

## React & JavaScript Questions

1. The correct format for assigning inline styles to elements in React.

- a) Strings
  - b) Plain text
  - c) JSON
  - d) CamelCase JavaScript objects (ans)
- 

2. The type of data that can be stored in state.

- a) Only strings
  - b) Any data type (ans)
  - c) Only numbers
  - d) Only objects
- 

3. The correct syntax for including a CSS stylesheet in React.

- a) Using require()
  - b) Using <link>
  - c) Using import './App.css' (ans)
  - d) Using useCSS()
- 

4. Identify the reason why keys should be unique in a list of React elements.

- a) To prevent style conflicts
  - b) To allow React to uniquely identify elements for updates (ans)
  - c) To link the elements with external data
  - d) To allow form submission
- 

5. The main features and advantages of react-bootstrap

- a) Converts Bootstrap to HTML
- b) Provides Bootstrap components for React (ans)

- c) Installs jQuery
  - d) Adds JavaScript routing
- 

**6. Identify the npm command used to add React Bootstrap to a project.**

- a) npm bootstrap-react
  - b) npm install bootstrap-react
  - c) **npm install react-bootstrap bootstrap (ans)**
  - d) npm add react-css
- 

**7. The purpose of React Table in managing data display.**

- a) Creating styled buttons
  - b) Managing layout
  - c) **Rendering flexible, customizable data tables (ans)**
  - d) Creating navigation
- 

**8. Identify the command used to install React Table.**

- a) npm i react-table
  - b) **npm install react-table (ans)** (Note: npm i is also a valid alias)
  - c) yarn table
  - d) npm get table
- 

**9. Identify the role props play in React for passing data.**

- a) A hook
  - b) A router
  - c) **A way to pass data to components (ans)**
  - d) A CSS file
- 

**10. What is the syntax used to send props to child components.**

- a) As attributes in JSX (ans)
  - b) Through context only
  - c) Using Redux
  - d) Through state only
- 

**11. What is the correct syntax to destructure props in a functional component.**

- a) function MyComponent(props) { const name = props.name; }
  - b) const name = this.props.name;
  - c) **function MyComponent({ name }) { return <h1>{name}</h1>; } (ans)**
  - d) component({ name }) => <h1>{name}</h1>
- 

**12. Which one is the purpose of using the map() function in a React component.**

- a) To loop through an array and render elements (ans)
  - b) To fetch data from APIs
  - c) To update component state
  - d) To validate props
- 

**13. Which one is the correct use of props.children in React from the following options.**

- a) Router element
  - b) **Nested elements passed between opening and closing component tags (ans)**
  - c) Component name
  - d) prop validation function
- 

**14. Select the statement that correctly describes props in React:**

- a) Props are mutable and can be changed within the child component.
- b) **Props are read-only and passed from parent to child. (ans)**
- c) Props are used to pass data from child to parent.

- d) Props are used to manage internal component state.
- 

**15. Identify the library most commonly used for validating component props in React.**

- a) redux-types
  - b) react-validate
  - c) **prop-types (ans)**
  - d) type-check
- 

**16. Identify the primary purpose of the filter() method in JavaScript when used in React.**

- a) To remove null values from props
  - b) **To create a new array based on a condition (ans)**
  - c) To convert an array into a string
  - d) To sort an array alphabetically
- 

**17. Which one is the primary difference between state and props in a functional component.**

- a) State is passed down
  - b) Props are mutable
  - c) **State is internal and mutable (ans)**
  - d) State is readonly
- 

**18. Select the correct use of filter() in a React component.**

- a) data.filter(item.active)
  - b) filter.data(item => item)
  - c) **data.filter(item => item.active) (ans)**
  - d) data.map(item => item.active === true)
- 

**19. The React hook responsible for managing state in function components.**

- a) useContext
  - b) useRef
  - c) **useState (ans)**
  - d) useEffect
- 

20. Identify the syntax used to update a state variable declared using the useState hook.

- a) state.value = 10
  - b) this.setState(10)
  - c) **setValue(10) (ans)**
  - d) value = setValue(10)
- 

21. Identify which one the filter() method returns.

- a) A boolean value
  - b) A modified version of the original array
  - c) **A new array with elements that meet the condition (ans)**
  - d) A list of components
- 

22. Identify the main purpose of using state in functional React components.

- a) Holds static values
  - b) **Tracks internal dynamic values of components (ans)**
  - c) Manages routing
  - d) Applies styling
- 

