction of Complexity: JavaScript frameworks simplify complex tasks like DOM manipulation and state management by providing higher-level APIs, making these operations more efficient.

Efficiency and Performance: JavaScript frameworks improve application performance through features like virtual DOM rendering or reactive updates, enhancing performance and enabling smoother, more responsive interfaces.

Component-based Architecture: Many frameworks promote a component-based architecture, where UI elements and their logic are encapsulated into reusable components. This approach enhances modularity and encourages the development of scalable applications.

Declarative Programming: Frameworks like React emphasize declarative programming paradigms, where developers describe how the UI should look based on state changes, rather than specifying every step of UI updates imperatively.

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

They hide away the imperative DOM mutations

Encapsulation

Declarative APIs

Normalization

Optimizations

Abstraction of State

Ecosystem and tools

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

Fundamental Concepts: this includes state management, components and routing.

Hands-on Practice: gain practical experience by building projects or applications.

Documentation: understand & make yourself familiar with the documentation of the framework

Updates & best practices: stay up to date with the latest updates of the framework you are working with.

Problem Solving: you can gain problem solving skills by debugging your code.