

[Updated] 300+ React Js Interview (MCQ) With Answers - December 2024

🔗 [React.js](#) is a widely popular open-source front-end 🔗 [JavaScript library](#) that is extensively used for developing user interfaces in web applications.

As the demand for skilled React developers continues to rise, it becomes essential to evaluate their knowledge and abilities effectively.

This is where React JS Multiple-Choice Interview Questions (MCQs) can be immensely helpful.

India to Malaysia

from India

India to Bangkok

from India

India to Phuket

from India

from Rs5

View Now

from Rs15

View Now

from Rs15

View Now

In the year 2024, we have curated a comprehensive collection of over 300+ React JS MCQs with their corresponding answers.

These MCQs have been thoughtfully designed to assist hiring managers, recruiters, and interviewers in assessing candidates' proficiency in React JS.

Whether you are a beginner or an advanced learner, these MCQs are suitable for anyone looking to enhance their understanding of React.js.

They provide an opportunity to both improve your knowledge and evaluate your grasp of React.js concepts.

Our goal is to provide you with valuable insights and help you prepare effectively for your upcoming React JS interview.



India to Phuket
from India

from Rs12,681

Vi

from Rs12,681

View Now

We hope that this post proves to be a helpful resource for you.

Most Asked 300+ React Js Interview Multiple Choice Questions With Answers



**300+ React Js Interview
Multiple Choice Questions with Answers**

www.fullformatoz.com

ReactJS Introduction

1. What is the use of the create-react-app command?



- a. Updates a React app.
- b. Creates a new React app.
- c. Installs dependencies.



Answer: B is the correct option.

It is used to create a new React app. The create-react-app command is used to set up a new React project with a preconfigured development environment.

If you want to update an existing React application, you typically use the following command:

```
npm update react-scripts  
or  
yarn upgrade react-scripts
```

2. What keyword initiates a class in JavaScript?

- a. Constructor
- b. Class
- c. Object
- d. DataObject

[View Answer](#) ^

Answer: B is the correct option.

In JavaScript, a class is a type of function that's initiated with the keyword "class" to create and define new objects with shared properties and methods.

3. What does ES6 stand for?

- a. ECMAScript 6
- b. ECMA 6
- c. ECMAJavaScript 6
- d. EJavaScript 6



[View Answer](#) ^

4. What will be the output of the following JSX code?

```
const name = "John";  
const element = <h1> Hello, {name}! </h1>
```

- a. "Hello, John!"
- b. "Hello, {name}!"
- c. The code will throw an error.
- d. None of the above.

[View Answer](#) 

Answer: A is the correct option.

The {name} expression will be evaluated to "John" at runtime and the resulting JSX element will render as "Hello, John!".

5. What is the correct name of React.js?

- a. React
- b. React.js
- c. ReactJS
- d. All of the above

[View Answer](#) 

Answer: D is the correct option.

React.js is known by all of these names and they all refer to the same library.

6. Which advantage does JSX provide?



- a. Enables server-side rendering.
- b. Reduces the size of the React.js library.

Answer: C is the correct option.

JSX in React.js allows for a more readable and easy-to-write code by combining HTML and JavaScript, making it an advantage rather than a disadvantage.

7. What is ReactJS?

- a. A programming language
- b. A framework for building user interfaces
- c. A database management system
- d. An operating system

View Answer ^

Answer: B is the correct option.

ReactJS is a JavaScript library and framework used for building user interfaces. It was developed by Facebook and is now maintained by a community of developers.

ReactJS allows developers to build reusable UI components and manage the state of their applications. It is widely used in web development and has become a popular choice for building single-page applications.

8. What command is used to install create-react-app globally?

- a. npm install -g create-react-app
- b. npm install create-react-app
- c. npm install -f create-react-app
- d. install -g create-react-app

View Answer ^



Answer: A is the correct option.

- a. Class
- b. Factory
- c. Render
- d. Props

[View Answer](#) 

Answer: D is the correct option.

Props act as input for class-based components, allowing data to be passed from parent to child components, facilitating reusability and modularity.

10. Why should component names start with capital letter?

- a. To differentiate from regular HTML tags.
- b. To make it easier to distinguish between components and elements.
- c. Both A and B are correct.
- d. None of the above.

[View Answer](#) 

Answer: C is the correct option.

Component names must start with a capital letter to differentiate them from regular HTML tags and to make it easier to distinguish between components and elements.

11. Which method is used to render a React element into the DOM

- a. `React.createElement()`
- b. `componentDidMount()`
- c. `renderToDOM()`
- d. `ReactDOM.render()`



[View Answer](#) 

12. What is the output of the following code snippet?

```
import React from 'react';

function App() {
  return (
    <div>
      <h1>Hello World!</h1>
      <p>This is a React app.</p>
    </div>
  );
}

export default App;
```

- a. It renders a div element with a heading and paragraph inside.
- b. It throws an error.
- c. It renders nothing.
- d. None of the above.

[View Answer](#) ^

Answer: A is the correct option.

The code defines a React functional component named App, which returns a div element with a heading and paragraph inside.

13. What is the purpose of the "key" prop in React.js?

- a. "Key" prop is used to look pretty, and there is no benefit whatsoever.
- b. "Key" prop is a way for React to identify a newly added item in a list and compare it during the "diffing" algorithm.
- c. It is one of the attributes in HTML.
- d. It is not commonly used in an array.

 [View Answer](#) ^

reorder the list items when changes occur.

For example, imagine you have a list of fruits in a React component:

```
const fruits = ['apple', 'banana', 'cherry'];
```

If you render this list in a React component without specifying a "key" prop, React might have difficulty distinguishing between items when the list changes. However, if you use the "key" prop, like so:

```
const fruits = [  
  { id: 1, name: 'apple' },  
  { id: 2, name: 'banana' },  
  { id: 3, name: 'cherry' }  
];
```

And render it like this:

```
<ul>  
  {fruits.map(fruit => (  
    <li key={fruit.id}>{fruit.name}</li>  
  ))}  
</ul>
```

React can now efficiently track changes in the list because each item has a unique "key" based on its "id" property. This ensures smoother and more accurate updates when items are added, removed, or reordered in the list.

14. What will be the output of the following JSX code?

```
const element = <button onClick={() => alert("Button clicked!")}>Click me</button>
```



- a. A button with the text "Click me" that triggers an alert when clicked.
- b. A syntax error due to the arrow function syntax.

Answer: A is the correct option.

The onClick prop is a special prop in React that allows you to attach a function to the click event of an element.

In this case, an arrow function is used to create the function inline and trigger an alert when the button is clicked.

15. Which of the following is a valid React Component?

A.

```
const myComponent = () => { return <h1>Hello World!</h1> }
```

B.

```
class MyComponent extends React.Component { render() { return <h1>Hello World!</h1> }
```

- a. A
- b. B
- c. Both A and B
- d. None of the above

View Answer 

Answer: C is the correct option.

Both a function component and a class component are valid ways to define a React component.

16. When was React.js initially released?

- a. May 29, 2013
- b. April 29, 2013



Answer: B is the correct option.

React.js was first released on April 29, 2013, by Facebook.

17. What is the virtual DOM in React.js used for?

- a. Handling user authentication
- b. Creating database queries
- c. Improving performance by minimizing DOM manipulation
- d. Styling [React](#) components

View Answer ^

Answer: C is the correct option.

The virtual DOM is used for improving performance by minimizing direct DOM manipulation.

React creates a virtual representation of the DOM and updates the actual DOM only when necessary to reduce rendering time and enhance application speed.

18. What is the purpose of the PureComponent class?

- a. To create a component that automatically implements `shouldComponentUpdate()` for improved performance.
- b. To create a component that does not implement `shouldComponentUpdate()` for simpler code.
- c. To create a component that does not require state management.
- d. None of the above.

View Answer ^

Answer: A. To create a component that automatically implements `shouldComponentUpdate()` for improved performance.



- A. react-router-dom is used for web [applications](#), while react-router-native is used for mobile applications.
- B. react-router-native is used for web applications, while react-router-dom is used for mobile applications.
- C. react-router-dom and react-router-native are the same thing.
- D. None of the above.

View Answer ^

Answer: A. react-router-dom is used for web applications, while react-router-native is used for mobile applications.

react-router-dom is a version of the React Router library that is designed for web applications, while react-router-native is a version of the library that is designed for mobile applications.

20. What is the output of the following React code snippet?

```
import React from 'react';
import ReactDOM from 'react-dom';

class App extends React.Component {
  render() {
    return <h1>Hello World!</h1>
  }
}

ReactDOM.render(<app>, document.getElementById('root'));
```

- a. It renders a heading element with the text "Hello World!" to the root element of the HTML document.
- b. It throws an error.
- c. It renders nothing.
- d. None of the above.



View Answer ^

The ReactDOM.render method is used to render the App component to the root element of the HTML document.

21. What is the primary benefit of virtual DOM in terms of performance optimization?

- a. It eliminates the need for the actual DOM, reducing memory usage.
- b. It completely eliminates the need for re-rendering components.
- c. It reduces the time it takes to fetch data from an API.
- d. It minimizes the number of updates required to the actual DOM.

[View Answer](#) ^

Answer: D is the correct option.

ReactJS's virtual DOM minimizes the number of updates needed to the actual DOM, thus optimizing performance by reducing unnecessary operations and improving rendering efficiency.

22. How to write comments in React?

- a. Use single-line comments //
- b. Use multi-line comments /* */
- c. Use JSX comments {/ */}
- d. All of the above

[View Answer](#) ^

Answer: D is the correct option.

- Single-line comments can be written using //
- Multi-line comments can be written using /* */
- JSX comments can be written using {/ /*}



```
import React from 'react';
```

```
return (  
  <div className="user-profile">  
    {/* JSX comment: This div holds the user's profile information. */}  
    <h2>Welcome, {username}! </h2>  
    {  
      /*  
        Multi-line comment: Inside the div, we display the user's  
        username and bio.  
      */  
    }  
    <p>{bio}</p>  
  </div>  
)  
};  
  
export default UserProfile;
```

23. What are the ways to handle data in react.js?

- a. state and services
- b. state and component
- c. state and props
- d. services and component

[View Answer](#) 

Answer: C is the correct option.

The ways to handle data are through state and props.

State is used for managing component-specific data while props are used for passing data between components.

24. Does React.js cover only the view layer of the app?



- a. Yes
- b. No

React.js is a JavaScript library that is primarily used for building user interfaces in the front-end of web applications, which includes only the view layer of the application.

25. What is the output of the following code snippet?

```
import React from 'react';

function Greeting(props) {
  return <h1>Hello, {props.name}!</h1>
}

export default Greeting;
```

- a. It defines a functional component that displays a greeting.
- b. It defines a class component that displays a greeting.
- c. It defines a higher-order component that displays a greeting.
- d. None of the above.

[View Answer](#) ^

Answer: A is the correct option.

The code defines a React functional component named Greeting, which takes a props object containing a name property, and returns a heading element with a greeting that includes the name.

26. Babel is a:

- a. Compiler
- b. Transpiler
- c. Both A and B are correct
- d. None of the above



[View Answer](#) ^

It can also be used to transpile code written in other languages that compile to JavaScript, like TypeScript.

27. Which of the following statements is true for controlled components?

- a. The source of truth is DOM.
- b. The source of truth can be anything.
- c. The source of truth is a component state.
- d. None of the above.

[View Answer](#) ^

Answer: C is the correct option.

Controlled components in React.js have the source of truth for user input in their component state, allowing for easier management of form data.

28. What is the term used to describe the process of converting ES6 (modern JavaScript) code into ES5 (older JavaScript) using Babel.js?

- a. ES6ification
- b. Transpilation
- c. Optimization
- d. Compilation

[View Answer](#) ^

Answer: B is the correct option.

The process of converting ES6 (modern JavaScript) code into ES5 (older JavaScript) using Babel.js is commonly referred to as Transpilation.



29. What is wrong with the following JSX code?



- b. The class prop should be style in JSX.
- c. The class prop is not a valid prop for the <p> element.
- d. None of the above.

[View Answer](#) ^

Answer: A is the correct option.

In JSX, the class attribute should be replaced with className to avoid conflicts with the class keyword in JavaScript.

30. What is the render() method in a React Component?

- a. A method that returns a React element
- b. A method that returns a DOM node
- c. A method that updates the state of a component
- d. None of the above

[View Answer](#) ^

Answer: A is the correct option.

The render() method in a React Component is a required method that returns a React element.

This element can be a DOM node or another React Component.

31. What is wrong with the following JSX code?

```
const element = 
```

- a. The img element is missing a closing tag.
- b. The src prop is not enclosed in quotes.
- c. The src prop should be source in JSX.
- d. None of the above.



This JSX code is correct and will render an `` element with the specified `src` prop.

32. What is the output of the following React code snippet?

```
import React from 'react';
import ReactDOM from 'react-dom';

function App() {
  return <h1>Hi</h1>
}

ReactDOM.render(, document.getElementById('root'));
```

- a. It renders a heading element with the text "Hi" to the root element of the HTML document.
- b. It throws an error.
- c. It renders nothing.
- d. None of the above.



View Answer 

Answer: A is the correct option.

The code defines a React functional component named `App`, which returns a heading element with the text "Hi".

The `ReactDOM.render` method is used to render the `App` component to the root element of the HTML document.

33. Which of the following statements about React's rendering process is true?

- a.  [React](#) always renders components synchronously in the order they are called.
- b. React uses a single rendering thread for all components.
- c. React may batch multiple component updates for performance reasons.
- d. React renders components in parallel with the main  [application](#) thread.



React may batch multiple component updates for performance reasons. This means that React can optimize rendering by grouping multiple updates together and applying them in a single batch for improved performance.

