

Chapter 04 - Talk is cheap, show me the code!

Assignment

- Is JSX mandatory for React?
No we can use create react element to make a whole application.
- Is ES6 mandatory for React?
No

- `{TitleComponent}` vs `{<TitleComponent/>}` vs `{<TitleComponent></TitleComponent>}` in JSX

- How can I write comments in JSX?
There are three types of comments in JSX

1. Single line comment.

```
...  
<> { /* hello */ } </>  
...
```

2. Multiline comment.

```
...  
<>  
  { /* <div>  
    <h1> hello </h1>  
    </div> */ }  
  </>  
...
```

3. comment outside jsx tag.

```
const Compo = ()=>{  
  // comment outside the comment using double slashes  
  return (  
    )  
  }  
}
```

- What is `<React.Fragment></React.Fragment>` and `<></>` ?
Both are used to group multiple elements together
the key difference is in `<React.Fragment></React.Fragment>` is we can add key
- What is Virtual DOM?
 - The fake dom which we create using js in browser in the JRE is called VirtualDOM
 - The Virtual DOM is a lightweight in-memory representation of the actual DOM elements. It is a concept implemented in libraries like React to optimize the rendering process in web applications. The Virtual DOM allows efficient updates to the user interface by minimizing direct interactions with the real DOM.

- What is Reconciliation in React?

Reconciliation is a process in which when there are some changes on the ui , The react core uses a diff algorithm to compare the changes and update the UI. This process of updating the dom with the latest state is called Reconciliation.

- What is React Fiber?

React Fiber is a complete rewrite of the React core algorithm. It was introduced to address the limitations of the original reconciliation algorithm and to enable new features in React. Fiber improves the performance and flexibility of React applications by breaking the rendering work into units of work that can be paused, aborted, or resumed.

- Why do we need keys in React? When do we need keys in React?

React keys are crucial for optimizing the rendering process and ensuring the UI elements are efficiently updated and corrected.

We need keys for the following

1. Rendering List
2. Rendering Fragments
3. Dynamic component in Array

- Can we use indexes as keys in React?

Yes we can but it is not recommended

- What are props in React? Ways to

- What is a Config Driven UI ?

In the Config Driven UI ,every component displayed on the UI is dependent on the configuration sent by api response configuration.

Coding Assignment:

- Build a Food Ordering App
 - Think of a cool name for your app
 - Build a AppLayout
 - Build a Header Component with Logo & Nav Items & Cart
 - Build a Body Component
 - Build RestaurantList Component
 - Build RestaurantCard Component
 - Use static data initially
 - Make your card dynamic(pass in props)
 - Props - passing arguments to a function - Use Destructuring & Spread operator
 - Render your cards with dynamic data of restaurants
 - Use Array.map to render all the restaurants

PS. Basically do everything that I did in the class, in the same sequence. Don't skip small things.

- repolink - <https://github.com/shubham12346/foodOrderingApp.github.io>

References

- Code Link: <https://bitbucket.org/namastedev/namaste-react-live/src/master/>
- React without JSX: <https://reactjs.org/docs/react-without-jsx.html>
- Virtual DOM: <https://reactjs.org/docs/faq-internals.html>
- Reconciliation: <https://reactjs.org/docs/reconciliation.html>

- React Fiber Architecture: <https://github.com/acdlite/react-fiber-architecture>
- React Without ES6: <https://reactjs.org/docs/react-without-es6.html>
- Index Keys as Anti-Pattern:
<https://robinpokorny.com/blog/index-as-a-key-is-an-anti-pattern/>