### **1. How Different is React-native from ReactJS ?**

* ****Usage Scope****  
  ReactJs - [React](https://www.interviewbit.com/react-interview-questions/) is a  JavaScript library for building Responsive User Interfaces for Building Web Application.  
  React Native - It is a framework for creating mobile applications with a native feel.
* ****Syntax****  
  Both React and React Native uses JSX (JavaScript XML)  syntax but React uses html tags like <div> <h1> <p> etc while React Native uses <view> <text> etc.
* ****Animation And Gestures****  
  React uses CSS animations on a major scale to achieve animations for a web page while  The recommended way to animate a component is to use the Animated API provided by React-Native.
* ****Routing Mechanism****  
  React uses a react-router for routing and does not have any inbuilt routing capabilities but React Native has a built-in Navigator library for navigating mobile applications.

| **REACT JS** | **REACT NATIVE** |
| --- | --- |
| It is used for developing web applications. | It is used for developing mobile applications. |
| It uses React-router for navigating web pages. | It has a built-in navigator library for navigating mobile applications. |
| It uses HTML tags. | It does not use HTML tags. |
| It provides high security. | It provides low security in comparison to ReactJS. |
| In this, the virtual DOM renders the browser code. | In this, Native uses its API to render code for mobile applications. |

### **2. What is Flexbox and describe any elaborate on its most used properties?**

It is a layout model that allows elements to align and distribute space within a container. With Flexbox when Using flexible widths and heights, all the inside the main container can be aligned to fill a space or distribute space between elements, which makes it a great tool to use for responsive design systems.

| **Property** | **Values** | **Description** |
| --- | --- | --- |
| flexDirection | ‘column’,'row' | Used to specify if elements will be aligned vertically or horizontally |
| justifyContent | ‘center’,'flex-start','flex-end','space-around','space-between' | Used to determine how should elements be distributed inside the container |
| alignItems | ‘center’,'flex-start','flex-end','stretched' | Used to determine how should elements be distributed inside the container along the secondary axis (opposite of flexDirection) |

### **3. Describe advantages of using React Native?**

There are multiple advantage of using React Native like,

* ****Large Community****  
  React Native is an Open Source Framework, it is completely community driven so any challenges can be resolved by getting online help from other developers.
* ****Reusability****  
  Code can be written once and can be used for both IOS and ANDROID, which helps in maintaining and as well debugging large complex applications as no separate teams are needed for supporting both the platforms, this also reduces the cost to a major extent.
* ****Live and Hot Reloading****  
  Live reloading reloads or refreshes the entire app when a file changes. For example, if you were four links deep into your navigation and saved a change, live reloading would restart the app and load the app back to the initial route.  
  Hot reloading only refreshes the files that were changed without losing the state of the app. For example, if you were four links deep into your navigation and saved a change to some styling, the state would not change, but the new styles would appear on the page without having to navigate back to the page you are on because you would still be on the same page.
* ****Additional Third-Party Plugins****  
  If the existing modules do not meet the business requirement in React Native we can also use Third Party plugins which may help to speed up the development process.

**4. What are threads in General ? and explain Different Threads in ReactNative with Use of Each ?**

The single sequential flow of control within a program can be controlled by a thread.  
  
****React Native right now uses 3 threads:****

* ****MAIN/UI  Thread**** — This is the main application thread on which your Android/iOS app is running. The UI of the application can be changed by the Main thread and it has access to it .
* ****Shadow Thread**** — layout created using React library in React Native can be calculated by this and it is a background thread.
* ****JavaScript Thread**** — The main Javascript code is executed by this thread.

### **5. Are default props available in React Native and if yes for what are they used and how are they used ?**

Yes, default props available in React Native as they are for React,  If for an instance we do not pass props value, the component will use the default props value.

import React, {Component} from 'react';

import {View, Text} from 'react-native';

class DefaultPropComponent extends Component {

render() {

return (

<View>

<Text>

{this.props.name}

</Text>

</View>

)

}

}

Demo.defaultProps = {

name: 'BOB'

}

export default DefaultPropComponent;

### **6. How is user Input Handled in React Native ?**

TextInput is a Core Component that allows the user to enter text. It has an onChangeText prop that takes a function to be called every time the text changes, and an onSubmitEditing prop that takes a function to be called when the text is submitted.

import React, { useState } from 'react';

import { Text, TextInput, View } from 'react-native';

const PizzaTranslator = () => {

const [text, setText] = useState('');

return (

<View style={{padding: 10}}>

<TextInput

style={{height: 40}}

placeholder="Type here to translate!"

onChangeText={text => setText(text)}

defaultValue={text}

/>

<Text style={{padding: 10, fontSize: 42}}>

{text.split(' ').map((word) => word && '

### **7. What is State and how is it used in React Native?**

It is used to control the components. The variable data can be stored in the state. It is mutable means a state can change the value at any time.

import React, {Component} from 'react';

import { Text, View } from 'react-native';

export default class App extends Component {

state = {

myState: 'State of Text Component'

}

updateState = () => this.setState({myState: 'The state is updated'})

render() {

return (

<View>

<Text onPress={this.updateState}> {this.state.myState} </Text>

</View>

); } }

Here we create a Text component with state data. The content of the Text component will be updated whenever we click on it. The state is updated by event onPress .

### **8. What is Redux in React Native and give important components of Redux used in React Native app ?**

Redux is a predictable state container for JavaScript apps. It helps write applications that run in different environments. This means the entire data flow of the app is handled within a single container while persisting previous state.

Actions: are payloads of information that send data from your application to your store. They are the only source of information for the store. This means if any state change is necessary the change required will be dispatched through the actions.  
  
Reducers: “Actions describe the fact that something happened, but don’t specify how the application’s state changes in response. This is the job of reducers.” when an action is dispatched for state change its the reducers duty to make the necessary changes to the state and return the new state of the application.  
  
Store: a store can be created with help of reducers which holds the entire state of the application. The recommended way is to use a single store for the entire application rather than having multiple stores which will violate the use of redux which only has a single store.  
  
Components: this is where the UI of the application is kept.

### **9. Describe Timers in React Native Application ?**

Timers are an important and integral part of any application and React Native implements the browser timers.  
  
****Timers****

* ****setTimeout, clearTimeout****

There may be business requirements to execute a certain piece of code after waiting for some time duration or after a delay setTimeout can be used in such cases, clearTimeout is simply used to clear the timer that is started.

setTimeout(() => {

yourFunction();

}, 3000);

* ****setInterval, clearInterval****

setInterval is a method that calls a function or runs some code after specific intervals of time, as specified through the second parameter.

setInterval(() => {

console.log('Interval triggered');

}, 1000);

A function or block of code that is bound to an interval executes until it is stopped. To stop an interval, we can use the clearInterval() method.

* ****setImmediate, clearImmediate****

Calling the function or execution as soon as possible.

var immediateID = setImmediate(function);

// The below code displays the alert dialog immediately.

var immediateId = setImmediate(

() => { alert('Immediate Alert');

}

clearImmediate  is used for Canceling the immediate actions that were set by setImmediate().

* ****requestAnimationFrame, cancelAnimationFrame****

It is the standard way to perform animations.

Calling a function to update an animation before the next animation frame.

var requestID = requestAnimationFrame(function);

// The following code performs the animation.

var requestId = requestAnimationFrame(

() => { // animate something}

)

cancelAnimationFrame is used for Canceling the function that was set by requestAnimationFrame().

### **10. How to debug React Native Applications and Name the Tools used for it ?**

In the React Native world, debugging may be done in different ways and with different tools, since React Native is composed of different environments (iOS and Android), which means there’s an assortment of problems and a variety of tools needed for debugging.

* ****The Developer Menu:****  
    
  Reload: reloads the app  
  Debug JS Remotely: opens a channel to a JavaScript debugger  
  Enable Live Reload: makes the app reload automatically on clicking Save  
  Enable Hot Reloading: watches for changes accrued in a changed file  
  Toggle Inspector: toggles an inspector interface, which allows us to inspect any UI element on the screen and its properties, and presents an interface that has other tabs like networking, which shows us the HTTP calls, and a tab for performance.
* ****Chrome’s DevTools:****  
    
  Chrome is possibly the first tool to think of for debugging React Native. It’s common to use Chrome’s DevTools to debug web apps, but we can also use them to debug React Native since it’s powered by JavaScript.To use Chrome’s DevTools with React Native, first make sure to connect to the same Wi-Fi, then press command + R if you’re using macOS, or Ctrl + M on Windows/Linux. In the developer menu, choose Debug Js Remotely. This will open the default JS debugger.
* ****React Developer Tools****  
  For Debugging React Native using React’s Developer Tools, you need to use the desktop app. In can installed it globally or locally in your project by just running the following command:  
  yarn add react-devtools  
  Or npm:  
  npm install react-devtools --save  
    
  React’s Developer Tools may be the best tool for debugging React Native for these two reasons:  
  It allows for debugging React components.  
  There is also a possibility to debug styles in React Native. There is also a new version that comes with this feature that also works with the inspector in the developer menu. Previously, it was a problem to write styles and have to wait for the app to reload to see the changes. Now we can debug and implement style properties and see the effect of the change instantly without reloading the app.
* ****React Native Debugger****  
  While using Redux in your React Native app, React Native Debugger is probably the right debugger for you. This is a standalone desktop app that works on macOS, Windows, and Linux. It even integrates both Redux’s DevTools and React’s Developer Tools in one app so you don’t have to work with two separate apps for debugging.

React Native CLI

You can use the React Native CLI to do some debugging as well. It can also be used for showing the logs of the app. Hitting react-native log-android will show you the logs of db logcat on Android, and to view the logs in iOS you can run react-native log-ios, and with console.log you can dispatch logs to the terminal:

console.log("some error

### **11. What is Props Drilling and how can we avoid it ?**

Props Drilling (Threading) is a concept that refers to the process you pass the data from the parent component to the exact child Component BUT in between, other components owning the props just to pass it down the chain.

****Steps to avoid it****

1. React Context API.  
2. Composition  
3. Render props  
4. HOC  
5. Redux or MobX

### **12. Describing Networking in React Native and how to make AJAX network calls in React Native?**

React Native provides the Fetch API for networking needs.   
To fetch content from an arbitrary URL, we can pass the URL to fetch:

fetch('https://mywebsite.com/endpoint/', {

method: 'POST',

headers: {

Accept: 'application/json',

'Content-Type': 'application/json'

},

body: JSON.stringify({

firstParam: 'yourValue',

secondParam: 'yourOtherValue'

})

});

Networking is an inherently asynchronous operation. Fetch methods will return a Promise that makes it straightforward to write code that works in an asynchronous manner:

const getMoviesFromApi = () => {

return fetch('https://reactnative.dev/movies.json')

.then((response) => response.json())

.then((json) => {

return json.movies;

})

.catch((error) => {

console.error(error);

});

};

The XMLHttpRequest API is built in to React Native  Since frisbee and Axios use XMLHttpRequest we can even use these libraries.

var request = new XMLHttpRequest();

request.onreadystatechange = (e) => {

if (request.readyState !== 4) {

return;

}

if (request.status === 200) {

console.log('success', request.responseText);

} else {

console.warn('error');

}

};

request.open('GET', 'https://mywebsite.com/endpoint/');

request.send();

### **13. List down Key Points to integrate React Native in an existing Android mobile application**

Primary points to note to integrating React Native components into your Android application are to:

* Set up React Native dependencies and directory structure.
* Develop your React Native components in JavaScript.
* Add a ReactRootView to your Android app. This view will serve as the container for your React Native component.
* Start the React Native server and run your native application.
* Lastly, we need to Verify that the React Native aspect of your application works as expected.

## **React Native Intermediate Interview Questions**

### **14. How is the entire React Native code processed to show the final output on a mobile screen**

* At the first start of the app, the main thread starts execution and starts loading JS bundles.
* When JavaScript code has been loaded successfully, the main thread sends it to another JS thread because when JS does some heavy calculations stuff the thread for a while, the UI thread will not suffer at all times.
* When React starts rendering, Reconciler starts “diffing”, and when it generates a new virtual DOM(layout) it sends changes to another thread(Shadow thread).
* Shadow thread calculates layout and then sends layout parameters/objects to the main(UI) thread. ( Here you may wonder why we call it “shadow”? It’s because it generates shadow nodes )
* Since only the main thread is able to render something on the screen, the shadow thread should send the generated layout to the main thread, and only then UI renders.

### **15. What is a bridge and why is it used in React Native ? Explain for both android and IOS ?**

Bridge in ReactNative is a layer or simply a connection that is responsible for gluing   
together Native and JavaScript environments.

****Consider Below diagram:****

* The layer which is closest to the device on which the application runs is the Native Layer.

The bridge is basically a transport layer which acts as a connection between Javascript and Native modules, it does the work of transporting asynchronous serialized batched response messages from JavaScript to Native modules.  
  
Now for an example, there is some state change that happens, because of which React Native will batch Update UI and send it to the Bridge. The bridge will pass this Serialized batched response to the Native layer, which will process all commands that it can distinguish from a serialized batched response and will update the User Interface accordingly.  
  
****IOS Pla****

### **16. Name core Components in React Native and the analogy of those components when compared with the web .**

The core components used in React Native are <View> , <Text> , <Image> , <ScrollView> , <TextInput>  
  
And analogy when compared Web can be explained by below diagram:

| **REACT NATIVE UI COMPONENT** | **ANDROID VIEW** | **IOS VIEW** | **WEB ANALOG** | **DESCRIPTION** |
| --- | --- | --- | --- | --- |
| <View> | <ViewGroup> | <UIView> | A non-scrolling <div> | A container that supports layout with flexbox style, some touch handling, and accessibility controls. |
| <Text> | <TextView> | <UITextView> | <p> | Displays, styles, and nests strings of text and even handles touch events. |
| <Image> | <ImageView> | <UIImageView> | <img> | Displays different types of images |
| <ScrollView> | <ScrollView> | <UIScrollView> | <div> | A generic scrolling container that can contain multiple components and views. |
| <TextInput> | <EditText> | <UITextField> | <input type="text"> | Allows the user to enter text |

### **17. What is ListView and describe its use in React Native ?**

React Native ListView is a view component that contains the list of items and displays it in a vertically scrollable list.

export default class MyListComponent extends Component {

constructor() {

super();

const ds = new ListView.DataSource({rowHasChanged: (r1, r2) => r1 !== r2});

this.state = {

dataSource: ds.cloneWithRows(['Android','iOS', 'Java','Php', 'Hadoop', 'Sap', 'Python','Ajax', 'C++']),

};

}

render() {

return (

<ListView

dataSource={this.state.dataSource}

renderRow={

(rowData) =>

<Text style={{fontSize: 30}}>{rowData}</Text>} />

); }

}

### **18. How can you write different code for IOS and Android in the same code base ? Is there any module available for this ?**

The platform module detects the platform in which the app is running.

import { Platform, Stylesheet } from 'react-native';

const styles = Stylesheet.create({

height: Platform.OS === 'IOS' ? 200 : 400

})

Additionally Platform.select method available that takes an object containing Platform.OS as keys and returns the value for the platform you are currently on.

import { Platform, StyleSheet } from 'react-native';

const styles = StyleSheet.create({

container: {

flex: 1,

...Platform.select({

ios: {

backgroundColor: 'red',

},

android: {

backgroundColor: 'green',

},

default: {

// other platforms, web for example

backgroundColor: 'blue',

}, }),

},

});

### **19. What are Touchable components in react Native and which one to use when ?**

Tapping gestures can be captured by Touchable components and can display feedback when a gesture is recognized.

Depending on what kind of feedback you want to provide we choose Touchable Components.

Generally, we use TouchableHighlight anywhere you would use a button or link on the web. The background of the view will be darkened when the user presses down on the button.

We can use TouchableNativeFeedback on Android to display ink surface reaction ripples that respond to the user's touch.

TouchableOpacity can be used to provide feedback by reducing the opacity of the button, allowing the background to be seen through while the user is pressing down.

If we need to handle a tap gesture but you don't want any feedback to be displayed, use TouchableWithoutFeedback.

import React, { Component } from 'react';

import { Platform, StyleSheet, Text, TouchableHighlight, TouchableOpacity, TouchableNativeFeedback, TouchableWithoutFeedback, View } from 'react-native';

export default class Touchables extends Component {

\_onPressButton() {

alert('You tapped the button!') }

\_onLongPressButton() {

alert('You long-pressed the button!')

}

render() {

return (

<View style={styles.container}>

<TouchableHighlight onPress={this.\_onPressButton} underlayColor="white">

<View style={styles.button}>

<Text style={styles.buttonText}>TouchableHighlight</Text>

</View>

</TouchableHighlight>

);}

}

### **20. Explain FlatList components, what are its key features along with a code sample ?**

The FlatList component displays similarly structured data in a scrollable list. It works well for large lists of data where the number of list items might change over time.

****Key Feature:****

The FlatList shows only those rendered elements which are currently displaying on the screen, not all the elements of the list at once.

import React, { Component } from 'react';

import { AppRegistry, FlatList,

StyleSheet, Text, View,Alert } from 'react-native';

export default class FlatListBasics extends Component {

renderSeparator = () => {

return (

<View

style={{

height: 1,

width: "100%",

backgroundColor: "#000",

}}

/>

);

};

//handling onPress action

getListViewItem = (item) => {

Alert.alert(item.key);

}

render() {

return (

<View style={styles.container}>

<FlatList

data={[

{key: 'Android'},{key: 'iOS'}, {key: 'Java'},{key: 'Swift'},

{key: 'Php'},{key: 'Hadoop'},{key: 'Sap'},

]}

renderItem={({item}) =>

<Text style={styles.item}

onPress={this.getListViewItem.bind(this, item)}>{item.key}</Text>}

ItemSeparatorComponent={this.renderSeparator}

/>

</View>

);

}

}

AppRegistry.registerComponent('AwesomeProject', () => FlatListBasics);

### **21. How To Use Routing with React Navigation in React Native ?**

One of the popular libraries for routing and navigation in a React Native application is React Navigation.