34. What is the purpose of `setState()` in React.js?

- a. Invoke code after the `setState` operation is done.
- b. Replace the state completely instead of the default merge action.
- c. Access the previous state before the `setState` operation.
- d. None of the above.

[View Answer](#) ^

Answer: C is the correct option.

In React.js, the `setState()` method serves the purpose of updating the state of a component, triggering a re-render with the updated state.

Additionally, it provides access to the previous state through a callback function, allowing for more controlled updates.

Here's the same concept implemented using a functional component with React hooks:

```
import React, { useState } from 'react';

function ExampleComponent()
  const [message, setMessage] = useState('Initial Message');

  const handleClick = () => {
    // Using setMessage to change the message
    setMessage('Updated Message');
  };

  return (
    <div>
      <p>{message}</p>
      <button onClick={handleClick}>Change Message </button>
    </div>
  );
```



35. What will happen if you remove the 'ReactDOM.render' call from the given React code snippet?

```
import React from 'react';
import ReactDOM from 'react-dom';
function App() {
  return <h1>Hello World!</h1>;
}
// ReactDOM.render(<App />, document.getElementById('root'));
```

- a. It will render nothing, and the page will be empty.
- b. It will result in a compilation error.
- c. It will still render the 'Hello World!' heading to the page.
- d. It will throw a runtime error.

[View Answer](#) ^

Answer: A is the correct option.

If you remove the 'ReactDOM.render' call from the given code snippet, it will render nothing, and the page will be empty.

The 'ReactDOM.render' call is responsible for rendering the 'App' component to the specified root element in the HTML document.

36. What are props in React and how are they used?

- a. Methods
- b. Injected
- c. Both 1 & 2
- d. All of the above

[View Answer](#) ^



They are used to customize the behavior of a component, and are read-only.

37. How many ways can you define variables in ES6?

- a. 1
- b. 3
- c. 4
- d. 5

View Answer ^

Answer: B is the correct option.

In ES6, there are three ways to define variables: using "var", "let", and "const", each with different scoping rules and behavior in various contexts.

38. How does React.js optimize performance when updating the user interface?

- a. By directly manipulating the original DOM elements.
- b. By using a lightweight representation of the actual DOM called the Virtual DOM.
- c. By relying on browser-specific optimizations for rendering.
- d. By increasing the size of the React.js library to handle updates more efficiently.

View Answer ^

Answer: B is the correct option.

React.js optimizes performance when updating the user interface by using a lightweight representation of the actual DOM called the Virtual DOM. This approach minimizes direct manipulation of the real DOM and allows for efficient updates.

39. How many elements can a valid React component return?

- a. 1
- b. 2



Answer: A is the correct option.

A valid React component can return only one element, which can be a single HTML element, a React fragment, or a component containing multiple elements.

40. What is the purpose of the constructor() method in a class component?

- a. Define the initial state
- b. Define structure and content
- c. Define lifecycle methods
- d. None of the above

[View Answer](#) ^

Answer: A is the correct option.

The constructor() method in a class component is used to initialize the state of the component.

41. What is the purpose of the ReactDOM.render() method?

- a. Render a React component to the DOM
- b. Create a new React component
- c. Update the state of a React component
- d. None of the above

[View Answer](#) ^

Answer: A. Render a React component to the DOM

The ReactDOM.render() method is used to render a component to the DOM, which inserts the component's HTML markup into the page.

42. What is the output of the following code snippet?



```
        setCount(count + 1);
    }
    return (
        <div>
            <p>You clicked {count} times.</p>
            <button onClick={handleClick}>Click me</button>
        </div>
    );
}

export default Counter;
```

- a. Defines a class component with an incrementable counter
- b. Defines a functional component with an incrementable counter
- c. Defines a higher-order component with an incrementable counter
- d. None of the above.

[View Answer](#) ^

Answer: B is the correct option.

The code defines a React functional component named Counter, which uses the useState hook to define a count state variable initialized to 0, and a setCount function to update the count variable.

The component also defines a handleClick method using an arrow function that calls the setCount function to increment the count variable.

The render method returns a div element with a paragraph element that displays the count variable, and a button element with an onClick event handler that calls the handleClick method to increment the count.

43. How can you set a default value for an uncontrolled form field?

- a. Using Value property
- b. using the defaultValue property
- c. Using the default property
- d. It is assigned automatically.



The `defaultValue` property can be used in React.js to set a default value for an uncontrolled form field, allowing for easier handling of user input.

44. What is the smallest building block of ReactJS?

- a. Props
- b. Elements
- c. Components
- d. None of the above

[View Answer](#) ^

Answer: B is the correct option.

An element is the smallest building block of ReactJS, which represents a single unit of a user interface, like a button or a text input.

45. In a component hierarchy, if a parent component passes a prop to a child component, can the child component modify the value of that prop directly?

- a. Yes, a child component can modify the value of a prop directly.
- b. No, a child component cannot modify the value of a prop directly.
- c. It depends on whether the prop is marked as mutable or immutable.
- d. Only if the parent component explicitly allows it.

[View Answer](#) ^

Answer: B is the correct option.

No, a child component cannot modify the value of a prop directly. Props in React are read-only and should not be mutated by the child component.



Modifying a prop directly can lead to unexpected behavior and should be avoided.

- c. Create a new React project with preconfigured settings.
- d. None of the above

[View Answer](#) ^

Answer: A. To create a new React project with preconfigured settings.

The create-react-app tool is used to create a new React project with preconfigured settings, such as a development server, a build system, and a set of recommended dependencies.

47. What is the function used to change the state of the component?

- a. this.setState
- b. this.setChangeState
- c. this.State{}
- d. None of the above.

[View Answer](#) ^

Answer: A is the correct option.

this.setState is used to update the state of the React component and re-render the component with the updated state.

48. Two primary ways to manage data

- a. State & Props
- b. Services & Components
- c. State & Services
- d. State & Component

[View Answer](#) ^

Answer: A is the correct option.

State and Props are the two primary methods for handling data in React.



49. Difference between a controlled component and an uncontrolled component

- a. A controlled component uses state to manage its data, while an uncontrolled component does not.
- b. An uncontrolled component uses state to manage its data, while a controlled component does not.
- c. Controlled components and uncontrolled components are the same thing.
- d. None of the above.

[View Answer](#) ^

Answer: A. A controlled component uses state to manage its data, while an uncontrolled component does not.

A controlled component is one that uses state to manage its data, while an uncontrolled component does not use state and instead relies on the DOM to manage its data.

50. What is the output of the following React code snippet?

```
import React, { Component } from 'react';

class Counter extends Component {
  constructor() {
    super();
    this.state = {
      count: 0
    };
  }

  handleIncrement = () => {
    this.setState({ count: this.state.count + 1 });
  };

  render() {
    return (
      <div>
        <p>You clicked {this.state.count} times.</p>
        <button onClick={this.handleIncrement}>Click me</button>
      </div>
    );
  }
}
```



```
export default Counter;
```

- a. Functional component with an incrementable counter.
- b. Class component with an incrementable counter.
- c. Higher-order component with an incrementable counter.
- d. None of the above.

[View Answer](#) 

Answer: A is the correct option.

The code defines a React functional component named `Counter`, which uses the `useState` hook to create a state variable named `count` and a function named `setCount` that can be used to update the count.

The component returns a `div` element with a paragraph element that displays the count, and a button element with an `onClick` event handler that calls the `setCount` function to increment the count.

51. In a class component, what method is used to update the state based on the previous state?

- a. Using `this.props`
- b. Using `setState()`
- c. Using `this.state`
- d. Using `state.update()`

[View Answer](#) 

Answer: B is the correct option.

In a class component, you should use the `setState()` method to update the state based on the previous state. This method allows you to pass a function as an argument and update the state using the previous state, ensuring safe and correct state updates.



^ 52. What is the output of the following code snippet?

```
}  
}  
  
export default Greeting;
```

- a. Class component displaying a greeting.
- b. Functional component displaying a greeting.
- c. Higher-order component displaying a greeting.
- d. None of the above.

View Answer ^

Answer: A is the correct option.

The code defines a React class component named Greeting, which takes a props object containing a name property, and returns a heading element with a greeting that includes the name.

53. How do you define a functional component?

- a. Extend [React.Component](#) class
- b. Use the class keyword
- c. Define a function returning JSX element
- d. Use the render() method

View Answer ^

Answer: C is the correct option.

Functional components in React are defined as functions that return a JSX element.

54. How do you define a class component?

- a. Extend `React.Component` class
- b. Use the class keyword
- c. Define a function returning JSX
- d. Use the render() method



Class components in React are defined by creating a new class that extends the `React.Component` class.

55. What does "state" represent?

- a. A permanent storage
- b. Internal storage of the component
- c. External storage of the component
- d. None of the above

View Answer ^

Answer: B is the correct option.

In React, state refers to the internal storage of a component that holds data specific to that component and can be changed over time, affecting the component's behavior and rendering.

56. What is the declarative method for rendering a list of components based on an array's values?

- a. Using the `reduce` array method
- b. Using the `<Each />` component
- c. Using the `Array.map()` method
- d. With a `for/while` loop

View Answer ^

Answer: C is the correct option.

The `Array.map()` method is a declarative way to render a dynamic list of components in React based on values in an array, allowing efficient iteration and transformation of the array elements.

57. What is the output of the following code snippet?

```
    return <h1>Hello World!</h1>
  }
}

ReactDOM.render(, document.getElementById('root'));
```

- a. Renders "Hello World!" in a heading to the root HTML element.
- b. Throws an error.
- c. Renders nothing.
- d. None of the above.

[View Answer](#) ^

Answer: A is the correct option.

The code defines a React class component named App, which defines a render method that returns a heading element with the text "Hello World!".

The ReactDOM.render method is used to render the App component to the root element of the HTML document.

58. What is the default port for the webpack-dev-server?

- a. 3000
- b. 8080
- c. 3030
- d. 6060

[View Answer](#) ^

Answer: B is the correct option.

The webpack-dev-server runs by default on port 8080, providing a development server with live reloading for a faster and smoother development experience.



59. Which keyword is used for class inheritance in JavaScript?

[View Answer](#) ^

Answer: C is the correct option.

The "extends" keyword is used in JavaScript to create class inheritance, allowing a new class to inherit properties and methods from an existing class.

60. Difference between stateful and stateless components

- a. Stateful components have state while stateless components do not have state.
- b. Stateful components use class components while stateless components use functional components.
- c. Stateful components have lifecycle methods while stateless components do not have lifecycle methods.
- d. Stateful components are used for rendering dynamic data while stateless components are used for rendering static data.

[View Answer](#) ^

Answer: A is the correct option.

Stateful components in ReactJS are components that have state, meaning they store and manage data that can change over time.

They are typically created using class components and can use lifecycle methods to manage their state.

61. Which of the following statement is true for uncontrolled components in React.js?

- a. The source of truth is a component state.
- b. The source of truth can be anything.
- c. The source of truth is DOM.
- d. None of the above.



Uncontrolled components rely on the DOM as the source of truth for user input, rather than managing state within the component.

62. What is the output of the following React code snippet?

```
import React from 'react';
import PropTypes from 'prop-types';
function Greeting(props) {
  return <h1>Hello, {props.name}!</h1>;
}

Greeting.propTypes = {
  name: PropTypes.string.isRequired
};

export default Greeting;
```

- a. It defines a functional component that displays a greeting.
- b. It defines a class component that displays a greeting.
- c. It defines a higher-order component that displays a greeting.
- d. None of the above.

[View Answer](#) ^

Answer: A is the correct option.

The code defines a React functional component named Greeting, which takes a props object containing a name property, and returns a heading element with a greeting that includes the name.

The component also defines a propTypes object using the PropTypes library, which specifies that the name property is required and must be a string.



63. What happens when the state of a React component is changed?

- a. It will do nothing; you have to call render method to render the component again.

[View Answer](#) ^

Answer: B is the correct option.

When the state of a React component is changed, the component will re-render to reflect the updated state.

64. Calls to `setState()` method in React.js are:

- a. Synchronous in nature.
- b. Asynchronous in nature.
- c. Are asynchronous but can be made synchronous when required.
- d. None of the above.

[View Answer](#) ^

Answer: B is the correct option.

Calls to `setState()` method in React.js are asynchronous in nature, but can be made synchronous when required using callback functions.

65. Which keyword is used to create a class inheritance in React?

- a. This
- b. Create
- c. Extends
- d. Inherits

[View Answer](#) ^

Answer: C is the correct option.

The "extends" keyword is used to create a class inheritance in React, which allows one class to inherit properties and methods from another class.



d. Props

[View Answer](#) ^

Answer: B is the correct option.

The Flux architecture pattern, which is often used in conjunction with React, helps to keep data flowing in a unidirectional manner, which simplifies the data flow and makes it easier to reason about the application's state.

67. What is the correct data flow sequence in the Flux architecture pattern?

- a. Dispatcher->Action->Store->View
- b. Action->Dispatcher->View->Store
- c. Action->Dispatcher->Store->View
- d. Action->Store->Dispatcher->View

[View Answer](#) ^

Answer: C is the correct option.

In the Flux architecture pattern, data flows in a unidirectional loop from the "Action" component to the "Dispatcher", then to the "Store", and finally to the "View" component.

68. What are the requirements for the keys given to a list of elements in React?

- a. Do not require to be unique
- b. Unique in the DOM
- c. Unique among the siblings only
- d. All of the above



[View Answer](#) ^

They do not need to be unique across the entire DOM.

69. How can you access the state of a component from inside of a member function?

- a. `this.getState()`
- b. `this.values`
- c. `this.prototype.stateValue`
- d. `this.state`

[View Answer](#) 

Answer: D is the correct option.

