## 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 handite of direct DOM routations over replacing LITM 2

1. What are the benefits of direct DOM mutations over replacing HTML?

Direct DOM mutations offer more control over the changes you want to make and target only those specific parts/elements of the existing DOM that need to be changed while replacing HTML involves recreating and rendering the entire DOM tree.

Recreating and rendering the entire DOM tree can be resource-intensive but only modifying the specific parts you need to change results in faster updates.

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

JavaScript frameworks allow developers to focus more on building application logic and user experiences, rather than getting caught up in the details of low-level noise, which implementation complexities such as DOM manipulation, event handling and state management.

3. What essence do JavaScript frameworks elevate?

JavaScript frameworks simplify web development by providing a structured and well-organized environment for building complex applications. They abstract away complexities, promote code reusability and simplify data management.

\_\_\_\_\_

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

Most JavaScript frameworks simplify web development by offering higher-level APIs and ready-to-use tools that hide the complexities of low-level tasks. They provide pre-built components, modules, and patterns, allowing developers to focus on building applications instead of dealing with very complicated details.

\_\_\_\_\_

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

The most important part of learning a JavaScript framework is understanding its core concepts and principles, and not just learning how to use the abstractions provided by the frameworks.