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?

- The DOM is a real in-memory representation of the document at any given time, whereas the actual HTML document is just text that tells the browser what the document looks like when it first loads. This means if you have to replace the html to update the page then you have to reload the page for every update but DOM manipulation allows you to dynamically change the contents of the page without reloading.
- DOM manipulation allows for the automation of user interface interaction without having to manually respond to each user input by changing the html then they have to reload the page.

---

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

- DOM manipulation - JS frameworks simplify the process of manipulating the DOM by providing higher-level abstractions and utilities for selecting elements, modifying attributes, event handling and updating the UI in real-time.
- Event handling - Frameworks handle the process of event handling by providing methods for attaching event listeners and handling different types of events.
- State management - Many JS frameworks offer state management ability that can help to abstract away the complexities of managing state. They have tools to simplify the definition and updating of state, can handle state transitions and synchronize the state update with the UI.

---

3. What essence do JavaScript frameworks elevate?

- **Productivity** - frameworks can increase productivity by providing ready-made tools, abstractions and rules to simplify common tasks in application building. This allows developers to write code more efficiently and focus on building unique features rather than waste time reinventing the wheel.
- **Maintainability** - frameworks help to promote maintainability by providing and enforcing structured code and since they usually follow established and accepted code architectural patterns, they make code easier to understand, reuse and test.
- **Scalability** - JS frameworks promote scalability by allowing for code reusability and component-based development. They provide a way to break an application down to smaller components that can be developed and tested independently. This allows teams to work on different parts of an application at the same time.
- **Cross browser compatibility** - JS frameworks abstract away browser-specific differences and allow for applications to behave the same across different browsers.

---

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

JS frameworks use the *state* of the application as an indicator of how the User Interface must look at a specific point in time, so that when the user takes action or inputs data, the application state changes accordingly. JS frameworks also encourage encapsulation and modulation so that each module or component hides its internal implementation, thereby reducing the complexity of the code. This makes abstraction

easier (which of course makes code more reusable, and reduces the chances of bugs being created).

---

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

I would say that the most important part of learning a JS framework is understanding the documentation that it uses. This means understanding how to create the abstractions and modules. If one does not understand this, it will be very difficult to apply and debug them.
Another important thing to learn about a framework is the type/s of state management solutions it uses, plus the workflow that they recommend - this will make programming easier and ensure compatibility.