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 offer several benefits over replacing HTML entirely.

Firstly, they improve performance by selectively modifying only the necessary parts of the DOM, reducing the workload on the browser. This results in faster updates and a more responsive user interface.

Secondly, direct DOM mutations minimise memory consumption since they operate on the existing DOM structure, avoiding the need for memory-intensive operations like parsing and re-rendering. This is particularly advantageous for complex and dynamic web applications.

Additionally, direct DOM mutations provide a smoother user experience by avoiding page flickering or re-rendering, as only the relevant portions of the page are updated.

They also offer fine-grained control, allowing developers to precisely define which parts of the DOM should be modified. Lastly, direct DOM mutations are compatible with various JavaScript frameworks, making them widely applicable in different web development ecosystems.

---

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

JavaScript frameworks abstract away low-level complexities such as DOM manipulation, event handling, AJAX requests, state management, and routing. This allows developers to focus on application logic, improves productivity, and enhances code organisation and maintainability.

---

3. What essence do JavaScript frameworks elevate?

JavaScript frameworks elevate the essence of building web applications by providing a structured and efficient approach to development. They prioritise concepts such as modularity, reusability, and maintainability, enabling developers to focus on creating robust and feature-rich applications.

JavaScript frameworks also promote a declarative programming style, allowing developers to define what they want the application to do without worrying too much about the underlying implementation details. By abstracting away low-level complexities, frameworks elevate the essence of simplicity, productivity, and scalability in web development.

---

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

Broadly speaking, most JS frameworks achieve abstraction through the use of APIs and a set of established conventions that help abstract low-level details and provide high-level structures and patterns for building complex applications.

This enables developers to create scalable and maintainable code, while also reducing the cognitive load required to manage the underlying technology stack and infrastructure. Some frameworks, like React, use a declarative programming model to abstract away the complexities of DOM manipulation and state management, while others, like Angular, rely on dependency injection and a reactive programming model to achieve similar levels of abstraction.

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

There are many different opinions on the most important parts of learning a JS framework. But in my opinion the most important aspect is learning the core concepts and API's specific to the framework you choose, another important point would be to be good/fairly proficient in Vanilla JavaScript before attempting to learn any frameworks(something which is yet to come for me at this moment). This all depends on where you are personally in your development journey and what your goal is with the framework you would like to learn.