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?

Easily access and manipulate tags, IDs, classes, Attributes, or Elements of HTML using commands or methods provided by the Document object and Performance boost
Preserving State by directly manipulating the DOM allows you to update specific elements without losing user input or changing the state of other parts of the page.
Animation and Smooth by directly manipulating the DOM gives you more control over the timing and sequence of changes.
Accessibility by directly manipulating the DOM can be more maintaining accessibility information associated with elements, ensuring that technologies continue to work as expected.
Network Efficiency by updating the DOM, you only need to send and receive the data required for the specific updates.

---

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

DOM Manipulation Complexity - Direct manipulation of the Document Object Model (DOM) can be cumbersome and error-prone.
Event Handling - Managing and handling browser events (e.g., click, input)
State Management - Managing application state across components can be challenging.
Routing - Proper handling of browser history and navigation.
Cross-Browser Compatibility - Dealing with inconsistencies in how different browsers interpret and render JavaScript and
writing code that works across various browser versions.
Updating the DOM, keeping track of what elements need to change.
Imperative updating of the DOM, keeping track of what elements need to change

---

3. What essence do JavaScript frameworks elevate?

Declarative Syntax - Frameworks introduce a more approach where developers describe the UI state, and the framework handles the underlying changes.
Component-Based Architecture - Frameworks a component-based architecture, where UIs are broken down into modular components with responsibilities. This reusability and maintainability.
Data Binding - Frameworks provide data-binding mechanisms that automatically synchronize the UI with the application state. Changes in data are reflected in the UI and vice versa, without manual updates.
Abstraction - Frameworks simplify operations with features like async/await syntax or the use of observable patterns and this makes code more readable and easier to reason about.
Routing Abstraction - Frameworks provide a routing mechanism that abstracts away the complexities of managing client-side navigation. Developers can define routes and associated components in a more structured manner.
State Management - Frameworks often include built-in solutions or patterns for managing state in a more organized and predictable way.
Developer Productivity - Frameworks aim to boost developer productivity by providing abstractions that handle common tasks, allowing developers to focus more on building features.

---

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

They hide away the imperative DOM mutations
JavaScript frameworks achieve abstraction by providing higher-level tools and interfaces
to simplify common web development tasks.
They use declarative syntax
Component-based architecture
Data binding
Abstraction of asynchronous operations to reduce manual
Low-level coding
This allows developers to focus on building features without getting deeply involved in intricate implementation details.

---

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

The most important part of learning a JavaScript framework is gaining a solid understanding of its core concepts and principles. While each framework may have its unique features and syntax, there are common foundational concepts that are to comprehend.