The state of a component can be accessed from inside a member function using the "this.state" syntax.

70. What is used in React.js to increase performance?

- a. Virtual DOM
- b. Original DOM
- c. Both original and virtual DOM
- d. None of the above

[View Answer](#) 

Answer: A is the correct option.

React.js uses a virtual DOM to improve performance, by allowing it to update only the parts of the real DOM that have changed, rather than updating the entire page.

This results in faster rendering and better user experience.



71. What is a state in React.js?



[View Answer](#) ^

Answer: B is the correct option.

A state in React.js is an object that stores data within a component.

It is used for managing component data that can change over time and affect the component's rendering.

72. Where are React.js components typically stored?

- a. Inside the js/components/ directory
- b. Inside the vendor/components/ directory
- c. Inside the external/components/ directory
- d. Inside the vendor/ directory

[View Answer](#) ^

Answer: A is the correct option.

React.js components are usually stored in the js/components/ directory, keeping them organized and easily accessible within a project structure.

73. What happens if the key attribute is not provided when looping through an array in JSX?

- a. The code will not [compile](#).
- b. [React](#) will automatically assign a default key to each element.
- c. Each element will have the same key, causing rendering issues.
- d. The element will not be rendered.

[View Answer](#) ^



Answer: C is the correct option.

performance.

74. How many elements can a valid React component return?

- a. 2
- b. 3
- c. 1
- d. 4

View Answer ^

Answer: C is the correct option.

A valid React component can only return a single element or a fragment, which can contain multiple elements.

75. How can a React app be created?

- a. install -g create-react-app
- b. npm install create-react-app
- c. npx create-react-app reactapp
- d. None of the above

View Answer ^

Answer: C is the correct option.

The npx command can be used to create a new React app with the create-react-app package.

The command "npx create-react-app [app-name]" creates a new React app with the given name.

76. What is the purpose of Babel in JavaScript development?

- a. A JavaScript transpiler
- b. A JavaScript interpreter
- c. A JavaScript compiler
- d. None of the above



Babel is a JavaScript tool used to transpile modern JavaScript code into an older, more widely-supported version.

This allows developers to write modern JavaScript syntax while still being compatible with older browsers and environments.

77. What is the purpose of the key attribute when looping through an array in JSX?

- a. To assign a unique identifier to each element.
- b. To change the order of elements in the array.
- c. To group elements together based on a shared property.
- d. To apply styles to individual elements.

[View Answer](#) 

Answer: A is the correct option.

The key attribute is used to provide a unique identifier for each element when looping through an array in JSX.

This helps React to optimize updates and avoid unnecessary re-rendering of elements.

78. What is the output of the following React code snippet?

```
import React from 'react';
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
    this.handleClick = this.handleClick.bind(this);
  }

  handleClick() {
    this.setState({ count: this.state.count + 1 });
  }
}
```



```
        </div>
      );
    }
  }

  export default Counter;
```

- a. It defines a class component that displays a counter that can be incremented using a button.
- b. It defines a functional component that displays a counter that can be incremented using a button.
- c. It defines a higher-order component that displays a counter that can be incremented using a button.
- d. None of the above.

[View Answer](#) ^

Answer: A is the correct option.

The code defines a React class component named Counter, which has a constructor that initializes the state with a count property of 0, and binds the handleClick method to the instance.

The handleClick method increments the count property of the state using the setState method.

The render method returns a div element with a paragraph element that displays the count property of the state, and a button element with an onClick event handler that calls the handleClick method to increment the count.

79. In which programming language is React.js written?

- a. JavaScript
- b. Python
- c. Java
- d. Php

[View Answer](#) ^



80. Which of the following React code snippets will render a component that displays a list of names from an array of strings?

A.

```
function NameList(props) {  
  const names = props.names;  
  const listItems = names.map((name) =>  
    <li>{name}</li>  
  );  
  return (  
    <ul>{listItems}</ul>  
  );  
}  
ReactDOM.render(  
  <NameList names={['Alice', 'Bob', 'Charlie']} />,  
  document.getElementById('root')  
)
```

B.

```
function NameList(props) {  
  const names = props.names;  
  const listItems = names.map((name) =>  
    <li>{name}</li>  
  );  
  return (  
    <ul>{listItems}</ul>  
  );  
}  
ReactDOM.render(  
  <NameList names={['Alice', 'Bob', 'Charlie']} />,  
  document.querySelector("#root")  
)
```

C.



```

        <li>{name}</li>
      );
      return (
        <ul>{listItems}</ul>
      );
    }
  }
}
ReactDOM.render(
  <NameList names={['Alice', 'Bob', 'Charlie']} />,
  document.getElementById('root')
);

```

D.

```

class NameList extends React.Component {
  render() {
    const names = this.props.names;
    const listItems = names.map((name) =>
      <li>{name}</li>
    );
    return (
      <ul>{listItems}</ul>
    );
  }
}
ReactDOM.render(
  <NameList names={['Alice', 'Bob', 'Charlie']} />,
  document.querySelector('#root')
);

```

- a. A
- b. B
- c. C
- d. D

[View Answer](#) ^



Answer: D is the correct option.



The correct code snippet is D because it defines a React class component named NameList that

81. How do you add a key to each element when looping through an array in JSX?

- a. Use the index of the element as the key.
- b. Use a unique identifier as the key.
- c. Keys are not necessary when looping through an array in JSX.
- d. Use a random number generator to create a key.

[View Answer](#) 

Answer: B is the correct option.

When looping through an array in JSX, it's important to provide a unique key for each element.

This helps React to optimize updates and avoid unnecessary re-rendering of elements.

82. How do you import a named export from a module in React?

- a. `import NamedExport from './Module';`
- b. `import { NamedExport } from './Module';`
- c. `import * as NamedExport from './Module';`
- d. `import { name as NamedExport } from './Module';`

[View Answer](#) 

Answer: B is the correct option.

When importing a named export from a module in React, you use the `import` keyword followed by braces around the name of the export you want to import, then `from` and the path to the module file.

83. How do you export a component as a named export in a React module?

- a. `export MyComponent from './MyComponent';`
- b. `export { MyComponent };`



Answer: B is the correct option.

To export a component as a named export in a React module, you use the export keyword followed by braces around the name of the component, then assign the component to the name.

84. How do you export a component as the default export in a React module?

- a. export default MyComponent;
- b. export MyComponent from './MyComponent';
- c. export { MyComponent };
- d. module.exports = MyComponent;

[View Answer](#) ^

Answer: A is the correct option.

To export a component as the default export in a React module, you use the export default syntax followed by the name of the component.

85. What is the correct syntax for importing a default export from a module in React?

- a. import DefaultComponent from './Component';
- b. import { DefaultComponent } from './Component';
- c. import * as DefaultComponent from './Component';
- d. import { default } from './Component';

[View Answer](#) ^

Answer: A is the correct option.



When importing a default export from a module in React, you use the import keyword followed by

- a. SetinitialComponent
- b. renderComponent
- c. render
- d. All of the above

View Answer ^

Answer:C is the correct option.

Every React.js component must include the "render" API, which is a required method that returns the component's JSX representation and dictates what will be displayed on the screen.

87. In the context of the MVC pattern, what role does React.js play?

- a. Middleware
- b. Controller
- c. Model
- d. View

View Answer ^

Answer:D is the correct option.

React.js is often used as the "View" component in the Model-View-Controller (MVC) pattern, which separates an application's data (Model), user interface (View), and control flow (Controller) into separate components.

88. Who originally developed React.js?

- a. Jordan Walke
- b. Rasmus Lerdorf
- c. Miško Hevery
- d. None of the above

View Answer ^



89. How can we pass data to a component from outside in React.js?

- a. SetState
- b. Render with arguments
- c. Props
- d. PropTypes

View Answer ^

Answer:C is the correct option.

Props (short for properties) are used to pass data from one component to another in React.js.

90. Which of the following lifecycle events do React components have at the highest level?

- a. Destruction Initialization State/Property Updates
- b. Initialization State/Property Updates Destruction
- c. State/Property Updates Initialization Destruction
- d. All of the above

View Answer ^

Answer:D is the correct option.

React components have several lifecycle events, including initialization, state/property updates, and destruction.

91. Which of the following methods is not a part of ReactDOM in React.js?

- a. ReactDOM.destroy()
- b. ReactDOM.hydrate()
- c. ReactDOM.createPortal()
- d. ReactDOM.findDOMNode()



ReactDOM.destroy() is not a part of ReactDOM in React.js.

The other methods are used for rendering and manipulating components in the DOM.

92. In which of the following condition, the React.js Lifecycle method static `getDerivedStateFromProps(props, state)` is called?

- a. The component is created for the first time.
- b. The state of the component is updated.
- c. Both of the above.
- d. None of the above.

[View Answer](#) ^

Answer: C is the correct option.

`getDerivedStateFromProps` is called when a component is created for the first time and when its state is updated.

93. What is the purpose of the `shouldComponentUpdate` method in ReactJS?

- a. It is used to update the component's state.
- b. It is used to determine whether the component should be re-rendered.
- c. It is used to handle user input events.
- d. It is used to fetch data from an external API.

[View Answer](#) ^

Answer: B is the correct option.

The `shouldComponentUpdate` method in ReactJS is used to determine whether the component should be re-rendered.

It is called before the component is re-rendered and should return a boolean value.



94. What is the `useLayoutEffect()` function used for in React?

- a. Completing the update
- b. Optimizing for all devices
- c. Changing the layout of the screen
- d. When we need the browser to paint before effects

[View Answer](#) ^

Answer: D is the correct option.

The `useLayoutEffect()` function is similar to `useEffect()`, but it is called synchronously after all DOM mutations have been applied, which makes it useful for scenarios where we need the browser to paint before executing the effect.

95. Which method is used to update a component every second in React?

- a. `componentDidUpdate()`
- b. `shouldComponentUpdate()`
- c. `componentDidMount()`
- d. `setInterval()`

[View Answer](#) ^

Answer: D is the correct option.

The `setInterval()` method is used to update a component every second in React by repeatedly calling a function after a certain amount of time.

96. What is the purpose of state in React?

- a. To store data that can be changed within the component
- b. To pass data from a parent component to a child component
- c. To render HTML content in the component
- d. None of the above



State is used to store data within a component that can change and cause the component to re-render.

97. Does React.js create a virtual DOM in memory?

- a. TRUE
- b. FALSE
- c. Can be true or false
- d. Cannot say

View Answer ^

Answer: A is the correct option.

React.js uses a virtual DOM (Document Object Model) as an abstraction of the real DOM, which is a representation of the HTML structure of a web page.

The virtual DOM allows React to efficiently update the real DOM only where necessary, resulting in better performance.

98. What is the difference between Imperative and Declarative in ReactJS?

- a. Imperative is used for describing how the [UI](#) should be updated while declarative is used for describing what the UI should look like.
- b. Imperative is used for describing what the UI should look like while declarative is used for describing how the UI should be updated.
- c. Imperative is used for passing data between components while declarative is used for managing component state.
- d. Imperative is used for handling user input events while declarative is used for rendering static data.

View Answer ^



Answer: A is the correct option.

instructions.

In contrast, Declarative programming involves describing what the UI should look like based on certain conditions or events, without specifying how the updates should be made.

This can often be accomplished using higher-level abstractions provided by ReactJS, like components, props, and state.

99. How to perform automatic redirect after login in React?

- a. Use the history prop to redirect the user.
- b. Use the setState method to update the page.
- c. Use the location prop to redirect the user.
- d. Use the Router component to redirect the user.

[View Answer](#) ^

Answer: A is the correct option.

To perform an automatic redirect after login in React, we can use the history object provided by the react-router-dom library.

After a successful login, we can push a new path to the history object to redirect the user to a new page.

For example, we can use `history.push('/dashboard')` to redirect the user to the dashboard page.

100. What method can be used to loop through an array in JSX?

- a. for-loop
- b. map()
- c. forEach()
- d. while-loop

[View Answer](#) ^



101. What is the syntax for looping through an array in JSX?

- a. `{for (let i = 0; i < array.length; i++) {}}`
- b. `{array.forEach((item) => {})}`
- c. `{array.map((item) => {})}`
- d. `{while (i < array.length) {}}`

View Answer 

Answer: C is the correct option.

In JSX, we can use curly braces to execute JavaScript expressions.

To loop through an array, we use the `map()` method and return a new array of elements.

102. What is the purpose of using StrictMode in React?

- a. To enforce best practices and detect potential problems.
- b. To disable warnings and errors in the console.
- c. To optimize the rendering performance.
- d. None of the above.


View Answer 

Answer: A is the correct option.

StrictMode is a developer tool that highlights potential problems in an application.

It enforces best practices and enables additional checks and warnings in the development mode, helping developers to detect and fix issues early in the development process.

103. Which of the following is not a limitation of React?

- a.  [React](#) can be difficult to learn for beginners.
- b. React has a steep learning curve.



Answer: C is the correct option.

React applications are typically faster than traditional server-side applications because they use a virtual DOM and only update the parts of the page that have changed.

This reduces the number of DOM manipulations required, which can significantly improve performance.

104. Which of the following is a limitation of React?

- a. React can only be used for client-side rendering.
- b. React does not support server-side rendering.
- c. React is not suitable for large-scale applications.
- d. React does not have built-in support for animations.

[View Answer](#) ^

Answer: B is the correct option.

One of the limitations of React is that it does not support server-side rendering out of the box.

This means that React applications cannot be rendered on the server and sent to the client as HTML, which can affect the performance of the application.

