Sean Cheng Recommended React Things to know

1. Revisit JS CSS and HTML basics
2. Understand why React was created and what problem does it solve

React builds single page application.

React was created by Facebook (now Meta) to address some of the challenges faced in building complex user interfaces for web applications. React solves the following key problems:

**Complexity of UI Development**: As web applications became more complex, with dynamic and interactive user interfaces, managing the state and updating the DOM efficiently became increasingly challenging. React provides a component-based architecture that makes it easier to build and maintain complex UIs.

**Performance Issues**: Updating the entire DOM for even small changes can be slow and inefficient. React introduced the concept of a "virtual DOM", which allows it to efficiently calculate and apply only the necessary changes to the actual DOM, resulting in better performance.

**Separation of Concerns**: React encourages a separation of concerns by allowing developers to encapsulate UI logic, state, and rendering in reusable components. This makes the codebase more modular, maintainable, and testable.

**Declarative Approach**: Instead of imperatively manipulating the DOM, React allows developers to describe the desired UI state, and React handles the necessary DOM updates. This declarative approach makes the code more readable and easier to reason about.

**Reusability**: React's component-based architecture promotes the creation of reusable UI components, which can be composed together to build complex user interfaces. This improves development speed and code consistency.

**Ecosystem and Community**: React has a large and active community that has contributed a vast ecosystem of libraries, tools, and resources, making it easier for developers to build and scale their applications.

In summary, React was created to address the challenges of building complex, performant, and maintainable user interfaces for modern web applications. By introducing a virtual DOM, a component-based architecture, and a declarative approach to UI development, React has become a popular choice for building web applications and has significantly influenced the front-end development landscape.

Create ReactJS project with Ant Design templates (See Ant Design Documentation)

1. Understand class and function component

Function component return JSX.element

Class component extend React Component

Understand class component life cycle (Skip for the moment, it is too complicated)

1. Use of function component

We cannot use state in our functional style react component, the props is going to pass over the first argument of the function. Writing functional style react component is essentially writing only the render method directly without the class

Function component use function parameter props and use React Hooks states

3 phases in the React Component LifeCycle:

1. Mounting Phase

2. Updating Phase

3. Un-Mounting Phase

1. Hook characteristics after v16.8

Hook is used to simplify life cycle management

React Hooks create state variable like “this” in class

React Hooks should not be put inside conditional and iterations. So it cannot be use inside if-else ,for-loop and while-loop.

React Hooks can only be used in function components. You cannot use them inside class component. If you are integrating some libraries that make use of React Hook, you would need to use Function component.

React Hooks must be run in exact order every times the render function is run.

1. How parameters are transferred between components: parent to child, child to parent, between same levels and multiple levels
2. Understand functions like: useState, useCallback, useMemo, useEffect, useRef, useContext, useReducer
3. Understand React router
4. Management of states of entire app  
   See entire app
5. Class component use Redux and Redux thunk, function component use zustand
6. Webpack
7. Node (LTS 20.16.0 use nvm node)
8. NPM
9. Antd
10. CSS extension use less and scss