23. What is the outcome of calling a state updater function (like setState or setCount) in React.

- a) Page reload
- b) Function call
- c) **Component re-render (ans)**

- 
- d) Component unmount
- 

**24. Select the correct outcome of using filter() followed by map() in a React render.**

- a) Filters out props before assigning
  - b) Combines arrays before rendering
  - **c) Filters and then renders only selected items (ans)**
  - d) Deletes state variables
- 

**25. Identify the traits that define a reusable component in React development.**

- a) Tied to one view
  - b) Fixed for single use
  - **c) Can be reused across the app with different data via props (ans)**
  - d) Applies global state
- 

**26. What is the React hook that enables updating of state variables within a functional component.**

- a) useEffect
  - b) useContext
  - **c) useState (ans)**
  - d) useUpdate
- 

**27. Identify the purpose of the key attribute in this code: {items.map((item) => <li key={item.id}>{item.name}</li>)}**

- a) It sets the CSS class
  - b) It adds an event listener
  - **c) It uniquely identifies elements for efficient rendering (ans)**
  - d) It adds a name to each item
-

28. What is the output of this code: `function ClickMe() { return <button onClick={() => alert("Clicked!")}>Click</button>; }`

- a) A paragraph with text "Clicked!"
  - b) A disabled button
  - c) A button that shows an alert when clicked (ans)
  - d) No output due to syntax error
- 

29. Identify what will be displayed: `function Status(props) { return <p>{props.online ? "Online" : "Offline"</p>; } Called as: <Status online={false} />`

- a) Online
  - b) Error
  - c) true
  - d) Offline (ans)
- 

30. Identify what the `fetch()` function returns in JavaScript.

- a) A JSON object
  - b) An array of data
  - c) A promise (ans)
  - d) A string
- 

31. Identify the correct syntax to define default props in React components.

- a) `static defaultProps = {} (ans)` (Used in class components;  
`MyComponent.defaultProps = {}` is used for functional components)
  - b) `props.defaults()`
  - c) `useDefaultProps()`
  - d) `component.default()`
- 

32. Select the correct sequence of steps when fetching data with `fetch()` in React.

- a) Fetch → Set Timeout → Console Log

- **b) Fetch → Convert to JSON → Use Data (ans)**
  - c) Fetch → Use Data → Convert to JSON
  - d) Use Data → Fetch → Console Log
- 

**33. Identify the correct syntax for writing an arrow function in JavaScript.**

- a) `function() => {}`
  - **b) () => {} (ans)**
  - c) `() = {}`
  - d) `=> () {}`
- 

**34. Identify scenarios where using arrow functions in event handlers prevents common issues like incorrect this binding.**

- **a) They automatically bind this (ans)**
  - b) They delay execution
  - c) They refresh the app
  - d) They fetch data
- 

**35. Identify the role of the handleChange function in managing input value changes in React.**

- a) Submits the form
  - **b) Calls a function on input change (ans)**
  - c) Prevents re-render
  - d) Binds event with HTML only
- 

**36. Identify the function of the name attribute in linking input fields to submitted form data.**

- a) Validation
- **b) Binding value to corresponding state (ans)**
- c) Styling

- d) Route identification
- 

**37. Identify the React hook commonly used for managing form state.**

- a) useState (ans)
  - b) useRef
  - c) useEffect
  - d) useContext
- 

**38. Determine the default parameter passed to event handlers in React and its purpose.**

- a) Event object (ans)
  - b) State object
  - c) Props
  - d) DOM node
- 

**39. Select the correct arrow function syntax for an event handler from the given options.**

- a) handleClick() => {}
  - b) () => { console.log("Clicked") } (ans)
  - c) = () {}
  - d) onClick = function => {}
- 

**40. Determine the React technique used to manage and access form data through component state.**

- a) From the DOM
  - b) Via event.target.value (ans)
  - c) Using a router
  - d) Through props only
-

**41. Identify key advantages of controlled components in ensuring consistent state management in forms.**

- a) Reduce performance
  - b) UI and state stay in sync (ans)
  - c) Force reload
  - d) Require Redux
- 

**42. Identify the default folder that contains the production build of a React application.**

- a) /public
  - b) /src
  - c) /build (ans)
  - d) /index
- 

**43. Select the platform(s) that support free React app deployment from the list provided.**

- a) Photoshop
  - b) VS Code
  - c) GitHub Pages (ans)
  - d) Android Studio
- 

**44. Choose the correct way to render a component conditionally:**

- a) {condition => <Component/>}
  - b) condition ? <Component/> : "" (Also correct)
  - c) {condition && <Component/>} (ans) (Also correct)
  - d) if (condition) return <Component /> (Also correct)
- 