This library helps solve the problem of navigating between multiple screens and sharing data between them.

import \* as React from 'react';

import { NavigationContainer } from '@react-navigation/native';

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

const Stack = createStackNavigator();

const MyStack = () => {

return (

<NavigationContainer>

<Stack.Navigator>

<Stack.Screen

name="Home"

component={HomeScreen}

options={{ title: 'Welcome' }}

/>

<Stack.Screen name="Profile" component={ProfileScreen} />

</Stack.Navigator>

</NavigationContainer>

);

};

### **22. What are the Different Ways to style React Native Application ?**

React Native give us two powerful ways by default to style our application :

****1 ) Style props****

You can add styling to your component using style props. You simply add style props to your element and it accepts an object of properties.

import React, {Component} from 'react';

import {Platform, StyleSheet, Text, View} from 'react-native';

export default class App extends Component<Props> {

render() {

return (

<View style={{flex:1,justifyContent:"center",backgroundColor:"#fff", alignItems:"center"}}>

<View style={{width:200,height:150,backgroundColor:"red",padding:10}}>

<Text style={{fontSize:20, color:"#666"}}>Styled with style props</Text>

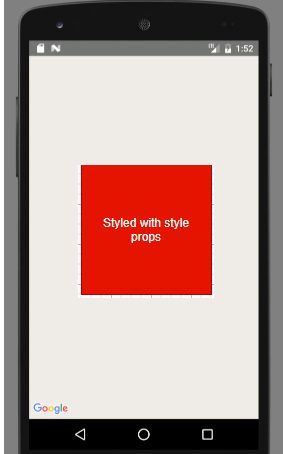
</View>

</View>

);

}

}



****2 )  Using StyleSheet****

For an extremely  large codebase or you want to set many properties to your elements, writing our styling rules directly inside style props will make our code more complex that’s why React Native give us another way that let us write a concise code using the StyleSheet method:

import { StyleSheet} from 'react-native';

const styles = StyleSheet.create({

container: {

flex:1,

justifyContent:"center",

backgroundColor:"#fff",

alignItems:"stretch"

},

title: {

fontSize:20,

color:"#fff"

},

item1: {

backgroundColor:"orange",

flex:1

},

item2: {

backgroundColor:"purple",

flex:1

},

item3: {

backgroundColor:"yellow",

flex:1

},

});

 We then  pass the styles object to our component via the style props:

<View style={styles.container}>

<View style={styles.item1}>

<Text style={{fontSize:20, color:"#fff"}}>Item number 1</Text>

</View>

<View style={styles.item2}>

<Text style={{fontSize:20, color:"#fff"}}>Item number 1</Text>

</View>

<View style={styles.item3}>

<Text style={{fontSize:20, color:"#fff"}}>Item number 1</Text>

</View>

<View style={styles.item4}>

<Text style={{fontSize:20, color:"#fff"}}>Item number 1</Text>

</View>

</View>

****3 ) styled-components in React Native****

We can also use styled-components with React native so you can write your styles in React Native as you write normal CSS. It is very easy to include it in your project and it doesn’t need any linking just run this following command inside the root directory of your app to install it:

yarn add styled-components

import React, {Component} from 'react';

import { StyleSheet,Text, View} from 'react-native';

import styled from 'styled-components'

const Container=styled.View`

flex:1;

padding:50px 0;

justify-content:center;

background-color:#f4f4f4;

align-items:center

`

const Title=styled.Text`

font-size:20px;

text-align:center;

color:red;

`

const Item=styled.View`

flex:1;

border:1px solid #ccc;

margin:2px 0;

border-radius:10px;

box-shadow:0 0 10px #ccc;

background-color:#fff;

width:80%;

padding:10px;

`

export default class App extends Component {

render() {

return (

<Container>

<Item >

<Title >Item number 1</Title>

</Item>

<Item >

<Title >Item number 2</Title>

</Item>

<Item >

<Title >Item number 3</Title>

</Item>

<Item >

<Title >Item number 4</Title>

</Item>

</Container>

);

}

### **23. Explain Async Storage in React Native and also define when to use it and when to not?**

* Async Storage is the React Native equivalent of Local Storage from the web.
* Async Storage is a community-maintained module for React Native that provides an asynchronous, unencrypted, key-value store. Async Storage is not shared between apps: every app has its own sandbox environment and has no access to data from other apps.

| **DO USE ASYNC STORAGE WHEN..** | **DON'T USE ASYNC STORAGE FOR..** |
| --- | --- |
| Persisting non-sensitive data across app runs | Token storage |
| Persisting Redux state | Secrets |
| Persisting GraphQL state |  |
| Storing global app-wide variables |  |

## **React Native Advanced Interview Questions**

### **24. What’s the real cause behind performance issues in React Native ?**

The real cause behind React Native performance issues is that each thread (i.e Native and JS thread) is blazingly fast. The performance bottleneck in React Native app occurs when you’re passing the components from one thread to another unnecessarily or more than required. A major thumb rule to avoid any kind of performance-related issue in React Native is to keep the passes over the bridge to a minimum.

* Native thread built for running Java/ Kotlin, Swift/ Objective C
* Javascript thread is the primary thread that runs everything from javascript-based animations to other UI components
* The bridge as the name suggests acts as an  intermediate communication point for the native and JS thread

### **25. List down some of the steps to optimize the application.**

* Use Proguard to minimize the application size.(It does this by stripping parts of the React Native Java bytecode (and its dependencies) that your app is not using)
* Create reduced-sized APK files for specific CPU architectures. When you do that, your app users will automatically get the relevant APK file for their specific phone’s architecture. This eliminates the need to keep JSCore binaries that support multiple architectures and consequently reduces the app size.
* Compress images and other graphic elements. Another option to reduce image size is using file types like APNG in place of PNG files.
* Don’t store raw JSON data,  eIther we need to Compress it or convert it into static object IDs.
* Optimize native libraries.
* Optimize the number of state operations and remember to use pure and memoized components when needed
* Use Global State wisely for example worst-case scenario is when state change of single control like TextInput or CheckBox propagates render of the whole application. Use libraries like Redux or Overmind.js to handle your state management in a more optimized way.
* Use key attribute on list items, it helps React Native to pick which list to update when rendering a long list of data
* Use VirtualizedList, FlatList, and SectionList for large data sets.
* Clear all the active timers which may lead to heavy memory leakage issues.

### **26. Describe Memory leak Issue in React Native , how can it be detected and resolved ?**

In JavaScript memory is managed automatically by Garbage Collector (GC). In short, Garbage Collector is a background process that periodically traverses the graph of allocated objects and their references. If it happens to encounter a part of the graph that is not being referenced directly or indirectly from root objects (e.g., variables on the stack or a global object like window or navigator) that whole part can be deallocated from the memory.  
  
In React Native world each JS module scope is attached to a root object. Many modules, including React Native core ones, declare variables that are kept in the main scope (e.g., when you define an object outside of a class or function in your JS module). Such variables may retain other objects and hence prevent them from being garbage collected.  
  
****Some Causes of Memory Leak:****

* Unreleased timers/listeners added in componentDidMount
* Closure scope leaks

****Detecting memory leaks for IOS:****  
  
In Xcode,  
  
Go to XCode → Product → Profile (⌘ + i)  
  
After that shows you all templates choose leaks.  
  
****Detecting memory leaks for Android :****  
  
Run React Native app normally (react-native run-android)  
Run Android Studio  
  
On the menu,  
click Tools → Android → Enable ADB Integration  
Click Tools → Android → Android Device Monitor  
When Android Device Monitor shows up, click Monitor → Preferences  
  
There is also one more way in Android  
Perf Monitor (Performance Monitor) is a good choice to use for android leak monitoring.

Import PerfMonitor from 'react-native/Libraries/Performance/RCTRenderingPerf';

PerfMonitor.toggle();

PerfMonitor.start();

setTimeout(() => {

PerfMonitor.stop();

}, 20000);

}, 5000);

### **27. Is there any out of the box way storing sensitive data in React ? If yes which and if not how can this be achieved ?**

React Native does not come bundled with any way of storing sensitive data. However, there are pre-existing solutions for Android and iOS platforms.

****iOS - Keychain Services****  
Keychain Services allows you to securely store small chunks of sensitive info for the user. This is an ideal place to store certificates, tokens, passwords, and any other sensitive information that doesn’t belong in Async Storage.  
  
****Android - Secure Shared Preferences#****  
Shared Preferences is the Android equivalent for a persistent key-value data store. Data in Shared Preferences is not encrypted by default, but Encrypted Shared Preferences wraps the Shared Preferences class for Android, and automatically encrypts keys and values.  
  
****Android - Keystore****  
The Android Keystore system lets you store cryptographic keys in a container to make it more difficult to extract from the device. In order to use iOS Keychain services or Android Secure Shared Preferences, you can either write a bridge yourself or use a library that wraps them for you and provides a unified API at your own risk. Some libraries to consider:

* Expo-secure-store
* React-native-keychain
* react-native-sensitive-info - secure for iOS, but uses Android Shared Preferences for Android (which is not secure by default). There is however a branch that uses Android Keystore.

### **28. What is Network Security and SSL Pinning?**

****Understanding of SSL:****

SSL (Secure Sockets Layer) and its successor, TLS (Transport Layer Security), are protocols for establishing authenticated and encrypted links between networked computers.  
SSL/TLS works by binding the identities of entities such as websites and companies to cryptographic key pairs via digital documents known as X.509 certificates. Each key pair consists of a private key and a public key. The private key is kept secure, and the public key can be widely distributed via a certificate.  
  
****Understanding of pinning****  
Pinning is an optional mechanism that can be used to improve the security of a service or site that relies on SSL Certificates. Pinning allows specifying a cryptographic identity that should be accepted by users visiting site/app  
           
****Why do we need SSL pinning?****

One of the inherent risks to the SSL ecosystem is mis-issuance. This is when an unauthorized certificate is issued for a domain/host you control. This can happen with both public and private PKIs (Public Key Infrastructure)

****How is SSL pinning used in Mobile applications?****  
When mobile applications communicate with the server, they typically use SSL to protect the transmitted data against tampering. By default SSL implementations used, apps trust any server with a certificate trusted by the Operating systems trust store, This store is a list of certificate authorities that are shipped with the operating system.

With SSL pinning, however, the application is configured to reject all but one or few predefined certificates, whenever the application connects to a server, it compares the server certificate with the pinned certificate(s) , if and only if they match the server is trusted and SSL connection is established.

### **29. Explain setNativeProps. Does it create Performance issues and how is it used ?**

It is sometimes necessary to make changes directly to a component without using state/props to trigger a re-render of the entire subtree. When using React in the browser, for example, you sometimes need to directly modify a DOM node, and the same is true for views in mobile apps. setNativeProps is the React Native equivalent to setting properties directly on a DOM node.  
Use setNativeProps when frequent re-rendering creates a performance bottleneck.  
  
Direct manipulation will not be a tool that you reach for frequently; you will typically only be using it for creating continuous animations to avoid the overhead of rendering the component hierarchy and reconciling many views. setNativeProps is imperative and stores state in the native layer (DOM, UIView, etc.) and not within your React components, which makes your code more difficult to reason about. Before you use it, try to solve your problem with setState and shouldComponentUpdate.

### **30. How to make your React Native app feel smooth on animations ?**

The primary reason and an important one why well-built native apps feel so smooth are by avoiding expensive operations during interactions and animations. React Native has a  limitation that there is only a single JS execution thread, but you can use InteractionManager to make sure long-running work is scheduled to start after any interactions/animations have completed.

Applications can schedule tasks to run after interactions with the following:

InteractionManager.runAfterInteractions(() => {

// ...long-running synchronous task...

});

### 1) Explain React Native?

React Native is an open-source JavaScript framework introduced by Facebook. It is used for developing a real, native mobile application for iOS and Android platforms. It uses only JavaScript to build a mobile app. It is like React, which uses native components rather than using web components as building blocks. It is cross-platform, which allows you to write code once and can run on any platform.

React Native application is based on React, a JavaScript library developed by Facebook and XML-Esque markup (JSX) for creating the user interface. It targets the mobile platform rather than the browser. It saves your development time as it allows you to build apps by using a single language JavaScript for both Android and iOS platforms.

### 2) What are the advantages of React Native?

React Native provides many advantages for building mobile applications. Some of the essential benefits of using React Native are given below:

* ****Cross-Platform:**** It offers the facility to "Write once and run everywhere." It is used to create apps for Android, iOS, and Windows platforms.
* ****Performance:**** The code written in React Native is compiled into native code, which enables it for all operating systems to provide closer native appearance and functions in the same way on all platforms.
* ****Community:**** React Native provides a large community of passionate developers who are always ready to help us to fix bugs, and issues occur at any instant.
* ****Hot Reloading:**** Making a few changes in your app's code immediately visible during development. If the business logic is changed, its reflection is live reloaded on screen.
* ****Faster Development:**** React Native helps to develop apps fast. It uses a common language to build an app for Android, iOS, and Windows platforms, which gives speedier app deployment, delivery, and quicker time-to-market.
* ****JavaScript:**** JavaScript knowledge is used to build native mobile apps.

### 3) What are the disadvantages of React Native?

Some of the big disadvantages of React Native for building mobile applications are given below:

* ****React Native is still new and immature:**** React Native is a new framework in Windows, Android, and iOS programming languages. It is still in the improvement stage, which can have a negative impact on the apps.
* ****Learning is tough:**** React Native cannot learn quickly, especially for a fresher in the app development field.
* ****It Lacks the Security Robustness:**** React Native is an open-source JavaScript framework, which is fragile and creates a gap in the security robustness. When you are creating banking and financial apps where data is highly confidential, experts advise not to choose React Native.
* ****It Takes More Time to Initialize:**** React Native takes a lot of time for initializing the runtime even if you are using the hi-tech gadgets and devices.
* ****Existence is Uncertain:**** As Facebook develop this framework, its presence is uncertain since it keeps all the rights to kill off the project anytime. As the popularity of React Native rises, it is unlikely to happen.

### 4) List the essential components of React Native.

These are the core components of React Native:

* ****View:**** It is the basic built-in component used to build UI of Mobile apps. The view is similar to the div in HTML. It is a content area where you can display your content.
* ****States:**** It is used to control the components. The variable data can be stored in the state. It is mutable means a state can change the value at any time.
* ****Props:**** Props are used to pass data to the different components. It is immutable means props cannot change the value. It provides a connection between the container component and a presentation component.
* ****Style:**** It is an essential component in the web or mobile, which makes the application attractive. React Native does not require any special language or syntax for styling. It can style the application using the JavaScript object.
* ****Text:**** This component displays text in the app. It uses the basic component textInput to take text input from the user.
* ****ScrollView:**** It is a scrolling container used to host multiple views. It can be used to render the large list or content in view with a scroll bar.

### 5) How many threads run in React Native?

The React Native app contains the following thread:

* ****React Native UI Thread (Main Thread):**** This thread is used for the layout of the mobile app.
* ****React Native JavaScript Thread:**** It is a thread where our business logic will run. It means JS thread is a place where our JavaScript code is executed.
* ****React Native Modules Thread:**** Sometimes, our app needs access to platform API, which happens as a part of native module thread.
* ****React Native Render Thread:**** This thread is used to generate actual OpenGL commands used to draw the app UI.

### 6) What is React Native Apps?

React Native Apps are not web applications. These types of apps are running on mobile devices, and cannot load over the browser. Also, they are not hybrid apps that build over Ionic, Phonegap, etc. which can run over WebView component. They are the real native apps built in a single language JavaScript with the native components to run on mobile devices.

### 7) List Step to Create and start a React Native App?

The following steps are necessary to create and start a React Native app:

****Step-1:**** Install Node.js

****Step-2:**** Install react-native environments by using the following command.

1. $ npm install -g create-react-**native**-app

****Step-3:**** Create a project by using the following command.

1. $ create-react-**native**-app MyProject

****Step-4:**** Next, navigate in your project by using the following command.

1. $ cd MyProject

****Step-5:**** Now, run the following command to start the project.

1. $ npm start

To read more information, [click here](https://www.javatpoint.com/react-native-environment-setup).

### 8) What are states in React Native?

It is used to control the components. The variable data can be stored in the state. It is mutable means a state can change the value at any time.

****Example****

Here, we are going to create a Text component with state data. The content of the Text component will be updated whenever we click on it. The event ****onPress**** calls the ****setState**** function, which updates the state with "****myState****" text.

1. **import** React, {Component} from 'react';
2. **import** { Text, View } from 'react-native';
4. export **default** **class** App **extends** Component {
5. state = {
6. myState: 'This is a text component, created using state data. It will change or updated on clicking it.'
7. }
8. updateState = () => **this**.setState({myState: 'The state is updated'})
9. render() {
10. **return** (
11. <View>
12. <Text onPress={**this**.updateState}> {**this**.state.myState} </Text>
13. </View>
14. );
15. }
16. }

To read more information, [click here](https://www.javatpoint.com/react-native-state).

### 9) What are props in React Native?

The properties of React Native components are pronounced as props. They are used to pass data to the different components. In React Native, several components are customized at the time of creation with different parameters, and these parameters are known as props. It is immutable means props cannot change the value. It provides a connection between the container component and a presentation component.

****Example****

Here, we have created a Heading component, with a message prop. The parent class App sends the prop to the child component Heading.

1. // Parent Component
2. export **default** **class** App **extends** Component {
3. render () {
4. **return** (
5. <View style={{alignItems: 'center' >
6. <Heading message={'Custom Heading **for** Parent **class**?}/>
7. </View>
8. )
9. }
10. }
12. // Child component
13. export **default** **class** Heading **extends** Component {
14. render () {
15. **return** (
16. <View style={{alignItems: 'center' >
17. <Text>{**this**.props.message}</Text>
18. </View>
19. )
20. }
21. }
22. **const** styles = StyleSheet.create({
23. welcome: {
24. fontSize: 30,
25. }
26. });
27. Heading.propTypes = {
28. message: PropTypes.string
29. }
30. Heading.defaultProps = {
31. message: 'Heading One'
32. }

To read more information, [click here](https://www.javatpoint.com/react-native-props).

### 10) List the users of React Native?

Today, thousands of React Native built-in apps are available in the market. Here is the list of users who uses React Native apps:

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

### 11) Are all React components usable in React Native?

React web components use DOM elements (ex. div, h1, table, etc.) to display on UI. But, these components are not supported in React Native. You will need to find libraries or components which is made specifically for React Native. It is very hard to find that there are components available, which support both. But, it should be easy to figure out that the given components are made for React Native or not. Thus, it makes clear that all components are not usable in the React Native.

### 12) How Virtual DOM works in React Native?

Virtual DOM is a lightweight JavaScript object, which is an in-memory representation of a real DOM. It is an intermediary step between the render function being called and the displaying of elements on the screen. It is similar to a node tree, which lists the elements, their attributes, and content as objects and their properties. The render function creates a node tree of the React components and then updates this node tree in response to the mutations in the data model caused by various actions done by the user or by the system.

****Virtual DOM works in three steps:****

* Whenever any data changes in the React App, the entire UI is re-rendered in Virtual DOM representation.
* Now, the difference between the previous DOM representation and the new DOM is calculated.
* Once the calculations are completed, the real DOM updated with only those things which are changed.

### 13) Can we combine native iOS or Android code in React Native?

Yes, we can combine the native iOS or Android code with React Native. It can combine the components written in Objective-C, Java, and Shift.

### 14) Do we use the same code base for Android and iOS?

Yes, we can use the same codebase for Android and iOS, and React takes care of all the native component translations. For example, a React Native ScrollView use ScrollView on Android and UiScrollView on iOS.

### 15) What is the difference between an Element and a Component in React Native?

The difference between an Element and a Component in React Native are:

|  |  |
| --- | --- |
| **React Element** | **React Component** |
| The React Element is a simple object, which describes a DOM node and its attributes or properties. It is an immutable object where you cannot apply any methods. | The React Component is a function or class that takes inputs and returns a React element. It contains references to its DOM nodes and the instances of the child components. |
| ****For Example:****  <button className = "green"></button> | ****For Example:****  const SignIn = () => (  <div>  <p>Sign In</p>  <button>Continue</button>  <button color='green'>Cancel</button>  </div>  ); |

### 16) What is the difference between React and React Native?

The essential differences between React and React Native are:

* React is a JavaScript library, whereas React Native is a JavaScript framework based on React.
* Tags can be used differently in both platforms.
* React is used for developing UI and Web applications, whereas React Native can be used to create cross-platform mobile apps.

To read more information, [click here](https://www.javatpoint.com/react-native-vs-react).

### 17) What is the difference between React Native and Ionic?

The essential differences between React Native and Ionic are:

* Ionic is a typical hybrid development framework. It mainly focuses on front-end user experience or UI interaction, which handles all the look and feel of your app. It is easy to learn and can integrate with other libraries or frameworks such as Angular, React, Cordova, etc. Its purpose is to write once and runs everywhere.
* React Native is an open-source JavaScript framework developed by Facebook to build a cross-platform mobile app. It is used for developing a mobile application for iOS, Android, and Windows. React Native is the same as React, but it uses native components instead of using web components as building blocks. It targets mobile platforms rather than the browser. Its purpose is to learn once and write anywhere.

To read more information, [click here](https://www.javatpoint.com/react-native-vs-ionic).

### 18) What are the differences between React Native and Native (Android and iOS)?

React Native allows you to write once and runs everywhere. It means we can reuse the React Native code on both Android and iOS platforms. Since we can reuse most of the React Native code between both platforms, but Android and iOS are different systems. Here, we are going to see these differences.

****Operating System****

You can build applications for both Android and iOS with React Native, but it is not an easy task to check that the app works on both systems if you are working on Windows systems. Windows do not allow to run XCode and its simulator, which is a macOS app. There are other tools available, but they are not official.

