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 boost

The main benefit of direct DOM mutations over replacing HTML is performance. When you replace HTML, the browser has to re-parse the entire DOM tree, which can be slow, especially for large DOM trees. Direct DOM mutations, on the other hand, only modify the parts of the DOM tree that are being changed, which can be much faster. Additionally, direct DOM mutations allow you to make changes to the DOM tree in a more granular way, which can be helpful when you only need to make small changes.

---

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

Imperative updating of the DOM, keeping track of what elements need to change

The main low-level noise that JavaScript frameworks like React abstract away is the imperative updating of the DOM. With React, you don't need to worry about updating the DOM in a step-by-step fashion, and you don't need to worry about keeping track of which elements have changed and which ones need to be updated. Instead, you can focus on defining your application in terms of high-level components, and the framework will take care of the low-level details. This can make it easier to write maintainable code that's easy to reason about.

---

3. What essence do JavaScript frameworks elevate?

JavaScript frameworks like React elevate the declarative essence of HTML. In HTML, you can specify the structure and content of a document without having to worry about the details of how it will be rendered. JavaScript frameworks take this idea to the next level, allowing you to create components that encapsulate both the structure and behavior of the component, without having to worry about how the component will be rendered in the browser. This makes it easier to create reusable and composable components that can be used to build complex user interfaces.

---

4. Very broadly speaking, how do most JS frameworks achieve abstraction?
They hide away the imperative DOM mutations

That's a very astute observation! Yes, most JS frameworks like React, Vue, Svelte, and Angular achieve abstraction by hiding away the imperative DOM mutations. Instead, they provide a declarative way to describe the desired state of the DOM, and they use a virtual DOM to keep track of the differences between the current state and the desired state. This allows the framework to efficiently update the DOM only when necessary. The virtual DOM is a lightweight representation of the actual DOM, so it's much faster to manipulate than the actual DOM.

---

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

In my opinion, the most important part of learning a JS framework is understanding its mental model. Each framework has its own mental model, which is the way it conceptualizes how the application is built and how it works under the hood. If you understand the mental model of the framework, you'll be able to use it more effectively and avoid common pitfalls. The mental model will also help you understand how to write clean and efficient code. Without understanding the mental model, you might just

be blindly following the framework's API without really understanding what's going on behind the scenes