

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?

- a. Performance: Direct DOM mutations are faster and more efficient because they only update the specific parts of the page that need to change, instead of rebuilding the entire HTML structure.
- b. Fine-grained control: With direct DOM mutations, you have more control over individual elements, allowing you to make targeted changes and updates as needed.
- c. Preservation of state: When you directly manipulate the DOM, you can preserve the current state of elements like user inputs, scroll positions, and dynamic data, which would otherwise be lost if you replace the entire HTML structure.

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

- a. Browser compatibility: JavaScript frameworks handle browser inconsistencies and provide a consistent API, so you don't have to worry about writing code that works across different browsers.
 - b. DOM manipulation complexities: Frameworks abstract away the complexities of directly interacting with the DOM, providing simpler and more intuitive methods for manipulating and updating the user interface.
 - c. Event handling: Frameworks often simplify event handling by providing abstractions that make it easier to attach event listeners to elements and handle user interactions.
-

3. What essence do JavaScript frameworks elevate?

Frameworks aim to optimize performance by providing mechanisms to efficiently update and render the user interface, minimizing unnecessary DOM manipulations and re-renders. It also helps developers write code faster and with less effort by providing high-level abstractions, reusable components, and built-in tools for common tasks like routing, state management, and data fetching.

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

a. Providing a higher-level API: Frameworks offer a simplified and expressive API that abstracts away the complexities of low-level operations, such as DOM manipulation, event handling, and AJAX requests.

b. Encapsulating functionality: Frameworks encapsulate common functionalities into reusable components, allowing developers to work with higher-level building blocks rather than manually implementing every aspect of an application.

c. Implementing patterns and conventions: Frameworks establish patterns and conventions that guide developers in structuring their code and organizing their applications, promoting consistency and reducing the cognitive load of decision-making.

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

- Understanding its core concepts: Familiarize yourself with the fundamental concepts of the framework, such as its component-based architecture, state management mechanisms, routing, and data flow.
- Reading the documentation: Thoroughly read and refer to the framework's official documentation, as it provides essential information on how to use the framework, its API, and best practices.
- Building projects and practicing: Gain hands-on experience by building projects using the framework. Practice implementing different features, exploring the framework's capabilities, and solving problems.
- Engaging with the community: Engaging with other developers, asking questions, and sharing knowledge will expand your understanding, provide insights, and keep you updated on the latest developments in the framework ecosystem.