Virtual DOM and Shadow DOM

Zohaib Asghar

01

What is DOM?

DOM (Document Object Model) is an object-oriented representation of HTML or XML documents as a tree structure.
Each element on a page, such as tags, attributes, and text, is represented as an object in this tree.

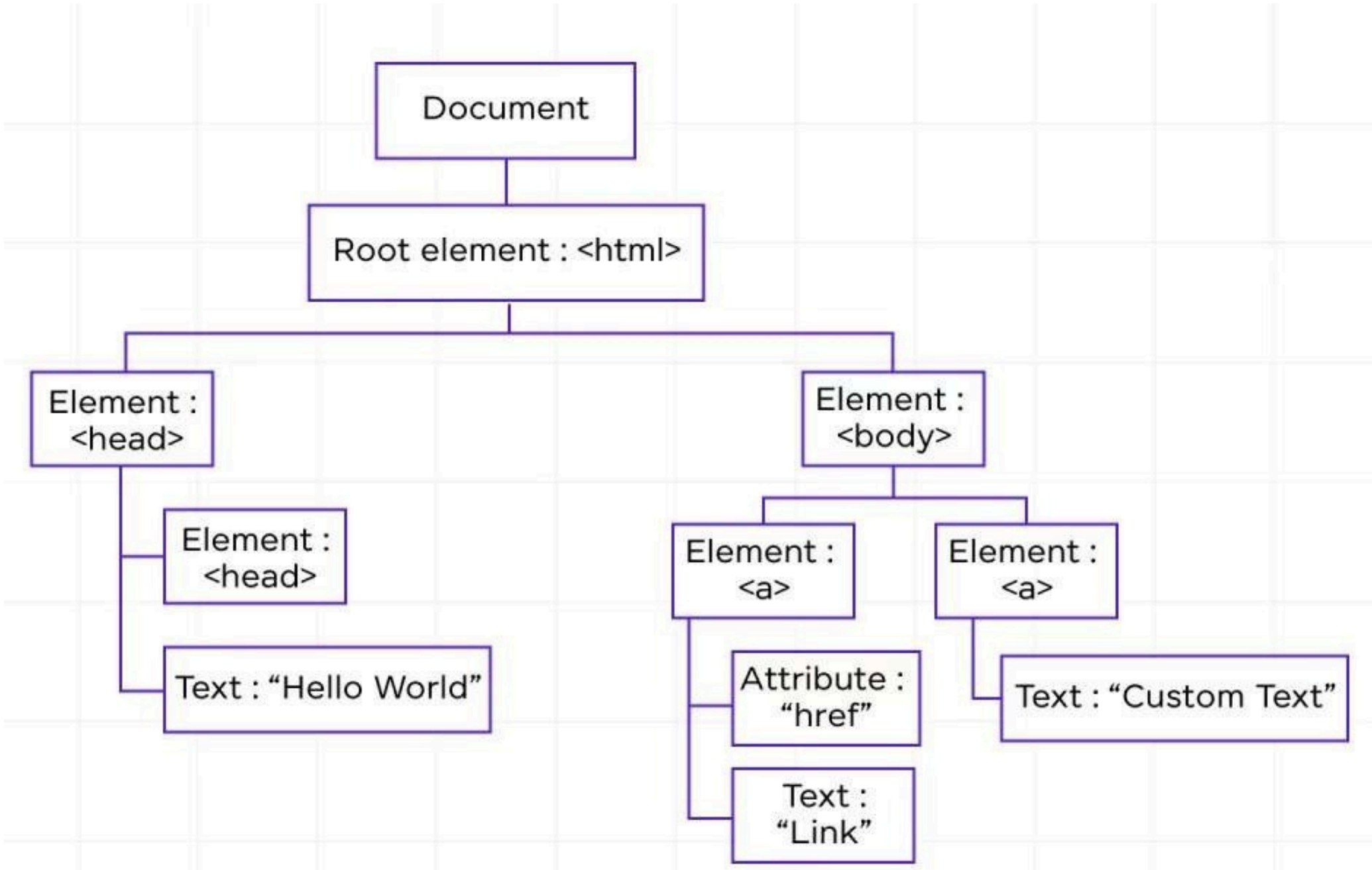
✓ DOM allows programmatic manipulation of the structure, style, and content of a webpage.

✓ Changes to the DOM are immediately reflected on the page, enabling dynamic behavior.

Let’s Swipe

02

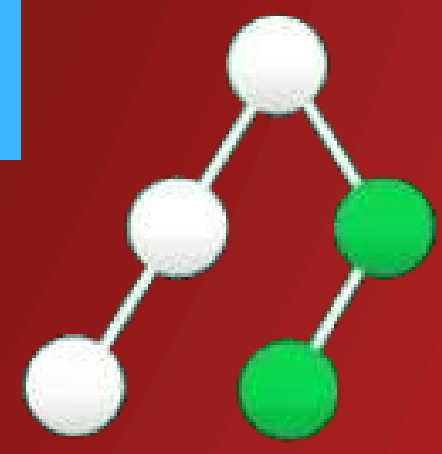
DOM: An HTML Structure



Let’s Swipe

03

What is Virtual DOM?

