

1. What is React Native?

React native is an open-source JavaScript framework designed by Facebook for native mobile applications development. It is based on a JavaScript library-React.

React Native saves your development time as it enables you to build real and native mobile apps within a single language – JavaScript for both Android and iOS platforms, such that code once, run that on any platform, and the React Native App is ready for use with native look and feel.

2. Why use React Native?

There is the following list of React Native features behind its use:

- Easy to use.
- Open-source framework
- Cross-platform compatibility
- Code Sharing
- Use a common language – JavaScript for cross-platform development.
- Faster Development
- Saves Time and efforts
- Gives Native look and feel

3. What are the advantages of React Native?

- React Native is based on “**Learn Once Write Everywhere**” approach to equip developers with a tool that only needs to be learned once, just in a single language and then can be reused on both iOS and Android mobile platform.
- React Native offers **cross-platform development** and a real experience to developers allowing them to build only one app with effectively 70% code sharing between different platforms.
- React Native helps in **faster development**. Building one app instead of two using a common language gives speedier app deployment, delivery, and quicker time-to-market.
- React Native exists with **essential components** for ending of native apps as the app development ends with native look and feel.
- React Native has a **large community** of developers for its security. The developers are always ready to fix bugs and issues that occur at any instant. They improve the performance of React Native from time to time with the best practices possible.
- React Native supports **Live and Hot Reloading**. Both are different features. Live Reloading is a tool that helps in compiling and reading the modified files. Hot Reloading is based on HMR (Hot Module Replacement) and helps to display the updated UI content.

4. List the essential components of React Native.

These are the following essential components of React Native:

- **View** is the basic built-in component used to build UI.
- **Text** component displays text in the app.
- **Image** component displays images in the app.
- **TextInput** is used to input text into the app via the keypad.
- **ScrollView** is a scrolling container used to host multiple views.

5. What are the cons of React Native?

- React Native is still a new development platform as compared to iOS and Android platforms. It is still immature, i.e., in an improvement stage and impacting negatively on apps.
- Sometimes, React Native built-in apps face performance problem if there is a requirement of advanced functionality. In that case, they don't perform well as compared to native apps.
- React Native has a steep learning curve for an average learner as it is not more comfortable in comparison to other cross-platform apps. It is because of existing JSX (JavaScript Syntax extension) in which HTML and JavaScript get combined and make learning challenging for average ones.
- React Native is based on JavaScript library which is fragile and creates a gap in the security robustness. As an expert's point of view, React Native is not secure and robust for building highly confidential data apps like Banking and Financial apps.

6. How many threads run in React Native?

There are two threads run in React Native:

1. JavaScript thread
2. Main UI thread

7. What are props in React Native?

props pronounced as the properties of React Native Components. props are the immutable parameters passed in Presentational Component to provide data.

8. What are React Native Apps?

React Native Apps are not web apps; they are the real and native mobile applications built-in a single language with the native components to run on mobile devices.

9. List the users of React Native?

There are thousands of React Native built-in apps. Here is the list of those apps:

- Facebook
- Facebook Ads Manager
- Instagram
- F8
- Airbnb
- Skype
- Tesla
- Bloomberg
- Gyroscope
- Myntra
- UberEats

10. For what purpose XHR module used in React Native?

XHR module implements XMLHttpRequest to post data on the server.

11. Can we use Native code alongside React Native?

Yes, we can use a native code alongside React Native for task completion, and several limitations can also overcome in previous versions like Titanium.

12. Are React Native built-in mobile apps like other Hybrid Apps which are slower in actual than Native ones?

React Native designed as a highly-optimized performance-based framework that builds real mobile apps with native components. Facebook is the best-suited example of high performance based app built-in React Native.

13. What is the difference between React and React Native?

- React is a JavaScript library while React native is a framework based on React.
- js used for building UI and web applications while React Native is used for creating cross-platform native mobile apps.

- Both uses synonymous tags such as <div> <h1> <h2> are the tags in React.js and <View> <Text> are the tags in React native.
- js uses DOM for path rendering of HTML tags while React Native uses AppRegistry for app registration.