****Native elements****

The elements perform different actions for the React Native and Native apps. React Native apps uses elements from React Native library, whereas Native apps do not use elements of React native libraries.

****Specific Styles-Shadows****

Shadows style is an essential term of differences between iOS and Android while working on cross-platform apps. Android does not support shadow; instead of this, it uses elevation property.

****Linking libraries****

Sometimes we want to use third-party libraries in our app. Most of the time, we add it as a dependency, but sometimes it requires manual linking for adding libraries. Linking libraries manually is not an easy task for developers, either web or native applications. Since the React Native is in the improvement stage, the libraries docs are not updated according to the latest framework.

****React Native vs. Native (Android and iOS)****

|  |  |  |  |
| --- | --- | --- | --- |
|  | **React Native** | **Android** | **iOS** |
| ****Language**** | JavaScript JSX | Java | Objective-C/Swift |
| ****Debugger**** | Text Editor, Chrome Debugger | Android Studio | XCode |
| ****Used By**** | Facebook, AirBnB | Airdroid, Chromer | GarageBand, iMovie |

### 19) What is the difference between React Native and Xamarin?

The essential differences between React Native and Ionic are:

* React Native is an open-source JavaScript framework developed by Facebook to build a cross-platform mobile app for iOS, Android, and Windows. React Native is the same as React, but it uses native components instead of using web components as building blocks. It targets mobile platforms rather than the browser. Its purpose is to learn once and write anywhere.
* Xamarin is an open-source, cross-platform development framework, which offers you to build android, iOS, Windows, and Mac apps by using the C# language. It is first launched in May 2011 by Xamarin Company. In 2016, Microsoft had signed an agreement to acquire Xamarin.

To read more information, [click here](https://www.javatpoint.com/react-native-vs-xamarin).

### 20) What does a StyleSheet.create do?

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

### 21) For what XHR Module is used in the React Native?

In React Native, the XHR module is used to implement the ****XMLHttpRequest****. It is an object for interacting with remote services. This object is consists of two parts, front-end and back-end, where the front-end allows interacting within JavaScript. It sends the request to the XHR back-end, which is responsible for a processing network request. The back-end part is called Networking.

### 22) Is React Native a Native mobile app?

Yes, React Native is a native mobile app, which compiles a native mobile app using native app components. It is neither a Hybrid mobile app that uses WebView to run the HTML5 app nor a mobile web app. The React Native framework builds a real mobile app, which is indistinguishable from an app built using Objective-C/Swift or Java.

### 23) Which language is used in React Native?

The language used in React Native is Java for Android applications and Objective-C/Swift for iOS apps.

### 24) What is style in React Native?

It is an essential component in the web or mobile, which makes the application attractive. React Native does not require any special language or syntax for styling. It can style the application using the JavaScript object.

To read more information, [click here](https://www.javatpoint.com/react-native-style).

### 25) What are Refs in React Native?

React refs are useful features that allow you to access DOM elements or component's instance directly within React. It comes handy in situations where you want to change the child of a component without using props or re-rendering the whole component.

To read more information, [click here](https://www.javatpoint.com/react-refs).

### 26) Why React Native app use keys?

Keys are a unique identifier. They are used to identify which items have changed, updated, or deleted from the lists. It should always use inside the array.

To read more information, [click here](https://www.javatpoint.com/react-keys).

### 27) What is meant by HOC in React Native?

HOC Stands for Higher-Order Component. It is a technique, which allows you to reuse the component logic. It is a function that takes a component and gives back a new component.

****Syntax****

1. **const** NewComponent = higherOrderComponent(WrappedComponent);

### 28) What is meant by InteractionManager, and why it is Important?

The ****InteractionManager**** is a native module in React Native, which is responsible for differing the execution of a function until an interaction has finished. To handle this deferral, we need to call ****InteractionManager.runAfterInteractions(() => {...})****.

The InteractionManager is important because React Native has ****two threads****. One is JavaScript UI thread, which handles drawing updates to the screen, and the second thread used for all task, not on the UI thread. Since React Native has only one thread for making UI updates, it can get overloaded and drop frames, especially in navigation screen animations. So, developers use the InteractionManager to ensure that the function is executed after these animations occur. As a result, we do not drop frames on the UI thread.

### 29) What are the differences between Class and Functional Component?

The essential differences between the class component and functional component are:

****Syntax:**** The declaration of both components is different. A functional component takes props, and returns React element, whereas the class components require to extend from React.

1. //Functional Component
2. function WelcomeMessage(props) {
3. **return** <h1>Welcome to the , {props.name}</h1>;
4. }
6. //Class Component
7. **class** MyComponent **extends** React.Component {
8. render() {
9. **return** (
10. <div>This is main component.</div>
11. );
12. }
13. }

****State:**** The class component has a state while the functional component is stateless.

****Lifecycle:**** The class component has a lifecycle, while the functional component does not have a lifecycle.

### 30) When would you prefer a class component over functional components?

We prefer class component when the component has a state and lifecycle; otherwise, the functional component should be used.

### 31) How React Native handle different screen sizes?

React Native provides many ways to handle screen sizes. Some of them are given below:

****1. Flexbox:**** It is used to provide a consistent layout on different screen sizes. It has three main properties:

* flexDirection
* justifyContent
* alignItems

****2. Pixel Ratio:**** It is used to get access to the device pixel density by using the ****PixelRatio**** class. We will get a higher resolution image if we are on a high pixel density device.

****3. Dimensions:**** It is used to handle different screen sizes and style the page precisely. It needs to write the code only once for working on any device.

****4. AspectRatio:**** It is used to set the height or vice versa. The aspectRatio is present only in React-Native, not a CSS standard.

****5. ScrollView:**** It is a scrolling container which contains multiple components and view. The scrollable items can be scroll both vertically and horizontally.

### 32) What is ListView?

ListView is a core component of React Native, which contains a list of items and displays in vertical scrollable lists.

To read more information, [click here](https://www.javatpoint.com/react-native-listview).

### 33) What are the best UI Components for React Native?

The best UI component for React Native are:

* Material UI
* Semantic UI
* React Bootstrap
* React Toolbox
* Ant Design

### 34) What are the similarities between React and React Native?

The most common similarities between React and React Native are:

* React Lifecycle Methods
* React Components
* React States and Props
* Redux Libraries

### 35) What are animations in React Native?

The animation is a method in which images are manipulated to appear as moving objects. React Native animations allows you to add extra effects, which provide great user experience in the app. We can use it with React Native API, Animated.parallel, Animated.decay, and Animated.stagger.

React Native has two types of animation, which are given below.

* ****Animated:**** This API is used to control specific values. It has a start and stops methods for controlling the time-based animation execution.
* ****LayoutAnimated:**** This API is used to animate the global layout transactions.

To read more information, [click here](https://www.javatpoint.com/react-native-animation).

### 36) How is data loaded on the server by React Native?

React Native uses Fetch API to fetched data for networking needs.

### 37) What is the storage system in the React Native?

React Native storage is a simple, unencrypted, asynchronous, persistent system, which stores the data globally in the app. It stores data in the form of a ****key-value**** pair. React Native provides AsyncStorage class to store data globally. Using the ****AsyncStorage**** class, we need to have a data backup and synchronization class. It is because data saved on the device is not permanent and not encrypted.

To read more information, [click here](https://www.javatpoint.com/react-native-asyncstorage).

### 38) Can you integrate more features in the existing app by React Native?

Yes, we can add new features to existing applications in React Native.

### 39) What is meant by Gesture Responder System?

It is an internal system of React Native, which is responsible for managing the lifecycle of gestures in the system. React Native provides several different types of gestures to the users, such as tapping, sliding, swiping, and zooming. The responder system negotiates these touch interactions. Usually, it is used with Animated API. Also, it is advised that they never create irreversible gestures.

### 40) What does React Native Packager do in the React Native?

The React Native Packager performs the following functionalities:

* The React Native Packager combines all the JavaScript code of your application into a single file and then translate any of the JavaScript code that your device won't understand like JSX.
* It also converts the assets (e.g., PNG file) used in your project into objects, which can be displayed by an Image component.

### 41) Why React Native use Redux?

Redux is a state container for JavaScript applications. It is a state management tool, which helps you to write applications that behave consistently, can run in a different environment, and are easy to test.

React Native use Redux because it allows developers to use one application state as a global state and interact easily with the state from any React component. It can combine with any framework or library.

### 42) How to update React Native with the latest version?

It is very important to upgrade the existing React Native with the latest version, which gives you access to more APIs, views, developer tools, and other latest features. The following steps need to be performed for upgrading the React Native with the latest versions.

1. Upgrade your ****expo project**** in ****package.json**** with the latest version of react-native, react, and expo package.

2. Set the latest version of SDK, which is compatible with the latest react-native in your ****app.json**** file.

3. Upgrade the React Native CLI to update the source file by using the following command.

1. $ react-**native** upgrade

4. Install the upgrade helper web tool that provides you to upgrade your apps between any two versions.

5. Upgrade your project files by running the following command.

1. $ react-**native** init

6. Last, you need to perform the troubleshoot activity to upgrade with React Native CLI.

### 43) What is API in React Native?

An API or Application Programming Interface is a software intermediary that lets in two applications to communicate with each other without having to know how they are implemented. Sometimes it is thought of as a contract, with documentation that represents an agreement between two parties. ****For example,**** each time when you use an app in the mobile like Facebook, it sends a message, or when you see the weather on your phone, these are the usage of an API.

React Native use the Fetch networking API to suit our needs. It simply calls the URL through Fetch, and then make requests to the server as needed. The React Native API mainly uses three lifecycle methods, which are constructor, componentDidMount, and Render.

### 44) How to use Axios in the React Native?

Axios is a popular library for making HTTP requests from the browser. It allows us to make GET, POST, PUT, and DELETE requests from the browser. Therefore, React Native uses Axios to make requests to an API, return data from the API, and then perform actions with that data in our React Native app. We can use Axios by adding the Axios plugin to our project using the following command.

1. # Yarn
2. $ yarn add axios
4. # npm
5. $ npm install axios --save

Axios have several features, which are listed below:

* It makes XMLHttpRequests from the browser.
* It makes Http requests from the React Native framework.
* It supports most of the React Native API.
* It offers a client-side feature that protects the application from XSRF.
* It automatically transforms response and request data with the browser.

### 45) Which database is best for React Native?

The most popular database for React Native is an SQLite database.

### 46) How to use firebase in react native?

Firebase is a popular tool for mobile and web app development platform. It provides many services to help you in building fast and high-quality apps, grow your user base, and earn more money without managing infrastructure. It is a powerful Database as a Service (DBaaS) tool, which provides a scalable cloud database to store and sync data for client and server-side development. Some of the key features of firebase are authentication, Real-time database, cloud messaging, crash reporting, and analytics. Firebase is a type of freemium model, not an open-source model. However, you can use its services free until you don't pass the limits of its free tier.

We can get started with firebase by using following steps:

* First, login into the firebase console and then create a project.
* Retrieve apikey, authDomian, DatabaseURL, and storage bucket from the console.
* Next, you need to create a new React Native project
* Install firebase plugin from npm
* Add firebase plugin into the React Native project

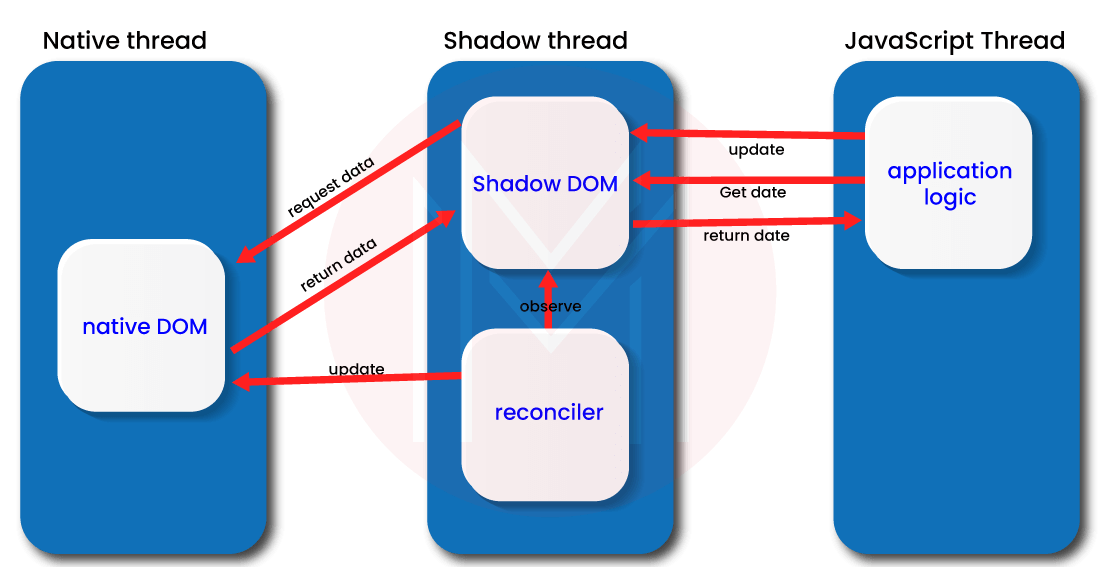
### **1. What is the general purpose of a thread in React Native?**

We can control a single sequential flow within a program using a thread in React Native

### **2. What are the threads used in React Native?**

React Native provides the following threads:

****Shadow Thread:**** it's a background thread. We can calculate a layout created using React library inside React Native by this thread.  
****MAIN/UI Thread:**** It's the main thread on which our app runs.  
****JavaScript Thread:**** it executes the main Javascript code.



### **3. Which type of database works best in React Native?**

SQLite is the most suited database for React Native.

|  |
| --- |
| ****Are you looking forward to becoming a Javascript Developer? Check out the**[Javascript Training](https://mindmajix.com/javascript-training" \o "Javascript Online Course" \t "https://mindmajix.com/_blank)**and get certified.**** |

### **4. Can apps for Android and iOS behave or look different?**

Yes! However, it comes at a cost as different code needs to be written for each of the platforms. But even then, it is relatively much cheaper than building 2 native apps. React Native has been developed in such a way that the iOS and Android logic is different. So the platforms can still share their functionality even when they look or behave differently.

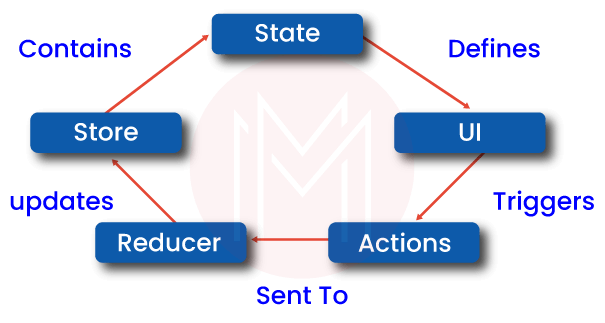
### **5. Elaborate Redux in React Native.**

Redux is a state container for JavaScript applications. We can develop applications able to run in various environments with the help of Redux. This means that we can handle the whole data flow of the application within a single container. And the previous state also persists alongside.

### **6. What are the components of Redux available in React Native app?**

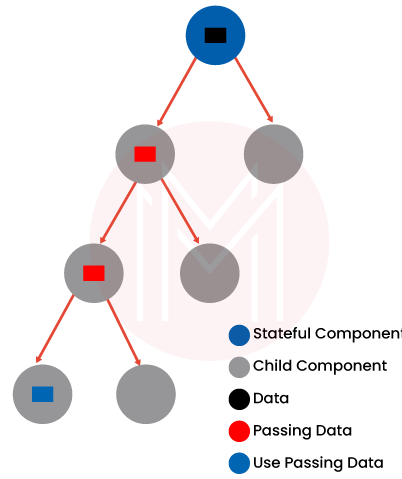
Redux has the following components:

* Actions
* Reducers
* Store
* Components



### **7. What is meant by Props drilling?**

Props drilling or threading is the process of passing down the data from the parent component to its exact child component. In between, it passes through other components which own the props.



### **8. List the steps to avoid Props drilling.**

The given steps can be used to avoid props drilling:

* React Context API
* Composition
* Render props
* HOC
* MobX or Redux

### **9. Explain about Timers available in React Native app.**

Timers are essential in any application. React Native provides the provision of the following timers:

#### **setInterval, clearInterval**

* The setInterval method is used to create a loop of a code running after a specific time interval as the second parameter specifies.
* clearInterval is used to stop the interval.

#### ****setTimeout, clearTimeout****

* setTimeout is used to schedule a piece of code to run at a specific scheduled time.
* clearTimeout is just used to clear that timer.

#### ****setImmediate, clearImmediate****

* setImmediate calls or execute the function as soon as possible.
* clearImmediate cancels the immediate actions set by setImmediate.

#### ****requsetAnimationFrame, cancelAnimationFrame****

* requsetAnimationFrame updates an animation before the next one.
* cancelAnimationFrame cancels the functions set by requsetAnimationFrame.

### **10. Name the tools that can be used to debug React Native apps.**

The following tools can be used to debug React Native apps:

* The developer menu
* Chrome's DevTools
* React developer tools
* React Native debugger

### **11. Are there any default props available in React Native? What are they used for?**

Yes, default props are available in React Native. The default props value is used by the component in case the props value doesn't pass.

### **12. What is the function of Flexbox in React Native?**

Flexbox is a layout model. The function of Flexbox is aligning and distributing the space within a container. It is a great tool that is used for responsive design systems. Its most used properties are flexDirection, justifyContent, and alignItems.

### **13. How does React Native differ from other mobile app development frameworks?**

1. Most other alternatives, such as ionic and Cordova run web apps. At the same time, React Native app runs a real mobile application and not a web app.
2. It provides better performance than others as the React Native app gets converted into machine code running on mobile.
3. React Native apps are relatively much closer to Native app development than the other Javascript frameworks.
4. Mobile apps developed by React Native have a smaller bundle size.

|  |
| --- |
| ****Related Articles:**[React JS Tutorial](https://mindmajix.com/react-js-tutorial" \t "https://mindmajix.com/_blank)** |

### **14. What is meant by the state in the React component?**

The state is another way of modifying a react component apart from the props function. Unlike props, the state value of React component changes in the component's life cycle. The react component's state value should not be changed directly. We should use the setState method of the React framework to change the state of a component.

### **15. Mention some of the best UI components in React Native.**

Some of the best UI components in React Native are given below:

* Semantic UI
* Material UI
* React toolbox
* React bootstrap
* Ant design

## **React Native Interview Questions for Experienced**

### **1. What is the function of a StyleSheet.create?**

The function of StyleSheet.create() in React Native is to ensure the immutability and opaqueness of the values. They are used for sending the style through the bridge only once. Hence, they help to avoid the passing of a new style object.

### **2. What is meant by Refs in React Native?**

Refs in React Native are the features used for accessing a component's instance or DOM elements directly within React. In situations where you wish to change a child's component without the component's re-rendering or the use of props, Refs come to your rescue.

### **3. Mention the similarities between React Native and React.**

Some of the similarities between React Native and React are given below:

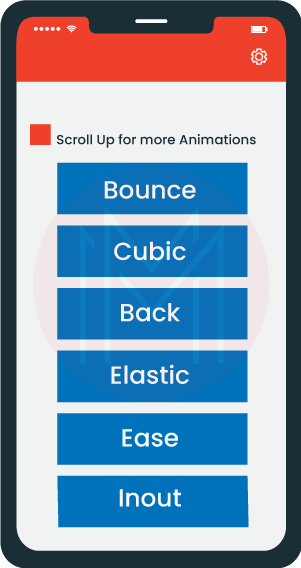
* React components
* React lifecycle methods
* Redux libraries
* React states and props

### **4. What are animations and their types in React Native?**

We can manipulate the images by using animations to make them appear as moving objects. For greater user experience in the application, extra effects can be added in React Native animations. It can be used with React Native API, Animated.decay, Animated.parallel, and Animated.stagger.