105. How do you validate an object with specific properties in React?

- a. Use the object validator
- b. Use the shape validator
- c. Use the arrayOf validator
- d. Objects are automatically validated in React

[View Answer](#) ^



Answer: B is the correct option.

boolean prop in React.

- a. bool
- b. number
- c. oneOf
- d. oneOfType

View Answer ^

Answer: B is the correct option.

The number validator is used for validating numeric props in React, not boolean props.

The correct validator for a boolean prop is bool.

107. Which method in React.js refers to the parent class?

- a. inherits()
- b. self()
- c. super()
- d. this()

View Answer ^

Answer: C is the correct option.

The super() method is used to call methods in the parent class in React.js, allowing for inheritance and the extension of functionality.

108. What is the result of rendering an input element with disabled={false} in React.js?

- a. It will be rendered as disabled.
- b. It will not be rendered at all.
- c. It will be rendered as enabled.
- d. You cannot set it false.



In React.js, the disabled attribute expects a boolean value. When set to false, it will render as enabled.

109. What is the purpose of the ReactJS Context API?

- a. It is used to pass data between components
- b. It is used to manage the component's state
- c. It is used to handle user input events
- d. It is used to fetch data from an external API

[View Answer](#) ^

Answer: A is the correct option.

The ReactJS Context API is used to pass data between components without having to pass the data through every intermediate component.

It provides a way to share data across the component tree without having to pass props down manually at every level.

110. What is the purpose of the "webpack" command in React.js?

- a. To transpile JavaScript into one file
- b. To run the local development server
- c. To bundle modules
- d. None of the above

[View Answer](#) ^

Answer: A is the correct option.

The "webpack" command is used to transpile all the JavaScript down into one file.



- c. The process of comparing and updating the state of a component
- d. The process of comparing and updating the props of a component

[View Answer](#) ^

Answer: A. The process of comparing and updating the virtual DOM

Reconciliation is the process of comparing and updating the virtual DOM.

112. When does reconciliation occur in ReactJS?

- a. When a component's state changes
- b. When a component's props change
- c. When a component mounts or unmounts
- d. All of the above

[View Answer](#) ^

Answer: D. All of the above

Reconciliation occurs when a component's state or props change, or when a component mounts or unmounts.

113. Which algorithm does ReactJS use for reconciliation?

- a. Breadth-first search
- b. Depth-first search
- c. Merge sort
- d. Quick sort

[View Answer](#) ^

Answer: B. Depth-first search



ReactJS uses a depth-first search algorithm for reconciliation.

- b. ReactJS throws an error.
- c. ReactJS updates the first component with the properties of the second component.
- d. ReactJS updates the second component with the properties of the first component.

View Answer ^

Answer: B. ReactJS throws an error.

When two components have the same key in ReactJS, ReactJS throws an error.

Each key must be unique.

115. Which method is used to compare two React elements in the reconciliation process?

- a. shouldUpdate()
- b. componentWillUpdate()
- c. componentDidUpdate()
- d. render()

View Answer ^

Answer: D. render()

The render() method is used to create a new tree of React elements during the reconciliation process, which is then compared with the previous tree to determine which elements need to be updated.

116. What is the difference between "shallow" and "deep" rendering in React testing?

- a. Shallow rendering only renders the top-level component, while deep rendering renders all child components as well.
- b. Shallow rendering only renders the virtual DOM, while deep rendering renders the real DOM.
- c. Shallow rendering only renders components that have been updated since the previous render, while deep rendering renders all components.
- d. There is no difference between shallow and deep rendering in React testing.



all child components as well.

Shallow rendering in React testing only renders the top-level component and none of its children, while deep rendering renders all child components as well.

117. What is a limitation of using React with server-side rendering?

- a. Server-side rendering can increase the load time of the application.
- b. Server-side rendering can make the [application](#) less scalable.
- c. Server-side rendering can cause compatibility issues with certain browsers.
- d. Server-side rendering can make it harder to debug the application.

View Answer ^

Answer: B. Server-side rendering can make the application less scalable.

While server-side rendering can improve the performance of React applications, it can also make the application less scalable because it requires more resources to render the application on the server.

This can cause issues such as slower load times and increased server load.

React Hooks

118. Which hook in React can be used to share state between components?

- a. useMemo
- b. useCallback
- c. useEffect
- d. useContext

View Answer ^



Answer: D is the correct option.

119. Which hook in React can be used to memoize a function?

- a. useMemo
- b. useCallback
- c. useEffect
- d. useState

[View Answer](#) ^

Answer: B is the correct option.

The useCallback hook in React can be used to memoize a function.

It memoizes the result of a function, and only re-creates the function if the inputs have changed.

120. The useEffect hook in React can be used to subscribe to events. Which cleanup function should be returned to unsubscribe?

- a. removeEventListener
- b. clearInterval
- c. clearTimeout
- d. off

[View Answer](#) ^

Answer: A is the correct option.

When subscribing to events using the useEffect hook in React, the cleanup function should be returned to unsubscribe.

The removeEventListener is used to unsubscribe.

121. Which hook in React can be used to optimize performance by preventing unnecessary renders?



[View Answer](#) ^

Answer: A is the correct option.

The useMemo hook in React can be used to optimize performance by preventing unnecessary renders.

It memoizes the result of a function, and only re-executes the function if the inputs have changed.

122. In the useEffect hook, the second argument is used to specify:

- a. The callback function to execute
- b. The dependencies to watch for changes
- c. The initial state of the component
- d. The time interval for the effect to run

[View Answer](#) ^

Answer: B is the correct option.

The second argument to the useEffect hook in React is an array of dependencies that the effect should watch for changes.

If any of the dependencies change, the effect will be re-executed. If the array is empty, the effect will only run once.

123. What is the purpose of the useContext hook in React?

- a. To share state between components without using props.
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.



[View Answer](#) ^

124. What is the purpose of the useCallback hook in React?

- a. To memoize a function to prevent unnecessary re-renders.
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.

View Answer 

Answer: A. To memoize a function to prevent unnecessary re-renders.

The useCallback hook in React is used to memoize a function to prevent unnecessary re-renders, by returning a memoized version of the function that only changes if its dependencies change.

125. What is the purpose of the useEffect hook in React?

- a. To perform side effects after rendering a component.
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.

View Answer 

Answer: A. To perform side effects after rendering a component.

The useEffect hook in React is used to perform side effects after rendering a component, such as fetching data from an API or subscribing to a WebSocket.

126. What is the purpose of the useLayoutEffect hook in React?

- a. To perform side effects before rendering a component.
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.



The `useLayoutEffect` hook in React is used to perform side effects before rendering a component, such as measuring the size of a DOM element or updating a ref.

127. What is the difference between `useMemo` and `useCallback` hooks in React?

- a. `useMemo` is used for memoizing values, while `useCallback` is used for memoizing functions.
- b. `useCallback` is used for memoizing values, while `useMemo` is used for memoizing functions.
- c. `useMemo` and `useCallback` are the same thing.
- d. None of the above.

[View Answer](#) ^

Answer: A. `useMemo` is used for memoizing values, while `useCallback` is used for memoizing functions.

The `useMemo` and `useCallback` hooks in React are used for memoizing values and functions, respectively.

`useMemo` is used for memoizing values, while `useCallback` is used for memoizing functions.

128. What is the purpose of the `useRef` hook in React?

- a. To create a mutable reference to a value that persists across renders.
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.

[View Answer](#) ^

Answer: A. To create a mutable reference to a value that persists across renders.

The `useRef` hook in React is used to create a mutable reference to a value that persists across renders, by returning an object with a `current` property that can be updated.



- c. To render a component to the DOM.
- d. None of the above.

View Answer ^

Answer: A. To expose a component's imperative API to its parent component.

The useImperativeHandle hook in React is used to expose a component's imperative API to its parent component, by defining functions that can be called from the parent component using a ref.

130. What is the purpose of the useDebugValue hook in React?

- a. To display custom labels in the [React DevTools](#).
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.

View Answer ^

Answer: A. To display custom labels in the React DevTools.

The useDebugValue hook in React is used to display custom labels in the React DevTools, by accepting a value and a formatter function that returns a label.

131. What is the purpose of the useTransition hook in React?

- a. To defer rendering of a component until after a certain time has elapsed.
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.

View Answer ^



Answer: A. To defer rendering of a component until after a certain time has elapsed.

132. What is the purpose of the useSubscription hook in React?

- a. To subscribe to an external event source and update the state accordingly.
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.

[View Answer](#) ^

Answer: A. To subscribe to an external event source and update the state accordingly.

The useSubscription hook in React is used to subscribe to an external event source and update the state accordingly, by accepting a callback that will be called with the event data.

133. What is the purpose of the useIntersect hook in React?

- a. To observe when an element intersects with the viewport and update the state accordingly.
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.

[View Answer](#) ^

Answer: A. To observe when an element intersects with the viewport and update the state accordingly.

The useIntersect hook in React is used to observe when an element intersects with the viewport and update the state accordingly, by accepting a callback that will be called when the intersection occurs.

React Router

134. Which library is used for routing in React applications?



- A. React Router
- B. [Redux](#)

Answer: A. React Router

React Router is a popular library used for routing in React applications. It allows you to handle client-side routing and keep your UI in sync with the URL.

135. What is React Router?

- a. A library for handling server-side routing in React
- b. A library for handling client-side routing in React
- c. A library for styling in React
- d. None of the above

View Answer ^

Answer: B is the correct option.

React Router is a library for handling client-side routing in React applications.

136. Which component is used to render a Route in React Router?

- a. BrowserRouter
- b. Route
- c. Link
- d. Switch

View Answer ^

Answer:B is the correct option.

The Route component is used to render a particular route in React Router.

137. How do you pass parameters in React Router?

- a. As query strings
- b. As URL parameters



Answer: B is the correct option.

You can pass parameters in React Router as URL parameters, which can be accessed in the component using `props.match.params`.

138. How can you implement a default or NotFound page using React Router?

- a. Use the component.
- b. Use the component.
- c. Use the component.
- d. Use the component.

[View Answer](#) ^

Answer: C is the correct option.

The component matches any route that is not matched by any other route.

This can be used to implement a default or NotFound page in your React Router application.

139. Which of the following is an advantage of using React Router?

- a. Improved performance
- b. Improved user experience
- c. Improved SEO
- d. All of the above

[View Answer](#) ^

Answer: D is the correct option.



Using React Router can improve performance, user experience, and SEO in React applications.

C. To render a component to the DOM.

D. None of the above.

[View Answer](#) ^

Answer: A. To provide a way to handle routing in React applications.

The React Router library is used to provide a way to handle routing in React applications, by allowing developers to define routes and render different components based on the URL.

141. How do you redirect to a new route in React Router?

- a. By using the Redirect component
- b. By using the Route component with the "redirect" attribute
- c. By using the Switch component with the "redirect" attribute
- d. None of the above

[View Answer](#) ^

Answer: A is the correct option.

The Redirect component can be used to redirect to a new route in React Router.

142. Which of the following is NOT a type of routing provided by React Router?

- a. Client-side routing
- b. Server-side routing
- c. Static routing
- d. Dynamic routing

[View Answer](#) ^



Answer: C is the correct option.

Router.

- a. BrowserRouter
- b. Route
- c. Link
- d. Switch

View Answer ^

Answer: B is the correct option.

The Route component can be used to create a nested route in React Router.

144. Which component is used to create a link in React Router?

- a. BrowserRouter
- b. Route
- c. Link
- d. Switch

View Answer ^

Answer: C is the correct option.

The Link component is used to create a link in React Router, which navigates to a specified route when clicked.

145. Which of the following is NOT a type of router provided by React Router?

- a. BrowserRouter
- b. HashRouter
- c. MemoryRouter
- d. HttpRouter

View Answer ^



146. What is the purpose of the Switch component in React Router?

- A. To render multiple components based on the current route.
- B. To render a single component based on the current route.
- C. To handle user authentication in React Router.
- D. None of the above.

[View Answer](#) ^

Answer: B. To render a single component based on the current route.

The Switch component in React Router is used to render a single component based on the current route, and is typically used to wrap multiple Route components.

147. What is the difference between a Route component and a NavLink component in React Router?

- A. A Route component is used to render a component based on the current route, while a NavLink component is used to create links to different routes.
- B. A NavLink component is used to render a component based on the current route, while a Route component is used to create links to different routes.
- C. Route components and NavLink components are the same thing.
- D. None of the above.

[View Answer](#) ^

Answer: A. A Route component is used to render a component based on the current route, while a NavLink component is used to create links to different routes.

A Route component in React Router is used to render a component based on the current route, while a NavLink component is used to create links to different routes that can be clicked to navigate to other parts of the application.



148. What is the purpose of the withRouter higher-order component in React Router?



[View Answer](#) ^

Answer: B. To enable access to the history object in a component.

The withRouter higher-order component in React Router is used to enable access to the history object in a component, which can be used to programmatically navigate to different routes.

149. What is the purpose of the Redirect component in React Router?

- A. To redirect the user to a different route.
- B. To render a component based on the current route.
- C. To handle user authentication in React Router.
- D. None of the above.

[View Answer](#) ^

Answer: A. To redirect the user to a different route.

The Redirect component in React Router is used to redirect the user to a different route based on certain conditions, such as if the user is not logged in or if the current URL is invalid.

150. What is the purpose of the Prompt component in React Router?

- A. To prompt the user before navigating to a different route.
- B. To render a component based on the current route.
- C. To handle user authentication in React Router.
- D. None of the above.

[View Answer](#) ^

Answer: A. To prompt the user before navigating to a different route.



The Prompt component in React Router is used to prompt the user with a message before

HashRouter components in React Router?

