**React**

**Day 5th**

**Basic Concepts of React**

React

Here in browser whenever a webpage is loaded it creates a real DOM. Every changes in the file will create a reload inside this but react doesn’t work like this. Inside react we have some internal components like reactDOM and virtual DOM which is also known as copy of real DOM it has the same data which real DOM holds.

Real DOM

Browser

Virtual DOM

Clone of Real DOM

ReactDOM

React Internals

**Virtual DOM**

Virtual DOM is a clone or copy of DOM, no rerendering takes place when the virtual DOM changes.

The Virtual DOM is a lightweight, in-memory representation of the actual DOM elements.

**Reconciliation Process**

Reconciliation is the process React uses to figure out how to efficiently update the DOM when changes occur in the UI. React’s goal is to update the page as efficiently as possible, without unnecessary re-rendering or slow performance.

Reconciliation helps in the following ways:

* **Minimizes unnecessary updates:** React only changes the parts of the UI that actually need to be updated, rather than re-rendering the entire page.
* **Improves Performance:** By optimizing the update process, react reduces the number of changes to the actual DOM, which improves the performance of your application.
* **Ensures consistency:** React ensures that the UI always matches the current state of your application, even as it changes over time.

**Diffing Algorithm**

The diffing algorithm in react allows for the efficient updates and rendering of these DOM elements. As changes occur within the application’s state or properties, React uses its diff algorithm to compare the new virtual DOM with the old one.

It compares two DOM old virtual DOM and new Virtual DOM.

**How DOM updates in React**

* On the first run, both virtual DOM and real DOM tree are created.
* React works on observable patterns, hence, whenever there is a change in the state, it updates the node in the virtual DOM.
* Then, react compares virtual DOM with the real DOM and updates the changes. This process is called reconciliation.

React uses a heuristic algorithm called the diffing algorithm for reconciliation based on these assumptions.

1. Elements of different types will produce different trees.
2. We can set which elements are static and do not need to be checked.

React checks the root elements for changes and the updates depend on the types of the root elements.

**What is Virtual DOM?**

Virtual DOM is a copy of Real DOM, you can say a xerox copy, whatever updation we do on the virtual DOM is not reflected on the real DOM or the screen directly.

It does all the manipulation within itself and renders those changes on the real DOM so this way we don’t need to update the Real DOM again and again.

It can change 200000 nodes/sec.

**Diffing**

React compares the virtual copy of the real DOM to an updated copy of the virtual DOM, compares or picks out the changes, and finally renders it to real DOM. This process is called diffing and the algorithm used is called diffing algorithm.

**Reconciliation**

When a component state or prop changes, React decides whether it should render the changes on real DOM or not. So, if the states/props of two nodes/components are not the same, then it renders the changes to real DOM. This process is called reconciliation.

Reconciliation is dependent on

1. Virtual DOM
2. Diffing Algorithm

In react when the state of component changes, the component needs to update its UI to reflect the new state. This process of updating the UI is called reconciliation.

React uses Virtual DOM to perform reconciliation, which is used to compare component’s current state and previous state.

**What is the virtual DOM**

The virtual DOM is a lightweight in memory representation of the actual DOM. When the state of a component changes, react compares the VDOM of the last and current states and calculates the minimum number of DOM operations required to update the actual DOM to match the current VDOM.

**The Catch**

This helps reduce the number of DOM manipulations and improve the application’s performance. However, there are specific scenarios where the reconciliation process can become inefficient.

For example suppose a component has many elments that need to be updated. In that case, the reconciliation process can take a long time and cause the UI to become responsive – this is where **Fiber comes** in

**What is Fiber??**

Fiber is a new reconciliation algorithm introduced in react 16.0 that aims to improve the performance of react applications by making the reconciliation process more efficient.

It does that by allowing the reconciliation process to be broken down into smaller chunks and scheduled over multiple frames rather than being completed in a single frame.

Fiber divides the reconciliation work into smaller units called “**fibers**”

Each fiber represents a single element in the VDOM tree, and the reconciliation process is performed on each fiber individually.

This allows react to prioritize the reconciliation of certain fibers over others, depending on the importance of the updates.

**Fiber in Asynchronous programming**

Fiber also introduces a new concept called suspense. Suspense allows react components to wait for a specific condition to be met before rendering.

This can improve the performance of applications that rely on asynchronous data, such as fetching data from a server.

**What is react fiber??**

React fiber is an internal engine change geared to make react faster and smarter. The fiber reconciler, which became the default reconciler for react 16 and above. It is a complete rewrite of react’s reconciliation algorithm to solve some long standing issues in react.

Because fiber is asynchronous react can

* Pause, resume and restart rendering work on components as new updates come in.
* Reuse previously completed work and even abort it if not needed
* Split work into chunks and prioritize tasks based on importance

**Purpose of react fiber??? (flexiple.com)**

**Features??**

**How it works??**