React Native provides the given two types of animations:

* ****Animated-****We use this API for controlling specific values. It provides the start and stops methods to control the animation execution on the basis of time.
* ****LayoutAnimated-**** We use this API for animating the global layout transactions.



### **5. What do you mean by JSX?**

The full form of JSX is JavaScript XML. It enables writing HTML elements in React and placing them in DOM without any /orappendChild() and craeteElement() methods. JSX also helps in converting HTML tags into react elements. And hence writing it becomes easier.

### **6. List the ways to write react components.**

There are two ways to write a react component:

* ****Class-based component****  
  A simple function is used here that returns the JSX
* ****Functional component****  
  It uses the class keyword, which is introduced in ES6. The render lifecycle method that returns the JSX is used here.

### **7. Mention the steps for creating and starting a React Native App.**

The following steps need to be followed to create and start an app:

* Install Node.js
* Use the following command for installing React Native environments.  
  $ npm install –g create-react-native-app
* Use the following command for creating a project.  
  $ create-react-native-app MyProject
* You can navigate the project with the following command.  
  $ cd MyProject
* Start the project by running the given command.  
  $ npm start

### **8. What is the Gesture Responder System in React Native?**

Gesture Responder System is React Native's internal system, which manages the lifecycle of the system of gestures. Sliding, tapping, zooming, and swiping are some of the various kinds of gestures that React Native provides. These touch interactions are negotiated by the responder system. We use the Animated API generally. And we can't reverse the gestures once made.

### **9. How can you use Axios in React Native?**

Axios is a famous library if you want to make HTTP requests from the browser. We can make the POST, GET, DELETE, AND PUT requests from the browser. That's why React Native uses the Axios for making requests to an API, returning data from API, and performing actions with the data in React Native application. By using the given command, we can use Axios via the addition of the Axios plugin to the project.

# Yarn

$ yarn add Axios

#npm

$npm install axios--save

### **10. List the features of Axios.**

There are several features of Axios which are given below:

* The HTTP requests are made from the React Native framework.
* The XML HTTP requests are made from the browser.
* A client-side feature that protects our app from XSRF is offered by Axios.
* Most React Native APIs are supported by Axios.
* The response is automatically transformed, and the data is requested by the browser.

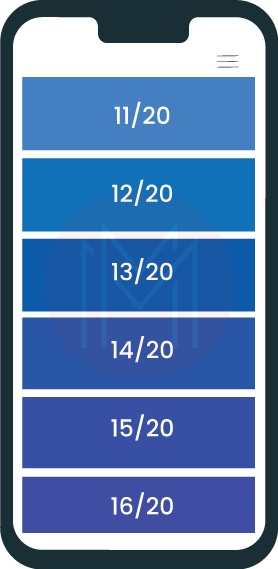
### **11. Mention some ways of handling different screen sizes.**

Several ways of handling screen sizes are listed below:

* Flexbox
* Dimensions
* Pixel ratio
* ScrollView
* AspectRatio

### **12. Explain the ScrollView component available in React Native.**

ScrollView component is a general scrolling container. It is capable of holding multiple views and components. We can scroll both horizontally and vertically. We can also use it for implementing zoom and pinch functions in iOS. And we can swipe horizontally between views on Android by the use of the ViewPagerAndroid component. Its best use is presenting a limited number of things. ScrollView renders all the views and elements even if they are not visible on the screen.

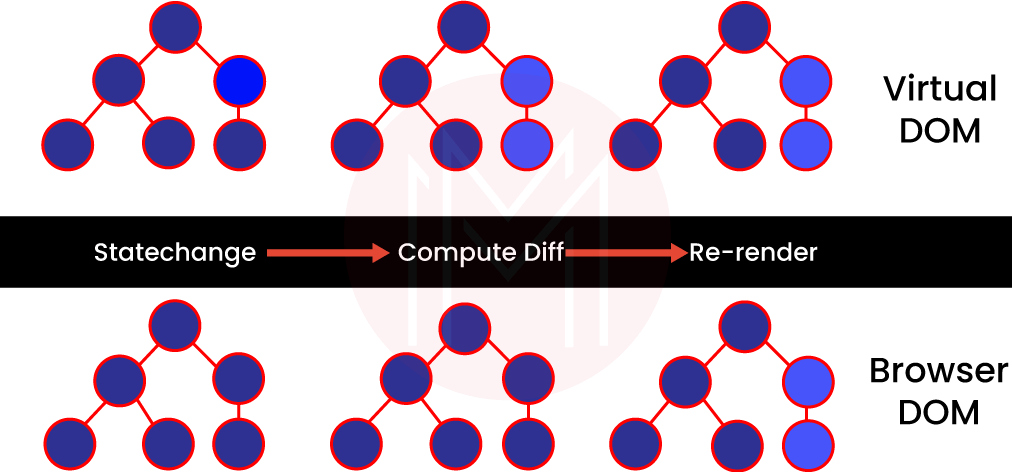


### **13. What is virtual DOM?**

Virtual DOM is real DOM's representation. And it is a lightweight JavaScript object. This step comes in between calling the render function and displaying the elements on the screen. It is like a node tree that makes the list of elements, attributes, and content into objects and properties. A node tree of React components is created by the render function, and then it's updated as per the changes in the data model.

### **14. List the steps of the working of Virtual DOM.**

* The virtual DOM representation re-renders the entire UI whenever there is a change in the data model.
* Then it calculates the difference between the previous and new DOM representation.
* The DOM gets updated once the calculations are completed.



### **15. Elaborate HOC in React Native.**

The full form of HOC is Higher-Order Component. We can reuse the component logic by using this technique. This function takes in a component and returns a new one.

****Syntax:****

Const NewComponent = higherOrderCompnent(WrappedComponent);

## **Frequently Asked Interview Questions on React Native:**

### **1. Which major companies do use React Native in their apps and UIs?**

Microsoft, Walmart, Instagram, Skype, Airbnb, Uber Eats, Pinterest, Facebook, Bloomberg, and Flipkart are some of the major companies using React Native.

### **2. What are the top alternatives to React Native, in your opinion?**

The following are the top alternatives to React Native, in my opinion:

* Flutter
* Xamarin
* Nativescript
* Ionic
* Apache Cordova

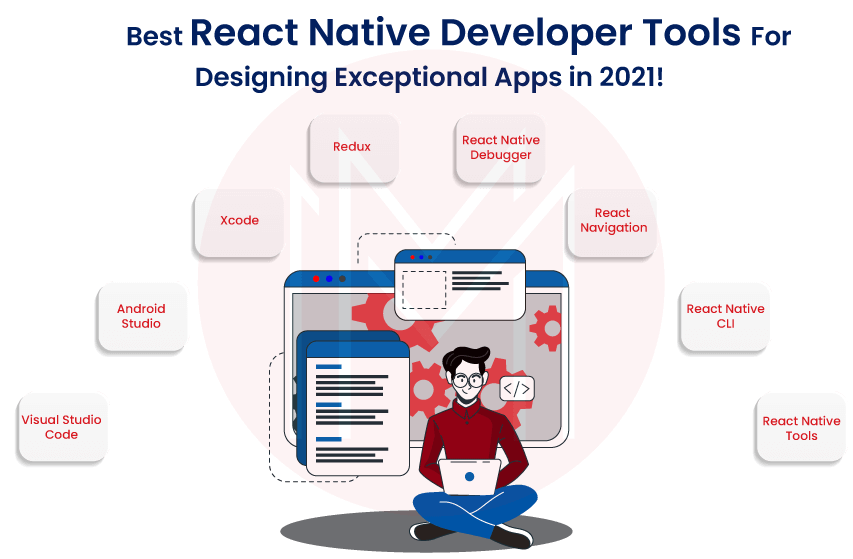
### **3. Which language is React Native based on?**

JavaScript is the language upon which React Native is based. However, its community-driven and open-source framework allow us to avail of online help as well.

### **4. What are the tools for development in React Native?**

The following are the tools for development in React Native:

1. ****Atom:**** It is a free text editor. And its features include cross-platform editing, a built-in package manager, user templates, themes, and multiple file navigation.
2. ****Visual Studio Code:**** A good set of plugins is provided here for supporting other languages like Python, Java, C#, C++, and PHP. Runtimes, including .NET and Unity, are also supported.
3. ****Expo:**** It is an open-source platform that provides tools to build, deploy, and iterate quickly on native iOS, Android, and web pages via the same JavaScript codebase. Its features include a rich native APIs library, reporting, and a shared native runtime
4. ****Other tools include Redux, React Navigation, Reduxsauce, and TypeScript/Flow.****

********

### **5. Differentiate between React and React Native.**

1. React is used to develop web pages, whereas React Native is used to develop mobile apps.
2. React is a library used for web development, whereas React native is a platform.
3. The focus of React is an improved User Interface. While React Native shares a common logic layer for all operating systems.
4. We have everything needed to start a project when we use React Native. At the same time, React is just a library we have chosen to develop a web application.

### **6. Mention some pros of React Native.**

Some pros of React Native are given below:

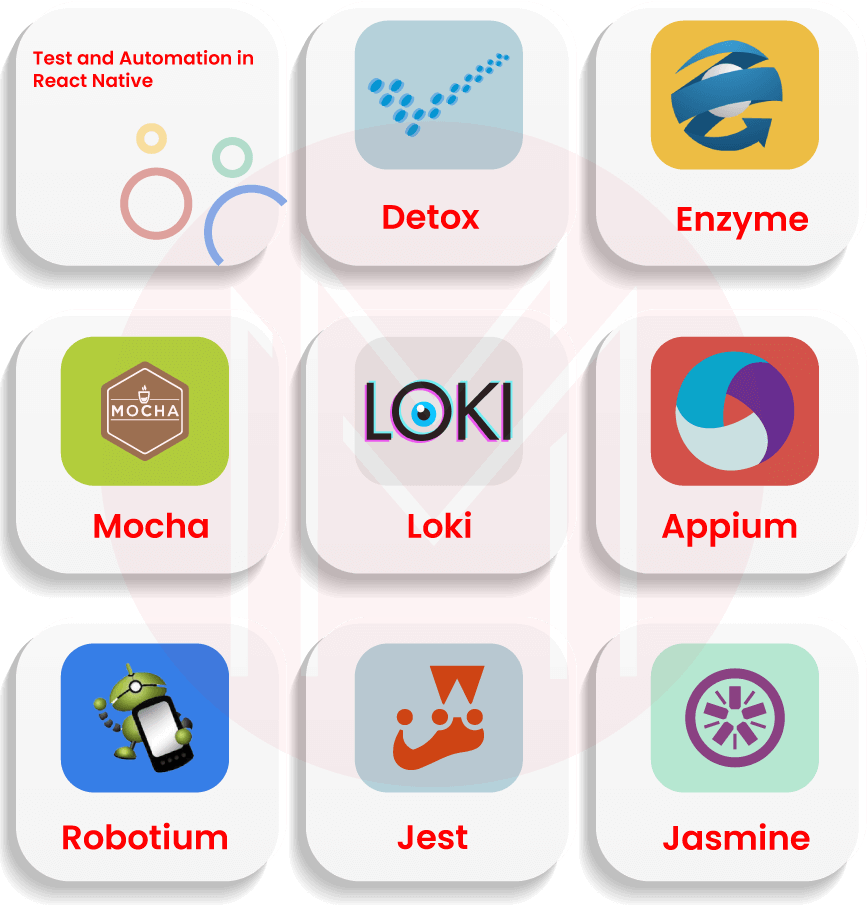
* Works everywhere, including Android, iOS, and Windows.
* We are able to reuse, recycle, and share an earlier developed codebase.
* Without a big investment, we can test our MVP in the market and get feedback much faster.
* It has a low maintenance cost.
* A great user experience is provided.
* The help is available on demand. We can easily find the solutions to our problem with the help of the massive community of React Native.
* Our apps are listed in Play Store and App Store, unlike PWA.
* We can update over the air without even needing to wait for Google Play/App Store to accept our changes.

### **7. Can you transfer the existing apps to React Native?**

We can transfer the apps to React Native as long as our program does not depend on the smartphone for performing heavy computational operations. React Native enables building more complicated and larger apps with its ever-expanding capabilities. Also, It becomes cheaper to develop an app on React Native.

### **8. What are the tools available for testing the codebase?**

Jest, Jasmine, Enzyme, Detox, Cypress, Karma, and Mocha are some tools available for the testing codebase.



### **9. Does React Native allow you to write code for iOS and Android in the same codebase?**

Yes, we can write the code for iOS and Android in the same codebase.

### **10. Mention the core components that React Native provides.**

* View
* States
* Props
* Style
* Text
* scrollView

### **Conclusion**

Creative and interactive User Interfaces, as well as apps, are the keys to the success of the company. That is because the users are the ultimate targets for any company's growth. Hence, for web developers having knowledge of React and JavaScript, React Native is an amazing platform to grow. This technology is supported by some of the biggest players in the market as well. This proves the presence of ample growth in this field. And we hope that we can help you in getting one step closer to your success through these React Native Interview Questions.

1. How Different is React-native from ReactJS?

#### **Answer**

| **React** | **React Native** |
| --- | --- |
| It is a JavaScript library, supporting front-end web and being run on a server, for web applications and building user interfaces. | It is a framework that compiles to native app components, allowing you to build native mobile applications for different platforms. |
| [React Js](https://www.bestinterviewquestion.com/react-js-interview-questions) is a Javascript Library where you can develop and run faster web applications. | React-Native is a framework where you can develop mobile applications. |
| React is for websites. | React Native is for mobile applications. |

The ReactJs is a JavaScript library to develop apps in HTML5 while using JavaScript as the developing language. Whereas, React Native is a JavaScript framework used to create native mobile applications while using JavaScript as the development language.

**Also Read:**[Difference between React and React Native](https://www.bestinterviewquestion.com/blog/difference-between-react-and-react-native" \o "difference between React and React Native)

2. How to install a specific version of React Native?

#### **Answer**

To install a specific version of React Native, use this following command:

$ react-native init newproject --version react-native@VersionNumber

**Note**: In the above, replace VersionNumber with the version of React you want to install.

3. How to use map function in React Native?

#### **Answer**

The map function is used to show a list of elements in an array. It can be used in React Native like the following example:

#### **Example**

import React, { Component } from "react";  
import { Text, View } from "react-native";  
  
export default class mapFunction extends Component {  
constructor(props) {  
super(props);  
this.state = {  
array: [  
{  
title: "example title 1",  
subtitle: "example subtitle 1"  
},  
{  
title: "example title 2",  
subtitle: "example subtitle 2"  
},  
{  
title: "example title 3",  
subtitle: "example subtitle 3"  
}  
]  
};  
}  
  
list = () => {  
return this.state.array.map(element => {  
return (  
<View style={{ margin: 10 }}>  
<Text>{element.title}</Text>  
<Text>{element.subtitle}</Text>  
</View>  
);  
});  
};  
  
render() {  
return <View>{this.list()}</View>;  
}  
}

4. What are Props and how is it used in React Native?

#### **Answer**

Props are means to the parameters that are used for customizing the component when the component is being created or re-rendered. They are more like the argument which is passed to the React component.

#### **Example**

import React, {Component} from 'react';  
import {View, Text} from 'react-native';  
class DefaultPropComponent extends Component {  
   render() {  
       return (  
           <View>  
             <Text>  
              {this.props.name}  
            </Text>  
          </View>  
       )  
   }  
}  
Demo.defaultProps = {  
   name: 'BOB'  
}

export default DefaultPropComponent;

5. What is HOC in React Native?

#### **Answer**

**Higher-Order-Component**, shortly known as HOC is an advanced React Native technique to reuse the component logic. The function obtains a component and returns a new element.

NOTE: If you are a **react native developer** then these questions & answers will help you to crack your interview easily.

#### **Example**

function HOC(Comp) {  
     return class NewComp extends Component {  
         render() {  
            return <comp>  
         }  
    }  
}

6. How to use the camera in React Native?

#### **Answer**

To use the camera to React Native, follow these steps carefully:

* On Android, camera permission must be asked:  
  <uses-permission android:name="android.permission.CAMERA" />
* Now, to enable video recording feature add the following code to the AndroidManifest.xml:  
  <uses-permission android:name="android.permission.RECORD\_AUDIO"/> <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" /> <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />
* For iOS, you must update the Info.plist with a usage description for camera  
  <key>NSCameraUsageDescription</key> <string>Your own description of the purpose</string>

7. What are pure components in React Native?

#### **Answer**

A React component can be viewed as pure on the chance that it renders a similar output for a similar state and props. Class components that extend the React are PureComponent class that is treated as pure components.

8. Can we combine Android or iOS code in react native?

#### **Answer**

Yes, we can combine Android or iOS code in react native. React Native helps to smoothly combines the components written in Java, Objective-C or Swift.

9. Write a program to display "Hello World" in react native?

#### **Answer**

Create a simple "Hello World" app by using modifying App.js file of FirstApp. Save the utility and reload by way of in reality pressing twice "R" or Ctrl+M (Reload).

#### **Example**

import React, {Component} from 'react';

import {Platform, StyleSheet, Text, View} from 'react-native';

type Props = {};

export default class App extends Component<Props> {

render() {

return (

      <View>

        <Text>Hello WorldText>

     View>

    );

  }

}

10. What is the use of the arrow function?

#### **Answer**

The use of Arrow functions in React Native is to help reduce your application's memory consumption. It is done by reducing the CPU time required to iterate the over loops to generate the components necessary for your lists.

Here’s an example of Arrow function in render:  
class Foo extends Component {  
   handleClick() {  
   console.log('Click happened');  
}  
render() {  
   return <button onClick={() => this.handleClick()}>Click Me</button>;  
}  
}

11. What is the difference between state and props?

#### **Answer**

| **State** | **Props** |
| --- | --- |
| States are mutual | Props are immutable |
| The state is set and updated by the object. | In props, you can pass properties from parent components. |
| The state can be modified | Props can’t be modified. |

12. What is the difference between ShadowDOM and VirtualDOM?

#### **Answer**

| **ShadowDOM** | **VirtualDOM** |
| --- | --- |
| Shadow DOM creates small pieces of the DOM object which has its own, isolated scope as they represent. | Virtual DOM creates a copy of the whole DOM object |
| It is a tool that is used in building the apps and websites that are based on components. | It is a concept of DOM which is being used by Vue.js and React.js |
| It comes in small pieces and doesn’t represent the whole Document Object Model. | It's done to improve the performance of the UI libraries |

13. How do you call a Web API in React Native?

#### **Answer**

Here’s an example to call a web service to React Native.

fetch('http://\*\*someurl\*\*', {  
method: 'POST',  
headers: {  
   'Accept': 'application/json',  
   'Content-Type': 'application/json',  
},  
body: JSON.stringify({  
   username : "admin1",  
   password: "admin13",  
})  
})

14. How to import components in React Native?

#### **Answer**

* Add this code in the component.js file destination file  
  import Title from './src/components/importcomponentdemo';
* Here’s the complete code:  
  import React from 'react';  
  import { AppRegistry } from 'react-native';  
  import App from './src/components/importcomponentdemo';  
  const App = () => (  
     <Title />  
  );  
  AppRegistry.registerComponent('ComponentDemo', () => App);

15. What are the advantages of using React Native?

#### **Answer**

#### **Advantages of using React native**

* Conveniently uses the client as well as server side
* Superbly cost effective and code reuse
* Better code readability because of JSX use
* Easy to integrate with other significant frameworks
* Easy to write UI test cases because of React

16. What is the use of Redux in React Native?

#### **Answer**

Redux is a standalone state management library present in React Native and can be used combined with any framework or library. With the use of Redux, app developers can use one application state as a global state and interact with the state from any react component will be easy.

**NOTE**: If you are looking [Redux Interview Questions](https://www.bestinterviewquestion.com/redux-interview-questions) then you can visit here.

17. What do you mean by interactionManager and why is it important?

#### **Answer**

The interaction Manager works on a long-running schedule after any interactions/animations have been completed. In particular, to run smoothly this allows Javascript animations.