- A. BrowserRouter uses browser history, while HashRouter uses hash history.
- B. HashRouter uses browser history, while BrowserRouter uses hash history.
- C. BrowserRouter and HashRouter are the same thing.
- D. None of the above.

[View Answer](#) 

Answer: A. BrowserRouter uses browser history, while HashRouter uses hash history.

The BrowserRouter and HashRouter components in React Router are used to handle routing in a React application.

The difference between them is that BrowserRouter uses the HTML5 history API for navigation, while HashRouter uses the URL hash for navigation.

152. What is the purpose of the Link component in React Router?

- A. To render a component based on the current route.
- B. To create links to different routes.
- C. To handle user authentication in React Router.
- D. None of the above.

[View Answer](#) 

Answer: B. To create links to different routes.

The Link component in React Router is used to create links to different routes in the application, which can be clicked to navigate to other parts of the application.

153. What is the difference between the exact and strict props in a Route component in React Router?

- A. exact checks for an exact match of the URL path, while strict checks for a trailing slash.
- B. strict checks for an exact match of the URL path, while exact checks for a trailing slash.



Answer: A. exact checks for an exact match of the URL path, while strict checks for a trailing slash.

The exact and strict props in a Route component in React Router are used to specify how the route should be matched.

exact checks for an exact match of the URL path, while strict checks for a trailing slash.

154. What is the purpose of the NavLink component in React Router?

- A. To create links to different routes.
- B. To render a component based on the current route.
- C. To handle user authentication in React Router.
- D. None of the above.

View Answer ^

Answer: A. To create links to different routes.

The NavLink component in React Router is used to create links to different routes in the application, which can be clicked to navigate to other parts of the application.

It is similar to the Link component, but with added functionality for styling the link based on the current route.

155. What is the purpose of the withRouter higher-order component in React Router?

- A. To provide access to the history, location, and match props in any component.
- B. To update the state of a component.
- C. To render a component to the DOM.
- D. None of the above.

View Answer ^



that has access to the Router context.

156. What is the purpose of the `useLocation` hook in React Router?

- A. To access the current location of the app.
- B. To update the state of a component.
- C. To render a component to the DOM.
- D. None of the above.

View Answer ^

Answer: A. To access the current location of the app.

The `useLocation` hook in React Router is used to access the current location of the app, by returning an object with `pathname`, `search`, `hash`, and `state` properties.

157. What is the purpose of the `useParams` hook in React Router?

- A. To access the URL parameters of the current route.
- B. To update the state of a component.
- C. To render a component to the DOM.
- D. None of the above.

View Answer ^

Answer: A. To access the URL parameters of the current route.

The `useParams` hook in React Router is used to access the URL parameters of the current route, by returning an object with the parameter values.

158. What is the purpose of the `useHistory` hook in React Router?

- A. To access the browser history object and navigate programmatically.
- B. To update the state of a component.
- C. To render a component to the DOM.
- D. None of the above.



The useHistory hook in React Router is used to access the browser history object and navigate programmatically, by returning an object with push, replace, and go methods.

159. What is the purpose of the useNavigate hook in React Router?

- A. To navigate programmatically without needing access to the history object.
- B. To update the state of a component.
- C. To render a component to the DOM.
- D. None of the above.

[View Answer](#) ^

Answer: A. To navigate programmatically without needing access to the history object.

The useNavigate hook in React Router is used to navigate programmatically without needing access to the history object, by returning a function that can be called with a string path or an object with properties.

160. What is the purpose of the useRouteMatch hook in React Router DOM?

- A. To match the current URL to a route configuration and extract information.
- B. To update the state of a component.
- C. To render a component to the DOM.
- D. None of the above.

[View Answer](#) ^

Answer: A. To match the current URL to a route configuration and extract information.

The useRouteMatch hook in React Router DOM is used to match the current URL to a route configuration and extract information, by returning an object with properties that depend on the configuration.



- c. JavaScript Serialized Notation
d. Java Serialized Object Notation

[View Answer](#) ^

Answer: A is the correct option.

JSON is a lightweight data-interchange format that is easy for humans to read and write and easy for machines to parse and generate.

162. Which of the following is a valid JSON data type?

- a. Date
- b. Function
- c. Undefined
- d. All of the above

[View Answer](#) ^

Answer: C is the correct option.

Undefined is a valid JSON data type. However, Date and Function are not valid JSON data types.

163. What is the syntax for creating an empty JSON object?

- A. {}
- B. []
- C. ()
- D. //

[View Answer](#) ^

Answer: A is the correct option.



An empty JSON object is created using curly braces ({}). Follow the above code format.

- B. JSON.parse()
- C. JSON.stringifyify()
- D. JSON.parseify()

[View Answer](#) ^

Answer: A. JSON.stringify()

The JSON.stringify() method converts a JavaScript object to a JSON string.

165. Which method is used to convert a JSON string to a JavaScript object?

- A. JSON.stringify()
- B. JSON.parse()
- C. JSON.stringifyify()
- D. JSON.parseify()

[View Answer](#) ^

Answer: B. JSON.parse()

The JSON.parse() method parses a JSON string and returns a JavaScript object.

166. Which of the following is a valid JSON object?

- A. {name:"John",age:30,city:"New York"}
- B. ['name':'John','age':30,'city':'New York']
- C. {"name":"John","age":30,"city":"New York"}
- D. {"name"="John","age"=30,"city"="New York"}

[View Answer](#) ^



Answer: C. {"name":"John","age":30,"city":"New York"}

- A. ["apple", "banana", "orange"]
- B. {fruit: ["apple", "banana", "orange"]}
- C. {"fruit": "apple", "fruit": "banana", "fruit": "orange"}
- D. All of the above

[View Answer](#) 

Answer: A. ["apple", "banana", "orange"]

A valid JSON array is a comma-separated list of values enclosed in square brackets [].

useReducer hook

168. What is the purpose of the useReducer hook in React?

- a. To manage more complex state with a reducer function.
- b. To manage simple state without a reducer function.
- c. To manage state with an asynchronous function.
- d. None of the above.

[View Answer](#) 

Answer: A. To manage more complex state with a reducer function.

The useReducer hook in React is used to manage more complex state with a reducer function, by dispatching actions to update the state.

169. What is the difference between the useState and useReducer hooks in React?

- a. useState is used for managing simple state, while useReducer is used for managing more complex state.
- b. useReducer is used for managing simple state, while useState is used for managing more complex state.
- c. useState and useReducer are the same thing.
- d. None of the above.



more complex state.

The useState and useReducer hooks in React are used for managing state in functional components.

useState is used for managing simple state, while useReducer is used for managing more complex state with a reducer function.

React Redux

170. Which of the following libraries is used to manage state in React applications?

- a. [Redux](#)
- b. jQuery
- c. D3
- d. Bootstrap

View Answer ^

Answer: A. Redux

Redux is a popular JavaScript library used to manage state in React applications.

It provides a predictable state container for JavaScript apps, making it easier to write and maintain complex applications.

171. What is Redux?

- a. A [JavaScript library](#) for building user interfaces.
- b. A state management library for JavaScript applications.
- c. A CSS framework for styling web pages.
- d. None of the above.

View Answer ^



172. What is a Redux store?

- a. An object that holds the state of an application.
- b. A function that returns the state of an application.
- c. A component that renders the state of an application.
- d. None of the above.

View Answer 

Answer: A. An object that holds the state of an application.

A Redux store is an object that holds the state of an application and provides methods to update and access that state.

173. What is a Redux action?

- a. An object that describes an event in the application.
- b. A function that updates the state of the application.
- c. A component that renders the state of the application.
- d. None of the above.

View Answer 

Answer: A. An object that describes an event in the application.

A Redux action is an object that describes an event in the application, typically triggered by user interaction or some other event.

174. What is a Redux reducer?

- a. A function that updates the state of the application in response to an action.
- b. A component that renders the state of the application.
- c. A CSS framework for styling web pages.
- d. None of the above.



A Redux reducer is a function that updates the state of the application in response to an action, by returning a new state object based on the previous state and the action.

175. What is the purpose of the connect function in Redux?

- a. To connect a React component to the Redux store.
- b. To update the state of the Redux store.
- c. To render a component to the DOM.
- d. None of the above.

View Answer 

Answer: A. To connect a React component to the Redux store.

The connect function in Redux is used to connect a React component to the Redux store, by providing it with access to the store state and any actions that can be dispatched.

176. What is the purpose of the Provider component in Redux?

- a. To provide the Redux store to a React component tree.
- b. To update the state of the Redux store.
- c. To render a component to the DOM.
- d. None of the above.

View Answer 

Answer: A. To provide the Redux store to a React component tree.

The Provider component in Redux is used to provide the Redux store to a React component tree, by wrapping the root component and passing the store as a prop.

177. What is the purpose of the bindActionCreatorsCreators function in Redux?

- a. To bind action creators to the dispatch function of the Redux store.
- b. To update the state of the Redux store.



Answer: A. To bind action creators to the dispatch function of the Redux store.

The `bindActionCreators` function in Redux is used to bind action creators to the dispatch function of the Redux store, by returning a new object with the same keys as the original object but with each action creator wrapped in a dispatch call.

178. What is a Redux middleware?

- a. A function that runs after a reducer is called.
- b. A function that runs before a reducer is called.
- c. A function that handles async actions.
- d. None of the above.

[View Answer](#) ^

Answer: B. A function that runs before a reducer is called.

A Redux middleware is a function that intercepts an action before it reaches the reducer, and can modify or stop the action as needed.

179. What is the purpose of the thunk middleware in Redux?

- a. To allow actions to return functions instead of objects.
- b. To update the state of the Redux store.
- c. To render a component to the DOM.
- d. None of the above.

[View Answer](#) ^

Answer: A. To allow actions to return functions instead of objects.

The thunk middleware in Redux is used to allow actions to return functions instead of objects, which can be used to perform asynchronous operations such as API calls.



a. Redux is used for local state management, while React state is used for global state management.

c. Redux and React state are the same thing.

d. None of the above.

View Answer ^

Answer: A. Redux is used for global state management, while React state is used for local state management.

Redux is used for managing the global state of an application, while React state is used for managing local state within a component.

181. What is the purpose of the Redux DevTools?

a. To debug Redux applications.

b. To design Redux applications.

c. To manage Redux state.

d. None of the above.

View Answer ^

Answer: A. To debug Redux applications.

The Redux DevTools is a browser extension that allows developers to debug and inspect the state changes of a Redux application.

182. What is the purpose of the selector function in Redux?

a. To derive data from the state of the Redux store.

b. To update the state of the Redux store.

c. To render a component to the DOM.

d. None of the above.



View Answer ^

183. What is the purpose of the useContext hook in Redux?

- a. To provide a way to pass data through the component tree without having to pass props down manually at every level.
- b. To manage complex state logic in a more concise and predictable way.
- c. To update the state of a component.
- d. None of the above.

View Answer ^

Answer: A. To provide a way to pass data through the component tree without having to pass props down manually at every level.

The useContext hook in Redux is used to provide a way to access the Redux store from any child component without having to pass props down manually at every level.

184. What is the purpose of the useDispatch hook in Redux?

- a. To dispatch actions to the [Redux](#) store.
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.

View Answer ^

Answer: A. To dispatch actions to the Redux store.

The useDispatch hook in Redux is used to dispatch actions to the Redux store, by returning a reference to the dispatch function.

185. What is the purpose of the useSelector hook in Redux?

- a. To select a subset of the state from the Redux store.
- b. To update the state of a component.
- c. To render a component to the DOM.



Answer: A. To select a subset of the state from the Redux store.

The useSelector hook in Redux is used to select a subset of the state from the Redux store, by returning the result of a selector function.

186. What is the purpose of the combineReducers function in Redux?

- a. To combine multiple reducer functions into a single reducer function.
- b. To update the state of the Redux store.
- c. To render a component to the DOM.
- d. None of the above.

[View Answer](#) ^

Answer: A. To combine multiple reducer functions into a single reducer function.

The combineReducers function in Redux is used to combine multiple reducer functions into a single reducer function, which can be passed to createStore to create a single store object.

187. What is the purpose of the createStore function in Redux?

- a. To create a Redux store object.
- b. To update the state of the Redux store.
- c. To render a component to the DOM.
- d. None of the above.

[View Answer](#) ^

Answer: A. To create a Redux store object.

The createStore function in Redux is used to create a Redux store object, by passing in a reducer function and optionally an initial state.



188. What is the purpose of the applyMiddleware function in Redux?

- a. To apply middleware to the Redux store.

[View Answer](#) ^

Answer: A. To apply middleware to the Redux store.

The `applyMiddleware` function in Redux is used to apply middleware to the Redux store, by passing in one or more middleware functions.

189. What is the purpose of the Redux DevTools extension?

- a. To provide a [UI](#) for debugging Redux applications.
- b. To update the state of the Redux store.
- c. To render a component to the DOM.
- d. None of the above.

[View Answer](#) ^

Answer: A. To provide a UI for debugging Redux applications.

The Redux DevTools extension is used to provide a UI for debugging Redux applications, by allowing developers to inspect the state and actions of the store.

190. What is the purpose of the Immutable.js library?

- a. To provide a collection of immutable data structures.
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.

[View Answer](#) ^

Answer: A. To provide a collection of immutable data structures.

The Immutable.js library is used to provide a collection of immutable data structures, which can be used to represent state in Redux applications.



- c. A function that creates new components.
- d. None of the above.

[View Answer](#) ^

Answer: A. A function that selects a portion of the state.

A Redux selector is a function that selects a portion of the state.

192. What is the purpose of the bindActionCreatorsCreators function in Redux?

- a. To bind action creators to the Redux store dispatch function.
- b. To handle async actions in Redux.
- c. To update the state of a component.
- d. None of the above.