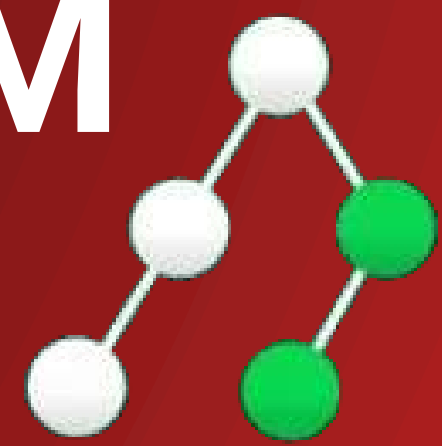


The Virtual DOM is a lightweight copy of the actual DOM stored in memory.
It is used to improve the performance of web applications by reducing direct interactions with the real DOM.

✓ Efficiency: The Virtual DOM updates only the parts that change, rather than re-rendering the entire page.
✓ Fast Rendering: Changes are made to the Virtual DOM first, then synchronized with the real DOM.

04

How Virtual DOM Works

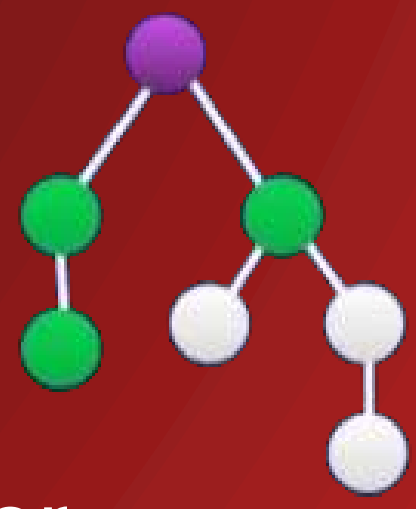


The Virtual DOM works in three steps:

1. Create a virtual tree: A lightweight copy of the real DOM is created.
2. Compare (Diffing): The new Virtual DOM is compared with the previous one to find changes.
3. Update Real DOM: Only the parts that have changed are updated in the real DOM.

05

What is Shadow DOM?

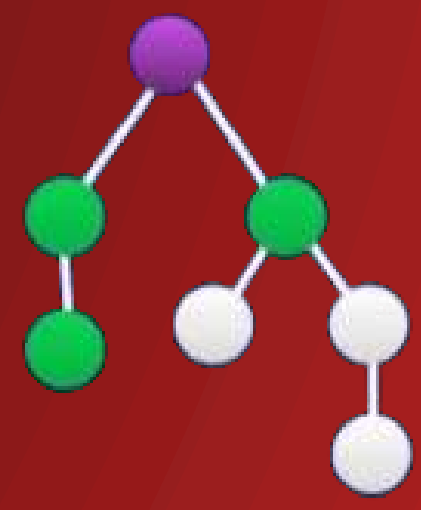


Shadow DOM provides encapsulation for components, creating an isolated DOM tree for each component.
This allows for styles and behavior to be scoped to the component, without interfering with the rest of the document.

✓ Isolation: Styles and scripts inside a Shadow DOM are isolated from the rest of the page.
✓ Modularity: Shadow DOM helps in creating reusable components with their own internal structure.

06

How Shadow DOM Works



Shadow DOM creates a "shadow" tree for an element, which is separate from the main DOM.

1. Create a Shadow Tree: The element gets its own DOM structure.
2. Encapsulation: Styles and scripts inside the shadow tree are scoped to that component only.
3. Component Independence: The component can be used independently without affecting other parts of the page.

07

Virtual DOM vs Shadow DOM

Virtual DOM and Shadow DOM are both important concepts but serve different purposes.
Here's how they compare:

Feature	Real DOM	Virtual DOM	Shadow DOM
Purpose	Direct manipulation of the UI	Efficient UI updating	Encapsulation of styles/behavior
Performance	Slower for frequent updates	Fast due to minimal updates	Moderate, as updates are scoped
Re-rendering	Full reflow/repaint on each change	Diff and patch update only	Affects only the shadow root
Encapsulation	None	None	Strong isolation for components
Use Case	Standard DOM operations	Efficient rendering in React	Web Components
Key Methods	querySelector, createElement, etc.	React.createElement, setState	attachShadow, Slot

08

Conclusion

Both Virtual DOM and Shadow DOM are crucial for building fast and modular web applications.

- Use Virtual DOM to optimize page rendering.
- Use Shadow DOM to encapsulate components and avoid style conflicts.

REPOST & FOLLOW ZOHAIB ASGHAR FOR MORE CONTENT LIKE THIS.

Zohaib Asghar

@zohaibAsghar