14. What is the difference between React Native and Native Script?

- React Native uses only a single core development language- JavaScript *while* Native Script can use any of these languages- Angular, Vuejs, TypeScript, and JavaScript.
- React Native has faster development speed than Native Script. React Native exists with reusable components that developed at once can be used at different mobile platforms and accelerates mobile app development *while* Native Script exists with a less number of plugins among which some pass improper verification.
- React Native performs high as compared to Native Script. React Native is React based and uses virtual Dom for faster UI updation *while* Native Script uses slower Angulars, Vuejs, and TypeScript.
- Native Script exists with a box of various themes that shorten the gap between the different platform UIs *while* React Native doesn't live with predefined themes; you get default look and feel by the devices.

15. Can we combine native codes of Android and iOS in React Native?

Yes, we can do this as React Native fluently combines the components of both iOS and Android written in Swift/ Objective-C or Java.

16. What is the point of StyleSheet.create() in react native?

In React Native, StyleSheet.create() send the style only once through the bridge to avoid passing new style object and ensures that values are immutable and opaque.

17. Why React Native has very clear animations?

The animated API of React Native was designed as serializable so that users can send animations to native without going through the bridge on every frame. Once the animation starts, a JS thread can be blocked, and the animations will still run fluently. As the code converted into native views before rendering, the animations in React native will run smoothly, and the users get bright animations.

18. Differentiate between the React component and the React element.

React component is a class or function that accepts input and returns a React element while React element displays the look of React Component Instance to be created.

19. Why React Native use Redux?

Redux is a standalone state management library that React Native use to simplify data flow within an app.

20. Which node_modules will run in React Native? How to test for this?

In React Native, node_modules as any pure JavaScript library that does not rely on Node.js runtime modules, and does not rely on web-specific concepts like **window.location.pathname** will run fine. But be conscious as there exists no way to test for this with Babel- it doesn't scan these libraries for offending dependencies. A module that uses **window.location.pathname** may fail at runtime in a different unexpected place.

21. What is Virtual DOM and how it works in React Native?

Virtual Dom is an in-memory tree representation of the real DOM with lightweight elements. It provides a declarative way of DOM representation for an app and allows to update UI whenever the state changes.

Working

Virtual DOM lists elements and their attributes and content. It renders the entire UI whenever any underlying data changes in React Native. Due to which React differentiates it with the previous DOM and a real DOM gets updated.

22. What is InteractionManager and what is its importance?

InteractionManager is a native module in React native that defers the execution of a function until an “interaction” finished.

Importance React Native has JavaScript UI thread as the only thread for making UI updates that can be overloaded and drop frames. In that case, InteractionManager helps by ensuring that the function is only executed after the animations occurred.

23. What is the point of the relationship between React Native and React?

React is a JavaScript library. React Native is a framework based on React that use JavaScript for building mobile applications. It uses React to construct its application UI and to define application business logic. React updates a real DOM by tree diffing and allows React Native to work.

24. What are the similarities between React Native and React?

React.js and React Native share

- same lifecycle methods like componentDidMount
- same state and prop variables
- same component architecture
- similar management libraries like Redux

25. Describe HOC.

HOC (High Order Components) in React is the functional programming methodology for reusing component logic.

- takes a component as an argument and
- returns a new component

It is a pattern evolved from React's compositional nature to decompose the logic into simpler and smaller reusable functions.

26. Define Native apps.

Native app is a software program for a specific mobile device that is developed on a particular platform in a specific programming language like Objective-C/Swift for iOS and Java for Android. It can use device-specific hardware and software as built on a particular device and its OS. It uses the latest technology such as GPS and provides optimized performance.

27. What are Hybrid Apps?

Hybrid applications are the web applications developed through HTML, CSS, JavaScript web standards and wrapped in a native container using a mobile WebView object. These apps are easier to maintain.

28. What are refs in React? When to use Refs?

Refs are escape hatch that provides a direct way to access DOM nodes or React elements created in the render method.