[View Answer](#) ^

Answer: A. To bind action creators to the Redux store dispatch function.

The bindActionCreatorsCreators function in Redux is used to bind action creators to the Redux store dispatch function, so that they can be easily called from a component.

193. What is the purpose of the applyMiddleware function in Redux?

- a. To apply middleware to the Redux store.
- b. To handle async actions in Redux.
- c. To update the state of a component.
- d. None of the above.

[View Answer](#) ^

Answer: A. To apply middleware to the Redux store.



The applyMiddleware function in Redux is used to apply middleware to the Redux store, by wrapping the store's dispatch method.

c. A function that returns a selector.

d. None of the above.

[View Answer](#) ^

Answer: A. A function that returns an action object.

A Redux action creator is a function that returns an action object, which is then dispatched to the Redux store.

195. What is the difference between Redux and Context API in React?

a. Redux is a separate library for state management, while Context API is built into React.

b. Redux is used for local state management, while Context API is used for global state management.

c. Redux and Context API are the same thing.

d. None of the above.

[View Answer](#) ^

Answer: A. Redux is a separate library for state management, while Context API is built into React.

Redux is a separate library that is commonly used for managing global state in a React application, while Context API is a built-in feature of React that can also be used for managing state, although it's typically used for smaller-scale applications.

196. What is the purpose of the createSelector() function in Redux?

a. To create a selector function for the Redux state.

b. To create a reducer function for the Redux store.

c. To create an action creator function for Redux.

d. None of the above.

[View Answer](#) ^



197. What is the purpose of the Redux Saga library?

- a. To handle async actions in Redux.
- b. To manage routing in a Redux app.
- c. To create new components in [React](#) Redux.
- d. None of the above.

View Answer 

198. What is the difference between a presentational component and a container component in React Redux?

- a. A presentational component is used for displaying data, while a container component is used for managing data.
- b. A presentational component is used for managing data, while a container component is used for displaying data.
- c. A presentational component and a container component are the same thing.
- d. None of the above.

View Answer 

Answer: A. A presentational component is used for displaying data, while a container component is used for managing data.

Presentational components are concerned with how things look and are responsible for rendering data to the UI, while container components are concerned with how things work and are responsible for managing the state of the application.

199. What is the purpose of the Redux DevTools extension?

- a. To help developers debug and analyze the Redux store and actions.
- b. To create new components in React Redux.
- c. To manage routing in a Redux app.
- d. None of the above.



The Redux DevTools extension is a browser extension that helps developers debug and analyze the Redux store and actions, providing features such as time travel debugging and state inspection.

200. What is the difference between a Redux action and a Redux thunk?

- a. A Redux action is a plain JavaScript object that describes a change to the state, while a Redux thunk is a function that allows for more complex asynchronous logic.
- b. A Redux action is a function that allows for more complex asynchronous logic, while a Redux thunk is a plain JavaScript object that describes a change to the state.
- c. A Redux action and a [Redux](#) thunk are the same thing.
- d. None of the above

View Answer ^

Answer: A. A Redux action is a plain JavaScript object that describes a change to the state, while a Redux thunk is a function that allows for more complex asynchronous logic.

A Redux action is a plain JavaScript object that describes a change to the state, while a Redux thunk is a function that allows for more complex asynchronous logic, such as making API calls or dispatching multiple actions.

201. What is the purpose of the connect function in React Redux?

- a. To connect a component to the Redux store.
- b. To handle async actions in Redux.
- c. To update the state of a component.
- d. None of the above.

View Answer ^

Answer: A. To connect a component to the Redux store.



The connect function in React Redux is used to connect a React component to the Redux store, by providing it with access to the store's state and actions.

- c. To update the state of a component.
- d. None of the above.

[View Answer](#) ^

Answer: A. To provide the Redux store to all components in a React application.

The Provider component in React Redux is used to provide the Redux store to all components in a React application, by wrapping the top-level component.

203. What is the difference between dispatching a plain object and a function in Redux?

- a. Dispatching a plain object updates the state immediately, while dispatching a function allows for more complex asynchronous logic.
- b. Dispatching a function updates the state immediately, while dispatching a plain object allows for more complex asynchronous logic.
- c. Dispatching a plain object and a function are the same thing.
- d. None of the above.

[View Answer](#) ^

Answer: A. Dispatching a plain object updates the state immediately, while dispatching a function allows for more complex asynchronous logic.

Dispatching a plain object to the Redux store will immediately update the state, while dispatching a function allows for more complex asynchronous logic, such as making API calls or dispatching multiple actions.

204. What is the purpose of the createSelector() function in the Reselect library?

- a. To generate selectors from the Redux state.
- b. To combine multiple selectors into a single selector function.
- c. To memoize selectors for performance optimization.
- d. None of the above.



The `createSelector()` function in the Reselect library is used to memoize selectors, which can improve the performance of the application by caching the results of the selector and only recalculating it when necessary.

205. How do you use React Router with Redux?

- a. By creating a new store for React Router
- b. By using the react-redux-router library
- c. By passing the router as a prop to each component
- d. None of the above

[View Answer](#) ^

Answer: B. By using the react-redux-router library

The react-redux-router library can be used to integrate React Router with Redux.

206. What is the difference between `mapDispatchToProps` and `mapStateToProps` in React Redux?

- a. `mapDispatchToProps` is used to map action creators to props, while `mapStateToProps` is used to map state to props.
- b. `mapDispatchToProps` is used to map state to props, while `mapStateToProps` is used to map action creators to props.
- c. `mapDispatchToProps` and `mapStateToProps` are the same thing.
- d. None of the above.

[View Answer](#) ^

Answer: A. `mapDispatchToProps` is used to map action creators to props, while `mapStateToProps` is used to map state to props.

`mapDispatchToProps` is used to map action creator functions to props, which allows components to dispatch actions to the Redux store, while `mapStateToProps` is used to map state to props, which allows components to access and display data from the Redux store.



- c. To enable immutable updates to the Redux state
- d. None of the above

View Answer ^

Answer: C. To enable immutable updates to the Redux state

The immer library is used in Redux to enable immutable updates to the state, which can simplify the process of updating nested data structures.

Redux Toolkit

208. What is the purpose of the Redux Toolkit?

- a. To simplify the process of creating Redux [applications](#).
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.

View Answer ^

Answer: A. To simplify the process of creating Redux applications.

The Redux Toolkit is used to simplify the process of creating Redux applications, by providing a set of utilities and abstractions that make it easier to write and manage Redux code.

209. What is the purpose of the createAsyncThunk function in the Redux Toolkit?

- a. To create an action creator that dispatches async actions.
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.



View Answer ^

fulfilled, and rejected actions.

210. What is the purpose of the createSlice() function in Redux Toolkit?

- a. To create a new slice of the Redux store.
- b. To create a new component in React Redux.
- c. To create a new middleware function for Redux.
- d. None of the above.

View Answer ^

Answer: A. To create a new slice of the Redux store.

The createSlice() function is used in Redux Toolkit to create a new slice of the Redux store, which includes a reducer function and action creators.

React Lifecycle

211. What is the lifecycle method used to make AJAX requests in React?

- a. componentWillMount()
- b. componentDidMount()
- c. componentWillUnmount()
- d. componentDidUpdate()

View Answer ^

Answer: B is the correct option.

The componentDidMount() method is used to make AJAX requests in React, typically to fetch data from an external API.



- b. componentWillMount()
- c. componentWillUnmount()
- d. render()

[View Answer](#) ^

Answer: B is the correct option.

The componentWillMount() method is called when a component is first initialized and before it is rendered for the first time.

213. The useEffect hook in React can be used to replace which class lifecycle method?

- a. componentWillMount
- b. componentWillUpdate
- c. componentDidUpdate
- d. shouldComponentUpdate

[View Answer](#) ^

Answer: C is the correct option.

The useEffect hook in React can be used to replace the componentDidUpdate class lifecycle method.

The effect will be executed after the component has updated.

214. What is the purpose of the shouldComponentUpdate() method in React?

- a. To determine if a component should be updated.
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.



The `shouldComponentUpdate()` method in React is used to determine if a component should be updated based on changes to its props or state, and can be used to optimize performance by avoiding unnecessary updates.

215. What is the purpose of the `componentWillUnmount()` method in React?

- a. To clean up resources used by a component before it is removed from the DOM.
- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.

[View Answer](#) 

Answer: A is the correct option.

The `componentWillUnmount()` method in React is used to clean up any resources used by a component before it is removed from the DOM, such as event listeners or timers.

216. Which lifecycle method is called before a component is removed from the DOM?

- a. `componentDidMount()`
- b. `componentWillMount()`
- c. `componentWillUnmount()`
- d. `componentDidUpdate()`

[View Answer](#) 

Answer: C is the correct option.

The `componentWillUnmount()` method is called before a component is removed from the DOM.



217. Which lifecycle method is called when a component is forced to re-render?



[View Answer](#) ^

Answer: A is the correct option.

The `componentWillUpdate()` method is called when a component is forced to re-render.

218. Which lifecycle method is called when a component receives new props?

- a. `componentWillReceiveProps()`
- b. `shouldComponentUpdate()`
- c. `componentDidUpdate()`
- d. `componentWillUnmount()`

[View Answer](#) ^

Answer: A is the correct option.

The `componentWillReceiveProps()` method is called when a component receives new props.

219. Which lifecycle method is called after a component is rendered for the first time?

- a. `componentDidMount()`
- b. `componentWillMount()`
- c. `componentWillUnmount()`
- d. `render()`

[View Answer](#) ^

Answer: A is the correct option.



The `componentDidMount()` method is called after a component is rendered for the first time.

- b. To update the state of a component.
- c. To render a component to the DOM.
- d. None of the above.

[View Answer](#) ^

Answer: A is the correct option.

The componentDidCatch() method in React is used to handle errors that occur during rendering, such as when a component throws an error or when a network request fails.

221. Which lifecycle method is used to optimize performance by avoiding unnecessary re-renders?

- a. shouldComponentUpdate()
- b. componentWillUpdate()
- c. componentDidUpdate()
- d. render()

[View Answer](#) ^

Answer: A is the correct option.

The shouldComponentUpdate() method is used to optimize performance by avoiding unnecessary re-renders.


222. Which lifecycle method is used to set initial state values?

- a. componentWillMount()
- b. componentDidMount()
- c. componentWillReceiveProps()
- d. constructor()

[View Answer](#) ^



223. Which React API method is used to render a React component to the DOM?

- a. ReactDOM.render()
- b.  [React](#).render()
- c. ReactDOM.component()
- d. React.component()

View Answer 

Answer: A is the correct option.

The ReactDOM.render() method is used to render a React component to the DOM.

224. Which React API method is used to create a new React component?

- a. React.createClass()
- b. ReactDOM.render()
- c. React.createElement()
- d. ReactDOM.component()

View Answer 

Answer: A is the correct option.

The React.createClass() method is used to create a new React component.

225. Which React API method is used to create a new React element?

- a. React.createClass()
- b. ReactDOM.render()
- c. React.createElement()
- d. ReactDOM.component()



The `React.createElement()` method is used to create a new React element.

226. Which React API method is used to access the current state of a component?

- a. `this.getState()`
- b. `this.values`
- c. `this.prototype.stateValue`
- d. `this.state`

[View Answer](#) ^

Answer: D is the correct option.

The `this.state` method is used to access the current state of a component.

227. Which React API method is used to handle events in React components?

- a. `eventHandler()`
- b. `handleEvent()`
- c. `handleOnClick()`
- d. `onClick()`

[View Answer](#) ^

Answer: D is the correct option.

The `onClick()` method is used to handle events in React components.

228. Which React API method is used to prevent default behavior in an event handler?

- a. `event.preventDefault()`
- b. `this.preventDefault()`



Answer: A is the correct option.

The `event.preventDefault()` method is used to prevent default behavior in an event handler.

229. Which React API method is used to create a new context object?

- a. `React.createContext()`
- b. `React.Context()`
- c. `createContext()`
- d. `createNewContext()`

[View Answer](#) ^

Answer: A is the correct option.

The `React.createContext()` method is used to create a new context object.

230. Which React API method is used to access the context object in a component?

- a. `this.context()`
- b. `this.contextValue()`
- c. `this.prototype.contextValue`
- d. `static contextType`

[View Answer](#) ^

Answer: D is the correct option.

The static `contextType` property is used to access the context object in a component.

231. Which React API method is used to update the component after rendering?

[View Answer](#) ^

Answer: C is the correct option.

The `componentDidUpdate()` method is used to update the component after rendering.

232. Which React API method is used to remove a component from the DOM?

- a. `componentWillUnmount()`
- b. `componentDidMount()`
- c. `componentDidUnmount()`
- d. `willUnmount()`

[View Answer](#) ^

Answer: A is the correct option.

The `componentWillUnmount()` method is used to remove a component from the DOM.

233. Which React API method is used to check if a component is mounted?

- a. `this.mounted()`
- b. `this.isMounted()`
- c. `this.componentMounted()`
- d. `this.isComponentMounted()`

[View Answer](#) ^

Answer: B is the correct option.



The `this.isMounted()` method is used to check if a component is mounted.

- b. this.update()
- c. this.render()
- d. this.refresh()

[View Answer](#) 

Answer: A is the correct option.

The this.forceUpdate() method is used to force a re-render of a component.

235. Which React API method is used to set the props of a component?

- a. this.setProps()
- b. this.updateProps()
- c. this.modifyProps()
- d. this.props()

[View Answer](#) 

Answer: A is the correct option.

The this.setProps() method is used to set the props of a component.