It is important because it creates an interaction 'handle' on animation start, and clearing it upon completion in applications to register animations.

18. How to create stack of screens in react native?

#### **Answer**

import React from 'react';  
import { View, Text } from 'react-native';  
import { createAppContainer } from 'react-navigation';  
import { createStackNavigator } from 'react-navigation-stack';  
class HomePage extends React.Component {  
render() {  
    return (  
       <view>  
         <text>Home Page</text></view>  
    );  
}  
}  
class DetailPage extends React.Component{  
    render(){  
        return (  
           <view>  
              <text>Detail Page</text></view>  
  
        )  
    }  
}  
const AppNavigator = createStackNavigator({  
   Home: HomePage,  
   Details: DetailPage,  
},  
{  
    initialRouteName: 'Home',  
});  
export default createAppContainer(AppNavigator);

19. What does StyleSheet.create do?What are refs in React Native?

#### **Answer**

The Stylesheet.create function in React Native is used for the following:

* It validates the keys and registers them to React
* It creates a StyleSheet style reference from the specified object.
* It allows you to send the style only once through the bridge while referring to all subsequent users through ID.

Short form for “reference”, Refs are used to access underlying DOM nodes or React elements within a React component.

20. What is JSX?

#### **Answer**

JSX stands for **J**ava**S**cript **X**ML

It allows us to write HTML elements in React and place them in the DOM without any createElement() and/orappendChild() methods.JSX also converts HTML tags into react elements and makes it easier to write.

#### **Example**

const myelement = <h1>React Interview Questions</h1>;  
ReactDOM.render(myelement, document).

21. What is AppRegistry react native?

#### **Answer**

It is an entry point of JavaScript which is used to run all the apps to React Native. All the components of the App root must register themselves with **AppRegistry.registerComponent()**, after that the native system will be able to load the bundle regarding the app. After that, they can run the app simply by invoking **AppRegistry.runApplication()**.

22. How to update react native with latest version?

#### **Answer**

It is important to upgrade the existing react-native to the latest version in order to access more featural aspects that involve views, APIs, classes, arrow functions, template string, import and export modules, array destructing and development tools.

**Following steps should need to be performed to update the react-native to the latest version:**

* Update react native, react and Expo package versions for upgrading the expo projects in package.json
* Install the latest SDK version which is compatible with the latest version of react-native in the app.json file.
* Upgrade React Native CLI using the command: react-native upgrade
* Install upgrade helper web tool for upgrading the applications
* Upgrade existing file using the command react-native init
* Perform troubleshoot activity to upgrade with react native CLI

23. What are the new features in React Native 0.61?

#### **Answer**

React Native 0.61 is the latest version of react native which is proficiently able to optimize the features of react-native version 0.60. the latest version of react-native comes with the latest and improved features.

The react native version 0.61 is incorporated with four advanced level features such as:

* React native 0.61 is provided with improved CocoaPods Support
* For dimension updates, new useWindowDimensions Hook feature is provided
* Fest processing and refreshing of the application
* React native is upgraded to 16.9 as it opposes old names for unsafe methods

24. How Axios works in React Native?

#### **Answer**

With the help of Axios, the user can send GET, POST, PUT, and DELETE requests to the REST API and render a response to the application. With the help of Axios, users can interact with the REST API. In general, Axios is a promise-based HTTP client which used by the react-native.

**Axios is associated with a number of features which are enlisted below:**

* It makes XMLHttpRequests from the browser
* From react native framework, it makes Http requests
* It supports react-native API’s
* It provides a client-side feature that protects the application from XSRF.
* It automatically transforms response and request data.

#### **How to use Axios**

* You have to install Axios using npm  
  npm install axios
* After that you have to import this module.  
  import axios from 'axios';

#### **Example**

Using POST method.

axios.post('/login', {  
       username: 'bestinterviewquestion',  
       password: 'admin@123'  
  })  
  .then(function (response) {  
        console.log(response);  
  })  
  .catch(function (error) {  
       console.log(error);  
  });

25. What are controlled and uncontrolled components in react native?

#### **Answer**

| **Controlled Components** | **Uncontrolled Components** |
| --- | --- |
| A controlled component is one that is bound to a value, and the changes inside it will be handled in code by using event-based callbacks. | This is similar to the traditional HTML form inputs, but, here, the form data is handled by the DOM itself. |
| It does not maintain its internal state. | It maintains its internal states. |
| Data is controlled by the parent component. | Here, the data is controlled by the DOM itself. |
| It accepts the current value as a prop. | A ref is used for their current values. |
| Has much efficient control over the form elements and data. | Has less control over the form elements and the data. |

26. How to use firebase in react native?

#### **Answer**

Firebase is a mobile platform that helps you quickly develop high-quality apps, grow your user base and earn more money. This is a tool and infrastructure that users need to build better apps and grow a successful business. In addition, firebase is made up of a complementary feature that users can mix-and-match to fit your needs. There are total give key features of firebase namely authentication, Realtime database, cloud messaging, crash reporting and analytics.

**Getting started with firebase by using following steps-**

* Create a firebase project in the Firebase console
* Retrieve apikey, authDomian, DatabaseURL and storage bucket from firebase console
* Create a new react-native project
* Install firebase from npm
* Add it into the react-native project

Is this helpful?  Yes  No

 14  2

27. Which command is used to run for installing react native?

#### **Answer**

npm install -g react-native-cli

28. How to use typescript in react native?

#### **Answer**

When a developer is working on a project and wants to maintain the project for a long period of time then the user should need TypeScript. In order to use the TypeScript user should need to configure the TypeScript because it gives an option to configure the compiler. Moreover, users can configure the TypeScript as per their requirements and can effectively prioritize the type of errors. Given below are the steps that will enable the developer to use TypeScript in react native-

* Creating a project in react native using-  
  **react-native init myapp --template typescript && node myapp/setup.js && cd myapp**
* Set up and configure TSlint  
  **npm install --save-dev tslint tslint-eslint-rules tslint-react tslint-config-prettier**
* Setting up Jest  
  **npm install --save-dev ts-jest**
* Setting up an enzyme  
  **npm install --save-dev enzyme enzyme-adapter-react-16 react-dom @types/enzyme @types/enzyme-adapter-react-16**
* Setting up a test library of react-native  
  **npm install --save-dev react-native-testing-library**

29. List the most common commands used in React Native?

#### **Answer**

Here you will discover a list of fundamental instructions to start creating iOS and Android apps the use of React Native.

#### **Example**

react-native init PROJECTNAME  
react-native run-android  
react-native run-ios  
watchman watch-del-all  
react-native link LIBRARYNAME

30. What is State and how is it used in React Native?

#### **Answer**

The state is used to control the components. State also allows the variable data to get stored in it. As states are mutable, they can change the values at any point in time.

In this example, we are creating the Text component with the help of state data. Content in the text component can be updated anytime by clicking on it. However, the state needs to be updated by the event “**onPress**”.

#### **Example**

import React, {Component} from 'react';      
import { Text, View } from 'react-native';      
export default class App extends Component {      
      state = {  
           myState: 'React Native Interview Questions'  
      }  
updateState = () => this.setState({myState: 'React Native Interview Questions and Answers'})  
render() {  
     return (  
                 
               {this.state.myState}       
            
     );  
} }

31. Why react native is better than native?

#### **Answer**

React Native is better than native because of React native as it is a faster framework and allows the developers to create mobile applications that further support iOS and Android. The react native increases the speed of mobile development effectively and it is quite easier for the developers to maintain the mobile applications which are developed to react-native. It is easier for the developer to identify the bug in the applications which are developed in react native.

32. What is Gesture Responder System in React Native?

#### **Answer**

**Gesture Responder System** is a system that manages the lifecycle of the applications which are built in React- Native framework. The applications automatically determine the intention of the users through touch. For example, the application determines slide on the widget, touch is scrolling or tapping. The Gesture Responder System allows the components to negotiate these touch interactions without integrating any sought of knowledge regarding their parent component and child component. React Native gesture responder system is the most powerful system, as it uses standard library to detect gestures such as scroll distance, single tap, double-tap, single tap confirmed and pinch distance.

33. How to store data in AsyncStorage in react native?

#### **Answer**

When developers develop applications, they need to store information permanently for an application. The react native applications use the database to remember all the information which is related to the applications and their respective users. React Native uses AsyncStorage for data storing purpose. AsyncStorage is a simple, unencrypted synchronous, persistent, key-value storage system that is global to the application. On iOS, AsyncStorage is backed by native code that stores small values in the serialized dictionary and large values in separate files. On Android, AsyncStorage will use either RocksDB or SQLite based on which is available.

##### **Four steps are needed to understand how to use AsyncStorage to React Native-**

* Don’t need to install any extra library. By default, it comes with React Native. Import {AsyncStorage} from 'react-native';
* AsyncStorage uses key-value pairs for saving the data for example- AsyncStorage.setItem ('myKey', myValue);
* To load the saved data run given the command- AsyncStorage.getItem('myKey'). then((myValue)=> {this.setState ({'myKey': myValue});});
* Since loading data is a time-consuming task, it is designed to be an asynchronous operation. So getItem returned the promise, which will invoke the call back function when the read operation is completed.

34. How to load data from server in React Native?

#### **Answer**

React Native gives the Fetch API which offers networking needs. React Native makes use of componentDidMount lifecycle method to load the records from server.  
fetch('https://bestinterviewquestion.com/menu.json')

35. What is the difference between FlatList and ScrollView?

#### **Answer**

| **FlatList** | **ScrollView** |
| --- | --- |
| It renders all the child components at once without impacting the performance. | It renders all the child components at once. However, it decreases the performance. |
| It provides header and footer support. | It does not provide any support for the header and footer. |
| It provides multiple column support, infinite scroll loading, and nth number of features. | It provides limited features. |
| It gives horizontal mode as an optional feature. | It only allows vertical mode. |
| It allows configuring viewability callbacks. | You can’t configure callbacks in ScrollView. |

36. What are the advantages of native apps over hybrid apps?

#### **Answer**

##### **Here are some of the advantages of native apps over hybrid apps:**

* Native app development has fewer dependencies for bugs to occur.
* In the native app, you are not relying on cross-platform tools.
* The processing speed of the native app is faster than the hybrid app.
* It provides great designs and user experience than a hybrid app.

37. What are hooks in react native?

#### **Answer**

Hooks are a new enhancement in React 16.8. They allow to utilize the state and other react features without involving the class. The major agenda behind using the hooks are to handle the side effects in the functional components of react.

38. What is the difference between Hot Reloading and Live Reloading in React Native?

#### **Answer**

| **Hot Reloading** | **Live Reloading** |
| --- | --- |
| It only refreshes the files that were changed without losing the state of the app. | It reloads or refreshes the entire app when a file changes. |
| It displays the code changes according to new code changes without restarting the app from start. | It requires the restart of the app in case of any change in the code |

39. What is fabric in react native?

#### **Answer**

Fabric is the latest architecture of React Native which has been proposed by the company to develop the close and better user experience of mobile apps than the native apps.

40. What are the difference between a component and an element in React.

#### **Answer**

**React Component**: A class or function which optionally accepts inputs and returns a React element via JSX.

**React Element**: It describes what the user wants to be displayed on the screen, Simplifying it, the react element is an object representation for some UI.

41. Is it possible to use the same code base for both Android and iOS in React Native?

#### **Answer**

Yes, it’s entirely possible. App developers use the same code base for iOS and Android as Reacts entirely takes care of the native components translations part. For example, a React Native ScrollView uses ScrollView on Android and native UiScrollView on IOS.

42. What are the disadvantages of React Native?

#### **Answer**

##### **The React Native has following limitations or cons**

* Security risks
* Single threaded interface
* Poor memory management
* Doesn’t support all native APIs
* Dependable on third-party libraries
* Not equivalent to true native apps written entirely on Java or Swift/Obj-C.

43. Is React Native a Mobile App?

#### **Answer**

Yes, React Native complies with ALL the requires of a native mobile app using native app components. It's neither a mobile web app nor a hybrid mobile app that uses WebView to run the HTML5 app successfully. It builds a real mobile app, utterly indistinguishable from apps built using Java or Objective-C.

44. What do you mean by Dispatch in react native?

#### **Answer**

Dispatch is a feature of the Redux store. You call store.dispatch to dispatch an action. This is the only way to set off a kingdom change. With React Redux, your factors never get admission to the keep without delay - connect does it for you.

45. How Virtual Dom works in React Native?

#### **Answer**

**Virtual DOM** is a copy of the real DOM. This node tree lists the elements, their attributions, contents, and properties also. Whenever any underlying data changes in React Native, the entire UI will be re-rendered in Virtual DOM representation. Then, the difference between previous DOM representation and virtual DOM will be counted. After that, the real DOM will be updated.

46. How do you add react navigation to react native?

#### **Answer**

You can install react navigation by the following command:  
  
yarn add react-navigation  
or npm install react-navigation

47. What are the other modules need for react native navigation?

#### **Answer**

##### **You have to install following modules in addition to react-navigation:**

* react-native-gesture-handler
* react-native-reanimated
* react-native-screens
* react-native-safe-area-context
* react-navigation-stack

48. How to create stackNavigator in react native?

#### **Answer**

const AppNavigator = createStackNavigator({  
      Home: {  
          screen: HomeScreen,  
      },  
});

| Quick Facts About React Native | |
| --- | --- |
| **When did React Native Initial release?** | 26th March 2015 |
| **What is the latest version of React Native?** | React Native v0.64.0 was released on 13th March 2021 |
| **React Native is Created By** | Facebook |
| **What language does React Native use?** | It is based on React js, which is written in JavaScript. |

#### **Conclusion**

In the end, we know an interview that has a Javascript word in it, is not very easy to crack. By going through various questions published on different websites wouldn’t open the door for you. Here are some bonus tips for you which you should also keep in mind-

* Pick a programming language and make a good command of it.
* Don’t mix concepts of ReactJS with React Native.
* Practice regularly some coding challenges and mock interviews.
* You should know different methods to solve a problem.
* Take time to think before answering, don’t hurry.

### **1. What are the advantages of using React Native?**

****Sample answer****:

Since its launch in 2015, React Native has built a reputation as a reliable and effective JavaScript framework. Some of its key strengths include:

* ****Cross-platform compatibility:**** Most of the code is cross-platform, meaning developers only have to create one app rather than two separate apps for both iOS and Android
* ****Real-time feedback:**** React Native offers a ‘hot reloading’ feature where developers can immediately view the changes they’ve made in a separate preview window
* ****Flexible user interface:**** React Native’s interface is slick and makes it easy for multiple developers to work on a project together
* ****Third-party plugins:**** React Native is compatible with many third-party plugins that can be used to support and improve the app development process
* ****Community****: As a popular open-source framework, React Native has a large community of developers that exchange knowledge

### **2. What are the key differences between React Native and ReactJS?**

****Sample answer****:

React Native is used to develop mobile apps for iOS and Android, whereas ReactJS is used to build web apps in a web browser.

Both use reusable JavaScript XML components, but the syntax varies: while React Native uses app-view components like <View> and <Text>, ReactJS uses HTML tags such as <div> and <h1>.

### **3. How do native apps differ from hybrid apps?**

****Sample answer****:

Hybrid apps are developed to be used across all platforms, whereas native apps are developed for a particular platform. React Native is used for the development of hybrid apps.

While hybrid apps are faster to develop and typically require less maintenance than native apps, they may perform slightly worse than their native counterparts.

### **4. What products and apps is React Native best used for?**

****Sample answer****:

React Native is a great option for developing a hybrid app that does not require extremely high performance.

Cross-platform compatibility means development teams can save lots of time when using React Native compared to a native framework.

However, it might not be suitable when designing complex apps or if developers aren’t already well-versed in React code.

### **5. What are the core components of React Native?**

****Sample answer****:

Components are the building blocks of React Native; when combined, they make up the app as a whole. Some of the most common components are:

* ****View,**** used to display the entire app layout
* ****Text,**** used to display text
* ****TextInput****, used to input text
* ****ScrollView,****used to insert a scrolling container
* ****StyleSheet,**** used to insert style objects
* ****Image,**** used to render images
* ****Button,**** used to insert buttons

### **6. What is component-driven development?**

****Sample answer****:

Component-driven development (CDD) is a development methodology where the build process is anchored around components rather than objects. Components are loosely coupled and each one serves its own purpose.

When put together, components (buttons, navigation bars, images) form the program as a whole. React Native is a component-driven framework.

### **7. What is the role of props in React Native?**

****Sample answer****:

Props provide properties to components inserted in a program, which makes components modifiable and customizable. For example, the same component might be used in different parts of an app. When we use props, we can alter the component’s appearance or behavior.

### **8. What is the role of AsyncStorage in React Native?**

****Sample answer****:

AsyncStorage is React Native’s key-value, unencrypted storage module that allows developers to store data for offline use. Typically, it’s used to store data when an app is not linked to a cloud service, or when specific features require data storage.

### **9. What is the role of Flexbox in React Native?**

****Sample answer****:

In React Native apps, Flexbox is used to provide a consistent layout across different screen types. The Flexbox algorithm helps to structure the positioning of different components and create a responsive UI for the end user.

### **10. What is the state in React Native?**

****Sample answer****:

In React Native, the state refers to information about a property at a given time. Unlike props, the state is mutable; it can change. Typically, this will occur when a user interacts with the component.

For example, if your app had a filling form that users are invited to complete, the state of that component would change when the user types something in.

### **11. How do you import components in React Native?**

****Sample answer****:

In React Native, you can import components from scratch, or also import ready-made ones from another file.

To import a component, you need to type <import { Component } from 'react-native'>, changing the word in brackets depending on the type of component you want to import.

### **12. What coding languages are compatible with React Native?**

****Sample answer****:

While React Native is generally used with JavaScript, compatibility with other coding languages, including Python, C++, and C, is also possible through the framework’s Java Native Interface (JNI).

### **13. What engine does React Native use?**

****Sample answer****:

In React Native, JavaScript code runs through two engines:

1. ****JavaScriptCore**** is used on iOS simulators and Android emulators; virtually all operations run through this engine
2. ****V8****is used when Chrome debugging is being performed

## **14 intermediate React Native interview questions**

You can use these intermediate React Native interview questions for mid-level React Native developer roles, where the candidate already has a few years of experience.

### **14. What are the main disadvantages of using React Native?**

****Sample answer****:

As with any software framework, React Native has its fair share of drawbacks. These include:

* ****Non-nativity:**** React Native isn’t a native solution, which means its apps may be slightly slower than native ones
* ****Debugging issues:**** React Native is built using Javascript, Objective-C, Java, and C or C ++, which can make debugging more difficult
* ****Memory management:**** Limitations on memory mean React Native is not suited to developing computation-intensive apps
* ****Low security:**** React Native’s open-source design leaves apps more exposed to threats, which is especially dangerous for apps containing sensitive information, such as banking services
* ****Learning curve:**** React Native is one of the more challenging software frameworks to learn, especially for junior developers

### **15. What are the main performance issues in React Native and what causes them?**

****Sample answer****:

Some of the most common performance issues in React Native include:

* ****High CPU usage:**** Offloading complex functions to the JavaScript thread can cause performance issues
* ****Memory leak:****Information can be lost in the Bridge during the transfer from the Primary to the React Native domains, especially in Android apps
* ****Slow navigation:**** Multiple thread bridging can also cause slower navigation times

### **16. How can app performance be optimized in React Native?**

****Sample answer****:

There are several techniques that we can use to optimize an app’s performance in React Native, such as:

* Remove all console statements
* Resize and scale down images
* Cache images internally
* Compress or convert raw JSON data
* Use code splitting for large lists
* Schedule animations
* Remove unnecessary libraries and features

#### **17. How is React Native code processed to display the final output on the screen?**

****Sample answer****:

The process for rendering code in React Native is the following:

1. When the app is opened, the main thread (or UI thread) starts execution by loading JavaScript bundles
2. Once the JavaScript code has been loaded successfully, the main thread sends it to the second JS thread where more calculations are performed
3. When React Native starts rendering, the reconciler algorithm generates a virtual DOM or layout, which is then sent to a third shadow thread
4. The shadow thread calculates a new DOM and sends the layout characteristics to the main UI thread
5. The UI thread then renders the received DOM for display on the mobile app

### **18. What is the role of the bridge in React Native?**

****Sample answer****:

The bridge acts as a transport layer between JavaScript and Native modules. In the rendering process:

* The bridge first receives the user response to open the app from the Native module
* It then passes the serialized payload to the JavaScript module
* Once the event has been processed and a virtual DOM has been generated in the JavaScript module, the bridge receives the serialized batched response
* The bridge passes the serialized batched response to the Native module for final rendering

### **19. What is the interaction manager in React Native?**

****Sample answer****:

In React Native, the interaction manager is used to defer the execution of a function until a specified ‘interaction’ has been completed.

