

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

-DOM mutations can be more efficient in terms of performance. When you manipulate the existing DOM elements, you avoid the overhead of recreating and rendering the entire HTML structure. This can be especially beneficial when dealing with complex or dynamic web applications where frequent updates occur

State preservation

It allows you to preserve the state of the existing elements. By modifying the DOM in place, you can retain any associated event listeners, data bindings, or other dynamic behavior attached to those elements. This can be crucial in scenarios where you want to maintain the current state or avoid resetting user interactions.

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

Event-handling

Frameworks simplify event handling by providing abstractions and mechanisms for registering and handling events. They often provide a centralized event system that handles event delegation, event binding, and event listeners, reducing the need for manual event management

State Management

Managing application states can be complex, especially in large-scale applications. Frameworks provide state management mechanisms that abstract away the complexities of state synchronization and propagation. They offer tools like reactive data bindings, centralized stores, or component-level state management, making it easier to manage and update application state

3. What essence do JavaScript frameworks elevate?

Declarative Programming

Frameworks promote a declarative programming paradigm, where developers define the desired outcome rather than specifying step-by-step instructions. With declarative syntax and data-driven rendering, frameworks allow developers to express the structure and behavior of the user interface (UI) in a more concise and intuitive manner. This approach increases the risk of bugs.

Ecosystem and Tooling

Frameworks come with robust ecosystems and a wide range of tools, libraries, and extensions. These ecosystems provide additional functionality, community support, and a rich set of resources. Developers can leverage pre-built components, plugins, and tools for tasks like testing, bundling, state management, routing, and more. The availability of a vibrant ecosystem enhances developer productivity and accelerates development timelines.

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

Most Javascript frameworks achieve abstractions by providing higher-level API's and abstractions that simplify and hide the underlying complexities of web developments

Encapsulation- They encourage encapsulation by promoting the use of components or modules. Components encapsulate their logic, style, and behavior, allowing developers to work with self-contained units of functionality. Encapsulation enables code organization, reusability, and isolation, abstracting away the implementation details and making components easier to understand and maintain.

Declarative syntax-they often use a declarative syntax to describe the desired state of the UI or the behavior of the application. Instead of manually manipulating the DOM or specifying step-by-step instructions.

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

The most important part of learning JS framework is understanding its underlying concepts and principle. While frameworks have their own syntax, APIs, and tools,

focusing on the core concepts will provide you with a solid foundation for effectively working with the framework.

1. Architecture and Design patterns

Gain a clear understanding of the architecture and design patterns used in the framework. This includes understanding how components/modules are structured, how data flows through the application, and how the framework handles state management and UI updates. Understanding the framework's architecture will help you make informed decisions when building applications and ensure you utilize the frameworks's features effectively

2. Component-Based development

Most JS frameworks are built around a component-based development approach. Learn how to create, reuse, and compose components effectively. Understanding how components encapsulate their logic, style, and behavior, and how they interact with each other. Pay attention to component hooks for executing code at specific stages of a component's lifecycle.