236. Which React API method is used to get the props of a component?

- a. this.getProps()
- b. this.props()
- c. this.retrieveProps()
- d. this.accessProps()

[View Answer](#) 



Answer: B is the correct option.

- a. `this.setState()`
- b. `this.modifyState()`
- c. `this.updateState()`
- d. `this.setComponentState()`

[View Answer](#) 

Answer: A is the correct option.

The `this.setState()` method is used to set the state of a component.

238. Which React API method is used to get the state of a component?

- a. `this.componentState()`
- b. `this.getCurrentState()`
- c. `this.state()`
- d. `this.retrieveState()`

[View Answer](#) 

Answer: C is the correct option.

The `this.state()` method is used to get the state of a component.

239. Which React API method is used to set the context of a component?

- a. `this.setContext()`
- b. `this.modifyContext()`
- c. `this.updateContext()`
- d. `this.context()`

[View Answer](#) 

Answer: A is the correct option.



- a. `this.refs()`
- b. `this.getRefs()`
- c. `this.retrieveRefs()`
- d. `this.componentRefs()`

[View Answer](#) ^

Answer: A is the correct option.

The `this.refs()` method is used to get the refs of a component.

241. What is the purpose of the "getSnapshotBeforeUpdate" lifecycle method in ReactJS?

- a. To capture information from the DOM before it is updated
- b. To update the state of a component
- c. To update the props of a component
- d. To update the DOM of a component

[View Answer](#) ^

Answer: A is the correct option.

The "getSnapshotBeforeUpdate" lifecycle method is used to capture information from the DOM before it is updated.

242. The `useEffect` hook in React can be used to replace which class lifecycle method?

- a. `componentDidMount`
- b. `componentWillUnmount`
- c. `componentWillReceiveProps`
- d. `render`

[View Answer](#) ^



The effect will be executed after the component has mounted.

React Testing

243. Jest is a testing framework built by Facebook for:

- a. React [applications](#)
- b. Angular applications
- c. Vue.js applications
- d. All of the above

View Answer 

Answer: A is the correct option.

Jest is a testing framework built by Facebook that is widely used for testing React applications.

244. What is the command to run Jest tests?

- a. npm test
- b. npm run jest
- c. npm start
- d. npm run test

View Answer 

Answer: A is the correct option.

The command "npm test" is used to run Jest tests in a React project.

245. What is the purpose of snapshots in Jest?



- a. To store test results
- b. To compare HTML output
- c. To generate code coverage reports



Answer: B is the correct option.

Snapshots in Jest are used to compare the HTML output of a component between test runs to ensure that it remains consistent.

246. What is the syntax for creating a Jest test suite?

- a. describe()
- b. it()
- c. test()
- d. all of the above

[View Answer](#) ^

Answer: A is the correct option.

The "describe()" function is used to create a test suite in Jest, and it can contain multiple test cases created with the "it()" or "test()" functions.

247. What is the purpose of the "beforeEach()" function in Jest?

- a. To run code after each test case
- b. To run code before each test suite
- c. To run code after each test suite
- d. To run code before each test case

[View Answer](#) ^

Answer: D is the correct option.

The "beforeEach()" function is used to run code before each individual test case in a Jest test suite.

248. What is the purpose of the "afterAll()" function in Jest?

- a. To run code before each test case
- b. To run code after each test case



Answer: D is the correct option.

The "afterAll()" function is used to run code after all test cases in a Jest test suite have completed.

249. What is the purpose of the "expect()" function in Jest?

- a. To define a test case
- b. To make assertions
- c. To run setup code
- d. To run teardown code

[View Answer](#) ^

Answer: B is the correct option.

The "expect()" function is used to make assertions in Jest tests, which allows for checking the results of a test case.

250. What is the purpose of the "toMatch()" function in Jest?

- a. To check if a value is truthy
- b. To compare strings using regular expressions
- c. To check if an object contains a property
- d. To compare arrays for equality

[View Answer](#) ^

Answer: B is the correct option.

The "toMatch()" function is used to compare strings using regular expressions in Jest tests.

251. Which library can be used to handle date and time in React applications?



- a. Moment.js

[View Answer](#) ^

Answer: A is the correct option.

Moment.js is a popular JavaScript library used to handle date and time in web applications.

It can be used in React applications to format, parse, and manipulate dates and times.

252. Which library can be used for internationalization in React applications?

- a. [Redux](#)
- b. Lodash
- c. React Intl
- d. Axios

[View Answer](#) ^

Answer: C is the correct option.

React Intl is a library used for internationalization (i18n) in React applications.

It provides a way to format dates, times, and numbers for different locales, and to handle translations.

253. Which library can be used for testing React components?

- a. Enzyme
- b. jQuery
- c. Axios
- d. Moment.js

[View Answer](#) ^



Answer: A is the correct option.

254. Which library can be used for state management in React applications instead of Redux?

- a. MobX
- b. Axios
- c. jQuery
- d. Lodash

[View Answer](#) 

Answer: A is the correct option.

MobX is a library used for state management in React applications.

It provides a way to manage state using observable objects and reactive programming.

255. Which library can be used for data visualization in React applications?

- a. React Motion
- b. D3.js
- c. Moment.js
- d. React Router

[View Answer](#) 

Answer: B is the correct option.

D3.js is a popular library used for data visualization in web applications.

It can be used in React applications to create charts, graphs, and other visualizations.

256. What is the recommended library for internationalization in React?



React?

- a. i18n.js



[View Answer](#) ^

Answer: C is the correct option.

React-intl is the recommended library for internationalization in React.

257. What is the purpose of the FormattedMessage component in react-intl?

- a. to translate text messages
- b. to format numbers
- c. to format dates and times
- d. to format currency values

[View Answer](#) ^

Answer: A is the correct option.

The FormattedMessage component is used to translate text messages in react-intl.

258. Which component from react-intl can be used to format dates and times?

- a. FormattedDate
- b. FormattedTime
- c. FormattedDateTime
- d. FormattedMessage

[View Answer](#) ^

Answer: C is the correct option.

The FormattedDateTime component can be used to format dates and times in react-intl.



- b. getLocale
- c. getUserLocale
- d. IntlProvider

[View Answer](#) 

Answer: B is the correct option.

The getLocale function can be used to get the user's locale in react-intl.

260. What is the purpose of the "toBe()" function in Jest?

- a. To check if a value is truthy
- b. To compare strings using regular expressions
- c. To check if an object contains a property
- d. To compare values for strict equality

[View Answer](#) 

Answer: D. To compare values for strict equality

The "toBe()" function is used to compare values for strict equality in Jest tests.

261. Which library can be used for server-side rendering in React applications?

- a. Next.js
- b. React Motion
- c. Lodash
- d. Axios

[View Answer](#) 



Answer: A. Next.js

Next.js is a framework used for server-side rendering (SSR) in React applications.



262. Which library can be used for managing forms in React applications?

- a. Formik
- b. React Motion
- c. Moment.js
- d. Redux

View Answer 

Answer: A. Formik

Formik is a library used for managing forms in React applications.

It provides a simple and intuitive way to handle form validation, submission, and state management.

263. Which package is commonly used for animation in React?

- A. react-transition-group
- B. react-router
- C. redux
- D. axios

View Answer 

Answer: A is the correct option.

react-transition-group is a commonly used package for animating React components.

264. Which of the following properties can be used to create staggered animations with react-transition-group?

- a. delay
- b. duration
- c. transitionTime



Answer: A is the correct option.

The delay property can be used to create staggered animations with react-transition-group, allowing you to create more complex animations.

265. Which method in react-transition-group is used to animate components when they mount?

- a. CSSTransition
- b. Transition
- c. animateOnMount
- d. None of the above

View Answer ^

Answer: A is the correct option.

The CSSTransition component is used to animate components when they mount, unmount, or change in React.

266. Which animation library can be used with React for more advanced animations?

- a. GreenSock
- b. jQuery
- c. Animate.css
- d. Bootstrap

View Answer ^

Answer: A is the correct option.

GreenSock is an animation library that can be used with React to create more advanced and complex animations.



- b. D3.js
- c. Moment.js
- d. React Router

[View Answer](#) 

Answer: B is the correct option.

D3.js is a popular library used for data visualization in web applications.

It can be used in React applications to create charts, graphs, and other visualizations.

268. What is the purpose of the React Spring library?

- a. To create smooth animations and transitions in React
- b. To create responsive and mobile-friendly UIs in React
- c. To manage state in React applications
- d. None of the above

[View Answer](#) 

Answer: A. To create smooth animations and transitions in React

React Spring is a library that provides tools for creating smooth and advanced animations and transitions in React applications.

269. What is a fragment in React?

- a. A special type of component
- b. A way to group multiple elements without adding extra nodes to the DOM
- c. A way to render elements outside the component tree
- d. A way to add comments to the component tree

[View Answer](#) 



They allow you to return multiple elements from a component's render method without having to wrap them in a parent element.

This can be useful for cases where you don't want to add an extra div or other container element to your markup.

270. What is PropTypes in React?

- a. A built-in method to validate props
- b. A library for working with JavaScript types
- c. A tool for unit testing React components
- d. A way to handle errors in React components

[View Answer](#) ^

Answer: A. A built-in method to validate props

PropTypes is a built-in method in React that allows you to validate the types of props passed to a component.

It helps you catch errors early in development by providing warnings in the console when a prop of the wrong type is passed.

271. Which of the following is not a valid PropTypes validator in React?

- a. string
- b. array
- c. function
- d. objectOfNumber

[View Answer](#) ^

Answer: D. objectOfNumber



There is no objectOfNumber validator in PropTypes.

✓ . ✓ . ✓ . ✓ . ✓ .

- a. Use the `isRequired` validator
- b. Use the `isRequiredProp` keyword
- c. Use the `required` attribute
- d. Required props are automatically validated in React

[View Answer](#) ^

Answer: A. Use the `isRequired` validator

To specify a required prop in React, you can use the `isRequired` validator in addition to the type validator.

For example, `PropTypes.string.isRequired` will specify that the string prop is required.

273. How do you specify a default value for a prop in React?

- a. Use the `default` keyword
- b. Use the `defaultValue` keyword
- c. Use the `defaultProp` attribute
- d. Default props are automatically set in React

[View Answer](#) ^

Answer: C. Use the `defaultProp` attribute

To specify a default value for a prop in React, you can use the `defaultProp` attribute on the component.

For example, `MyComponent.defaultProps = { prop1: 'default value' }` will set the default value of `prop1` to 'default value'.

274. Which of the following is not a valid `PropTypes` validator for a function prop in React?



- a. `func`
- b. `shape`

^

Answer: B. shape

The shape validator is used for validating the shape of an object prop in React, not a function prop.

The correct validator for a function prop is func.

275. How do you validate an array of a specific type in React?

- a. Use the arrayOf validator
- b. Use the instanceof validator
- c. Use the shape validator
- d. Arrays are automatically validated in React

[View Answer](#) ^

Answer: A. Use the arrayOf validator

To validate an array of a specific type in React, you can use the arrayOf validator.

For example, `PropTypes.arrayOf(PropTypes.string)` will validate that the prop is an array of strings.

276. Which of the following is a limitation of React when it comes to testing?

- a. React does not have built-in testing capabilities.
- b. React testing can be difficult to set up and configure.
- c. React tests can be slow to run.
- d. React tests can be prone to false positives.

[View Answer](#) ^

Answer: B. React testing can be difficult to set up and configure.

While React has a number of testing frameworks available, setting up and configuring tests can be challenging, especially for developers who are new to the platform.



occur in which of the following?

- a. Render methods
- b. Constructor methods
- c. Lifecycle methods
- d. All of the above

View Answer ^

Answer: A. Render methods

Error boundaries in React are used to catch and handle errors that occur during rendering of components.

278. Which of the following is the correct way to catch errors in React?

- a. Using try-catch blocks
- b. Using error boundaries
- c. Using conditional statements
- d. Using promises

View Answer ^

Answer: B. Using error boundaries

Error boundaries are the recommended way to catch and handle errors in React.

279. Which of the following is the correct way to handle errors in React?

- a. By ignoring the errors
- b. By logging the errors to the console
- c. By displaying a user-friendly error message
- d. None of the above



View Answer ^



Ignoring errors or logging them to the console is not sufficient.

280. In React, what is the purpose of the error object that is passed to the componentDidCatch method?

- a. To provide information about the error that occurred
- b. To provide a stack trace of the error
- c. To provide a way to retry the rendering of the component
- d. None of the above

View Answer 

Answer: A. To provide information about the error that occurred

The error object that is passed to the componentDidCatch method contains information about the error that occurred, such as the error message and the stack trace.

281. What is the best practice for handling errors in React?

- a. To use error boundaries
- b. To use try-catch blocks
- c. To ignore the errors
- d. To log the errors to the console

View Answer 

Answer: A. To use error boundaries

Error boundaries are the recommended way to catch and handle errors in React.

282. In React, what is the purpose of the error boundary component?

- a. To catch and handle errors that occur in child components
- b. To catch and handle errors that occur in parent components
- c. To prevent errors from occurring in the first place
- d. None of the above



Error boundary components in React are used to catch and handle errors that occur in child components.

283. How can you automatically format code using Prettier in a React project when saving files in the editor?

- a. Prettier does not support automatic formatting on file save in [React](#) projects.
- b. Prettier automatically formats code on file save without any additional configuration.
- c. Prettier can be configured to format code on file save by using the ``.prettiignore`` file.
- d. Prettier can be configured to format code on file save by using the ``.prettierrc`` file along with editor-specific configuration.

View Answer 

Answer: D. Prettier can be configured to format code on file save by using the ``.prettierrc`` file along with editor-specific configuration.

