



React Native Essential Cheatsheet (2026)

Complete Quick Revision Notes for Mobile App Development *Focused on practical concepts used in real-world apps*

1. What is React Native?

React Native is a framework developed by entity["company","Meta",""] that allows developers to build mobile applications using JavaScript and React.

- Single codebase for Android + iOS
 - Uses native components internally
 - Faster development compared to native apps
 - Large ecosystem and community support
-

2. Basic Project Creation

Expo (Beginner Friendly)

```
npx create-expo-app MyApp  
cd MyApp  
npm start
```

React Native CLI

```
npx react-native init MyApp  
cd MyApp  
npx react-native run-android
```

3. Folder Structure (Recommended)

```
src/  
├── components/  
├── screens/  
├── navigation/  
├── services/  
├── hooks/  
└── context/
```

```
|— redux/  
|— utils/  
|— assets/  
|— constants/
```

4. Core Components

```
import {  
  View,  
  Text,  
  Image,  
  Button,  
  TextInput,  
  ScrollView,  
  FlatList,  
  TouchableOpacity,  
  Pressable,  
  SafeAreaView  
} from 'react-native'
```

Most Important Components

- **View** → Container
- **Text** → Display text
- **Image** → Show images
- **TextInput** → User input
- **Button** → Basic button
- **TouchableOpacity** → Custom touchable button
- **ScrollView** → Scrollable content
- **FlatList** → Optimized list rendering
- **SafeAreaView** → Prevent overlap with notches
- **Pressable** → Modern touch handler

5. JSX Basics

```
const App = () => {  
  return (  
    <View>  
      <Text>Hello World</Text>  
    </View>  
  )  
}
```

Dynamic Values

```
const name = 'Zaid'  
  
<Text>{name}</Text>
```

Conditional Rendering

```
{isLoggedIn ? <Home /> : <Login />}
```

6. Styling

StyleSheet

```
import { StyleSheet } from 'react-native'  
  
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    justifyContent: 'center',  
    alignItems: 'center',  
    backgroundColor: '#fff'  
  }  
})
```

Important Styling Properties

- flex
- flexDirection
- justifyContent
- alignItems
- padding

- `margin`
- `borderRadius`
- `position`
- `backgroundColor`
- `fontSize`
- `fontWeight`

Flexbox Directions

```
flexDirection: 'row'  
flexDirection: 'column'
```

7. State Management

useState

```
const [count, setCount] = useState(0)
```

Updating State

```
setCount(count + 1)
```

useEffect

```
useEffect(() => {  
  fetchData()  
}, [])
```

Dependency Example

```
useEffect(() => {  
  console.log(count)  
}, [count])
```

8. Props

Parent Component

```
<MyButton title='Login' />
```

Child Component

```
const MyButton = ({ title }) => {  
  return <Text>{title}</Text>  
}
```

9. Lists Rendering

map()

```
{items.map(item => (  
  <Text key={item.id}>{item.name}</Text>  
))}
```

FlatList (Recommended)

```
<FlatList  
  data={data}  
  keyExtractor={({item}) => item.id}  
  renderItem={({item}) => (  
    <Text>{item.name}</Text>  
  )}  
</>
```

10. Navigation (Very Important)

Install

```
npm install @react-navigation/native
```

Stack Navigation

```
<Stack.Navigator>
  <Stack.Screen name='Home' component={HomeScreen} />
  <Stack.Screen name='Profile' component={ProfileScreen} />
</Stack.Navigator>
```

Navigate

```
navigation.navigate('Profile')
```

Go Back

```
navigation.goBack()
```

Common Navigation Types

- Stack Navigation
- Bottom Tabs
- Drawer Navigation

11. API Calls

fetch()

```
const getData = async () => {
  const response = await fetch('https://api.example.com/users')
  const data = await response.json()
  console.log(data)
}
```

axios

```
npm install axios
```

```
import axios from 'axios'

const response = await axios.get(url)
```

12. Forms Handling

Controlled Input

```
const [email, setEmail] = useState('')

<TextInput
  value={email}
  onChangeText={setEmail}
/>
```

Useful Libraries

- Formik
- React Hook Form
- Yup (validation)

13. Async Storage

Install

```
npm install @react-native-async-storage/async-storage
```

Save Data

```
await AsyncStorage.setItem('token', token)
```

Get Data

```
const token = await AsyncStorage.getItem('token')
```

14. Redux Toolkit (Important for Large Apps)

Install

```
npm install @reduxjs/toolkit react-redux
```

Store

```
const store = configureStore({  
  reducer: {  
    auth: authReducer  
  }  
})
```

useSelector

```
const user = useSelector(state => state.auth.user)
```

useDispatch

```
const dispatch = useDispatch()
```

15. Performance Optimization

Important Techniques

- Use `FlatList` instead of `ScrollView` for large lists
- Use `React.memo()` for reusable components
- Avoid unnecessary re-renders
- Use lazy loading when possible
- Optimize images

useMemo

```
const memoizedValue = useMemo(() => {  
  return expensiveCalculation(data)  
}, [data])
```

useCallback


```
const handleClick = useCallback(() => {  
  doSomething()  
}, [])
```

16. Commonly Used Libraries

UI Libraries

- NativeWind
- React Native Paper
- React Native Elements
- Tamagui

Utility Libraries

- Axios
- Lodash
- Moment.js / Day.js
- React Query / TanStack Query
- Zustand

17. Native Features

Common Features

- Camera
- Location
- Notifications
- File upload
- Permissions
- Biometrics
- Deep linking

Popular Packages

- react-native-image-picker
 - expo-camera
 - react-native-maps
 - react-native-permissions
 - expo-notifications
-

18. Debugging

Console

```
console.log(data)
```

Useful Tools

- React Native Debugger
 - Flipper
 - Chrome DevTools
-

19. Build & Deployment

Android APK

```
cd android  
./gradlew assembleRelease
```

Expo Build

```
eas build -p android
```

Publish Stores

- Google Play Store
 - Apple App Store
-

21. Most Important Hooks

```
useState()  
useEffect()  
useRef()  
useMemo()  
useCallback()  
useContext()  
useReducer()
```

22. Best Practices

- Keep components reusable
 - Avoid very large components
 - Use environment variables
 - Use constants for colors/fonts
 - Separate API logic from UI
 - Use TypeScript for large projects
 - Keep clean folder structure
 - Write meaningful component names
-

23. Frequently Used Commands

Start Metro

```
npx react-native start
```

Run Android

```
npx react-native run-android
```

Clear Cache

```
npx react-native start --reset-cache
```

Install Dependency

```
npm install package-name
```



Recommended Modern Stack (2026)

Beginner to Intermediate

- React Native + Expo
- NativeWind

- React Navigation
- Axios
- AsyncStorage

Professional Production Stack

- React Native CLI
 - TypeScript
 - Redux Toolkit / Zustand
 - TanStack Query
 - NativeWind / Tamagui
 - Firebase / Supabase
 - React Hook Form + Yup
-

developerzaid