

REDUX & NEXT.JS



&

NEXT.JS

100
INTERVIEW
QUESTIONS
& ANSWERS

1. Redux Q1: What are the core principles of Redux?

Single source of truth, state is read-only, and changes are made using pure functions.

2. Redux Q2: What is an action in Redux?

An action is a plain JavaScript object that describes what happened. It must have a type property.

3. Redux Q3: What is a reducer in Redux?

A reducer is a pure function that takes the current state and an action and returns the new state.

4. Redux Q4: What is the Redux store?

The store is a single object that holds the entire application state in Redux.

5. Redux Q5: What is Redux?

Redux is a predictable state container for JavaScript apps, commonly used with React for managing global state.

6. Redux Q6: What are the core principles of Redux?

Single source of truth, state is read-only, and changes are made using pure functions.

7. Redux Q7: What is an action in Redux?

An action is a plain JavaScript object that describes what happened. It must have a type property.

8. Redux Q8: What is a reducer in Redux?

A reducer is a pure function that takes the current state and an action and returns the new state.

9. Redux Q9: What is the Redux store?

The store is a single object that holds the entire application state in Redux.

10. Redux Q10: What is Redux?

Redux is a predictable state container for JavaScript apps, commonly used with React for managing global state.

11. Redux Q11: What are the core principles of Redux?

Single source of truth, state is read-only, and changes are made using pure functions.

12. Redux Q12: What is an action in Redux?

An action is a plain JavaScript object that describes what happened. It must have a type property.

13. Redux Q13: What is a reducer in Redux?

A reducer is a pure function that takes the current state and an action and returns the new state.

14. Redux Q14: What is the Redux store?

The store is a single object that holds the entire application state in Redux.

15. Redux Q15: What is Redux?

Redux is a predictable state container for JavaScript apps, commonly used with React for managing global state.

16. Redux Q16: What are the core principles of Redux?

Single source of truth, state is read-only, and changes are made using pure functions.

17. Redux Q17: What is an action in Redux?

An action is a plain JavaScript object that describes what happened. It must have a type property.

18. Redux Q18: What is a reducer in Redux?

A reducer is a pure function that takes the current state and an action and returns the new state.

19. Redux Q19: What is the Redux store?

The store is a single object that holds the entire application state in Redux.

20. Redux Q20: What is Redux?

Redux is a predictable state container for JavaScript apps, commonly used with React for managing global state.

21. Redux Q21: What are the core principles of Redux?

Single source of truth, state is read-only, and changes are made using pure functions.

22. Redux Q22: What is an action in Redux?

An action is a plain JavaScript object that describes what happened. It must have a type property.

23. Redux Q23: What is a reducer in Redux?

A reducer is a pure function that takes the current state and an action and returns the new state.

24. Redux Q24: What is the Redux store?

The store is a single object that holds the entire application state in Redux.

25. Redux Q25: What is Redux?

Redux is a predictable state container for JavaScript apps, commonly used with React for managing global state.

26. Redux Q26: What are the core principles of Redux?

Single source of truth, state is read-only, and changes are made using pure functions.

27. Redux Q27: What is an action in Redux?

An action is a plain JavaScript object that describes what happened. It must have a type property.

28. Redux Q28: What is a reducer in Redux?

A reducer is a pure function that takes the current state and an action and returns the new state.

29. Redux Q29: What is the Redux store?

The store is a single object that holds the entire application state in Redux.

30. Redux Q30: What is Redux?

Redux is a predictable state container for JavaScript apps, commonly used with React for managing global state.

31. Redux Q31: What are the core principles of Redux?

Single source of truth, state is read-only, and changes are made using pure functions.

32. Redux Q32: What is an action in Redux?

An action is a plain JavaScript object that describes what happened. It must have a type property.

33. Redux Q33: What is a reducer in Redux?

A reducer is a pure function that takes the current state and an action and returns the new state.

34. Redux Q34: What is the Redux store?

The store is a single object that holds the entire application state in Redux.

35. Redux Q35: What is Redux?

Redux is a predictable state container for JavaScript apps, commonly used with React for managing global state.

36. Redux Q36: What are the core principles of Redux?

Single source of truth, state is read-only, and changes are made using pure functions.

