

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: with direct DOM you only update specific parts of the DOM that have changed, reducing the overall processing time and improving the performance of your application. It's better than completely reparsing and re-rendering the entire HTML structure.

Efficiency: incremental DOM updates are more efficient in terms of memory usage. With direct DOM mutations you can retain the existing event listeners and styles, preserving the state and avoiding unnecessary memory allocations.

Fine-grained control: directly manipulating the DOM, this allows you to make specific changes to elements or attributes, add or remove elements, modify CSS properties. This reduces the risk of unintended side effects and improves the maintainability of code.

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

JavaScript frameworks abstract away several low-level noise, providing higher-level abstractions and simplifying development. Some of the low-level noise that frameworks handle include:

DOM manipulation - They provide APIs and abstractions that allow developers to interact with the DOM in a more declarative and intuitive manner. This eliminates the need to manually manipulate the DOM tree.

Event handling - Frameworks simplify event handling by providing mechanisms to attach event listeners to DOM elements and handle events in a consistent and unified way.

Browser compatibility - JavaScript frameworks handle browser compatibility issues by providing abstractions and polyfills. They abstract away the differences in browser APIs and behaviors, allowing developers to write code that works consistently across multiple browsers.

By abstracting away these low-level concerns, JavaScript frameworks enable developers to focus more on application logic and higher-level development tasks.

3. What essence do JavaScript frameworks elevate?

Abstractions - reduces the amount of boilerplate code required and simplifies common tasks by providing higher-level APIs, abstractions, and patterns.

Developer productivity - Frameworks aim to boost developer productivity by offering tools, utilities, and conventions that streamline development processes. This increased productivity allows developers to build applications more efficiently and effectively.

Maintainability - enforcing structure and organization within applications. Frameworks provide guidelines and best practices that help developers write code that is easier to understand, debug, and maintain over time.

Other essence are Reusability, Performance optimizations, Cross-browser compatibility, Ecosystem and community support.

JavaScript frameworks empower developers to build complex, scalable, and maintainable web applications more efficiently.

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

By providing higher level APIs, conventions and patterns that encapsulate complex functionality and hide low level implementation details.

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

Gaining a deep understanding of its core concepts and principles will allow you to make informed design decisions and utilize best practices. This includes grasping the framework's architecture, data flow patterns, component model, state management, and any unique features it offers

Try to always stay updated with the framework's ecosystem. Being aware of the latest updates, trends, and community practices can help you stay proficient and make the most of the framework's capabilities.