


CELEBRATING
13 YEARS

QualityThought®



React JS

ReactJS 18.2.0 Training Overview

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.

It is used for creating dynamic and interactive user interfaces for mobile and web applications. It is highly flexible, declarative and efficient for developing scalable, simple, and fast front-end for web & mobile applications. In simple terms, React JS effectively handles the view layer of mobile and web application.

React can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks like Next.js. However, ReactJS is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality

React is flexible in such a way that, in any application, we can use as little or as much React as you need. For example, react can be used in any existing web application to develop a new feature or even the application's entire UI.

ReactJS has become highly popular across the globe because of its extra simplicity and flexibility. Many people are even referring to ReactJS as the future of web development.

Part of this huge popularity comes from the fact that top corporations such as Facebook, PayPal, Uber, Instagram, Airbnb etc use it to develop the user interfaces.

Objectives of the Course

This is to provide in-depth insights about ReactJS and keep updated yourself with latest ReactJS concepts and it's integrated other technologies according to updated industry standards.

⇒ The main objective of React Training is to make you as a top-class React.js Developer able to design and build modern Interactive user interface components to enhance application performance.

Who should do the Course

⇒ Anyone who is trying to learn fastest growing UI framework and one of the top 3 frameworks. It is growing fast in terms of opportunities too.

Prerequisites

- ⇒ HTML
- ⇒ JavaScript
- ⇒ CSS

ES6: JavaScript concepts related to ReactJS

- ⇒ History of JavaScript
- ⇒ Features
- ⇒ let & const with example
- ⇒ Arrow Functions
- ⇒ Alternative

Corporate Training

Complete ReactJS (In-Depth)

- ⇒ Tips and Arrow Functions
- ⇒ Exports and Imports
- ⇒ Tips for exports and imports
- ⇒ Classes with example
- ⇒ Classes example
- ⇒ Inheritance
- ⇒ Spread and rest Parameter
- ⇒ Destructuring

ReactJS: Introduction

- ⇒ What is ReactJS?
- ⇒ Version history.
- ⇒ Why ReactJS?
- ⇒ Advantages of React JS
- ⇒ Internals of ReactJS.
- ⇒ Work flow of React JS
- ⇒ Scope of React JS
- ⇒ Software's required.

ReactJS: Setup – First Application

- ⇒ About NodeJS
- ⇒ Installation
- ⇒ Npm – managing the packages
- ⇒ Create First ReactJS Application
- ⇒ Build and Execute
- ⇒ About Visual Studio Code
- ⇒ Installation of Visual Studio Code
- ⇒ Managing the ReactJS application using Visual Studio Code

ReactJS: JSX

- ⇒ Overview of JSX
- ⇒ Jsx syntax
- ⇒ Difference between JS and JSX.
- ⇒ Jsx Parser
- ⇒ Comments in jsx
- ⇒ Introduction of Virtual DOM.
- ⇒ Rendering an Element into the DOM
- ⇒ Naming Conventions
- ⇒ JSX Transpilation to React Code Example
- ⇒ Running the Transpiled Code
- ⇒ What are the JavaScript expressions available in JSX?

ReactJS: All about Components

- ⇒ React Components
- ⇒ Creating a Functional Component Example
- ⇒ Component Names Must Be Capitalized
- ⇒ Components vs Elements
- ⇒ Elements Are Immutable
- ⇒ Properties
- ⇒ Property Naming Convention
- ⇒ Properties Default to 'True'
- ⇒ Spread Attributes (an ES6 Feature)
- ⇒ Expressions
- ⇒ Fragments
- ⇒ Functional Components
- ⇒ Nesting JSX Elements
- ⇒ Example of JSX Nesting
- ⇒ Comments in JSX Code
- ⇒ Setting CSS Styles Using Classes
- ⇒ Setting CSS Styles Directly
- ⇒ JSX Escapes Values
- ⇒ Working with Lists of Items - Iterating through Lists
- ⇒ Keys in Lists
- ⇒ Example List with Key
- ⇒ State
- ⇒ State in React
- ⇒ Setting state
- ⇒ Types of State Data
- ⇒ State Hierarchy
- ⇒ Lifting State Up
- ⇒ Props vs. State
- ⇒ Pass Down a Function
- ⇒ Immutability
- ⇒ Immutability – Why?
- ⇒ Virtual DOM and State
- ⇒ Updating Input fields
- ⇒ Passing Props to Components
- ⇒ Passing Functions to Components
- ⇒ Event Handling
- ⇒ Event Handler Example
- ⇒ Event Binding - Dos
- ⇒ Event Binding – Don'ts
- ⇒ Passing Parameters to Event Handlers
- ⇒ Component Life-cycle
- ⇒ Initial Render
- ⇒ Props Change
- ⇒ State Change
- ⇒ Component willMount
- ⇒ Component didMount
- ⇒ Component Unmount
- ⇒ Life-cycle in Functional Components
- ⇒ Applying Different Lifecycles in the Application.
- ⇒ When to choose Appropriate lifecycles.
- ⇒ Interaction in between components
- ⇒ Component Communication using Context

