

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?

Performance: direct DOM mutations are faster and more efficient because instead of re-rendering the entire page, specific parts of the DOM are updated when necessary which results in faster and smoother updates and user experience.

Less network needed: By only updating specific parts of the DOM, then amount of data that needs to be transmitted over the network is minimized and this significantly reduces the bandwidth usage and decrease the processing overhead on both the client and server sides.

Improved interactivity: It enables real-time updates and dynamic user interactions because it applies changes directly to the existing DOM, which results in a more immediate response from the website.

State preservation: when the HTML is replaced, that state of the application is lost and needs to be re-initialized. Direct DOM mutation allows a seamless continuation of user interaction without disruption as it preserves the state rather than re-initializing it.

Compatibility and maintainability: In some cases, having to replace or re-initialize the DOM means that if you have event listeners, you are most likely to have to reattach them after replacement.

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

Despite the Benefits listed above, direct DOM manipulation requires more manual handling and can be more complex to implement. This is where Frameworks come in, they help by abstracting away all of this and gives you a more efficient way of doing things.

Frameworks often provide data binding capabilities which means there will be an automatic two-way synchrony between the application state and DOM.

3. What essence do JavaScript frameworks elevate?

In a simple way, by providing a structured framework and reusable components they eliminated reinventing the wheel for common tasks which results in more productive developers.

Frameworks are presented on a structured manner. This enforces patterns and best practices for organizing code, separating concerns, and prompting maintainability.

JavaScript frameworks encourage building reusable components, which can be shared across different parts of an application or even between projects.

They also abstract away browser-specific quirks and inconsistencies, providing a unified programming interface.

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

Most JavaScript frameworks achieve abstraction by providing a layer of abstraction over the underlying technologies and APIs, such as the DOM, browser APIs, and JavaScript itself.

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

Understanding the Core concept is important because it means you will learn the component architecture, data flow, state management, routing, and any other fundamental concepts that the framework is built upon. This initializes a strong foundation and will allow you to effectively leverage the framework's capabilities and design patterns.