**NAMASTE REACT-AKSHAY SAINI**

**CHAPTER 4- TALK IS CHEAP, SHOW ME THE CODE!**

1. **Is ISX is mandatory for react?**

Ans:- No.But it makes React faster. JSX is not a requirement for using React. Using React without JSX is especially convenient when you don’t want to set up compilation in your build environment.

Refer: <https://reactjs.org/docs/react-without-jsx.html#:~:text=JSX%20is%20not%20a%20requirement,%2C%20...children)%20>.

1. **Is ES6 mandatory for React ?**

Ans:- No.

Refer: <https://reactjs.org/docs/react-without-es6.html>

<https://medium.com/@rashmishehana_48965/must-know-es6-features-for-react-97ad57468623>

1. **{TitleComponent} vs {<TitleComponenet/>} vs {<TitleComponenet><TitleComponent/>}**

Ans:-

First one is the way of rendering react element which gets converted to JavaScript variable behind the scenes. Second and Third is the way of rendering functional components in react.

1. **How can I write comment in JSX ?**

Ans:- You can write it normally using “//” or multiline comments as “/\* ……… \*/” .

1. **What is <React.Fragment></React.Fragment> and <></> ?**

Ans:-

<React.Fragment></React.Fragment> and <></> are same.

They are empty tags which we have to use when we don’t want to create any extra parent node inside JSX/Functional component. This React.fragment are invisible inside our DOM.

React Fragment is a feature in React that allows you to return multiple elements from a React component by allowing you to group a list of children without adding extra nodes to the DOM.

To return multiple elements from a React component, you'll need to wrap the element in a root element.

Refer: <https://refine.dev/blog/how-react-fragments-is-works/#:~:text=Fragment%20in%20Action-,What%20is%20React%20Fragment%3F,element%20in%20a%20root%20element>.

1. **What is Virtual DOM and Reconciliation?**

Ans:-

Virtual DOM is the representation of actual DOM tree which reacts uses for faster rendering.

Reconciliation is the feature of React which is based of DIFF algorithm that finds out difference between one tree and the other tree and only renders the thing which is new to React.

Ref: <https://reactjs.org/docs/faq-internals.html> <https://reactjs.org/docs/reconciliation.html>

[Reconciliation](https://reactjs.org/docs/reconciliation.html):-

As explained in the official [React documentation](https://reactjs.org/docs/reconciliation.html), reconciliation is the algorithm for diffing two DOM trees. When the UI renders for the first time, React creates a tree of nodes. Every individual node represents the React element. It creates a virtual tree (which is known as virtualDOM) that’s a copy of the rendered DOM tree. After any update from the UI, it recursively compares every tree node from two trees. The cumulative changes are then passed to the renderer.

1. **What is React Fiber ?**

Ans:-

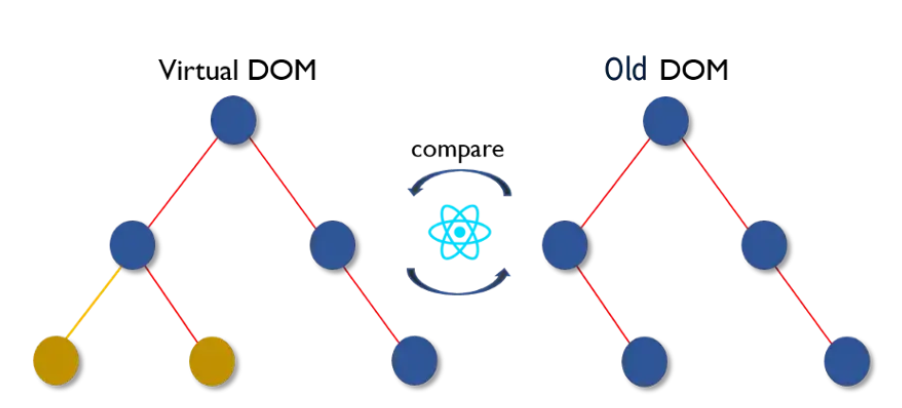
React Fiber is a new reconciliation engine which has been introduced in React 16(current version is React 18) and it is the one which is responsible for “Diff” algorithm.

Refer: <https://www.velotio.com/engineering-blog/react-fiber-algorithm> <https://github.com/acdlite/react-fiber-architecture>

1. **Why we need keys in React ? When do we need keys in React ?**

Ans:-

Why?- With the help of key React understands exact location where the element needs to be added/updated using Diffing Algorithm.



When? – We require keys whenever we are dealing with multiple children of same tag (eg:div) inside a node. Thus by recognizing these unique key React can find exact location of node which got changed.

 Refer: <https://betterprogramming.pub/why-react-keys-matter-an-introduction-136b7447cefc>

<https://lo-victoria.com/why-react-keys-matter-an-introduction>

1. **Can we use index as keys in React ?**

Ans:-

Using index as your key will result in issues when certain changes are made to the list. React don't recommend using indexes for keys if the order of items may change.

Still Index is the valid key and can be used if we don’t have any unique key to apply. But recommendation is not to use index.

No key(Not recommended) <<<<< index key(Use only if you don’t have anything) <<<<< Unique key(Best Practice)

1. **What is props in React ?**

Ans:-

We use props in React to pass data from one component to another (from a parent component to a child component(s)). Props is just a shorter way of saying properties. They are useful when you want the flow of data in your app to be dynamic.

They can pass as parameter in React Component.

Pops are read data only , which means data coming from parent should not be changed by child.

Refer: <https://reactjs.org/docs/components-and-props.html> <https://www.freecodecamp.org/news/how-to-use-props-in-react/>

1. **What is Config Driven UI?**

Ans:-

Mostly bigger companies like Swiggy nowadays follows Project based on Config Driven UI.

Is the the concept where backend data bind with UI conditionally. Here back end decides which UI we need to display as per the condition.

For eg: Swiggy dispaying offers for cities like Pune,Bangalore,Hyderabad but not for Kolkata,Delhi,Surat.

Refer: <https://medium.com/captech-corner/an-intro-to-configuration-driven-development-cdd-48a1c088baa9>

<https://iamrajatsingh1.medium.com/config-driven-ui-c8e93b730993>

<https://techmusings.dev/buildingAConfigDrivenUiInReactPart1>

1. **Can we use React.Fragment inside React.Fragment?**

Ans:-

“ Fragments are syntax that allows us to add multiple elements to a React component without wrapping them in an extra DOM node.”

Yes we can wrap one <React.Fragment> inside other.

Refer: https://www.knowledgehut.com/blog/web-development/understanding-react-fragments