Class Work

5th of February, 2025

❖ What is the difference b/w Virtual Dom & Real Dom?

Virtual DOM	Real DOM
It is a ? of the original DOM	It is a ? representation of HTML elements
It is maintained by ? Libraries.	It is maintained by the ? after parsing HTML elements
After manipulation it only re-renders ? .	After manipulation, it re-render the ? .
Updates are lightweight	Updates are heavyweight
Performance is ? and UX is optimised	Performance is ? and the UX quality is low
Highly efficient as it performs ? .	Less efficient due to re-rendering of DOM after ? .

Answer:

Virtual Dom:

1. It is a **copy** of the original DOM.
2. It is mentioned by **React** libraries.
3. After manipulation, it only re-renders the **changed parts**.
4. Performance is **improved** and UX is optimized.
5. Highly efficient as it performs **minimal DOM mutations**.

Real Dom:

1. It is a **tree-like** representation of HTML elements.
2. It is maintained by the **browser** after parsing HTML elements.
3. After manipulation, it re-renders **the entire DOM**.
4. Performance is **slower** and the UX quality is low.
5. Less efficient due to re-rendering of DOM after **every update**.

❖ What does JSX stands for? Also, SSR and CSR?

Answer

JSX:

JSX stands for “JavaScript Syntax Extension”. It’s the syntax combination of JavaScript along with the JSX.

JSX is a syntax extension for JavaScript that allows HTML-like code in JavaScript files. It's used for building user interfaces, especially with React.

Example: const element = <h1>Hello, world!</h1>;

SSR:

Server-Side Rendering (SSR) renders web pages on the server, sending fully formed HTML to the client's browser. This approach provides faster initial page loads and better SEO.

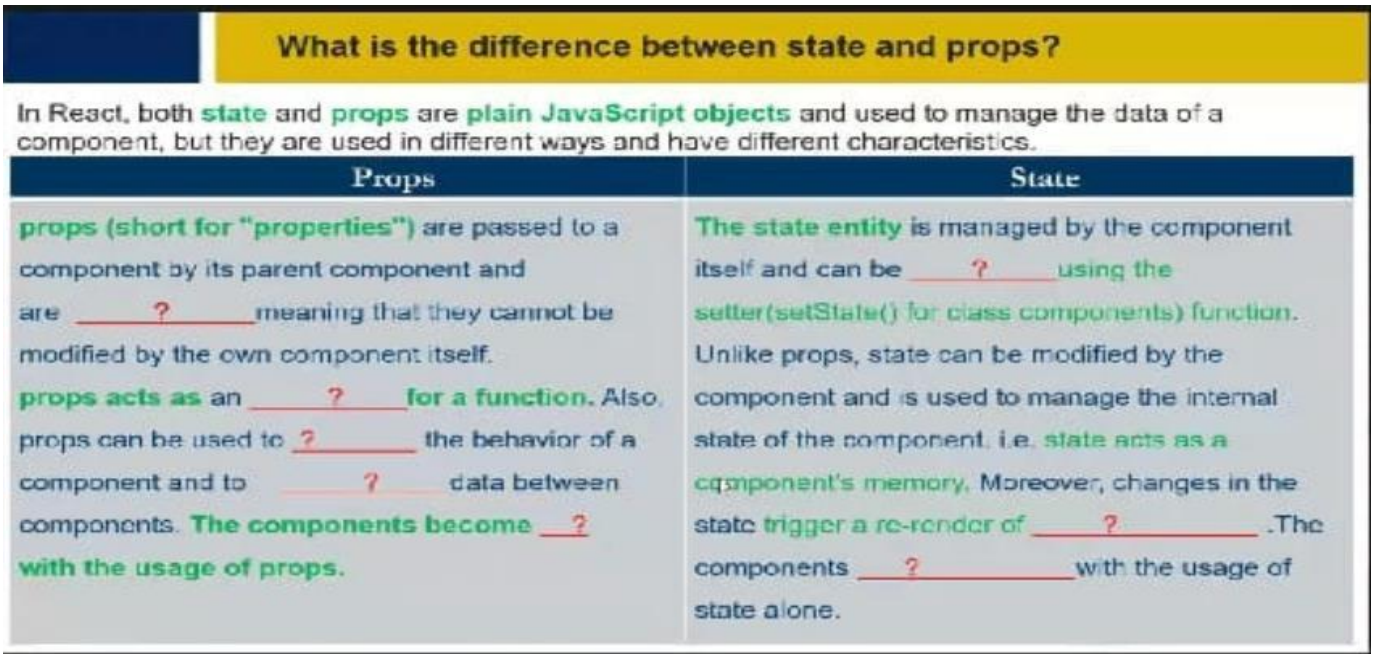
Example: Next.js uses SSR to pre-render pages.

CSR:

Client-Side Rendering (CSR) renders web pages on the client's browser using JavaScript. This approach provides dynamic updates and interactive web pages.

Example: React, Angular, and Vue.js use CSR to render and update web pages.

❖ What is the difference b/w the State and Props?



What is the difference between state and props?

In React, both state and props are plain JavaScript objects and used to manage the data of a component, but they are used in different ways and have different characteristics.

Props	State
props (short for "properties") are passed to a component by its parent component and are ? meaning that they cannot be modified by the own component itself. props acts as an ? for a function. Also, props can be used to ? the behavior of a component and to ? data between components. The components become ? with the usage of props.	The state entity is managed by the component itself and can be ? using the setter(setState() for class components) function. Unlike props, state can be modified by the component and is used to manage the internal state of the component, i.e. state acts as a component's memory. Moreover, changes in the state trigger a re-render of ? .The components ? with the usage of state alone.

Answer:

Props:

1. Props (short for “properties”) are passed to the component by its parent component and are **immutable**, meaning that they cannot be modified by the own component itself.
2. Props act as an **argument** for a function.
3. Also, props can be used to **customize** the behavior of the component and to **pass** data between components.
4. The components become **reusable** with the usage of props.

State:

1. The state entity is managed by the component itself and can be **updated** using the setter (setState() for class components) function.
2. Unlike props, state can be modified by the component and is used to manage the internal state of the component, i.e., state as a component's memory.
3. Moreover, changes in the state trigger a re-render of **the component**.
4. The components **become dynamic** with the usage of state alone.

❖ What does DOM stand for?

The Document Object Model (DOM) is a tree-like structure of HTML elements. It represents the structure and content of a web page. The DOM allows JavaScript to interact with and modify the web page's elements. It's a dynamic representation of the web page.