30-Days React.js Learning Plan 🚀

by GreatStack

To complete this **30-day React.js plan**, you should dedicate **3 to 4 hours per day** on average. Here's a suggested breakdown:

- **1.5 to 2 hours** – Studying concepts and reading resources.
- **1.5 to 2 hours** – Practicing exercises and building mini-projects.

For the **final week (Week 4)**, where you work on projects, consider spending **4 to 5 hours per day** to implement and refine your skills effectively.

Let me know if you need any help by sending DM on our Instagram:
https://www.instagram.com/greatstackdev/

🎯 React.js Basics: https://www.youtube.com/basics
🎯 React.js Portfolio Website: https://www.youtube.com/portfolio

⚡ 10+ React js Projects to practice: https://www.yt.com/react-projects

Week 1: React Fundamentals

Day 1: Introduction to React

- What is React? Why React?
- Understanding Single Page Applications (SPAs)
- Setting up the environment: Node.js, npm, Vite or Create React App
- JSX overview
- Exercise: Create a simple “Hello, React” component

Day 2: Components & Props

- Functional components
- Props and dynamic content
- Component reuse
- Exercise: Create a reusable Card component

Day 3: State & useState Hook

- What is state?
- Updating state
- Handling input elements
- Exercise: Create a counter app

Day 4: Event Handling

- Handling events in React
- Synthetic events
- Exercise: Create a form that shows live preview as you type

Day 5: Conditional Rendering

- if-else with JSX
- Ternary operator
- Short-circuit evaluation
- Exercise: Show/hide content based on a toggle

Day 6: Lists and Keys

- Rendering lists
- Using map()
- Unique keys and performance
- Exercise: Display a list of users from a local array

Day 7: Forms in React

- Controlled vs uncontrolled inputs
- Form submission
- Form validation basics
- Exercise: Build a contact form with basic validation

Week 2: Intermediate React

Day 8: useEffect Hook

- Side effects in React
- Fetching data, timers, etc.
- Dependency array
- Exercise: Fetch and display random users from API

Day 9: Conditional Effects & Cleanup

- Cleanup functions
- Memory leak prevention
- Exercise: Timer with cleanup on unmount

Day 10: Component Lifecycle (Hooks-based)

- useEffect as lifecycle replacement
- Mounting, updating, unmounting
- Exercise: Log lifecycle steps of a component

Day 11: Lifting State Up

- Sharing state between components
- Prop drilling
- Exercise: Build a temperature converter (Celsius to Fahrenheit)

Day 12: React Developer Tools & Debugging

- Installing and using React DevTools
- Debugging tips
- Exercise: Debug a broken component (intentional bugs)

Day 13: Styling in React

- Inline styles, CSS Modules, Tailwind (optional)
- Styled Components (optional)
- Exercise: Style a product card using CSS modules

Day 14: Project - Basic To-Do App

- Add, delete, filter tasks
- Store in localStorage

- Build it all from scratch!

Week 3: Advanced React Concepts

Day 15: Context API

- Avoiding prop drilling
- Creating global state
- Exercise: Create a theme switcher using context

Day 16: Advanced Hooks

- useRef (Accessing DOM Elements, Persisting Values)
- useReducer (Alternative to useState, State Reducers)
- useMemo (Performance Optimization, Memoization)
- useCallback (Memoizing Functions)
- useLayoutEffect (DOM Mutations Before Paint)
- useImperativeHandle (Customizing Ref Behavior)
- All hooks explained: https://youtu.be/AllReactHooks

Day 17: Custom Hooks

- Creating reusable logic
- Naming conventions
- Exercise: Create a custom useFetch hook

Day 18: React Router (v6)

- Setting up routing
- Dynamic routes
- useNavigate, useParams
- Exercise: Multi-page blog site with detail pages

Day 19: Forms with Libraries

- Introduction to react-hook-form or Formik
- Better validations
- Exercise: Build a sign-up form with validation

Day 20: Error Boundaries & Suspense

- Error handling in components
- Lazy loading components
- Exercise: Lazy load a heavy component with Suspense

Day 21: Project - Weather App (API + useEffect)

- Fetch weather by city name
- Display dynamic data and handle errors

Week 4: Projects & Real-World Practice

Day 22: Mini Project – Expense Tracker

- CRUD operations
- State management

Day 23: Mini Project – Quiz App

- Show questions
- Calculate score

Day 24: Mini Project – Movie Search App

- OMDB API
- Search & display

Day 25: Mini Project – Notes App

- Local storage
- Color-coded notes

Day 26: Mini Project – Image Gallery

- Grid view
- Modal
- Unsplash API

Day 27: Final Project – Portfolio Website

- Multi-page with React Router
- About, Projects, Contact sections

- Responsive design

Day 28: Optimization & Best Practices

- Code splitting
- Performance optimization
- Reusable components

Day 29: Testing React Apps

- Basics of Jest and React Testing Library
- Writing unit tests for components
- Resources: Testing Library Docs

Day 30: Review and Next Steps

- Revise tricky topics
- Explore Redux, Zustand or other state libs
- Start learning Next.js for SSR
- Resource: Roadmap.sh

🎯 Tips for Success:

- Practice every day - consistency is key.
- Push your mini-projects to GitHub.

Important Links to learn and practice:

🎯 React.js Basics: https://www.youtube.com/basics
🎯 React.js Portfolio Website: https://www.youtube.com/portfolio

⚡ 10+ React js Projects to practice: https://www.yt.com/react-projects

Conclusion:

This plan ensures progressive learning from basics to advanced React concepts, including hands-on projects to solidify knowledge. Keep practicing and build real-world projects to master react.js.