Forms and UI

- ⇒ Lists of Form components.
- ⇒ Setup Controlled and Uncontrolled form components.
- ⇒ Control Input elements.
- ⇒ How to set default values on all formats of Input elements.
- ⇒ React JS Form validations.
- ⇒ How to write Styles? ⇒ Animations overview
- ⇒ Create a React Form.
- ⇒ Client-side form validation.
- ⇒ Applying form components.
- ⇒ Submit and reset the form.

Working with Rest API

- ⇒ Calling REST API
- ⇒ Sending POST request
- ⇒ CRUD Operations example

React Router

- ⇒ Routing and Navigation
- ⇒ react-router
- ⇒ Creating a react-router based project
- ⇒ A Basic Routed Component
- ⇒ Router vs. BrowserRouter
- ⇒ The Route component
- ⇒ Redirect Route
- ⇒ Navigating with
- ⇒ Route Parameters
- ⇒ Retrieving Route Parameters
- ⇒ QueryString Parameters
- ⇒ Create a Single Page Application.
- ⇒ Applying Routing.
- ⇒ Dynamically render the components based on the url

State Management for React

- ⇒ React State Basics – Props and State
- ⇒ Props
- ⇒ State in Class Based Components
- ⇒ Managing State with Hooks in Functional Components
- ⇒ The Problem with Props and State
- ⇒ Redux State Library
- ⇒ Redux Advantages
- ⇒ Redux Disadvantages
- ⇒ Basic Rules for State Management
- ⇒ Types of State ⇒ Data State
- ⇒ Communication State

- ⇒ Control State ⇒ Session State
- ⇒ Location State
- ⇒ Location State Side Effects

Event handling in JSX

- ⇒ on Blur, onKeyUp, onChange and other useful primary events in React JS.
- ⇒ How to Sharing events between the components?
- ⇒ Communicate Data between components.
- ⇒ Applying all lists of events

Styles in ReactJS

- ⇒ CSS and inline styles in React JS overview.
- ⇒ Introduction to styled components
- ⇒ Styling the application using styled component
- ⇒ How to use Animations in the Application

Building React Apps with Redux

- ⇒ Redux
- ⇒ Redux Terminology
- ⇒ Redux Principles
- ⇒ Redux: Actions
- ⇒ Redux Action Types
- ⇒ Action Creators
- ⇒ Dispatching Actions
- ⇒ Data Flow Basics
- ⇒ Redux Reducers
- ⇒ Pure Functions
- ⇒ Reducer Example
- ⇒ Returning Default State
- ⇒ Creating a Development Environment with create-react-app
- ⇒ Using Redux with React
- ⇒ Initializing the Store
- ⇒ Immutability
- ⇒ Benefits of Immutable State
- ⇒ Mutability of Standard types
- ⇒ Copying Objects in JavaScript
- ⇒ Copying Arrays in JavaScript
- ⇒ One Store - Multiple Reducers
- ⇒ Combining Reducers
- ⇒ Components and Redux
- ⇒ The React-Redux Package
- ⇒ Wrapping App with Provider
- ⇒ mapStateToProps
- ⇒ mapDispatchToProps
- ⇒ Using Mapped Properties and Methods
- ⇒ Wrapping Components with Connect
- ⇒ Configure Store
- ⇒ Programming Advice - MultiTab Console

Flux

- ⇒ What is Flux Architecture?
- ⇒ What are the Flux Components available?
- ⇒ How Flux works?
- ⇒ Flux and React works together.