This is important as React Native is single-threaded in nature, meaning queued animations on the UI can become congested. The interaction manager helps with this issue and ensures that animations run smoothly in a scheduled manner.

### **20. What is the role of fabric in React Native?**

****Sample answer****:

Fabric is a modern type of architecture that was first created in 2018 and aims to address some of React Native’s performance issues. Fabric modernizes the framework’s rendering layer by allowing specified priority tasks to be executed synchronously and, therefore, quicker.

### **21. How do you debug React Native apps?**

****Sample answer****:

In React Native, we can perform debugging in two ways:

1. Remote debugging

* For iOS, open the menu and select ‘Debug remotely’
* For Android, open the menu and select ‘Debug remotely’
* The ‘Debug remotely’ button will launch the Chrome Developer Tools’ Debugger tool

1. In-app debugging

* For iOS, run the <react-native log-ios> command
* For Android, run the <react-native log-android> command
* These commands will launch the in-app debugging tools in the relevant operating system

### **22. How do you create basic text input in React Native?**

****Sample answer****:

The insertion of basic text in React Native apps is handled by the Text and TextInput components. TextInput allows users to type on the app. We can implement it using the following syntax: <import { Text, TextInput, View } from 'react-native'>.

### **23. How can you optimize the performance of images in React Native?**

****Sample answer****:

There are several useful tricks for optimizing the performance of images in React Native. These include:

* Using image caching tools
* Using PNG or WEBP formats rather than JPEG
* Using smaller images
* Reducing the number of renders

### **24. What is the role of timers in a React Native app?**

****Sample answer****:

In React Native, timers allow developers to manipulate the order in which events in a program occur. There are four different types of timers, each one serving a different purpose:

1. ****Timeout**** implements a delay
2. ****Interval****allows repeat actions to occur at given intervals
3. ****Immediate****allows actions to occur as soon as possible
4. ****Animation****allows animations to display when the program is ready to render frames

### **25. What is the role of hooks in React Native?**

****Sample answer****:

Hooks allow developers to ‘hook into’ existing components and access their state and lifestyle features. Previously, these would not be accessible for use elsewhere. With hooks, developers can now tap into a component’s state and lifestyle features without having to write a new class.

### **26. How do you create a basic button in React Native?**

****Sample answer****:

We can create basic buttons using the following syntax: <import { View, Button, StyleSheet } from "react-native">. Basic buttons support a minimal level of customization and can be modified using TouchableOpacity or TouchableWithoutFeedback.

### **27. What is the role of fast refresh in React Native?**

****Sample answer****:

Fast refresh allows developers to get near-instant feedback on recent changes in their app. Once ‘Enable fast refresh’ in the developer menu is toggled, any new edits in the program become visible within a few seconds for an easy evaluation.

## **13 advanced React Native interview questions**

You can use the advanced interview questions below when hiring a senior React Native developer with several years of experience.

### **28. How can sensitive data be stored securely in React Native?**

****Sample answer****:

Most React Native data is stored in Async Storage. As an unencrypted, local form of storage, it’s not suitable for storing sensitive data such as tokens and passwords.

Alternatively, React Native Keychain offers a secure form of storage that also works similarly to Async Storage. For iOS, Keychain storage can be used to protect sensitive data, while Android developers can use Facebook Conceal and Android Keystone.

### **29. How can you resolve common React Native performance issues?**

****Sample answer****:

When performance issues occur, there are several solutions available to developers. For example, here are some solutions for a few common performance issues:

* ****High CPU usage:**** Optimizing apps by compressing data, cutting out wasteful renders, and using cache storage will improve performance speed
* ****Memory leak:**** Memory leak can be avoided by debugging, avoiding the use of console statements, and regularly checking code for inconsistencies
* ****Slow navigation:**** Using React Navigation instead of other navigation tools like Navigator or NavigationExperimental will help soothe navigational issues

### **30. What are some differences between using React Native for iOS and for Android?**

****Sample answer****:

Around 85% of React Native code is cross-platform, which means that most processes are the same on both iOS and Android. However, there are a few minor differences. These include:

* iOS development uses Mac and Xcode, whereas Android development draws on Android SDK and an emulator
* Third-party plugins that don’t offer native functionality will need to be used differently
* The bridging process can be slightly different when developing complex apps

### **31. How do you ensure animations run smoothly in React Native apps?**

****Sample answer****:

Several steps can be taken to optimize animations in React Native. These include:

* ****Use lazy loading**** so components are only rendered when in use
* ****Remove animated values from the state**** to avoid unnecessary overhead
* ****Use shouldComponentUpdate**** to fast-track the rendering process
* ****Use useNativeDriver for Android**** to transfer all the animating work to the native layer

### **32. What steps would you take in React Native if you have an app that crashes continually?**

****Sample answer****:

Use a third-party error reporting integration to pull up an error report and further diagnose the bug. These plugins help collect, organize, and analyze crash reports and also provide quick fixes so the app can get back up and running. Popular error reporting plugins include:

* Bugsnag
* Crashlytics
* Sentry
* TestFairy
* Rollbar

### **33. How can Redux be implemented in React Native?**

****Sample answer****:

Redux is a useful state management tool. It can be implemented by following these steps:

1. Install the necessary Redux packages online
2. Create a Redux folder at the root of the app
3. Within the Redux folder, add three files: actions.js, reducer.js, and store.js
4. In the new reducer.js file, import <combineReducers> from 'redux'
5. In the store.js file, import <createStore> from 'redux' to create the store
6. Import the store to the React Native app

### **34. What is the role of TouchableOpacity in React Native?**

****Sample answer****:

In React Native, TouchableOpacity is a wrapper used to change the transparency of a button. When used on a button, opacity reduces in response to touch, allowing users to see the background whenever they press it.

### **35. What are the differences between Flexbox in browser and in React Native?**

****Sample answer****:

Flexbox generally works in the same way in React Native as it does in CSS on the web. However, there are a few minor differences in the values. These include:

* The default value in React Native is column, whereas the default value for CSS is row
* The default value in React Native is flex-start, whereas the default value for CSS is stretch
* The default value in React Native is 0, whereas the default value for CSS is 1

### **36. What steps can you take to resolve persistent memory leak issues?**

****Sample answer****:

A memory leak occurs when memory that is no longer needed by an app remains in the app rather than being returned to the operating system. This is one of the most common causes of performance issues.

In theory, memory management is handled automatically by the garbage collector. However, this process is still prone to errors. Debugging tools can be used to identify memory leak issues. Some of the most common causes for issues are:

* Timers and listeners in componentDidMount
* Inline styles
* Closure scope leaks
* The use of console.log

Debugging can identify the root of the memory leak; once it has been removed, the issue should be resolved.

### **37. How can you optimize FlatList items in React Native?**

****Sample answer****:

There are several techniques for optimizing the performance of FlatList items. For example, we can:

* Avoid using 1080P HD images
* Optimize the maxToRenderPerBatch prop
* Use the getItemLayout prop
* Use the keyExtractor prop
* Use fewer views
* Optimize the windowSize prop

### **38. How are hot reloading and live reloading in React Native different?**

****Sample answer****:

Live reloading in React Native refreshes the entire app when a file changes, whereas hot reloading only refreshes the files that were changed.

When hot reloading is used on an app, the state remains the same and the developer is returned to the page they started on. The opposite is true for live reloading.

### **39. When would you use ScrollView over FlatList and vice versa?**

****Sample answer****:

ScrollView loads all data items on one screen for scrolling purposes. All the data is stored on RAM, which can cause performance issues for large amounts of data.

FlatList only displays items that are currently shown on the screen (10 by default), thus sidestepping any performance issues.

Therefore, it is best to use FlatList for large datasets, whereas ScrollView can be used for smaller datasets.

### **40. How are vector icons used in React Native?**

****Sample answer****:

Vectors are an integral part of app design in React Native. They can initially be installed by running the following command: <npm install react-native-vector-icons>. Selected vector icons can then be imported into the React Native app for use.

### **41. When should setNativeProps be used in React Native?**

****Sample answer****:

In React Native, setNativeProps is used to change a component directly on a DOM node rather than in the app’s state. While this can help to solve issues, it also makes code more difficult to work with in the future.

It’s considered a backup solution to performance issues linked to animations, and should only be used if setState and shouldComponent fail to resolve the problem.

## **At which stage of the hiring process should you use React Native interview questions?**

Conducting an effective interview is an important step when [hiring a new app developer](https://www.testgorilla.com/blog/6-steps-to-take-before-you-hire-app-developers/)—but several other stages should come before that. The technical nature of React Native means that it’s essential to test the candidate’s proficiency first.

For example, you can create a React Native timed task for candidates to complete, such as to identify problems with a sample piece of code. Tailor the task to your business needs where possible.

*Alternatively, you can use TestGorilla’s [React Native test](https://www.testgorilla.com/test-library/programming-skills-tests/react-native-test/) to quickly evaluate candidates’ skills and knowledge, and shortlist the best applicants, who you can then invite to an interview. This enables you to simplify and optimize the screening stage, and allows you to evaluate a candidate’s expertise even if you [lack coding knowledge yourself](https://www.testgorilla.com/blog/how-to-hire-a-javascript-developer-even-if-youre-not-a-coder/).*

Performing a skills assessment early in the selection process will help you identify the most skilled candidates. You can then spend more time on shortlisted applicants: instead of conducting tens of interviews and coding assessments, you can concentrate only on a select few.

This allows you to confidently compare applicants’ experience against your business needs to gauge whether they will be a good fit.

## **Finding your next React Native developer**

By using the interview questions above alongside skills assessments, you can identify the best candidates for your company. With the right React Native developer on your team, you’ll be able to compete in the constantly evolving and extremely lucrative mobile app market, now estimated to be worth [$700 billion](https://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/" \t "https://www.testgorilla.com/blog/react-native-interview-questions/_blank).

Remember to adapt your questions to the right experience level and to your business requirements. The selection process isn’t about finding the best all-round React Native developer, but the one who’s the right match for your company’s needs.

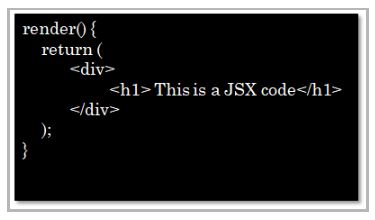
TestGorilla’s [React Native test](https://www.testgorilla.com/test-library/programming-skills-tests/react-native-test/) can help you find the best React Native developers fast and bias-free.  [Get started for free today](https://app.testgorilla.com/register/customer?utm_plan=get-started) and start making better hiring decisions.

### ****1. What are the features of React?****

|  |  |
| --- | --- |
| IMG_256 | **JSX:** JSX is a syntax extension to JavaScript. It is used with React to describe what the user interface should look like. By using JSX, we can write [HTML](https://www.simplilearn.com/tutorials/html-tutorial/what-is-html" \o "HTML" \t "https://www.simplilearn.com/tutorials/reactjs-tutorial/_blank) structures in the same file that contains [JavaScript](https://www.simplilearn.com/tutorials/javascript-tutorial/introduction-to-javascript" \o "JavaScript" \t "https://www.simplilearn.com/tutorials/reactjs-tutorial/_blank) code. |
| IMG_257 | **Components:**[Components](https://www.simplilearn.com/tutorials/reactjs-tutorial/reactjs-components" \o "Components" \t "https://www.simplilearn.com/tutorials/reactjs-tutorial/_blank) are the building blocks of any React application, and a single app usually consists of multiple components. It splits the user interface into independent, reusable parts that can be processed separately. |
| IMG_258 | **Virtual DOM:**React keeps a lightweight representation of the real DOM in the memory, and that is known as the virtual DOM. When the state of an object changes, virtual DOM changes only that object in the real DOM, rather than updating all the objects. |
| IMG_259 | **One-way data-binding:**React’s one-way [data binding](https://www.simplilearn.com/tutorials/angular-tutorial/angular-data-binding" \o "data binding" \t "https://www.simplilearn.com/tutorials/reactjs-tutorial/_blank) keeps everything modular and fast. A unidirectional data flow means that when designing a React app, you often nest child components within parent components. |
| IMG_260 | **High performance:**React updates only those components that have changed, rather than updating all the components at once. This results in much faster web applications. |

### ****2. What is JSX?****

JSX is a syntax extension of JavaScript. It is used with React to describe what the user interface should look like. By using JSX, we can write HTML structures in the same file that contains JavaScript code.



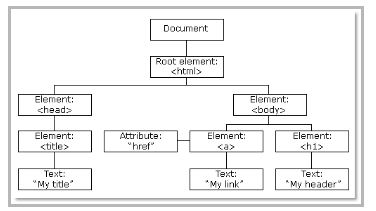
### ****3. Can web browsers read JSX directly?****

* Web browsers cannot read JSX directly. This is because they are built to only read regular JS objects and JSX is not a regular JavaScript object
* For a web browser to read a JSX file, the file needs to be transformed into a regular JavaScript object. For this, we use Babel

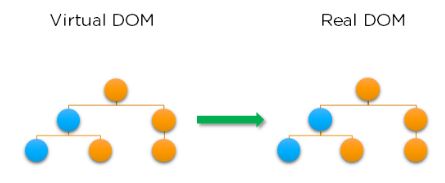


### ****4. What is the virtual DOM?****

DOM stands for Document Object Model. The DOM represents an HTML document with a logical tree structure. Each branch of the tree ends in a node, and each node contains objects.



React keeps a lightweight representation of the real DOM in the memory, and that is known as the virtual DOM. When the state of an object changes, the virtual DOM changes only that object in the real DOM, rather than updating all the objects. The following are some of the most frequently asked react interview questions.



### ****5. Why use React instead of other frameworks, like Angular?****

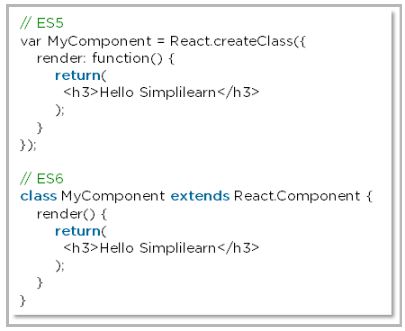
|  |  |
| --- | --- |
| IMG_267 | **Easy creation of dynamic applications:** React makes it easier to create dynamic web applications because it provides less coding and provides more functionality, whereas, with JavaScript applications, code tends to get complex very quickly. |
| IMG_268 | **Improved performance:**React uses virtual DOM, which makes web applications perform faster. Virtual DOM compares its previous state and updates only those components in the real DOM, whose states have changed, rather than updating all the components — like conventional web applications. |
| IMG_269 | **Reusable components:**Components are the building blocks of any React application, and a single app usually consists of multiple components. These components have their own logic and controls, and they can be reused through the application, which, in turn, dramatically reduces the development time of an application. |
| IMG_270 | **Unidirectional data flow:**React follows a unidirectional data flow. This means that when designing a React app, we often nest child components within parent components. And since the data flows in a single direction, it becomes easier to debug errors and know where the problem occurs in an application at the moment. |
| IMG_271 | **Dedicated tools for easy debugging:**Facebook has released a chrome extension that we can use to debug React applications. This makes the process of debugging React to web applications faster and easier. |

### 6. What is the difference between the ES6 and ES5 standards?

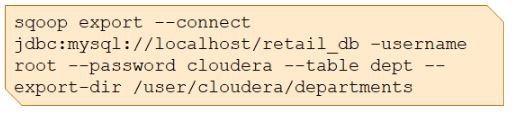
This is one of the most frequently asked react interview questions.

These are the few instances where ES6 syntax has changed from ES5 syntax:

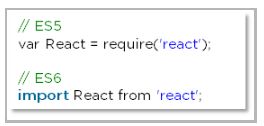
* **Components and Function**



* **exports vs export**



* **require vs import**



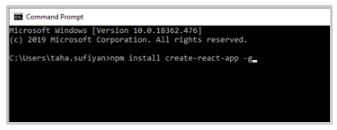
### ****7. How do you create a React app?****

These are the steps for creating a React app:

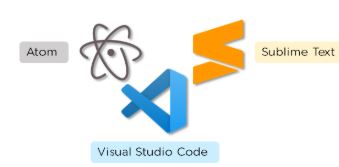
* Install [NodeJS](https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs" \o "NodeJS" \t "https://www.simplilearn.com/tutorials/reactjs-tutorial/_blank) on the computer because we need npm to install the React library. Npm is the node package manager that contains many JavaScript libraries, including React.



* Install the **create-react-app** package using the command prompt or terminal.



* Install a text editor of your choice, like VS Code or Sublime Text.



We have put together a set of [Node.js interview questions](https://www.simplilearn.com/tutorials/nodejs-tutorial/nodejs-interview-questions" \o "Node.js interview questions" \t "https://www.simplilearn.com/tutorials/reactjs-tutorial/_blank) in case you would like to explore them.Please note, This is one of the most frequently asked react interview questions.

### ****8. What is an event in React?****

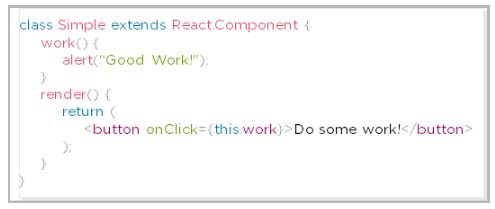
An event is an action that a user or system may trigger, such as pressing a key, a mouse click, etc.

* React events are named using camelCase, rather than lowercase in HTML.
* With JSX, you pass a function as the event handler, rather than a string in HTML.

|  |
| --- |
| <Button onPress={lightItUp} /> |

### ****9. How do you create an event in React?****

A React event can be created by doing the following:



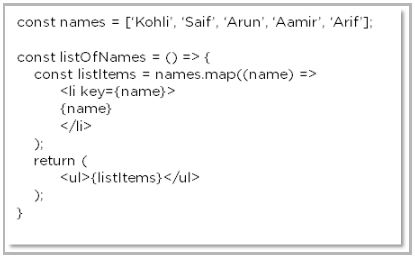
### ****10. What are synthetic events in React?****

* Synthetic events combine the response of different browser's native events into one API, ensuring that the events are consistent across different browsers.
* The application is consistent regardless of the browser it is running in. Here, **preventDefault**is a synthetic event.



### ****11. Explain how lists work in React****

* We create lists in React as we do in regular JavaScript. Lists display data in an ordered format
* The traversal of lists is done using the map() function



### ****12. Why is there a need for using keys in Lists?****

Keys are very important in lists for the following reasons:

* A key is a unique identifier and it is used to identify which items have changed, been updated or deleted from the lists
* It also helps to determine which components need to be re-rendered instead of re-rendering all the components every time. Therefore, it increases performance, as only the updated components are re-rendered

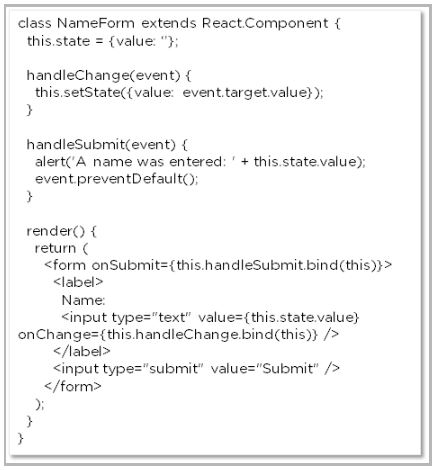
### ****13. What are forms in React?****

React employs forms to enable users to interact with web applications.