Refs get in use when

- To manage focus, text selection, or media playback
- To trigger imperative animations
- To integrate with third-party DOM libraries

29. What does a react native packager do?

A react native packager does a few things:

- It combines all JavaScript code into a single file
- It translates any JavaScript code that your device don't understand (e.g., JSX or some of the newer JS syntax)
- It converts assets like PNG files into objects that can be displayed by an **Image**

30. What is NPM in React Native?

npm installs the command line interface in React Native.

npm install -g react-native-cli

31. What are “props” and “state”?

“**Props**” and “**state**” are both plain JavaScript objects used to control data inside the components.

props

1. short for “properties.”
2. Immutable parameters -> *Unchangeable* inside the component.
 - Set by their parent and fixed for the whole lifetime of a component.
1. Get passed to the Presentational component.

state

1. Mutable parameters -> *Changeable* inside the component.
2. Get handled within the container component.
3. Can't be accessed and modified outside the component.

32. What is Style?

The style prop is a plain JavaScript object use to making style of React Native application. There are two ways to style your element in React Native application.

1. **style** property: adds styles inline.
1. external **Stylesheet**: enables us to write concise code.

33. How To Handling Multiple Platforms?

React Native smoothly handles multiple platforms. The large numbers of the React Native APIs are cross-platform so that one React Native component will work seamlessly on both iOS and Android. It provides the ways using which you can easily organize your code and separate it by platform:

- *Platform module* to detect the platform in which the app is running and
- *platform-specific file extensions* to load the relevant platform file.

34. When would you use a class component over a functional component?

We use class component if our component has state or a lifecycle method(s). Otherwise, we use a Functional component.

35. How React Native handle different screen size?

React Native offers many ways to handle different screen sizes:

1. **Flexbox** Flexbox is designed to provide a consistent layout on different screen sizes. It offers three main properties:
 - `flexDirection`
 - `justifyContent`
 - `alignItems`
2. **Pixel Ratio** exists in the official documentation with the definition such that we can get access to the device pixel density by using `PixelRatio` class. We will get a higher resolution image if we are on a high pixel density device. An ethical principle is that multiply the size of the image we display by the pixel ratio.
3. **Dimensions** easily handle different screen sizes and style the page precisely. It needs to write the code only once for working on any device.

36. Are all React components usable in React Native?

Web React components use DOM elements to display (ex. `div`, `h1`, `table`, etc.), but React Native does not support these. We will need to find libraries/components made specifically for React Native.

But today React is focusing on components that can be shared between the web version of React and React Native. This concept has been formalized since React v0.14.

37. What is the challenge with React Native?

Working across separate Android and iOS codebases is challenging.

38. Does React Native use the same code base for Android and iOS?

Yes, React Native uses the same code base for Android and IOS. React takes cares of the native components translations.

For example A React Native ScrollView uses native ScrollView on Android and UIScrollView on iOS.

39. Thus React Native is a native Mobile App?

Yes, React Native compiles a native mobile app using native app components. React Native builds a real mobile app that is indistinguishable from an app built using Objective-C or Java.

40. What is Gesture Responder System?

The gesture responder system manages the lifecycle of gestures in the app.

Users interact with mobile apps mainly through touch. They can use a combination of gestures, such as tapping on a button, zooming on a map, sliding on a widget or scrolling a list. The touch responder system is required to allow components to negotiate these touch interactions without any additional knowledge of their parent or child components.

41. How can React Native integrate more features on the existing app?

React Native is great to start a new application from scratch. However, React Native works well to add new features to an existing native app. It needs some steps to add new React Native based features, screen, views, etc. The specific steps are different for different platform you're targeting.

- Set up directory structure.
- Install JavaScript dependencies.
- Configuring permissions.
- Code integration.
- Test your integration.

42. What is the storage system in React Native?

React Native uses **AsyncStorage** class to store data in key-value pair which is global to all app. AsyncStorage is a JavaScript code which is a simple, unencrypted, asynchronous and persistent. React Native also uses separate files for iOS and RocksDB or SQLite for Android. Using AsyncStorage class, you must have a data backup, and synchronization classes as data saved on the device is not permanent and not encrypted.