

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 provide performance benefits, granularity, preservation of event listeners and data binding, compatibility, and integration with third-party libraries compared to replacing HTML.

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

- JavaScript frameworks abstract away low-level noise such as manual DOM manipulation, browser inconsistencies, event handling, state management, and complex data binding, allowing developers to focus on application logic and higher-level abstractions.

3. What essence do JavaScript frameworks elevate?

- JavaScript frameworks elevate the essence of building web applications by providing structured architectural patterns, modular components, efficient data

flow management, and simplified APIs, enabling developers to focus on creating scalable, maintainable, and feature-rich applications with ease

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

- JavaScript frameworks simplify web development by offering a higher-level API and predefined patterns that hide low-level complexities like DOM manipulation, event handling, data management, and component composition. They provide reusable components and tools for state management, routing, and data binding, allowing developers to focus on application logic and user experience rather than implementation details.
-

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

- The most important part of learning a JavaScript framework is understanding its core concepts and principles, such as its architecture, component structure, data flow, and API usage. By grasping these foundational aspects, you can effectively leverage the framework's features, design patterns, and best practices to build robust and scalable applications. Additionally, exploring the documentation, tutorials, and examples provided by the framework's creators and community will greatly aid in understanding and applying the framework effectively. Continuous practice, hands-on experimentation, and building projects using the framework are also crucial for solidifying your understanding and proficiency with the tool.