37. Redux Q37: What is an action in Redux?

An action is a plain JavaScript object that describes what happened. It must have a type property.

38. Redux Q38: What is a reducer in Redux?

A reducer is a pure function that takes the current state and an action and returns the new state.

39. Redux Q39: What is the Redux store?

The store is a single object that holds the entire application state in Redux.

40. Redux Q40: What is Redux?

Redux is a predictable state container for JavaScript apps, commonly used with React for managing global state.

41. Redux Q41: What are the core principles of Redux?

Single source of truth, state is read-only, and changes are made using pure functions.

42. Redux Q42: What is an action in Redux?

An action is a plain JavaScript object that describes what happened. It must have a type property.

43. Redux Q43: What is a reducer in Redux?

A reducer is a pure function that takes the current state and an action and returns the new state.

44. Redux Q44: What is the Redux store?

The store is a single object that holds the entire application state in Redux.

45. Redux Q45: What is Redux?

Redux is a predictable state container for JavaScript apps, commonly used with React for managing global state.

46. Redux Q46: What are the core principles of Redux?

Single source of truth, state is read-only, and changes are made using pure functions.

47. Redux Q47: What is an action in Redux?

An action is a plain JavaScript object that describes what happened. It must have a type property.

48. Redux Q48: What is a reducer in Redux?

A reducer is a pure function that takes the current state and an action and returns the new state.

49. Redux Q49: What is the Redux store?

The store is a single object that holds the entire application state in Redux.

50. Redux Q50: What is Redux?

Redux is a predictable state container for JavaScript apps, commonly used with React for managing global state.

51. Next.js Q1: What are the key features of Next.js?

Features include SSR, SSG, API routes, image optimization, file-based routing, and middleware support.

52. Next.js Q2: What is server-side rendering (SSR) in Next.js?

SSR allows pages to be rendered on the server at request time, improving SEO and performance.

53. Next.js Q3: What is static site generation (SSG) in Next.js?

SSG pre-renders pages at build time and serves them as static HTML for better speed and scalability.

54. Next.js Q4: What is `getStaticProps` in Next.js?

A function used to fetch data at build time for static site generation.

55. Next.js Q5: What is Next.js?

Next.js is a React framework that enables server-side rendering, static site generation, and API routes for building modern web apps.

56. Next.js Q6: What are the key features of Next.js?

Features include SSR, SSG, API routes, image optimization, file-based routing, and middleware support.

57. Next.js Q7: What is server-side rendering (SSR) in Next.js?

SSR allows pages to be rendered on the server at request time, improving SEO and performance.

58. Next.js Q8: What is static site generation (SSG) in Next.js?

SSG pre-renders pages at build time and serves them as static HTML for better speed and scalability.

59. Next.js Q9: What is `getStaticProps` in Next.js?

A function used to fetch data at build time for static site generation.

60. Next.js Q10: What is Next.js?

Next.js is a React framework that enables server-side rendering, static site generation, and API routes for building modern web apps.

61. Next.js Q11: What are the key features of Next.js?

Features include SSR, SSG, API routes, image optimization, file-based routing, and middleware support.

62. Next.js Q12: What is server-side rendering (SSR) in Next.js?

SSR allows pages to be rendered on the server at request time, improving SEO and performance.

63. Next.js Q13: What is static site generation (SSG) in Next.js?

SSG pre-renders pages at build time and serves them as static HTML for better speed and scalability.

64. Next.js Q14: What is `getStaticProps` in Next.js?

A function used to fetch data at build time for static site generation.

65. Next.js Q15: What is Next.js?

Next.js is a React framework that enables server-side rendering, static site generation, and API routes for building modern web apps.

66. Next.js Q16: What are the key features of Next.js?

Features include SSR, SSG, API routes, image optimization, file-based routing, and middleware support.

67. Next.js Q17: What is server-side rendering (SSR) in Next.js?

SSR allows pages to be rendered on the server at request time, improving SEO and performance.

68. Next.js Q18: What is static site generation (SSG) in Next.js?

SSG pre-renders pages at build time and serves them as static HTML for better speed and scalability.

69. Next.js Q19: What is `getStaticProps` in Next.js?

A function used to fetch data at build time for static site generation.

