React Notes

1. **What is react?**
    - React is a free and open-source front-end JavaScript library developed by Facebook.
    - It allows developers to create large web applications that can update and render efficiently in response to data changes.
    - Primarily used for creating a Single Page Application (SPA).

2. **What are the features of react?**
    - Component-based.
    - jsx syntax.
    - Virtual DOM.
    - Unidirectional data flow or one way binding.
    - Hooks.

3. **What is jsx?**
    - Jsx stands for JavaScript XML.
    - JSX allows you to write HTML elements directly within JavaScript code, making it easier to create and manage UI components.
    - You can embed JavaScript expressions within JSX by wrapping them in curly braces {}. This allows for dynamic content rendering.

4. **What is Real DOM vs Virtual DOM vs Shadow DOM?**
    - **Real DOM**:
        - The Document Object Model (DOM) is a tree-like structure representing the HTML elements of a webpage.
        - Directly manipulated by JavaScript.
        - Updates are slow and expensive because the entire DOM is re-rendered whenever a change occurs.
    - **Virtual DOM**:
        - A lightweight copy of the Real DOM, maintained in memory by React.
        - React uses the Virtual DOM to optimize updates by only re-rendering parts of the DOM that have changed.
        - When a change occurs, React creates a new Virtual DOM **tree** and compares it with the previous old Virtual DOM **tree**. This **process called "diffing".**
        - Only the differences are updated in the Real DOM, making updates faster and more efficient.
        - **Example**: When a component's state changes, React updates the Virtual DOM first, then applies only the necessary changes to the Real DOM.
    - **Shadow DOM:**
        - A browser technology used to encapsulate the styles and structure of web components.
        - Allows for the creation of isolated DOM trees within elements, preventing style and script conflicts.
        - Commonly used in custom elements and web component.
        - Changes in the Shadow DOM do not affect the Real DOM and vice versa.

    - **Real DOM:** The actual DOM structure of the webpage, slow to update.
    - **Virtual DOM:** An in-memory representation of the Real DOM used by React to optimize updates.
    - **Shadow DOM:** Encapsulated DOM trees used for web components to avoid conflicts.

5. **What is One way data binding?**
    - One way data binding refers to the flow of data in a single direction, from the component's state or props to the user interface (UI). This means that any changes in the component's state will automatically update the UI, but changes in the UI will not automatically update the state.

6. **What is component?**
    - components are the building blocks of the user interface.
    - In react components are of two types they are
        i) **Class component**:
            - These were more common in older React codebases. They are written as ES6 classes and extend from React.Component. They have a render method that returns HTML.
            - Class is state-full component

        ii) Function component:
            - These are simpler and more commonly used in modern React applications. They are written as JavaScript functions and can use hooks to manage state and lifecycle events.
            - Function is a state-less component.

7. **Difference between npm and yarn?**

npm	yarn
Node Package Manager	Yarn
Bundle of libraries in one single place	Yarn
Slower	Fast
Installation will be one by one	Installation will be parallel.

8. **What is npx?**
    - npx stands for Node package execute.
    - It's a command-line tool that comes bundled with npm (starting from version 5.2.0).
    - The main purpose of npx is to allow developers to execute Node.js packages directly from the npm registry without needing to install them globally on their system.

9. **How to create the react app?**
    - npx create-react-app my-app
        - ➢ This can be used with npm 5.2+ and higher. But it was deprecated.
    - npm init react-app my-app
        - ➢ This was available after npm 6+.
    - yarn create react-app my-app
        - ➢ This was for yarn.

10. **What is package.json and package-lock.json?**
    - **Package.json:**
        - This file serves as the main configuration file for your project.
        - It includes metadata about the project (like name, version, and description), scripts for various tasks, and a list of dependencies required for the project.
        - Lists the packages your project depends on, with version ranges specified using semantic versioning (e.g., ^1.0.0 for minor updates, ~1.0.0 for patch updates).
    - Package-lock.json:
        - This file locks the exact versions of dependencies and their sub-dependencies, ensuring consistent installations across different environments.
        - It provides a detailed tree of all dependencies, including their exact versions.
        - Ensures that everyone working on the project, as well as CI/CD systems, use the exact same versions of dependencies, preventing the "it works on my machine" problem.

