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?

- **Faster Performance:** direct DOM mutations are faster and more precise compared to completely replacing HTML since the entire HTML structure doesn't need to be created and rebuilt each time. Updates are made to specific elements or attributes, thus resulting in quicker updates and a more responsive UI.
- **Separation of State:** with direct DOM mutations, the UI and underlying data states can be updated independently. This separation between the UI and data states avoids the need to synchronize both states at the same time, thus reducing the chances of performance issues.
- **The Concept of v=f(s):** direct DOM mutations maintain a direct mapping between the application's functionality and its underlying state. By updating the DOM based on changes in the application's state, it becomes easier to reason about the behavior of the UI and debug potential issues.

---

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

- **DOM Manipulation:** frameworks abstract away the manual manipulation of the DOM by providing higher-level APIs to update and interact with the UI. They handle tasks such as element creation, insertion, removal, and attribute manipulation.

---

3. What essence do JavaScript frameworks elevate?

- **Abstraction of the DOM:** JavaScript frameworks provide an abstraction layer over the Document Object Model (DOM), allowing developers to interact with the UI in a more intuitive and simplified manner. They offer higher-level syntax that hides the complexities of directly manipulating the DOM, making UI development more efficient and less prone to errors.

- **Performant HTML Updates:** instead of rebuilding the entire HTML structure, frameworks identify and apply only the necessary changes to the UI based on updates in the application's state. This approach improves performance by minimizing unnecessary updates and rendering only what has changed.

---

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

- **Managing State:** JavaScript frameworks abstract the management of the application state by allowing developers to define and update state variables that represent the data of the application. This offers a centralized and predictable mechanism to handle the flow of data and ensures that the UI always reflects the most up-to-date state of the application.
- **Declarative Syntax:** Frameworks often encourage a declarative programming style that requires developers to describe the desired outcome or state of the UI, rather than specifying the steps to achieve it. In doing so, frameworks abstract away the low-level DOM manipulation details and provide a higher-level interface to define the UI structure.

---

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

The most important part of learning any framework is learning the key concepts and principles of the said framework as a foundation for its eventual utilization.

Additionally, familiarising yourself with the framework’s documentation, projects, libraries and plugins will lead you to use it more efficiently in your own projects.