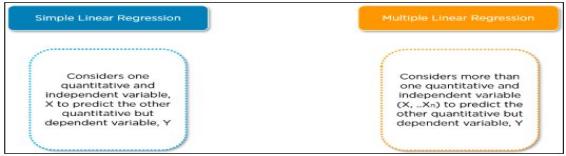
* Using forms, users can interact with the application and enter the required information whenever needed. Form contain certain elements, such as text fields, buttons, checkboxes, radio buttons, etc
* Forms are used for many different tasks such as user authentication, searching, filtering, indexing, etc

### ****14. How do you create forms in React?****

We create forms in React by doing the following:



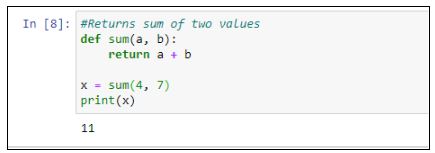
The above code will yield an input field with the label **Name**and a submit button. It will also alert the user when the submit button is pressed.



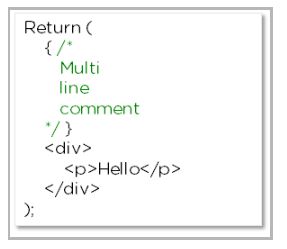
### ****15. How do you write comments in React?****

There are basically two ways in which we can write comments:

* Single-line comments

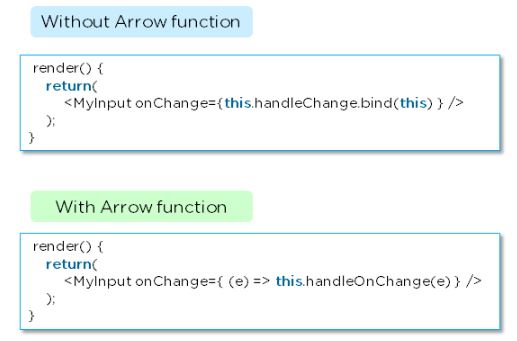


* Multi-line comments



### ****16. What is an arrow function and how is it used in React?****

* An arrow function is a short way of writing a function to React.
* It is unnecessary to bind **‘this’**inside the constructor when using an arrow function. This prevents bugs caused by the use of **‘this’**in React callbacks.



### ****17. How is React different from React Native?****

|  |  |  |
| --- | --- | --- |
|  | **React** | **React Native** |
| Release | 2013 | 2015 |
| Platform | Web | Mobile – Android, iOS |
| HTML | Yes | No |
| CSS | Yes | No |
| Prerequisites | JavaScript, HTML, CSS | React.js |

### ****18. How is React different from Angular?****

|  |  |  |
| --- | --- | --- |
|  | **Angular** | **React** |
| Author | Google | Facebook |
| Architecture | Complete MVC | View layer of MVC |
| DOM | Real DOM | Virtual DOM |
| Data-Binding | Bi-directional | Uni-directional |
| Rendering | Client-Side | Server-Side |
| Performance | Comparatively slow | Faster due to Virtual DOM |

In case you have any doubts about these Basic React interview questions and answers, please leave your questions in the comment section below.

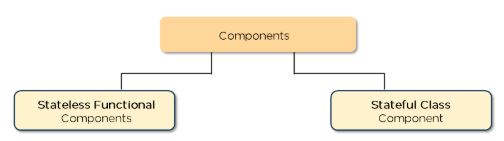
## ****ReactJS Interview Questions on Components****

Here are some React Interview Questions on components.

### ****19. What are the components in React?****

Components are the building blocks of any React application, and a single app usually consists of multiple components. A component is essentially a piece of the user interface. It splits the user interface into independent, reusable parts that can be processed separately.

There are two types of components in React:



* **Functional Components:**These types of components have no state of their own and only contain render methods, and therefore are also called **stateless components**. They may derive data from other components as props (properties).

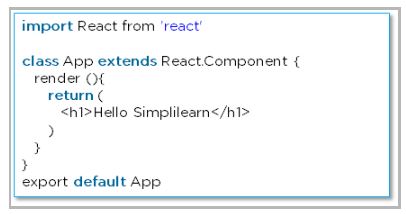
|  |
| --- |
| function Greeting(props) {    return <h1>Welcome to {props.name}</h1>;  } |

* **Class Components:**These types of components can hold and manage their own state and have a separate render method to return JSX on the screen. They are also called Stateful components as they can have a state.

|  |
| --- |
| class Greeting extends React.Component {    render() {      return <h1>Welcome to {this.props.name}</h1>;    }  } |

### ****20. What is the use of render() in React?****

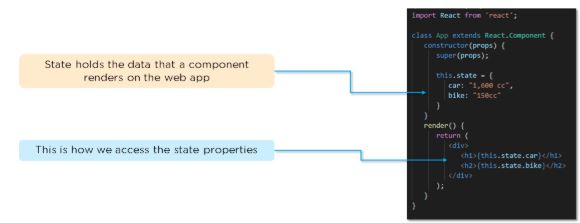
* It is required for each component to have a render() function. This function returns the HTML, which is to be displayed in the component.
* If you need to render more than one element, all of the elements must be inside one parent tag like <div>, <form>.



### ****21. What is a state in React?****

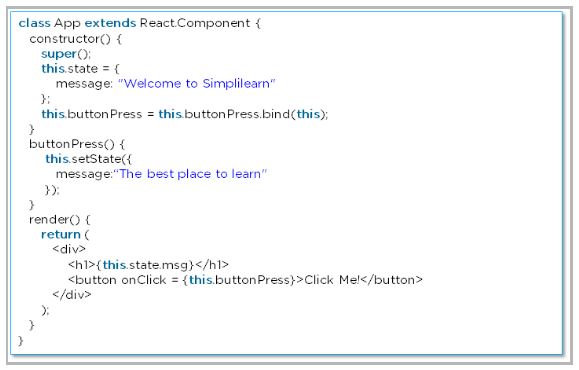
* The state is a built-in React object that is used to contain data or information about the component. The state in a component can change over time, and whenever it changes, the component re-renders.
* The change in state can happen as a response to user action or system-generated events. It determines the behavior of the component and how it will render.

### ****22. How do you implement state in React?****



### ****23. How do you update the state of a component?****

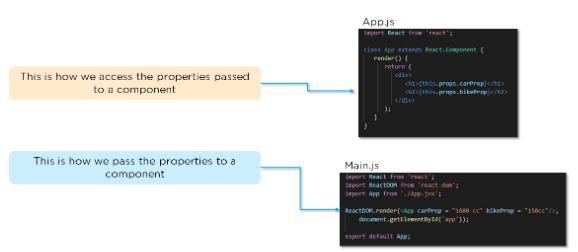
We can update the state of a component by using the built-in **‘setState()’**method:



### ****24. What are props in React?****

* [Props](https://www.simplilearn.com/tutorials/reactjs-tutorial/react-props" \o "Props" \t "https://www.simplilearn.com/tutorials/reactjs-tutorial/_blank) are short for Properties. It is a React built-in object that stores the value of attributes of a tag and works similarly to HTML attributes.
* Props provide a way to pass data from one component to another component. Props are passed to the component in the same way as arguments are passed in a function.

### ****25. How do you pass props between components?****



### ****26. What are the differences between state and props?****

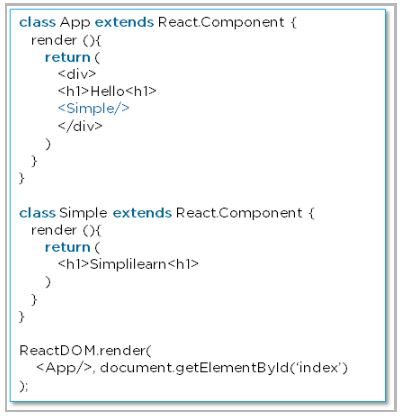
|  |  |  |
| --- | --- | --- |
|  | **State** | **Props** |
| Use | Holds information about the components | Allows to pass data from one component to other components as an argument |
| Mutability | Is mutable | Are immutable |
| Read-Only | Can be changed | Are read-only |
| Child components | Child components cannot access | Child component can access |
| Stateless components | Cannot have state | Can have props |

### ****27. What is a higher-order component in React?****

A higher-order component acts as a container for other components. This helps to keep components simple and enables re-usability. They are generally used when multiple components have to use a common logic.

### ****28. How can you embed two or more components into one?****

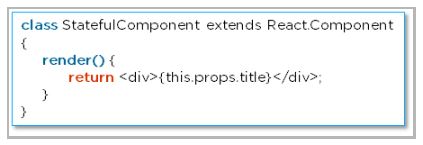
We can embed two or more components into one using this method:



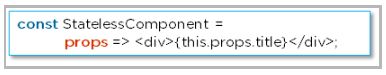
### ****29. What are the differences between class and functional components?****

|  |  |  |
| --- | --- | --- |
|  | **Class Components** | **Functional Components** |
| State | Can hold or manage state | Cannot hold or manage state |
| Simplicity | Complex as compared to the stateless component | Simple and easy to understand |
| Lifecycle methods | Can work with all lifecycle methods | Does not work with any lifecycle method |
| Reusability | Can be reused | Cannot be reused |

* **Class components example:**



* **Functional components example:**



### ****30. Explain the lifecycle methods of components.****

* **getInitialState():** This is executed before the creation of the component.
* **componentDidMount():** Is executed when the component gets rendered and placed on the DOM.
* **shouldComponentUpdate():** Is invoked when a component determines changes to the DOM and returns a “true” or “false” value based on certain conditions.
* **componentDidUpdate():** Is invoked immediately after rendering takes place.
* **componentWillUnmount():** Is invoked immediately before a component is destroyed and unmounted permanently.

So far, if you have any doubts about the above React interview questions and answers, please ask your questions in the section below.

## ****ReactJS Redux Interview Questions****

Here are some ReactJS Interview Questions on the ReactJS Redux concept.

### ****31. What is Redux?****

[Redux](https://www.simplilearn.com/tutorials/reactjs-tutorial/react-with-redux" \o "Redux" \t "https://www.simplilearn.com/tutorials/reactjs-tutorial/_blank) is an open-source, JavaScript library used to manage the application state. React uses Redux to build the user interface. It is a predictable state container for JavaScript applications and is used for the entire application’s state management.

### ****32. What are the components of Redux?****

* **Store:** Holds the state of the application.
* **Action:** The source information for the store.
* **Reducer:** Specifies how the application's state changes in response to actions sent to the store.



### ****33. What is the Flux?****

* Flux is the application architecture that Facebook uses for building web applications. It is a method of handling complex data inside a client-side application and manages how data flows in a React application.



* There is a single source of data (the store) and triggering certain actions is the only way way to update them.The actions call the dispatcher, and then the store is triggered and updated with their own data accordingly.



* When a dispatch has been triggered, and the store updates, it will emit a change event that the views can rerender accordingly.



### ****34. How is Redux different from Flux?****

|  |  |  |
| --- | --- | --- |
| **SN** | **Redux** | **Flux** |
| 1. | Redux is an open-source JavaScript library used to manage application State | Flux is an architecture and not a framework or library |
| 2. | Store’s state is immutable | Store’s state is mutable |
| 3. | Can only have a single-store | Can have multiple stores |
| 4. | Uses the concept of reducer | Uses the concept of the dispatcher |

So far, if you have any doubts about these React interview questions and answers, please leave your questions in the section below.

## ****ReactJS Router Questions****

Here are some ReactJS Interview Questions on React Router concepts.

### ****35. What is React Router?****

React Router is a routing library built on top of React, which is used to create routes in a React application. This is one of the most frequently asked react interview questions.

### ****36. Why do we need to React Router?****

* It maintains consistent structure and behavior and is used to develop single-page web applications.
* Enables multiple views in a single application by defining multiple routes in the React application.

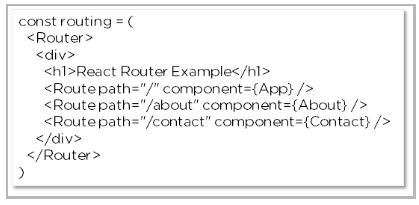
### ****37. How is React routing different from conventional routing?****

|  |  |  |
| --- | --- | --- |
| **SN** | **React Routing** | **Conventional routing** |
| 1. | Single HTML page | Each view is a new HTML file |
| 2. | The user navigates multiple views in the same file | The user navigates multiple files for each view |
| 3. | The page does not refresh since it is a single file | The page refreshes every time user navigates |
| 4. | Improved performance | Slower performance |

### ****38. How do you implement React routing?****

We can implement routing in our React application using this method:

Considering we have the components **App**, **About**, and **Contact** in our application:



Hope you have no doubts about this ReactJS interview questions article, in case of any difficulty, please leave your problems in the section below.

### ****1.  Differentiate between Real DOM and Virtual DOM.****

|  |  |
| --- | --- |
| Real DOM vs Virtual DOM ****Real DOM**** | ****Virtual  DOM**** |
| 1. It updates slow. | 1. It updates faster. |
| 2. Can directly update HTML. | 2. Can’t directly update HTML. |
| 3. Creates a new DOM if element updates. | 3. Updates the JSX if element updates. |
| 4. DOM manipulation is very expensive. | 4. DOM manipulation is very easy. |
| 5. Too much of memory wastage. | 5. No memory wastage. |

### ****2. What is React?****

* React is a front-end JavaScript library developed by Facebook in 2011.
* It follows the component based approach which helps in building reusable UI components.
* It is used for developing complex and interactive web and mobile UI.
* Even though it was open-sourced only in 2015, it has one of the largest communities supporting it.

### ****3. What are the features of React?****

Major features of React are listed below:

[](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=curriculum" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)

### **[React JS Certification Training Course](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=curriculum" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

* *[Instructor-led Sessions](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=curriculum" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)*
* *[Assessments](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=curriculum" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)*
* *[Assignments](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=curriculum" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)*
* *[Lifetime Access](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=curriculum" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)*

[Explore Curriculum](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=curriculum" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)

1. It uses the ****virtual DOM**** instead of the real DOM.
2. It uses ****server-side rendering****.
3. It follows ****uni-directional data flow**** or data binding.

### ****4. List some of the major advantages of React.****

Some of the major advantages of React are:

1. It increases the application’s performance
2. It can be conveniently used on the client as well as server side
3. Because of JSX, code’s readability increases
4. React is easy to integrate with other frameworks like Meteor, Angular, etc
5. Using React, writing UI test cases become extremely easy

### ****5. What are the limitations of React?****

Limitations of React are listed below:

1. React is just a library, not a full-blown framework
2. Its library is very large and takes time to understand
3. It can be little difficult for the novice programmers to understand
4. Coding gets complex as it uses inline templating and JSX

## ****React Interview Questions and Answers in 2023 | Edureka****



### ****6. What is JSX?****

JSX is a shorthand for JavaScript XML. This is a type of file used by React which utilizes the expressiveness of JavaScript along with HTML like template syntax. This makes the HTML file really easy to understand. This file makes applications robust and boosts its performance. Below is an example of JSX:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | render(){  **return**(    <div>    <h1> Hello World from Edureka!!</h1>             </div>        );  } |

### ****7. What do you understand by Virtual DOM? Explain its works.****

A virtual DOM is a lightweight JavaScript object which originally is just a copy of the real DOM. It is a node tree that lists the elements, their attributes and content as Objects and their properties. React’s render function creates a node tree out of the React components. It then updates this tree in response to the mutations in the data model which is caused by various actions done by the user or by the system. Check out this [Web developer course online](https://www.edureka.co/masters-program/full-stack-developer-training) to learn more about react.

This Virtual DOM works in three simple steps.

1. Whenever any underlying data changes, the entire UI is re-rendered in Virtual DOM representation.
2. Then the difference between the previous DOM representation and the new one is calculated.
3. Once the calculations are done, the real DOM will be updated with only the things that have actually changed. 

### **Front End Web Development Training**

### **[ANGULAR CERTIFICATION TRAINING COURSE](https://www.edureka.co/angular-training?utm_source=blogbanner&utm_campaign=trending_course" \o "Angular Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

### **[Angular Certification Training Course](https://www.edureka.co/angular-training?utm_source=blogbanner&utm_campaign=trending_course" \o "Angular Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

*[Reviews](https://www.edureka.co/angular-training?utm_source=blogbanner&utm_campaign=trending_course" \o "Angular Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)*

**[5](https://www.edureka.co/angular-training?utm_source=blogbanner&utm_campaign=trending_course" \o "Angular Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**[(19487)](https://www.edureka.co/angular-training?utm_source=blogbanner&utm_campaign=trending_course" \o "Angular Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)

### **[REACT JS CERTIFICATION TRAINING COURSE](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "React JS Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

### **[React JS Certification Training Course](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "React JS Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

*[Reviews](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "React JS Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)*

**[5](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "React JS Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**[(8779)](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "React JS Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)

### **[WEB DEVELOPER CERTIFICATION TRAINING COURSE](https://www.edureka.co/complete-web-developer?utm_source=blogbanner&utm_campaign=trending_course" \o "Web Developer Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

### **[Web Developer Certification Training Course](https://www.edureka.co/complete-web-developer?utm_source=blogbanner&utm_campaign=trending_course" \o "Web Developer Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

*[Reviews](https://www.edureka.co/complete-web-developer?utm_source=blogbanner&utm_campaign=trending_course" \o "Web Developer Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)*

**[5](https://www.edureka.co/complete-web-developer?utm_source=blogbanner&utm_campaign=trending_course" \o "Web Developer Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**[(9316)](https://www.edureka.co/complete-web-developer?utm_source=blogbanner&utm_campaign=trending_course" \o "Web Developer Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)

### **[HTML CERTIFICATION TRAINING COURSE](https://www.edureka.co/html-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o " HTML Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

### **[HTML Certification Training Course](https://www.edureka.co/html-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o " HTML Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

*[Reviews](https://www.edureka.co/html-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o " HTML Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)*

**[5](https://www.edureka.co/html-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o " HTML Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**[(1009)](https://www.edureka.co/html-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o " HTML Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)

### **[JAVASCRIPT CERTIFICATION TRAINING COURSE](https://www.edureka.co/javascript-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "JavaScript Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

### **[JavaScript Certification Training Course](https://www.edureka.co/javascript-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "JavaScript Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

*[Reviews](https://www.edureka.co/javascript-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "JavaScript Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)*

**[5](https://www.edureka.co/javascript-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "JavaScript Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**[(3746)](https://www.edureka.co/javascript-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "JavaScript Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)

### **[CSS CERTIFICATION TRAINING COURSE](https://www.edureka.co/css-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o "CSS Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

### **[CSS Certification Training Course](https://www.edureka.co/css-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o "CSS Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

*[Reviews](https://www.edureka.co/css-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o "CSS Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)*

**[5](https://www.edureka.co/css-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o "CSS Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**[(406)](https://www.edureka.co/css-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o "CSS Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)

### **[JQUERY CERTIFICATION TRAINING COURSE](https://www.edureka.co/jquery-training?utm_source=blogbanner&utm_campaign=trending_course" \o "jQuery Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

### **[jQuery Certification Training Course](https://www.edureka.co/jquery-training?utm_source=blogbanner&utm_campaign=trending_course" \o "jQuery Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

*[Reviews](https://www.edureka.co/jquery-training?utm_source=blogbanner&utm_campaign=trending_course" \o "jQuery Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)*

**[5](https://www.edureka.co/jquery-training?utm_source=blogbanner&utm_campaign=trending_course" \o "jQuery Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**[(1148)](https://www.edureka.co/jquery-training?utm_source=blogbanner&utm_campaign=trending_course" \o "jQuery Certification Training Course" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)

Next

### ****8. Why can’t browsers read JSX?****

Browsers can only read JavaScript objects but JSX in not a regular JavaScript object. Thus to enable a browser to read JSX, first, we need to transform JSX file into a JavaScript object using JSX transformers like Babel and then pass it to the browser.