**45. Identify the correct way to demonstrate the use of onClick in JSX:**

- a) onclick="handle()"
- b) onClick="handle()"

- c) `onClick={handleClick}` (ans)
  - d) `click={handle}`
- 

46. Choose the correct way to use routing in a React app:

- a) `<router> </router>`
  - b) `<BrowserRouter> <Routes>... </Routes> </BrowserRouter>` (ans)
  - c) `<RouteRouter />`
  - d) `<RouterMap />`
- 

47. Choose the correct interpretation of the use of `<Link>` from react-router-dom:

- a) Client-side navigation (ans)
  - b) Navigation with full page reload
  - c) Redirect to external site
  - d) Form submission
- 

## API, PHP & MVC Questions

48. Choose the correct function used to fetch data in modern JavaScript applications.

- a) XMLHttpRequest
  - b) getJSON()
  - c) `fetch()` (ans)
  - d) load()
- 

49. Choose the method commonly used to send data in a fetch POST request.

- a) `data: {}`
  - b) payload
  - c) **body (ans)**
  - d) content
-

**50. Choose the correct programming language used to develop the backend API in this module.**

- a) JavaScript
  - b) Java
  - c) PHP (ans)
  - d) Angular
- 

**51. Choose the commonly used method in PHP to establish a connection with a MySQL database.**

- a) PDO or mysqli (ans)
  - b) MongoDB
  - c) SQLServer
  - d) pg\_connect()
- 

**52. Choose the appropriate HTTP method used for submitting new data to an API.**

- a) GET
  - b) PUT
  - c) POST (ans)
  - d) DELETE
- 

**53. Choose the appropriate header to set when sending JSON in an API request.**

- a) Accept-Type
  - b) Content-Type: text/plain
  - c) Content-Type: application/json (ans)
  - d) Content-Length
- 

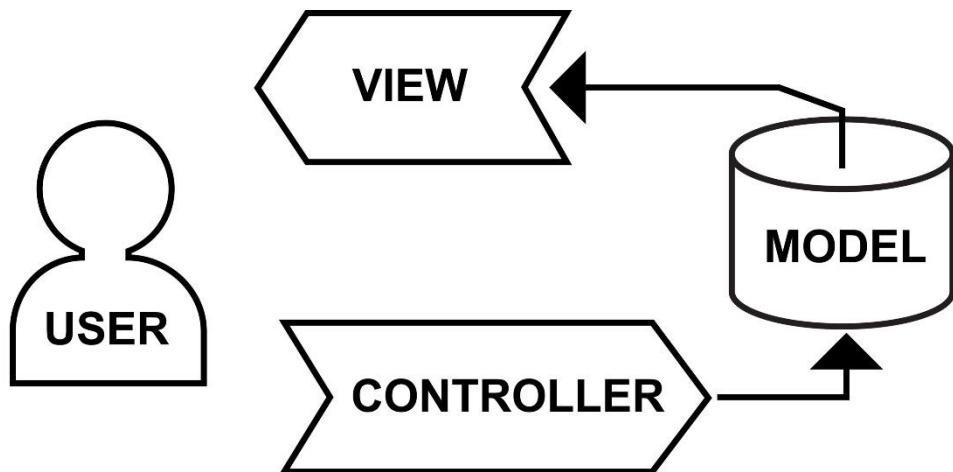
**54. Choose the purpose of JSON.stringify() when used in a fetch request.**

- a) Parses JSON
- b) Converts JS object to string (ans)

- c) Minifies HTML
  - d) Compresses data
- 

55. Choose the three core components that make up the MVC architecture.

- a) Middleware, Views, Classes
- b) Model, View, Controller (ans)
- c) Main, Virtual, Compiler
- d) Map, Validator, Controller



### MODEL - VIEW - CONTROLLER PATTERN

Shutterstock

---

56. Which one is the correct interpretation of the role of a REST API:

- a) Allows communication between systems (ans)
  - b) Serves HTML pages
  - c) Transfers data using XML
  - d) Used for compiling PHP files
- 

57. Which one is the correct line to sketch a basic API endpoint in PHP:

- a) function api() { return; }

- b) `header('Content-Type: application/json');` (ans)
  - c) `print("API")`
  - d) `open('api.php')`
- 

