**<https://learn.codevolution.dev/p/preparing-for-a-frontend-interview/?product_id=2637606&coupon_code=LAUNCH50>**

**What is React?**

React is a free and open-source front-end JavaScript library for building user interfaces based on UI components.

**What is JSX ?**

JSX stands for JavaScript XML. JSX allows us to write HTML in React. JSX makes it easier to write and add HTML in React.

**What is Components ?**

Components are independent and reusable bits of code.

**What is Pure Components ?**

Pure Components in React are the components which do not re-renders when the value of state and props has been updated with the same values.

**What is Props ?**

Props are arguments passed into React components. Props are used to communicate between components.

**What is PropTypes?**

PropTypes are a mechanism to ensure that components use the correct data type and pass the right data.

**What is prop drilling in react.**

Prop drilling refers to the process of sending props from a higher-level component to a lower-level component.

**what is hooks in react**

React **hooks are** used for state and lifecycle methods inside functional components

**What is useState ?**

useSt ate is a Hook that allows you to have state variables in functional components.

**What is useMemo ?**

useMemo is a React hook that memorizes the output of a function.

**What is useSelector ?**

useSelector is a function that takes the current state as an argument and returns whatever data you want from it.

**What is useDispatch ?**

This hook returns a reference to the dispatch function from the Redux store. You may use it to dispatch actions as needed.

**What is HOC ?**

**HOC** is an advanced technique for reusing component logic. It is a function that takes a component and returns a new component.

**What is Router ?**

React Router plays an important role to display multiple views in a single page application.

**What is virtual DOM ?**

virtual DOM is just a copy of the original DOM kept in the memory and synced with the real DOM by libraries such as ReactDOM**.**

**What is lazy loading ?**

lazy loading means that a component or a part of code must get loaded when it is required.

**What is Redux?**

Redux is a predictable state container for JavaScript apps based on the Flux design pattern.

**What is Redux lifecycle methods?**

componentDidMount(), componentDidUpdate(),componentWillUnmount()

**What is reducer Redux?**

The reducer is a pure function that takes the previous state and an action, and returns the next state.

**What is Store Redux?**

The entire state of an application lists at a place called Store.

**What is action Redux?**

An action is a pure object which is sent or dispatched from the view.

**what is connect in react-redux**

The **connect**() function connects a React component to a Redux store.

**Is middleware necessary in Redux?**

According to the docs, "Without middleware, Redux store only supports synchronous data flow". ( **Middleware ex: -** saga and thunk)

**What is async and await**

async before a function makes the function return a promise and await before a function makes the function wait for a promise. The await keyword can only be used inside an async function.

**What** promise**?**

A promise is a special JavaScript object. Promises are used to handle asynchronous operations in JavaScript.

**What Synchronous and Asynchronous ?**

Synchronous allows you to perform on task one time.

Asynchronous programming allows you to perform multiple requests simultaneously

**What is prototype**

The JavaScript prototype property allows you to add new properties to object constructors

**What is closure**

a closure is a function that references variables in the outer scope from its inner scope.

**How to copy an object in JavaScript?**

1. In ES6 can copy using spread operator (…Object), **2.** JSON.parse(Object)

**What is callback?**

A **callback** is just a regular function, it is called after another function has finished executing.

**What is callback hell?**

Callback hell is a phenomenon that afflicts a JavaScript developer when he tries to execute multiple asynchronous operations one after the other. By nesting callbacks in such a way, we easily end up with error-prone, hard to read, and hard to maintain code.

**Callback hell avoid**

We can avoid the callback hell with the help of Promises. Promises in JavaScript are a way to handle asynchronous operations in Node. js. It allows us to return a value from an asynchronous function like synchronous functions.

## **What is this?**

In JavaScript, the this keyword refers to an object.

**In an object method**, this refers to the object.

**Alone**, this refers to the global object.

**In a function**, this refers to the global object.

**In a function**, in strict mode, this is undefined.

**In an event**, this refers to the element that received the event.

**Methods like call(), apply(), and bind()** can refer this to any object.

**What is a Composite API?**

Composite APIs are a design approach to batch API requests sequentially into a single API call. Rather than multiple round trips to a server, a client can make one API request with a chain of calls and receive one response.

**Pure functions**

Reducers are just **pure functions** that take the previous state and an action, and return the next state. **Map** is also a **pure function** meaning that it does not change the original array, instead, it returns a new array.

**What's the Difference between mapstatetoprops and mapdispatchtoprops**

mapStateToProps is a function that you would use to provide the store data to your component, whereas mapDispatchToProps is something that you will use to provide the action creators as props to your component.

If mapStateToProps argument is specified, the new component will subscribe to Redux store updates. This means that any time the store is updated, mapStateToProps will be called. The results of mapStateToProps must be a plain object, which will be merged into the component’s props.

With mapDispatchToProps every action creator wrapped into a dispatch call so they may be invoked directly, will be merged into the component’s props.