11. **What are node modules?**
    - The node_modules directory is crucial as it contains all the dependencies required to build and run your application. When you create a new React app, this folder is automatically generated and populated with numerous packages.
    - Dependency Storage:
        - The node_modules folder stores all the packages listed in your package.json file. This includes essential libraries like React and React-DOM, as well as build tools like Webpack or Vite, and linters like ESLint or Prettier.
    - **Not Included in Version Control**:
        - Due to its large size, the node_modules directory is typically excluded from version control systems like Git. Instead, the package.json file is tracked, which lists all the dependencies. Other developers can then use this file to regenerate the node_modules directory by running npm install.
    - **Third-Party Modules**:
        - It also includes third-party modules that provide additional functionality to your application. These can be installed using npm (Node Package Manager) or yarn.

12. **What is Single Page Application (SPA)?**
    - you can navigate over several different pages without having to load the page as an entirely new entity. In short, a single-page application dynamically rewrites the current page with new data from the web server, instead of the default method of loading entire new pages on its own.

13. **What is react fragment?**
    - Wrap elements in <Fragment> to group them together in situations where you need a single element. Grouping elements in Fragment has no effect on the resulting DOM; it is the same as if the elements were not grouped. The empty JSX tag <></> is shorthand for <Fragment></Fragment> in most cases.

14. **What is component composition?**
    - Component composition in React refers to combining smaller or simpler components to build bigger or more complex user interfaces. Components in React are building blocks that are used to create user interfaces. By accepting props, which are to React components what parameters are to functions, we can make components more generic. Component composition is the name for passing components as props to other components, thus creating new components with other components.

15. What is props in react?
    - React components use props to communicate with each other. Every parent component can pass some information to its child components by giving them props. Props might remind you of HTML attributes, but you can pass any JavaScript value through them, including objects, arrays, and functions.

16. What is props trill?
    - Prop drilling, sometimes called "threading props" or "component chaining," is the method of passing data from a parent component through a series of nested child components using props. This happens when you need to send a prop through multiple levels of components to get it to a deeply nested child component that requires it.

17. What is setState in react?
    - In React, **state** refers to an object that holds information about the component that may change over time. The state is crucial for creating dynamic and interactive user interfaces, as it allows components to respond to user input, server responses, or other changes.
    - State in React is typically initialized in the constructor of a class component. Here's a basic example of how to set up state in a React class component:
    - With the introduction of **hooks in React 16.8**, functional components can now manage state using the useState hook. Here's how you can use state in a functional component
    - Best Practices
        - Always use this.setState() or the state update function from useState to change the state.
        - Treat state as immutable. Do not modify it directly.
        - When the new state depends on the previous state, use a function with this.setState() or the updater function from useState.

18. What is useState?
    - The useState hook is a fundamental aspect of React that allows you to add state variables to functional components. Before hooks were introduced, stateful logic could only be used within class components. However, useState provides the functionality to store and manipulate state within functional components.

19. What is useRef in the react?
    - The useRef hook in React is a powerful tool that allows you to create a reference to a value that persists across renders without causing a re-render when updated. This hook is particularly useful for accessing and manipulating DOM elements directly, as well as for storing mutable values that do not affect the visual output of your component.

20. What is redux in react?
    - Redux is a **predictable state container** for JavaScript applications, commonly used with React to manage the state of the application efficiently. It helps in maintaining the state of the application in a single store, making it easier to manage and debug.

21. What is middleware?
    - Middleware in React JS is a software service that provides a third-party extension point between dispatching an action and handing the action off to the reducer. Middleware allows for side effects to be run without blocking state updates. For example, middleware can be used to run API requests or logging in response to specific or every action that is dispatched.

22. What is redux thunk in react?
    - Redux Thunk is a **middleware library for Redux**, a state management library for React applications. It allows you to return functions from action creators, which enables you to handle asynchronous operations in a more elegant way.
23. What is react lifef cycle?|
    - React components go through a series of phases during their existence, known as the **React Lifecycle**. These phases include **Mounting**, **Updating**, and **Unmounting**. Each phase has specific methods that can be overridden to execute custom logic at different points in the component's lifecycle.
    - **Mounting Phase**
        - The **Mounting** phase involves putting elements into the DOM. The methods called during this phase are:
            i) constructore();
            ii) getDerivedStateFromProps();
            iii) render();
            iv) ComponentDidMount();
    - **Updating phase**
        - The **Updating** phase occurs when a component's state or props change. The methods called during this phase are:
            i) getDerivedStateFromProps();
            ii) shouldComponentUpdate();
            iii) render();
            iv) getSnapshotBeforeUpdate();
            v) ComponentDidUpdate();
    - **Unmounting phase**
        - The Unmounting phase occurs when a component is removed from the DOM. The method called during this phase is:
            i) componentWillUnmount();
24.