##

React JS By Prasad

##

Introduction to React.js:

React.js, commonly referred to as React, is an open-source JavaScript library used for building user interfaces (UIs) and single-page applications.

Developed by Facebook, React allows developers to create reusable UI components and efficiently update and render data changes in real-time.

Key Concepts:

Component-Based Architecture:

React follows a component-based architecture, where UIs are divided into smaller, reusable components.

Each component encapsulates its own logic and rendering, making it easier to manage complex UI structures.

Virtual DOM (Document Object Model):

React uses a virtual DOM to enhance performance. The virtual DOM is a lightweight copy of the actual DOM.

When data changes occur, React calculates the difference between the virtual DOM and actual DOM (reconciliation), updating only the necessary parts to minimize performance bottlenecks.

JSX (JavaScript XML):

JSX is a syntax extension that allows you to write HTML-like code within JavaScript.

JSX makes it easier to define component structures and renders in a more declarative manner.

State and Props:

State: Each React component can have its own state, which represents mutable data that can change over time. State updates trigger component re-renders.

Props (Properties): Props are immutable data passed from parent components to child components. They help maintain the flow of data and communication between components.

One-Way Data Binding:

React enforces one-way data binding, meaning data flows in a single direction—from parent to child components.

This simplifies debugging and understanding how data changes propagate through the UI.

Lifecycle Methods:

Components have lifecycle methods that allow developers to control behavior at various stages, such as component creation, update, and destruction.

Common lifecycle methods include componentDidMount, componentDidUpdate, and componentWillUnmount.

React Hooks:

Hooks are functions that allow you to "hook into" React state and lifecycle features from functional components, eliminating the need for class components in many cases.

Examples of hooks include useState for managing state and useEffect for handling side effects.

Advantages of React:

Reusable Components: Components can be reused across different parts of an application, promoting code modularity and reducing duplication.

Efficient Updates: React's virtual DOM enables efficient updates by minimizing direct interactions with the actual DOM.

Declarative Syntax: JSX's declarative syntax makes it easier to understand and visualize UI components.

Strong Ecosystem: React has a vast ecosystem of libraries, tools, and community support, making development faster and more efficient.

Unidirectional Data Flow: One-way data flow simplifies data management and minimizes unexpected behavior.

Installation:

Downloading and Installing Visual Studio Code (VS Code):

- **Visit the VS Code Website:** Go to the official Visual Studio Code website: https://code.visualstudio.com/
- **Download VS Code:**
  Click on the "Download for Windows" button if you're using Windows, "Download for Mac" if you're using macOS, or "Download for Linux" if you're using a Linux distribution.
- **Install VS Code:**
  Once the download is complete, run the installer executable file you downloaded. Follow the on-screen instructions to install VS Code on your computer.
- **Launch VS Code:**
  After the installation is complete, you can launch VS Code from your computer's application menu or desktop shortcut. (install extensions Live Server and Preetier)

Downloading and Installing Node.js:

- **Visit the Node.js Website:** Go to the official Node.js website: https://nodejs.org/
- **Download Node.js:** By default, the Node.js website will recommend the LTS (Long Term Support) version of Node.js. This is a stable and reliable option for most users. Click on the "LTS" button to download the recommended version.
- **Run the Installer:** After the download is complete, run the installer executable file you downloaded. Follow the on-screen instructions to install Node.js on your computer.
- **Check Installation:** To confirm that Node.js and its package manager (npm) have been installed successfully, open a command prompt (Windows) or terminal (macOS/Linux) and run the following commands:

  ```
  node -v
  ```

```
npm -v
```

If the installation was successful, these commands will display the installed versions of Node.js and npm.

Install Chrome Extension “REACT Developer Tools”

Attaching React through a Content Delivery Network (CDN) is a quick way to start using React without needing to set up a development environment with Node.js and a build process. Here's how you can attach React and ReactDOM using CDNs:

HTML Setup:

Create an HTML file where you want to use React and ReactDOM.

Include React and ReactDOM via CDNs:

Inside the <head> section of your HTML file, include the following <script> tags to attach React and ReactDOM using CDNs:

```
<!DOCTYPE html>
<html>
<head>
  <title>React via CDN</title>
  <script src="https://cdn.jsdelivr.net/npm/react@17.0.2/umd/react.development.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/react-dom@17.0.2/umd/react-dom.development.js"></script>
</head>
<body>
  <div id="root"></div>
  <!-- Your React components will be rendered here -->
</body>
</html>
```

Write React Components:

In the <body> section of your HTML file or in a separate <script> tag, you can write your React components using the React.createElement function.

```
<script>
  const element = React.createElement('h1', null, 'Hello, React via CDN!');
  ReactDOM.render(element, document.getElementById('root'));
</script>
```

Open the HTML File:

Save the HTML file and open it in a web browser. You should see the rendered React component.

Please note that while using CDNs is convenient for quick experiments, it's not recommended for production applications due to potential performance and reliability issues. For production use, it's better to set up a development environment using Node.js, npm, and a bundler like Webpack or Parcel to manage your React application.

Creating a React App with create-react-app:

create-react-app is a command-line tool provided by Facebook to set up a new React project with a default configuration. It simplifies the process of creating a modern React application by handling build configurations, development server, and more. Here's how to create a React app using create-react-app:

Install Node.js:

Make sure you have Node.js and npm (Node Package Manager) installed on your computer. You can download and install them from the official Node.js website: https://nodejs.org/

Install create-react-app:

Open your command-line interface (Terminal, Command Prompt, etc.) and run the following command to install create-react-app globally:

npm install -g create-react-app

Create a New React App:

Once create-react-app is installed, you can create a new React app by running the following command:

npx create-react-app my-react-app

Replace my-react-app with the name you want for your app. This command will create a new directory with the specified name and set up the React app inside it.

Navigate to App Directory:

Change your working directory to the newly created app directory:

cd my-react-app

Start the Development Server:

To start the development server and view your React app in the browser, run the following command:

npm start

This will launch a development server, and your app will be available at http://localhost:3000.

Explore and Modify:

Open the app directory in your code editor. The main source code is located in the src folder. You can modify the src/App.js file to edit the default content of your app.

Build for Production:

When you're ready to deploy your app, you can build a production-ready bundle using the following command:

npm run build

This command creates an optimized build of your app in the build directory.

Notes:

create-react-app abstracts away the complex build configurations, making it easier to focus on writing React code.

The generated app comes with a development server, hot reloading, and other development tools out of the box.

You can customize your app's configuration by ejecting from create-react-app, but this is recommended only if you're familiar with the underlying build tools.

To use additional packages, you can install them using npm install or yarn within your app directory.

Creating a React app with create-react-app is a great starting point for beginners and experienced developers alike. It saves time and provides a solid foundation for building modern, efficient, and maintainable React applications.