

Introuction to React JS

By

Narasimha Rao T

Microsoft.Net FSD Trainer

Professional Development Trainer

tnrao.trainer@gmail.com

1. Introduction to Client-Side Technologies

- **Definition:** Client-side technologies are tools and frameworks that run on the user's browser rather than the server.
- **Purpose:**
 - Improve **user experience** with interactive UI.
 - Reduce **server load** by handling rendering and logic on the client side.
 - Enable **faster updates** and dynamic web applications.

2. Significance of Client-Side Technologies in Modern Web Development

- Create rich, dynamic, and responsive applications.
- Enhance performance by minimizing server round trips.
- Support Single Page Applications (SPAs).
- Allow component reusability for faster development.
- Enable cross-platform compatibility with mobile and desktop apps.

3. Examples of Client-Side Technologies

- HTML, CSS, JavaScript (core building blocks).
- Frameworks & Libraries:
 - Angular
 - React
 - Vue.js
 - Bootstrap
 - jQuery

4. What is React?

- **React:** A JavaScript library for building user interfaces.
- Developed and maintained by **Facebook (Meta)**.
- Focuses on **component-based architecture** and efficient **DOM updates** with the **Virtual DOM**.

5. History of React

- **2011:** Created by Jordan Walke (Facebook engineer).
- **2013:** Open-sourced to the public.
- **2015 onwards:** Adoption grew due to its simplicity and efficiency.
- Now one of the most widely used **frontend libraries**.

6. Versions of React

- React 0.3 (2013): First release.
- React 15 (2016): Improved rendering performance.
- React 16 (2017): Introduced **Fiber architecture** and **Error Boundaries**.
- React 17 (2020): Gradual upgrades and improved event handling.
- React 18 (2022): Introduced **Concurrent Rendering** and **Automatic Batching**.

7. React Features

- **Component-based:** Build UI with reusable components.
- **Virtual DOM:** Efficient rendering by updating only changed elements.
- **JSX (JavaScript XML):** Write HTML-like syntax inside JavaScript.
- **One-way Data Binding:** Ensures data flow is predictable.
- **Fast Rendering:** Optimized UI updates.
- **Strong Community:** Large ecosystem of tools and libraries.

8. React Component-Based Architecture

- **Component** = Independent, reusable piece of UI.
- Types:
 - **Functional Components** (preferred).
 - **Class Components**.
- Components can be **nested, reused, and managed** easily.

9. Setting Up with Create-React-App

- Create React App (CRA) is a tool to quickly scaffold a React project.
- Steps:

```
npx create-react-app my-app  
cd my-app  
npm start
```

- Opens app at <http://localhost:3000/>.

10. Set Up First React Project

- Install **Node.js** (includes npm).
- Use **create-react-app** or alternatives like **Vite**.
- Start development server and modify **App.js**.

11. Folder Structure Overview (CRA)

```
my-app/
├── node_modules/ → Dependencies
├── public/ → Static files (index.html, favicon)
├── src/ → React code
│   ├── App.js → Main component
│   ├── index.js → Entry point
│   └── App.css → Styling
└── package.json → Project metadata & dependencies
```

12. Create a Simple “Hello World” Component

```
// HelloWorld.js
import React from "react";

function HelloWorld() {
  return <h1>Hello, World!</h1>;
}

export default HelloWorld;
```

```
// In App.js
import HelloWorld from "../HelloWorld";

function App() {
  return (
    <div>
      <HelloWorld />
    </div>
  );
}

export default App;
```

13. What is JSX?

- **JSX** = JavaScript XML.
- Syntax extension that lets us write **HTML-like code** inside JavaScript.
- Example:

```
const element = <h1>Hello JSX!</h1>;
```

14. JSX Syntax and Rules

- Must return a single root element.
- Use `className` instead of `class`.
- JSX expressions enclosed in `{}`.
- Elements must be properly closed.
- Example:

```
function Greeting() {  
  const name = "John";  
  return <h2>Hello, {name}</h2>;  
}
```


Understanding Virtual DOM in React JS

Self Check Questions

Important Quiz Questions

1. What is the significance of client-side technologies in modern web apps?
2. Who developed React and in which year?
3. Explain the difference between Real DOM and Virtual DOM.
4. What are the main features of React?
5. What are the types of React components?

Important Quiz Questions

6. How do you create a new React project using create-react-app?
7. What is the role of `index.js` in a React project?
8. Write a simple JSX code that displays your name.
9. What are the rules of JSX syntax?
10. Explain one-way data binding in React.

Q & A

Narasimha Rao T

tnrao.trainer@gmail.com