

Ans.1

React JS is an open-source JavaScript library for building user interfaces, especially single-page applications. Developed by Facebook, it uses a component-based architecture and a virtual DOM to efficiently update the UI. Key features include:

Component-Based: Reusable UI elements.

Virtual DOM: Improves performance by minimizing direct DOM manipulation.

Declarative: Simplifies UI development by letting developers describe how the UI should look.

JSX: Syntax extension that combines JavaScript and HTML.

State and Props: Manage dynamic data and component properties.

Ans.2

NPM (Node Package Manager) in React JS is used to manage and install packages (libraries or modules) that React projects depend on. It helps with:

Dependency Management: Installing and updating packages like React.

Running Scripts: Commands like `npm start` (start development server) and `npm run build` (build project for production).

Project Initialization: Tools like Create React App use NPM to set up new React projects.

Ans.3

Node.js plays a crucial role in React JS development by:

Package Management: Using NPM to install and manage dependencies.

Development Tools: Supporting tools like Create React App, Webpack, and Babel for setting up and building projects.

Server-Side Rendering (SSR): Enabling SSR for better performance and SEO with frameworks like Next.js.

Development Server: Running local servers for testing and debugging React applications.

Ans.4

In React JS, CLI (Command Line Interface) commands are used to perform various tasks related to project setup, development, and management. Key CLI commands include:

Create React App: Initializes a new React project.

```
npx codecreate-react-app my-app
```

Start Development Server: Runs the local development server.

```
npm start
```

Build Project: Compiles the project for production.

```
npm run build
```

Run Tests: Executes tests for the application.

```
npm test
```

Ans.5

In React JS, components are the building blocks of a React application. They are reusable, self-contained pieces of UI that encapsulate their own structure, styling, and behavior. There are two main types of components:

Functional Components: Defined as JavaScript functions. They receive props and return JSX.

```
function Greeting(props) {  
  return <h1>Hello, {props.name}!</h1>;  
}
```

```
function Greeting(props) { return <h1>Hello, {props.name}!</h1>; }
```

Class Components: Defined using ES6 classes. They can have state and lifecycle methods.

```
class Greeting extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}!</h1>;  
  }  
}
```

```
class Greeting extends React.Component { render() { return <h1>Hello, {this.props.name}!</h1>; } }
```

Ans.6

In React JS, Header and Content components are examples of UI components that represent specific sections of a web page.

Header Component

Purpose: Represents the top section of a web page, typically containing navigation links, logos, and titles.

Example:

```
javascript
```

```
function Header()
```

```
{
  return (
    <header> <h1>My Website</h1>
    <nav> <a href="#home">Home</a>
    <a href="#about">About</a>
    <a href="#contact">Contact</a>
    </nav>
    </header> );
}
```

Content Component

Purpose: Represents the main content area of a web page, where the primary information or functionality is displayed.

Example:

```
function Content() {
  return (
    <main>
      <h2>Welcome to My Website</h2>
      <p>This is the main content section.</p>
    </main>
  );
}
```

```
function Content() { return ( <main> <h2>Welcome to My Website</h2> <p>This is the main content section.</p> </main> ); }
```

ans.7

Install Node.js (which includes NPM):

Download and install Node.js from nodejs.org.

Create a React App:

```
npx create-react-app my-app
```

```
cd my-app
```

```
npm start
```

Install Node.js and NPM:

Using a package manager:

```
sudo apt update
```

```
sudo apt install nodejs npm
```

Or use Node Version Manager (NVM) for managing multiple Node versions:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash
```

```
source ~/.nvm/nvm.sh
```

```
nvm install node
```

Create a React App:

```
npx create-react-app my-app
```

```
cd my-app
```

```
npm start
```

Installing NPM

NPM is included with Node.js. When you install Node.js, NPM is installed automatically.

Checking NPM Version

```
npm -v
```

Ans.8

To check the version of React JS installed in your project, you can use the following command in your project directory:

```
npm list react
```

This command will display the version of React JS installed locally in your project's node_modules directory.

Alternatively, if you're using React in your application code, you can output the React version directly in your code like this:

```
import React from 'react';
```

```
console.log(React.version);
```

Ans.9

Locate the Component: Identify the component file you want to modify.

Edit the Component: Open the component file in your code editor and make the necessary changes to the JSX, state, props, or any other logic within the component.

Save Changes: Save the changes you've made to the component file.

View Changes: If you're running a development server (npm start), the changes will be automatically reflected in your browser. Otherwise, you'll need to rebuild your project (npm run build) and view the changes in your deployed application.

Ans.10

```
import { useState } from 'react';
```

```
function App() {
```

```
  let [count, setCount] = useState(0); // usestate always two value return do.(array ) and your variable  
  namre
```

```
  const Increment = () => {
```

```
    if (count >= 15) {
```

```
      setCount(count);
```

```
    } else {
```

```
      setCount(count + 1);
```

```
    }
```

```
  };
```

```
  const Decrement = () => {
```

```
    if (count <= 0) {
```

```
      setCount(count);
```

```
    } else {
```

```
      setCount(count - 1);
```

```
    }
```

```
  };
```

```
  const reset = () => {
```

```
    setCount(count = 0);
```

```
};
```

```
return (
```

```
  <div>
```

```
    <h1>Count:{count}</h1>
```

```
    <button onClick={Increment}>Increment</button>
```

```
    <button onClick={Decrement}>Decrement</button>
```

```
    <button onClick={reset}>reset</button>
```

```
  </div>
```

```
);
```

```
}
```

```
export default App;
```