58. Choose the correct way to use PHP to send a JSON response from an API:

- a) `echo "JSON";`
  - b) `json_encode($data)` (ans) (This function creates the JSON string, which is then typically echoed)
  - c) `send_json($data)`
  - d) `return data;`
- 

**A** Angular & AngularJS Questions

59. Identify what the templateUrl in `@Component` does: (This is Angular 2+)

- a) Binds event handlers
  - b) Specifies the path to the HTML template (ans)
  - c) Links to CSS
  - d) Registers the module
- 

60. Identify the role of AngularJS in building single-page applications.

- a) Backend library
  - b) JavaScript framework for building SPAs (ans)
  - c) CSS framework
  - d) PHP engine
- 

61. Choose the correct distinctions between AngularJS and Angular (2+).

- a) Based on TypeScript
- b) Uses JSX
- c) Runs on Node.js

- d) AngularJS is older and based on JS (ans)
- 

62. Choose the correct HTML tag or directive used to define an AngularJS app.

- a) <html>
  - b) <body>
  - c) ng-app (ans)
  - d) <angular>
- 

63. Choose the correct syntax to output data using AngularJS expressions in a web page.

- a) [[ name ]]
  - b) { name }
  - c) {{ name }} (ans)
  - d) <%= name %>
- 

64. Choose the correct AngularJS expression used to bind an object property in the view.

- a) {{ object[property] }} (ans) (Used when the property name is a variable)
  - b) [[ object.property ]]
  - c) {{ object::property }}
  - d) object->property
- 

65. Select the AngularJS directive used to loop through items in an array.

- a) ng-if
  - b) ng-model
  - c) ng-loop
  - d) ng-repeat (ans)
- 

66. Select the AngularJS directive used to bind an app to the HTML document.

- a) **ng-app (ans)**
  - b) ng-start
  - c) app-ng
  - d) angular-init
- 

67. Select the part of the MVC architecture in AngularJS that handles data logic.

- a) View
  - b) **Model (ans)**
  - c) Controller
  - d) DOM
- 

68. Which one is the correct file needed to include AngularJS in an HTML document.

- a) angular.html
  - b) ng.module.js
  - c) **angular.min.js (ans)**
  - d) module.angular.js
- 

69. Which one is the default delimiter used for AngularJS expressions.

- a) <% %>
  - b) { }
  - c) {{ }} (ans)
  - d) [[ ]]
- 

70. Select the correct AngularJS expression that represents a valid numeric operation.

- a) {{ 3 + 5 }} (ans)
- b) {3+5}
- c) <%= 3+5 %>
- d) [[3+5]]

---

**71. Choose the correct AngularJS expression that displays the full name from an object.**

- a) {{ user.fullname }} (ans)
  - b) {{ fullname.user }}
  - c) [[ user.fullname ]]
  - d) <%= user.fullname %>
- 

**72. Choose the correct explanation of object binding in AngularJS.**

- a) Combining multiple views
  - b) Assigning string to class
  - c) **Binding object data to UI (ans)**
  - d) Creating nested routes
- 

**73. What is the correct behavior of AngularJS when it encounters an invalid expression.**

- a) Throws fatal error
  - b) **Ignores it silently (ans)**
  - c) Shows warning popup
  - d) Removes scope
- 

**74. Which one is the correct output of the AngularJS expression {{ 'Hello' + ' AngularJS' }}.**

- a) HelloAngularJS
  - b) "Hello AngularJS"
  - c) **Hello AngularJS (ans)**
  - d) {{ "Hello AngularJS" }}
- 

**75. What is the correct decorator to define a component in Angular: (This is Angular 2+)**

- a) @NgModule

- b) `@Injectable`
  - c) `@Directive`
  - d) `@Component (ans)`
- 

76. Select the key feature of AngularJS expressions from the following options.

- a) Can evaluate complex logic (ans) (Relative to plain HTML, but simpler than full JS)
  - b) Cannot use operators
  - c) Run outside the scope
  - d) Allow only strings
- 

77. Choose the correct purpose of the `ng-bind` directive in AngularJS.

- a) Loops through values
  - b) Binds content of an HTML element to scope variable (ans)
  - c) Triggers click events
  - d) Declares filters
- 

78. Choose the correct syntax for binding a value using the `ng-bind` directive in AngularJS.

- a) `<p ng-bind="message"></p>` (ans)
  - b) `<p bind="message">`
  - c) `<ng-bind="message">`
  - d) `bind.message()`
- 