Using React Hooks

- ⇒ Functional Component Shortcomings
- ⇒ Hooks Overview
- ⇒ Hook Rules

Using React Hooks

- ⇒ Functional Component Shortcomings
- ⇒ Hooks Overview
- ⇒ Hook Rules
- ⇒ React Linter Example
- ⇒ Functional Component Props
- ⇒ The useState Hook
- ⇒ Functional Component using the useState hook
- ⇒ useState with Multiple Variables
- ⇒ useState can also be used with Objects
- ⇒ The useEffect Hook
- ⇒ useEffect Hook Example
- ⇒ Using useEffect Hook to Load Data
- ⇒ Restricting when useEffect is Called
- ⇒ The useContext Hook
- ⇒ Additional Hooks
- ⇒ The useReducer Hook
- ⇒ An Example Reducer Function
- ⇒ Calling and using useReducer
- ⇒ The useMemo Hook
- ⇒ useMemo Example
- ⇒ The useCallback Hook
- ⇒ useCallback Example
- ⇒ The useRef Hook
- ⇒ Using useRef to Hold Values
- ⇒ The useImperativeHandle Hook
- ⇒ useImperativeHandle Hook Example
- ⇒ The useLayoutEffect Hook

Creating Custom React Hooks

- ⇒ Custom Hooks
- ⇒ Custom Message Hook
- ⇒ Using the Custom Message Hook
- ⇒ A Custom useList Hook
- ⇒ Using the useList Custom Hook
- ⇒ The built-in useDebugValue Hook
- ⇒ Viewing the Effect of the useDebugValue Hook

Unit Testing React with React Testing Library

- ⇒ What are the necessary Tools required for Unit Testing?
- ⇒ React Unit Testing overview
- ⇒ Introduction to JEST.
- ⇒ Using Jest with Test Utils from React-DOM
- ⇒ Using Jest with the React Testing Library
- ⇒ Using Jest with Enzyme
- ⇒ How to Test React Component?
- ⇒ How to Test React Router?
- ⇒ Features
- ⇒ Snapshot Testing
- ⇒ Code Coverage
- ⇒ Interactive Mode
- ⇒ Projects created with create-react-app
- ⇒ Default App Component Test
- ⇒ Unit Tests
- ⇒ Anatomy of a Unit Test
- ⇒ Common Matchers
- ⇒ Combining Tests
- ⇒ Running Tests
- ⇒ Testing Promise based async code with 'done'
- ⇒ Setup and Teardown
- ⇒ react-testing-library
- ⇒ A Simple Component Test
- ⇒ A Simple Snapshot Test
- ⇒ Running and Updating Snapshot Tests
- ⇒ Building Component Tests
- ⇒ Calling Render
- ⇒ Render Properties
- ⇒ Simulating Events
- ⇒ Testing Results
- ⇒ Using Query Functions
- ⇒ Text Matching
- ⇒ Counter Component
- ⇒ counter-test.js

Integration with Other Libraries

- ⇒ Gulp & Browserify
- ⇒ React with jQuery
- ⇒ React & AJAX

Server Integration & Deployment

- ⇒ https
- ⇒ httpster
- ⇒ npm

Code Splitting

- ⇒ Code splitting & Suspense
- ⇒ Route Based Code Splitting
- ⇒ Lazy Loading

Isomorphic React

- ⇒ Server-Side Rendering
- ⇒ SSR with React - Setup & Server
- ⇒ SSR with React - The Toolchain

New Features of React 16,17 and 18 Webpack Primer AND Isomorphic React

Isomorphic React

- ⇒ Webpack and its use
- ⇒ Setting up and installing Webpack
- ⇒ Working with the configuration file of Webpack
- ⇒ Working with loaders
- ⇒ Quick word on lazy loading, code splitting, and tree shaking
- ⇒ Setting up a hot module replacement
- ⇒ Server-side rendering (SSR)
- ⇒ Working with renderToStatic Markup and renderToString methods

Fetch Data Using GraphQL

- ⇒ What is GraphQL?
- ⇒ Cons of Rest API
- ⇒ Pros of GraphQL
- ⇒ Frontend backend communication using GraphQL
- ⇒ Type system
- ⇒ GraphQL datatypes
- ⇒ Modifiers
- ⇒ Schemas
- ⇒ GraphQL tool
- ⇒ Express framework
- ⇒ NPM libraries to build server side of GraphQL
- ⇒ Build a GraphQL API
- ⇒ NPM libraries to build client side of GraphQL



Our Clients