### ****9. How different is React’s ES6 syntax when compared to ES5?****

Syntax has changed from ES5 to ES6 in the following aspects:

1. require vs import

|  |  |
| --- | --- |
| 1  2  3  4  5 | // ES5  var React = require('react');    // ES6  **import** React from 'react'; |

1. export vs exports

|  |  |
| --- | --- |
| 1  2  3  4  5 | // ES5  module.exports = Component;    // ES6  export **default** Component; |

1. component and function

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | // ES5  var MyComponent = React.createClass({      render: function() {  **return**    <h3>Hello Edureka!</h3>  ;      }  });    // ES6  **class** MyComponent **extends** React.Component {      render() {  **return**    <h3>Hello Edureka!</h3>  ;      }  } |

1. props

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | // ES5  var App = React.createClass({      propTypes: { name: React.PropTypes.string },      render: function() {  **return**    <h3>Hello, {**this**.props.name}!</h3>  ;      }  });    // ES6  **class** App **extends** React.Component {      render() {  **return**    <h3>Hello, {**this**.props.name}!</h3>  ;      }  } |

1. state

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | // ES5  var App = React.createClass({      getInitialState: function() {  **return** { name: 'world' };      },      render: function() {  **return**    <h3>Hello, {**this**.state.name}!</h3>  ;      }  });    // ES6  **class** App **extends** React.Component {      constructor() {  **super**();  **this**.state = { name: 'world' };      }      render() {  **return**    <h3>Hello, {**this**.state.name}!</h3>  ;      }  } |

### ****10. How is React different from Angular?****

|  |  |  |
| --- | --- | --- |
| React vs Angular ****TOPIC**** | ****REACT**** | ****ANGULAR**** |
| 1. ARCHITECTURE | Only the View of MVC | Complete MVC |
| 2. RENDERING | Server-side rendering | Client-side rendering |
| 3. DOM | Uses virtual DOM | Uses real DOM |
| 4. DATA BINDING | One-way data binding | Two-way data binding |
| 5. DEBUGGING | Compile time debugging | Runtime debugging |
| 6. AUTHOR | Facebook | Google |

## ****React Components – React Interview Questions****

### ****11. “In React, everything is a component.” Explain.****

Components are the building blocks of a React application’s UI. These components split up the entire UI into small independent and reusable pieces. Then it renders each of these components independent of each other without affecting the rest of the UI.

### ****12. What is the purpose of render() in React.****

Each React component must have a ****render()****mandatorily. It returns a single React element which is the representation of the native DOM component. If more than one HTML element needs to be rendered, then they must be grouped together inside one enclosing tag such as ****<form>, <group>,<div>**** etc. This function must be kept pure i.e., it must return the same result each time it is invoked.

### ****13. How can you embed two or more components into one?****

We can embed components into one in the following way:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | **class** MyComponent **extends** React.Component{      render(){  **return**(    <div>    <h1>Hello</h1>                    <Header/>              </div>            );      }  }  **class** Header **extends** React.Component{      render(){  **return**    <h1>Header Component</h1>       };  }  ReactDOM.render(      <MyComponent/>, document.getElementById('content')  ); |

### ****14. What is Props?****

Props is the shorthand for Properties in React. They are read-only components which must be kept pure i.e. immutable. They are always passed down from the parent to the child components throughout the application. A child component can never send a prop back to the parent component. This help in maintaining the unidirectional data flow and are generally used to render the dynamically generated data.

### ****15. What is a state in React and how is it used?****

States are the heart of React components. States are the source of data and must be kept as simple as possible. Basically, states are the objects which determine components rendering and behavior. They are mutable unlike the props and create dynamic and interactive components. They are accessed via ****this.state().****

### ****16. Differentiate between states and props.****

|  |  |  |
| --- | --- | --- |
| States vs Props ****Conditions**** | ****State**** | ****Props**** |
| 1. Receive initial value from parent component | Yes | Yes |
| 2. Parent component can change value | No | Yes |
| 3. Set default values inside component | Yes | Yes |
| 4. Changes inside component | Yes | No |
| 5. Set initial value for child components | Yes | Yes |
| 6. Changes inside child components | No | Yes |

### ****17. How can you update the state of a component?****

State of a component can be updated using this.setState().

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | **class** MyComponent **extends** React.Component {      constructor() {  **super**();  **this**.state = {              name: 'Maxx',              id: '101'          }      }      render()          {              setTimeout(()=>{**this**.setState({name:'Jaeha', id:'222'})},2000)  **return** (    <div>    <h1>Hello {**this**.state.name}</h1>    <h2>Your Id is {**this**.state.id}</h2>                       </div>                );          }      }  ReactDOM.render(      <MyComponent/>, document.getElementById('content')  ); |

### ****18. What is arrow function in React? How is it used?****

Arrow functions are more of brief syntax for writing the function expression. They are also called ‘fat arrow‘ (****=>****) the functions. These functions allow to bind the context of the components properly since in ES6 auto binding is not available by default. Arrow functions are mostly useful while working with the higher order functions.

[](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=batches" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)

### **[React JS Certification Training Course](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=batches" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)**

[Weekday / Weekend BatchesSee Batch Details](https://www.edureka.co/reactjs-redux-certification-training?utm_source=blogbanner&utm_campaign=batches" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | //General way  render() {  **return**(          <MyInput onChange={**this**.handleChange.bind(**this**) } />      );  }  //With Arrow Function  render() {  **return**(          <MyInput onChange={ (e) => **this**.handleOnChange(e) } />      );  } |

### ****19. Differentiate between stateful and stateless components.****

|  |  |
| --- | --- |
| Stateful vs Stateless ****Stateful Component**** | ****Stateless Component**** |
| 1. Stores info about component’s state change in memory | 1. Calculates the internal state of the components |
| 2. Have authority to change state | 2. Do not have the authority to change state |
| 3. Contains the knowledge of past, current and possible future changes in state | 3. Contains no knowledge of past, current and possible future state changes |
| 4. Stateless components notify them about the requirement of the state change, then they send down the props to them. | 4. They receive the props from the Stateful components and treat them as callback functions. |

### ****20. What are the different phases of React component’s lifecycle?****

There are three different phases of React component’s lifecycle:

1. Initial Rendering Phase: This is the phase when the component is about to start its life journey and make its way to the DOM.
2. Updating Phase: Once the component gets added to the DOM, it can potentially update and re-render only when a prop or state change occurs. That happens only in this phase.
3. Unmounting Phase: This is the final phase of a component’s life cycle in which the component is destroyed and removed from the DOM.

Want to upskill yourself to get ahead in your career? Check out this video

## ****Top 10 Technologies to Learn in 2023| Edureka****

### ****21. Explain the lifecycle methods of React components in detail.****

Some of the most important lifecycle methods are:

1. **componentWillMount()**–Executed just before rendering takes place both on the client as well as server-side.
2. **componentDidMount()**–Executed on the client side only after the first render.
3. **componentWillReceiveProps()**– Invoked as soon as the props are received from the parent class and before another render is called.
4. **shouldComponentUpdate()**–Returns true or false value based on certain conditions. If you want your component to update, return **true** else return **false**. By default, it returns false.
5. **componentWillUpdate()**– Called just before rendering takes place in the DOM.
6. **componentDidUpdate()**–Called immediately after rendering takes place.
7. **componentWillUnmount()**– Called after the component is unmounted from the DOM. It is used to clear up the memory spaces.

### ****22. What is an event in React?****

In React, events are the triggered reactions to specific actions like mouse hover, mouse click, key press, etc. Handling these events are similar to handling events in DOM elements. But there are some syntactical differences like:

1. Events are named using camel case instead of just using the lowercase.
2. Events are passed as functions instead of strings.

The event argument contains a set of properties, which are specific to an event. Each event type contains its own properties and behavior which can be accessed via its event handler only.

### ****23. How do you create an event in React?****

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | **class** Display **extends** React.Component({      show(evt) {          // code      },      render() {          // Render the div with an onClick prop (value is a function)  **return** (    <div onClick={**this**.show}>Click Me!</div>            );      }  }); |

### ****24. What are synthetic events in React?****

Synthetic events are the objects which act as a cross-browser wrapper around the browser’s native event. They combine the behavior of different browsers into one API. This is done to make sure that the events show consistent properties across different browsers.

### ****25. What do you understand by refs in React?****

Refs is the short hand for References in React. It is an attribute which helps to store a reference to a particular React element or component, which will be returned by the components render configuration function. It is used to return references to a particular element or component returned by render(). They come in handy when we need DOM measurements or to add methods to the components.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | **class** ReferenceDemo **extends** React.Component{       display() {  **const** name = **this**.inputDemo.value;           document.getElementById('disp').innerHTML = name;       }  render() {  **return**(    <div>              Name: <input type="text" ref={input => **this**.inputDemo = input} />              <button name="Click" onClick={**this**.display}>Click</button>    <h2>Hello <span id="disp"></span> !!!</h2>          </div>      );     }   } |

### ****26. List some of the cases when you should use Refs.****

Following are the cases when refs should be used:

* When you need to manage focus, select text or media playback
* To trigger imperative animations
* Integrate with third-party DOM libraries

### ****27. How do you modularize code in React?****

We can modularize code by using the export and import properties. They help in writing the components separately in different files.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | //ChildComponent.jsx  export **default** **class** ChildComponent **extends** React.Component {      render() {  **return**(    <div>    <h1>This is a child component</h1>               </div>            );      }  }    //ParentComponent.jsx  **import** ChildComponent from './childcomponent.js';  **class** ParentComponent **extends** React.Component {      render() {  **return**(    <div>                  <App />              </div>            );      }  } |

### ****28. How**** ****are forms created in React?****

React forms are similar to HTML forms. But in React, the state is contained in the state property of the component and is only updated via setState(). Thus the elements can’t directly update their state and their submission is handled by a JavaScript function. This function has full access to the data that is entered by the user into a form.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | handleSubmit(event) {      alert('A name was submitted: ' + **this**.state.value);      event.preventDefault();  }    render() {  **return** (    <form onSubmit={**this**.handleSubmit}>              <label>                  Name:                  <input type="text" value={**this**.state.value} onChange={**this**.handleSubmit} />              </label>              <input type="submit" value="Submit" />          </form>        );  } |

### ****29. What do you know about controlled and uncontrolled components?****

|  |  |
| --- | --- |
| Controlled vs Uncontrolled Components ****Controlled Components**** | ****Uncontrolled Components**** |
| 1. They do not maintain their own state | 1. They maintain their own state |
| 2. Data is controlled by the parent component | 2. Data is controlled by the DOM |
| 3. They take in the current values through props and then notify the changes via callbacks | 3. Refs are used to get their current values |

In case you are facing any challenges with these React interview questions, please comment on your problems in the section below.

## ****React Interview Questions****

### ****30. What are Higher Order Components(HOC)?****

Higher Order Component is an advanced way of reusing the component logic. Basically, it’s a pattern that is derived from React’s compositional nature. HOC are custom components which wrap another component within it. They can accept any dynamically provided child component but they won’t modify or copy any behavior from their input components. You can say that HOC are ‘pure’ components.

### ****31. What can you do with HOC?****

HOC can be used for many tasks like:

* Code reuse, logic and bootstrap abstraction
* Render High jacking
* State abstraction and manipulation
* Props manipulation

### ****32. What are Pure Components?****

*Pure*components are the simplest and fastest components which can be written. They can replace any component which only has a **render().**These components enhance the simplicity of the code and performance of the application.

### ****33. What is the significance of keys in React?****

Keys are used for identifying unique Virtual DOM Elements with their corresponding data driving the UI. They help React to optimize the rendering by recycling all the existing elements in the DOM. These keys must be a unique number or string, using which React just reorders the elements instead of re-rendering them. This leads to increase in application’s performance.

****React Redux – React Interview Questions****

### ****34. What were the major problems with MVC framework?****

Following are some of the major problems with MVC framework:

* DOM manipulation was very expensive
* Applications were slow and inefficient
* There was huge memory wastage
* Because of circular dependencies, a complicated model was created around models and views

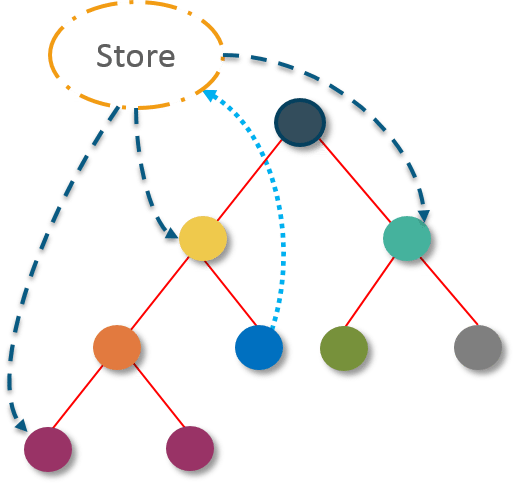
### ****35. Explain Flux.****

Flux is an architectural pattern which enforces the uni-directional data flow. It controls derived data and enables communication between multiple components using a central Store which has authority for all data. Any update in data throughout the application must occur here only. Flux provides stability to the application and reduces run-time errors.

### ****36.**** ****What is Redux?****

Redux is one of the most trending libraries for front-end development in today’s marketplace. It is a predictable state container for JavaScript applications and is used for the entire applications state management. Applications developed with Redux are easy to test and can run in different environments showing consistent behavior.

### ****37. What are the three principles that Redux follows?****

1. ****Single source of truth:****The state of the entire application is stored in an object/ state tree within a single store. The single state tree makes it easier to keep track of changes over time and debug or inspect the application.
2. **State is read-only:**The only way to change the state is to trigger an action. An action is a plain JS object describing the change. Just like state is the minimal representation of data, the action is the minimal representation of the change to that data.
3. ****Changes are made with pure functions:**** In order to specify how the state tree is transformed by actions, you need pure functions. Pure functions are those whose return value depends solely on the values of their arguments.

### ****38. What do you understand by “Single source of truth”?****

Redux uses ‘Store’ for storing the application’s entire state at one place. So all the component’s state are stored in the Store and they receive updates from the Store itself. The single state tree makes it easier to keep track of changes over time and debug or inspect the application.

### ****39. List down the components of Redux.****

Redux is composed of the following components:

1. **Action** – It’s an object that describes what happened.
2. **Reducer**–  It is a place to determine how the state will change.
3. **Store** – State/ Object tree of the entire application is saved in the Store.
4. **View** – Simply displays the data provided by the Store.

In case you are facing any challenges with these React interview questions, please comment on your problems in the section below.

### ****40. Show how the data flows through Redux?****



### ****41. How are Actions defined in Redux?****

Actions in React must have a type property that indicates the type of ACTION being performed. They must be defined as a String constant and you can add more properties to it as well. In Redux, actions are created using the functions called Action Creators. Below is an example of Action and Action Creator:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | function addTodo(text) {  **return** {                  type: ADD\_TODO,                   text      }  } |

### ****42. Explain the role of Reducer.****

Reducers are pure functions which specify how the application’s state changes in response to an ACTION. Reducers work by taking in the previous state and action, and then it returns a new state. It determines what sort of update needs to be done based on the type of the action, and then returns new values. It returns the previous state as it is, if no work needs to be done.

### ****43. What is the significance of Store in Redux?****

A store is a JavaScript object which can hold the application’s state and provide a few helper methods to access the state, dispatch actions and register listeners. The entire state/ object tree of an application is saved in a single store. As a result of this, Redux is very simple and predictable. We can pass middleware to the store to handle the processing of data as well as to keep a log of various actions that change the state of stores. All the actions return a new state via reducers.

### ****44. How is Redux different from Flux?****

|  |  |
| --- | --- |
| Flux vs Redux ****Flux**** | ****Redux**** |
| 1. The Store contains state and change logic | 1. Store and change logic are separate |
| 2. There are multiple stores | 2. There is only one store |
| 3. All the stores are disconnected and flat | 3. Single store with hierarchical reducers |
| 4. Has singleton dispatcher | 4. No concept of dispatcher |
| 5. React components subscribe to the store | 5. Container components utilize connect |
| 6. State is mutable | 6. State is immutable |

In case you are facing any challenges with these React interview questions, please comment on your problems in the section below.

## ****React Interview Questions****

### ****45. What are the advantages of Redux?****

Advantages of Redux are listed below:

* ****Predictability of outcome –****Since there is always one source of truth, i.e. the store, there is no confusion about how to sync the current state with actions and other parts of the application.
* ****Maintainability –****The code becomes easier to maintain with a predictable outcome and strict structure.
* ****Server-side rendering –**** You just need to pass the store created on the server, to the client side. This is very useful for initial render and provides a better user experience as it optimizes the application performance.
* ****Developer tools –****From actions to state changes, developers can track everything going on in the application in real time.
* ****Community and ecosystem –****Redux has a huge community behind it which makes it even more captivating to use. A large community of talented individuals contribute to the betterment of the library and develop various applications with it.
* ****Ease of testing –****Redux’s code is mostly functions which are small, pure and isolated. This makes the code testable and independent.
* ****Organization –****Redux is precise about how code should be organized, this makes the code more consistent and easier when a team works with it.

****React Router – React Interview Questions****

### ****46. What is React Router?****

React Router is a powerful routing library built on top of React, which helps in adding new screens and flows to the application. This keeps the URL in sync with data that’s being displayed on the web page. It maintains a standardized structure and behavior and is used for developing single page web applications. React Router has a simple API.

### ****47. Why**** ****is switch keyword used in React Router v4?****

Although a ****<div>**** is used to encapsulate multiple routes inside the Router. The ‘switch’ keyword is used when you want to display only a single route to be rendered amongst the several defined routes. The ****<switch>****tag when in use matches the typed URL with the defined routes in sequential order. When the first match is found, it renders the specified route. Thereby bypassing the remaining routes.

### ****48. Why do we need a Router in React?****

A Router is used to define multiple routes and when a user types a specific URL, if this URL matches the path of any ‘route’ defined inside the router, then the user is redirected to that particular route. So basically, we need to add a Router library to our app that allows creating multiple routes with each leading to us a unique view.

|  |  |
| --- | --- |
| 1  2  3  4  5 | <**switch**>      <route exact path=’/’ component={Home}/>      <route path=’/posts/:id’ component={Newpost}/>      <route path=’/posts’   component={Post}/>  </**switch**> |

### ****49. List down the advantages of React Router.****

Few advantages are:

1. Just like how React is based on components, in React Router v4, the API is ‘All About Components’. A Router can be visualized as a single root component (****<BrowserRouter>****) in which we enclose the specific child routes (****<route>****).
2. No need to manually set History value: In React Router v4, all we need to do is wrap our routes within the ****<BrowserRouter>**** component.
3. The packages are split: Three packages one each for Web, Native and Core. This supports the compact size of our application. It is easy to switch over based on a similar coding style.

### ****50. How is React Router different from conventional routing?****

|  |  |  |
| --- | --- | --- |
| Conventional Routing vs React Routing ****Topic**** | ****Conventional Routing**** | ****React Routing**** |
| ****PAGES INVOLVED**** | Each view corresponds to a new file | Only single HTML page is involved |
| ****URL CHANGES**** | A HTTP request is sent to a server and corresponding HTML page is received | Only the History attribute is changed |
| ****FEEL**** | User actually navigates across different pages for each view | User is duped thinking he is navigating across different pages |

I hope this set of [React](https://reactjs.org/" \t "https://www.edureka.co/blog/interview-questions/react-interview-questions/_blank) Interview Questions and Answers will help you in preparing for your interviews. All the best!

If you want to get trained in web development and wish to develop interesting UI’s on your own, then check out the [Web Development Certification](https://www.edureka.co/masters-program/web-developer-training) Training by Edureka, a trusted online learning company with a network of more than 250,000 satisfied learners spread across the globe.

Got a question for us? Please mention it in the comments section and we will get back to you.