

React Native Development Guide for Beginners

Build Cross-Platform Mobile Apps with JavaScript

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>

Introduction

React Native allows developers to **build native Android and iOS apps using JavaScript**. This guide covers **setup, core concepts, navigation, state management, and mini-projects** for beginners.

1. Environment Setup

Option 1: Expo CLI

- Easier setup, ideal for beginners
- Handles builds and development environment
- Install: `npm install -g expo-cli`

Option 2: React Native CLI

- Full control over native code
 - Requires Android Studio and Xcode for iOS
 - Install: `npm install -g react-native-cli`
-

2. Project Structure

- **App.js** – Main entry point
 - **Components/** – Reusable UI components
 - **Screens/** – Each screen of the app
 - **Assets/** – Images, fonts, icons
 - **Navigation/** – App routing logic
-

3. Core Concepts

JSX

- Syntax for designing UI

```
<Text>Hello, React Native!</Text>
```

Components

- **Functional Components:** Preferred modern approach
- **Class Components:** Legacy, less used now

Props & State

- Props: Data passed to components
 - State: Internal data managed by component
-

4. Styling

- Uses JavaScript objects
- Example:

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: 'center',
    justifyContent: 'center'
  }
});
```

- Flexbox for layout management
-

5. Navigation

- **React Navigation** library
- Types:
 - Stack Navigator
 - Tab Navigator
 - Drawer Navigator

```
import { createStackNavigator } from '@react-navigation/stack';
```

6. API Integration

- Fetch data using `fetch` OR `axios`
- Example:

```
axios.get('https://api.example.com/data')
  .then(response => setData(response.data))
  .catch(error => console.log(error));
```

- Handle async with `async/await`
-

7. State Management

- **Context API** – Lightweight global state
 - **Redux** – Scalable, complex applications
 - **Recoil / Zustand** – Modern alternatives
-

8. Debugging & Testing

- React Native Debugger
 - Console logs
 - Unit Testing: Jest
 - UI Testing: Detox
-

9. Deployment

Android

- Build APK using React Native CLI or Expo
- Submit to Google Play Store

iOS

- Use Xcode or Expo build service
 - Submit to Apple App Store
-

10. Miscellaneous

10. Mini Project Ideas

1. **To-Do App** – Navigation, state management
 2. **Weather App** – API integration, display data
 3. **Expense Tracker** – Charts, local storage
 4. **Social Feed App** – Scrollable lists, API integration
-

Best Practices

- Keep components modular
 - Reuse styles and components
 - Use proper state management for large apps
 - Optimize app performance
-

Key Takeaways

- React Native allows **cross-platform app development** using JavaScript
 - Proper project structure and navigation management are essential
 - API integration, state management, and deployment are key skills
 - Mini-projects are the fastest way to learn and build portfolio
-

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>