79. Choose the correct output of the AngularJS expression `{{ 10 / 2 }}`

- a) 2
- b) 5 (ans)
- c) 0.5
- d) 45698

---

**80. Choose the correct AngularJS expression that includes both a string and a number.**

- a) {{ "Count: " + 5 }} (ans)
  - b) [{"Count" + 5}]
  - c) {{ 5 + "Count" }}
  - d) 5 + "Count"
- 

**81. What is the correct way to use both @if and @else blocks in a card: (This is Angular 17+ control flow)**

- a) @if (isLoggedIn) { ... } else { ... }
  - b) @if (isLoggedIn) { Welcome } @else { Please Login } (ans)
  - c) \*ngIf="isLoggedIn; else notLoggedIn"
  - d) @if(isLoggedIn): true else false
- 

**82. Choose the correct AngularJS directive that conditionally adds an element to the DOM.**

- a) ng-show
  - b) ng-if (ans)
  - c) ng-bind
  - d) ng-model
- 

**83. Choose the correct syntax for using @if in a template: (This is Angular 17+ control flow)**

- a) \*ngIf="condition"
  - b) @if (condition) { <div>Yes</div> } (ans)
  - c) @if condition { <div>Yes</div> }
  - d) <div @if="condition">Yes</div>
-

**84. Choose the correct statement that describes the primary role of a controller in AngularJS.**

- a) Styling
  - b) **Managing scope and logic (ans)**
  - c) Handling HTTP calls
  - d) Compiling templates
- 

**85. Choose the correct definition and role of \$scope in AngularJS.**

- a) A built-in CSS class
  - b) A filter
  - c) **A bridge between controller and view (ans)**
  - d) A model validator
- 

**86. Which one is the correct outcome of assigning a function to \$scope within an AngularJS controller.**

- a) It becomes inaccessible in view
  - b) It throws an error
  - c) **It can be called from the view (ans)**
  - d) It deletes the controller
- 

**87. Choose the correct syntax to loop over a list using @for: (This is Angular 17+ control flow)**

- a) <li \*ngFor="let item of items">{{ item }}</li>
  - b) @for (item in items) { <li>{{ item }}</li> }
  - c) **@for (item of items) { <li>{{ item }}</li> } (ans)**
  - d) for (item of items) { <li>{{ item }}</li> }
- 

**88. \*Select the primary benefit of using @for over ngFor in Angular: (This is Angular 17+ control flow)**

- a) Easier styling
  - b) Supports Promises
  - c) **Cleaner syntax and better performance (ans)**
  - d) Replaces ngIf
- 

89. Choose the correct AngularJS filter that transforms text to uppercase.

- a) upperCase
  - b) capital
  - c) **uppercase (ans)**
  - d) toUpperCase
- 

90. Choose the AngularJS filter that displays a subset of array items.

- a) Slice
  - b) **filter (ans)**
  - c) Select
  - d) ng-subset
- 

91. Choose the correct way to define a custom filter in AngularJS.

- a) app.directive()
  - b) **app.filter('name', function() {})(ans)**
  - c) app.ngFilter()
  - d) scope.filter()
- 

92. Choose the correct option that is not a built-in filter in AngularJS.

- a) filter
- b) limitTo
- c) **sort (ans)** (The filter is orderBy, not sort)
- d) Date

---

**93. What is the correct way to operate a simple form with AngularJS:**

- a) Use form-submit directive
  - b) Forms aren't supported
  - c) **Use ng-model with inputs (ans)**
  - d) Use only vanilla JS
- 

**94. Which one is the correct directive used in AngularJS to apply a CSS class based on a condition.**

- a) Using ng-if
  - b) Using ng-style
  - c) **Using ng-class (ans)**
  - d) Using ng-model
- 

**95. Choose the correct way in which ng-switch handles multiple conditions in AngularJS:**

- a) It loops over options
  - b) **It uses ng-switch-when for condition blocks (ans)**
  - c) It uses ng-show internally
  - d) It switches styles
- 

**96. What is the correct syntax to apply the ng-if directive for conditionally displaying content when the variable isVisible is true:**

- a) <div if="isVisible">Hello</div>
- b) **<div ng-if="isVisible">Hello</div> (ans)**
- c) <div ng-show="isVisible">Hello</div>
- d) <div \*ngIf="isVisible">Hello</div>