# Module – 9) ReactJS Intro

Q1) What is React JS?

Ans: React.js is a JavaScript library for building user interfaces, primarily used for creating interactive web applications.

-> React.js is a popular open-source JavaScript library developed by Facebook.

-> It's designed for building user interfaces with a component-based architecture.

-> React allows developers to create reusable UI components.

-> These components manage their own state, making applications more predictable and easier to maintain.

-> React uses a virtual DOM (Document Object Model) for efficient rendering.

-> Changes to the virtual DOM are compared with the actual DOM, and only the differences are updated, improving performance.

-> JSX (JavaScript XML) is a syntax extension used in React to write HTML-like code within JavaScript.

-> JSX simplifies the creation of React elements and makes code more readable.

-> It's widely adopted by companies like Facebook, Instagram, Airbnb, and many others.

Q2) What is NPM in React JS?

Ans: In React.js, NPM stands for Node Package Manager. NPM is a package manager for JavaScript programming language, primarily used for managing project dependencies and distributing JavaScript modules

-> Dependency Management: NPM is used in React.js to manage project dependencies. Developers can specify the required packages and their versions in the package.json file, and NPM installs and manages these dependencies automatically.

-> Package Installation: React.js developers use NPM to install various packages and libraries needed for their projects. For example, popular React libraries like React Router, Redux, Axios, etc., are typically installed using NPM

-> Script Management: NPM allows developers to define custom scripts in the package.json file, which can automate tasks like building the project, running tests, and deploying the application

-> Version Control: NPM helps in version control by allowing developers to lock the dependencies to specific versions. This ensures consistency across different development environments and helps prevent unexpected issues due to dependency changes.

Q3) What is Role of Node Js in react Js?

Ans: Node.js plays several important roles in the React.js development ecosystem:

-> Server-side Rendering (SSR): Node.js can be used to implement server-side rendering for React applications. SSR improves performance and search engine optimization by generating HTML on the server and sending it to the client, rather than relying on client-side rendering alone.

-> Development Environment: Node.js provides a powerful runtime environment for building and running React applications locally. Developers can use tools like Create React App, Next.js, or Gatsby, which are built on top of Node.js, to scaffold, develop, and test React applications efficiently.

-> Middleware and APIs: Node.js can serve as a backend server for React applications, allowing developers to create RESTful APIs, handle database operations, and implement business logic. Express.js, a popular Node.js framework, is often used for building such server-side components.

-> Build Tools: Node.js is commonly used in conjunction with build tools like Webpack, Babel, and ESLint to compile, bundle, and optimize React code. These tools help streamline the development workflow and ensure compatibility across different browsers.

-> Package Management: Node.js powers npm (Node Package Manager), which is used to manage dependencies and packages in React.js projects. Developers can use npm to install, update, and share reusable code libraries and modules.

-> Deployment: Node.js is often used to deploy React applications to production environments. Platforms like Heroku, AWS, and DigitalOcean support Node.js, allowing developers to easily deploy and scale their React applications.

-> Proxy Servers: During development, Node.js can act as a proxy server to handle cross-origin requests and avoid CORS (Cross-Origin Resource Sharing) issues when making API calls to backend services.

Q4) What is CLI command In React Js?

Ans: In React.js, CLI (Command Line Interface) commands are tools provided by frameworks like Create React App (CRA) or Next.js to streamline the development process. These commands are executed in the terminal or command prompt and help developers scaffold, build, test, and deploy React applications. Here are some common CLI commands used in React.js development:

-> create-react-app: This command is used to create a new React project using Create React App, a popular toolchain for building React applications. It sets up the project structure, configuration files, and initial dependencies.

-> npm start: This command starts the development server for a React application. It compiles the code, launches a local server, and opens the application in the default web browser. Developers can then view and test their application in a development environment.

-> npm run build: This command generates a production-ready build of the React application. It compiles the code, optimizes assets, and creates a bundle that can be deployed to a web server. This build is typically used for production deployment.

Q5) What is Components in React Js?

Ans: Components in React.js are modular UI building blocks, encapsulating reusable code for specific elements. Functional components are simple functions that accept props, while class components manage state and lifecycle. Props pass data from parent to child components. Components enhance code organization, reusability, and scalability in React development.

Q6) What is Header and Content Components in React Js?

Ans: In React.js, Header and Content components are common UI elements used to structure web applications.

-> Header Component: Typically placed at the top of a page, the Header component contains branding, navigation links, or any other content that remains consistent across multiple pages.

-> Content Component: This component houses the main content of a web page, including text, images, forms, or other interactive elements. It dynamically renders based on user interactions or data changes.

-> These components aid in organizing and separating concerns within the application, promoting modularity and reusability.

Q7) How to install React Js on Windows, Linux Operating System? How to Install NPM and How to check version of NPM?

Ans: Installing React.js on Windows and Linux:

-> Install Node.js and npm:

Visit the official Node.js website: <https://nodejs.org/>

Download the appropriate installer for your operating system (Windows or Linux).

Follow the installation instructions to install Node.js and npm.

-> Create a New React Project:

Open a terminal or command prompt.

Use the following npm command to create a new React project: npx create-react-app my-react-app

Q8) How to check version of React Js?

Ans: To check the version of React.js installed in your project, you can navigate to your project directory and look for the package.json file, where dependencies including React are listed along with their versions. Alternatively, you can run the following command in the terminal or command prompt within your project directory: npm list react

This command will display the version of React.js installed in your project. If you have multiple versions of React installed, it will display the version used by your project.