70. Next.js Q20: What is Next.js?

Next.js is a React framework that enables server-side rendering, static site generation, and API routes for building modern web apps.

71. Next.js Q21: What are the key features of Next.js?

Features include SSR, SSG, API routes, image optimization, file-based routing, and middleware support.

72. Next.js Q22: What is server-side rendering (SSR) in Next.js?

SSR allows pages to be rendered on the server at request time, improving SEO and performance.

73. Next.js Q23: What is static site generation (SSG) in Next.js?

SSG pre-renders pages at build time and serves them as static HTML for better speed and scalability.

74. Next.js Q24: What is `getStaticProps` in Next.js?

A function used to fetch data at build time for static site generation.

75. Next.js Q25: What is Next.js?

Next.js is a React framework that enables server-side rendering, static site generation, and API routes for building modern web apps.

76. Next.js Q26: What are the key features of Next.js?

Features include SSR, SSG, API routes, image optimization, file-based routing, and middleware support.

77. Next.js Q27: What is server-side rendering (SSR) in Next.js?

SSR allows pages to be rendered on the server at request time, improving SEO and performance.

78. Next.js Q28: What is static site generation (SSG) in Next.js?

SSG pre-renders pages at build time and serves them as static HTML for better speed and scalability.

79. Next.js Q29: What is `getStaticProps` in Next.js?

A function used to fetch data at build time for static site generation.

80. Next.js Q30: What is Next.js?

Next.js is a React framework that enables server-side rendering, static site generation, and API routes for building modern web apps.

81. Next.js Q31: What are the key features of Next.js?

Features include SSR, SSG, API routes, image optimization, file-based routing, and middleware support.

82. Next.js Q32: What is server-side rendering (SSR) in Next.js?

SSR allows pages to be rendered on the server at request time, improving SEO and performance.

83. Next.js Q33: What is static site generation (SSG) in Next.js?

SSG pre-renders pages at build time and serves them as static HTML for better speed and scalability.

84. Next.js Q34: What is `getStaticProps` in Next.js?

A function used to fetch data at build time for static site generation.

85. Next.js Q35: What is Next.js?

Next.js is a React framework that enables server-side rendering, static site generation, and API routes for building modern web apps.

86. Next.js Q36: What are the key features of Next.js?

Features include SSR, SSG, API routes, image optimization, file-based routing, and middleware support.

87. Next.js Q37: What is server-side rendering (SSR) in Next.js?

SSR allows pages to be rendered on the server at request time, improving SEO and performance.

88. Next.js Q38: What is static site generation (SSG) in Next.js?

SSG pre-renders pages at build time and serves them as static HTML for better speed and scalability.

89. Next.js Q39: What is `getStaticProps` in Next.js?

A function used to fetch data at build time for static site generation.

90. Next.js Q40: What is Next.js?

Next.js is a React framework that enables server-side rendering, static site generation, and API routes for building modern web apps.

91. Next.js Q41: What are the key features of Next.js?

Features include SSR, SSG, API routes, image optimization, file-based routing, and middleware support.

92. Next.js Q42: What is server-side rendering (SSR) in Next.js?

SSR allows pages to be rendered on the server at request time, improving SEO and performance.

93. Next.js Q43: What is static site generation (SSG) in Next.js?

SSG pre-renders pages at build time and serves them as static HTML for better speed and scalability.

94. Next.js Q44: What is `getStaticProps` in Next.js?

A function used to fetch data at build time for static site generation.

95. Next.js Q45: What is Next.js?

Next.js is a React framework that enables server-side rendering, static site generation, and API routes for building modern web apps.

96. Next.js Q46: What are the key features of Next.js?

Features include SSR, SSG, API routes, image optimization, file-based routing, and middleware support.

97. Next.js Q47: What is server-side rendering (SSR) in Next.js?

SSR allows pages to be rendered on the server at request time, improving SEO and performance.

98. Next.js Q48: What is static site generation (SSG) in Next.js?

SSG pre-renders pages at build time and serves them as static HTML for better speed and scalability.

99. Next.js Q49: What is `getStaticProps` in Next.js?

A function used to fetch data at build time for static site generation.

100. Next.js Q50: What is Next.js?

Next.js is a React framework that enables server-side rendering, static site generation, and API routes for building modern web apps.