Prettier can be set up to automatically format code on file save by configuring it in the ``.prettierrc`` file and enabling editor-specific configuration like ESLint or editor extensions to trigger formatting on save.

284. How can you configure ESLint and Prettier to work with an editor or IDE in a React project?

- a. ESLint and Prettier do not have editor or IDE integrations for React projects.
- b. ESLint and Prettier automatically integrate with the editor or IDE without any additional configuration.
- c. ESLint and Prettier can be integrated with an editor or IDE by installing their respective extensions and configuring the extension settings.
- d. ESLint and Prettier can be integrated with an editor or IDE by modifying the global editor configuration file.

View Answer 



extensions (e.g., "ESLint" and "Prettier - Code Formatter" in VS Code) and configure the extension settings to use the project-specific ESLint and Prettier configurations.

285. What is Prettier and how does it differ from ESLint in React development?

- a. Prettier is a code formatter that automatically formats code, while ESLint focuses on catching errors and enforcing coding standards.
- b. Prettier and ESLint are two different names for the same code formatting tool used in React development.
- c. Prettier is a tool for type-checking in React [applications](#), while ESLint is a code formatter.
- d. Prettier is a build tool that optimizes React components for production, while ESLint is a linter for catching errors.

View Answer 

Answer: A. Prettier is a code formatter that automatically formats code, while ESLint focuses on catching errors and enforcing coding standards.

Prettier is a code formatter that helps maintain consistent code style automatically.

It focuses on code formatting aspects like indentation, line wrapping, and more, while ESLint primarily catches errors and enforces coding conventions.

286. What is the purpose of the `eslint-plugin-react` plugin in ESLint?

- a. `eslint-plugin-react` is used for enabling React-specific linting rules in ESLint.
- b. `eslint-plugin-react` is used for automatically fixing linting issues in React components.
- c. `eslint-plugin-react` is used for generating React component documentation from code comments.
- d. `eslint-plugin-react` is a deprecated plugin and should not be used in modern React applications.

View Answer 



One example rule provided by this plugin is `react/jsx-props-no-spreading`, which warns against using the spread operator for passing props in JSX.

287. In React, which of the following is a recommended way to handle errors that occur during data fetching?

- a. Using try-catch blocks
- b. Using error boundaries
- c. Ignoring the errors
- d. Logging the errors to the console

[View Answer](#) ^

Answer: B. Using error boundaries

Error boundaries are the recommended way to catch and handle errors that occur during data fetching in React.

288. What is a limitation of using React with legacy codebases?

- a. React does not have built-in support for integrating with legacy code.
- b. React can be difficult to integrate with legacy code.
- c. React can cause compatibility issues with older browsers.
- d. React can make legacy code more difficult to maintain.

[View Answer](#) ^

Answer: B. React can be difficult to integrate with legacy code.

Because React is a relatively new technology, it can be challenging to integrate it with legacy codebases that were built using older technologies or development practices.

This can require additional work and can make it harder to maintain the application over time.



- b. Using try-catch blocks
- c. Ignoring the errors
- d. Logging the errors to the console

[View Answer](#) 

Answer: C. Ignoring the errors

Ignoring errors is not a recommended way to handle errors in React.

It is important to handle errors in a way that is user-friendly and informative.

290. What is ESLint and what is its purpose in React development?

- a. ESLint is a tool used for type-checking in React applications.
- b. ESLint is a code formatter specifically designed for React code.
- c. ESLint is a linter that helps catch and fix code errors and enforce coding standards in React applications.
- d. ESLint is a utility for optimizing and bundling React components for production.

[View Answer](#) 

Answer: C. ESLint is a linter that helps catch and fix code errors and enforce coding standards in React applications.

ESLint is a widely used JavaScript linter that can be configured to enforce coding conventions, find potential errors, and improve code quality in React projects.

291. How can you ignore certain files or directories from being linted by ESLint or formatted by Prettier in a React project?

- a. You cannot exclude files or directories from ESLint or Prettier in a React project.
- b. You can exclude files or directories from ESLint by using the ``.eslintignore`` file and from Prettier by using the ``.prettierrignore`` file.
- c. You can exclude files or directories from both ESLint and Prettier by using the ``.ignore`` file.
- d. You can exclude files or directories from ESLint and Prettier by specifying exclusions in the respective configuration files (``.eslintrc`` and ``.prettierrc``).



Prettier by using the `.prettierrignore` file.

To exclude specific files or directories from being linted by ESLint or formatted by Prettier, you can use the `.eslintignore` file for ESLint and the `.prettierrignore` file for Prettier.

These configuration files allow you to specify patterns to exclude from the respective tools' operations.

292. What is the purpose of the `eslint-plugin-import` plugin in ESLint?

- a. `eslint-plugin-import` is used for enabling linting rules related to import statements in ESLint.
- b. `eslint-plugin-import` is used for automatically fixing import-related linting issues in ESLint.
- c. `eslint-plugin-import` is used for generating import statement documentation from code comments.
- d. `eslint-plugin-import` is a deprecated plugin and should not be used in modern [React](#) applications.

View Answer ^

Answer: A. `eslint-plugin-import` is used for enabling linting rules related to import statements in ESLint.

The `eslint-plugin-import` plugin extends ESLint with import-related linting rules.

One example rule provided by this plugin is `import/no-unresolved`, which warns against importing modules that cannot be resolved by the module system.

293. What is the purpose of React Context in ReactJS?

- a. React Context is used for managing component state in ReactJS.
- b. React Context is used for handling routing and navigation in ReactJS.
- c. React Context is used for providing global data that can be accessed by multiple components in a ReactJS application.
- d. React Context is used for server-side rendering of React components.

View Answer ^

at every level.

It is particularly useful when you want to pass data to multiple components that are not directly connected in the component hierarchy.

294. What is the purpose of React hooks in ReactJS?

- a. React hooks are used for creating custom HTML elements in ReactJS.
- b. React hooks are used for managing asynchronous operations in [ReactJS](#).
- c. React hooks are used for managing component state and lifecycle in functional components in ReactJS.
- d. React hooks are used for handling form validation in ReactJS.

View Answer ^

Answer: C. React hooks are used for managing component state and lifecycle in functional components in ReactJS.

React hooks are functions that allow functional components to use state and other React features.

They provide a way to manage component state and lifecycle methods without using class components.

React Toastify

295. What is React Toastify?

- a. A JavaScript library for creating pop-up notifications
- b. A CSS framework for styling toast messages
- c. A server-side rendering solution for React applications
- d. A state management library for React

View Answer ^



Answer: A is the correct option.

296. How can you install React Toastify in a React project?

- a. By including a CDN link in the HTML file
- b. By running a command in the terminal
- c. By manually downloading and linking the library
- d. React Toastify is included by default in React projects

[View Answer](#) ^

Answer: B is the correct option.

To install React Toastify in a React project, you need to run a command in the terminal.

You can use a package manager like npm or yarn to install the library.

For example, if you're using npm, you can run the following command:

```
npm install react-toastify
```

This will download and install the React Toastify package, making it available for use in your project.

297. How do you display a toast notification using React Toastify?

- a. By importing the `ToastContainer` component and rendering it in the component where you want to show the notification
- b. By calling the `toast` function and passing the notification message as an argument
- c. By adding a custom CSS class to the desired element
- d. React Toastify automatically displays toast notifications without any additional code

[View Answer](#) ^



Answer: A is the correct option.

298. How can you customize the appearance of toast notifications in React Toastify?

- a. By passing options as an argument to the ``toast`` function
- b. By applying CSS styles to the ``ToastContainer`` component
- c. By modifying the source code of the React Toastify library
- d. React Toastify does not support customization of toast appearance

[View Answer](#) ^

Answer: A is the correct option.

You can customize the appearance of toast notifications in React Toastify by passing options as an argument to the ``toast`` function.

The options allow you to modify various aspects of the toast, such as its position, duration, animation, and styling.

299. How can you handle user interactions with toast notifications in React Toastify?

- a. React Toastify does not support user interactions with toast notifications
- b. By adding event listeners to the ``ToastContainer`` component
- c. By providing callbacks through the options argument of the ``toast`` function
- d. By using React's built-in event system and capturing events from the toast elements

[View Answer](#) ^

Answer: C is the correct option.

You can handle user interactions with toast notifications in React Toastify by providing callbacks through the options argument of the ``toast`` function.

The options object allows you to specify event handlers for different user interactions, such as `onClick`, `onClose`, `onOpen`, etc.



```
toast.warning("Task completed");
```

B.

```
toast.info("Task completed");
```

C.

```
toast.error("Task completed");
```

D.

```
toast.success("Task completed");
```

- a. A
- b. B
- c. C
- d. D

[View Answer](#) ^

Answer: D is the correct option.

It is important to note that JSX comments must be written inside the curly braces {} to be interpreted correctly by React.

- Single-line comments can be written using //, for example:

```
// This is a single-line comment
```



- Multi-line comments can be written using /* */, for example:

- JSX comments can be written using `{/ /}`, for example:

```
{/* This is a JSX comment */}
```

```
import React from 'react';

function UserProfile({ username, bio }) {
  // This component displays a user's profile information.

  // Single-line comment: Here, we receive the username and bio as props.
  return (
    <div classname="user-profile">
      {/* JSX comment: This div holds the user's profile information. */}
      <h2>Welcome, {username}!</h2>
      {
        /*
          Multi-line comment: Inside the div, we display the user's
          username and bio.
        */
      }
      <p>{bio}</p>
    </div>
  );
}

export default UserProfile;
```

301. What is the purpose of the `toast.success` method in React Toastify?

- a. It displays a success toast notification with a green color theme.
- b. It removes an existing toast notification from the screen.
- c. It opens a new browser tab with additional information about the success.
- d. It triggers an error message in the console.



The `toast.success` method in React Toastify is used to display a success toast notification with a predefined green color theme.

It's a convenient way to show positive feedback messages to users when certain actions are successfully completed.

302. How can you add a custom icon to a toast notification in React Toastify?

- a. React Toastify does not support custom icons in toast notifications.
- b. By using the `icon` property in the `toast` function options.
- c. By adding an HTML `img` element inside the toast message content.
- d. By modifying the React Toastify source code.


View Answer 

Answer: B is the correct option.

You can add a custom icon to a toast notification in React Toastify by using the `icon` property in the `toast` function options.

This allows you to specify a custom icon or image URL to be displayed alongside the toast message, providing visual context to the notification.

303. How can you configure the position of toast notifications in React Toastify?

- a. By setting the `position` property of the `ToastContainer` component
- b. By passing options to the `toast` function with the desired position value
- c.  [React](#) Toastify does not provide options for positioning toast notifications
- d. By manually adjusting the CSS of the toast notification elements

View Answer 



The options object allows you to specify the `position` property, which can be set to values like `top-right`, `top-left`, `bottom-right`, `bottom-left`, `top-center`, `bottom-center`, etc.

This controls where the toast notifications will appear on the screen.

304. How can you handle multiple toast notifications in React Toastify?

- a. By rendering multiple `ToastContainer` components in different parts of your application
- b. By using a single `ToastContainer` component and calling the `toast` function multiple times
- c. React Toastify does not support displaying multiple toast notifications simultaneously
- d. By manually managing an array of toast messages and rendering them conditionally

[View Answer](#) 

Answer: B is the correct option.

You can handle multiple toast notifications in React Toastify by using a single `ToastContainer` component and calling the `toast` function multiple times.

Each call to `toast` will display a separate toast notification. React Toastify manages the queue of notifications internally and displays them in the order they are triggered.

305. Can you customize the appearance of individual toast notifications in React Toastify?

- a. Yes, by passing options as an argument to each `toast` function call
- b. No, all toast notifications have the same default appearance
- c. By modifying the CSS styles of the `ToastContainer` component
- d. React Toastify does not provide customization options for individual toasts

[View Answer](#) 



Answer: A is the correct option.

custom styles, or even custom react components to be rendered within the toast.

306. Can you programmatically dismiss or remove a toast notification in React Toastify?

- a. Yes, by calling the `toast.dismiss` method with the toast ID
- b. No, toast notifications are automatically dismissed after a certain duration
- c. By triggering a specific event on the `ToastContainer` component
- d. React Toastify does not support manual dismissal of toast notifications

[View Answer](#) ^

Answer: A is the correct option.

You can programmatically dismiss or remove a toast notification in React Toastify by calling the `toast.dismiss` method with the toast ID.

When a toast notification is displayed, React Toastify returns a unique ID for that toast.

By using this ID, you can manually dismiss the toast before its auto-close duration or remove it based on specific events or user interactions in your application.

307. How can you control the duration of toast notifications in React Toastify?

- a. By specifying the `autoClose` option in the `toast` function call
- b. By adjusting the default timing settings in the `ToastContainer` component
- c. React Toastify does not provide control over the duration of toast notifications
- d. By implementing a custom timer using JavaScript's `setTimeout` function

[View Answer](#) ^

Answer: A is the correct option.




You can control the duration of toast notifications in React Toastify by specifying the `autoClose` option in the `toast` function call.

Continue...

Conclusions

In this post, we have compiled a list of more than 300+ [React JS](#) interview questions with answers.

These questions cover various topics, including basics, components, state management, hooks, and advanced concepts.



Malaysia to Sandakan

from Malaysia

from Rs1,802

View Now

Our goal is to provide you with valuable insights and help you prepare effectively for your upcoming React JS interview.

Also Read : [Node Js Interview \(MCQ\) With Answers](#)

If you're interested in further enhancing your React.js skills, consider exploring the resources provided by [Examlabs](#).

We hope that this post proves to be a helpful resource for you.

If you find this post useful, please consider sharing it with your coding friends.

Thank you for reading!

