

REACT Foundations

Duration: 2 Days Course Code: GK840204

Overview:

Learn to build dynamic user interfaces with React in our comprehensive foundation course.

React Foundation offers a thorough introduction to React, a widely-used JavaScript library for building user interfaces. This course covers essential and advanced topics, including component-based architecture, state management with hooks, and styling techniques. Participants will learn to set up a React development environment, create reusable components, manage data flow, and handle side effects with the `useEffect` hook.

By the end of the course, you will learn to construct JSX elements, implement prop patterns, manage state, and debug React applications effectively. You will be equipped to build high-performance, scalable, and maintainable web applications.

This course includes hands-on practice through labs and exercises, reinforcing the concepts covered in the lectures - a balanced mix of theoretical knowledge and practical application, ensuring a comprehensive learning experience.

Target Audience:

- Front-End Developers
 - Web Developers
 - Software Engineers
 - Technical professionals
-

Objectives:

- | | |
|--|---|
| ■ Set up a React development environment and create an initial application structure. | ■ Implement container and presentational component patterns to organize React applications effectively. |
| ■ Construct JSX elements to build React component structures and handle dynamic content. | ■ Apply styling using CSS modules and styling organization patterns. |
| ■ Build reusable React components applying composition patterns. | ■ Use the <code>useEffect</code> hook to manage side effects and component lifecycle needs. |
| ■ Implement prop patterns to pass data and event handlers between components. | ■ Build form interfaces using both controlled and uncontrolled form patterns. |
| ■ Use state hooks to manage component data and handle user interactions. | ■ Use React Developer Tools to debug component issues and analyze application performance. |
| ■ Integrate data flow patterns to manage component communication and state sharing. | |
-

Prerequisites:

This is an intermediate level course for those looking to expand their skill set and master REACT. Learners should have proficiency in JavaScript, HTML, and CSS.

- GK840201 - Fundamentals of JavaScript
-

Content:

1 Introduction to React

What is React and its history

- Definition and differentiation
- Key benefits and use cases
- Component-based architecture for building UI

Virtual DOM

- Definition of the virtual DOM
- How React uses the virtual DOM (conceptually)
- Simple comparison with traditional web development
- Why this matters for modern web apps

React Development Setup

- Development environment setup
- Quick start with Create React App
- Development tools associated with React (React Developer Tools)

Project Structure

- Generating first React app with Create React App
- Folder structure: Component organization and file name conventions

2 JSX

Getting Started

- HTML entry point (root div)
- createRoot and render
- Basic React component structure with JSX
- Understanding the relationship between files

Fundamentals

- What is JSX and why do we use it?
- Rules of JSX (parent element, closing tags)

Writing JSX

- Basic syntax and attributes
- Adding JavaScript expressions with {}

Exploring JSX use cases

- Variables and props
- Simple conditional rendering

3 React Components

Component Basics

More about React Components

- Logic in a React component
- Conditions in a React component
- Handling user interactions in components

Component Composition

- Putting components in other components
- Reusing components
- Component organization

4 Props

Prop basics

- What are props?
- Passing data to components
- Receiving props
- Prop types and defaults
- Children prop
- Rendering a component's props
- Passing props from component to component
- Dynamic rendering with props

More complex prop patterns

- Pass an event handler as a prop
- Receive an event handler as a prop
- Event handler naming conventions

6 State and Events

Understanding State

- What is state?
- "stateful" components
- When to use state
- When not to use state
- Component re-rendering basics

useState Hook

- What is a hook?
- Initializing state
- State updates
- Multiple state values

Brief recap of content covered in Day 1, focusing on creating components and state

1 Component Patterns

Component Architecture

- Container vs Presentational Components
- When to separate components
- Rendering presentational components in

2 Styling in React

- Regular CSS with className
- CSS modules basics
- Styling organization
- Component-specific styling

3 Effects and Lifecycle

Working with Effects

- What are side effects in React?
- Why we need to manage effects
- Basic syntax and usage of useEffect hook
- When effects run
- Effect dependencies
- What is cleaning up an effect?
- How to write cleanup functions

Lifecycles

- Component phases (initial, updates, cleanup)
- Using effects for lifecycle needs

4 Forms

Form Fundamentals

- HTML forms vs React forms
- React's two approaches: Controlled vs Uncontrolled
- When to use each approach

Uncontrolled Forms

- useRef hook with forms
- Accessing input values
- Creating a basic uncontrolled form
- Handling submit events
- Getting form values
- Use cases for uncontrolled forms

Controlled Forms

- Basic form elements
- useState with single inputs
- Managing multiple form fields
- Handling different input types
- Handling form submission
- Preventing default behavior
- Gathering form data
- Basic validation
- Success/error states

5 Debugging React Applications

React Developer Tools

- Components tab

- What is a component?
- Creating your first component
- Single line and multiline JSX in a component
- Importing and exporting components
- Rendering a component

containers

React Data Flow

- One-way data flow concept
- Parent/Child communication patterns
- Sibling/Sibling communication
- Understanding prop drilling
- Lifting state up as a solution

- Props and state inspection
- Component tree navigation

Debugging React Applications

- Common React errors and solutions

Further Information:

For More information, or to book your course, please call us on Head Office 01189 123456 / Northern Office 0113 242 5931

info@globalknowledge.co.uk

www.globalknowledge.com/en-gb/

Global Knowledge, Mulberry Business Park, Fishponds Road, Wokingham Berkshire RG41 2GY UK