96 Blazor interview questions to hire skilled programmers

Questions

1. What is Blazor, in simple terms?

2. Can you explain the difference between Blazor Server and Blazor WebAssembly?

3. What is a component in Blazor?

4. How do you create a simple button in Blazor that displays an alert when clicked?

5. What is data binding in Blazor?

6. How do you pass data from a parent component to a child component in Blazor?

7. What are Blazor directives? Give a simple example.

8. Explain what dependency injection is and why it's useful in Blazor.

9. How do you handle events in Blazor?

10. What is the purpose of the ̀@codè block in a Blazor component?

11. How can you use CSS to style a Blazor component?

12. What is routing in Blazor, and how does it work?

13. What is the purpose of a ̀.razor̀ file?

14. Explain how you would display a list of items in a Blazor component.

15. What are lifecycle methods in Blazor components? Can you name a few?

16. How would you make an HTTP request to an API from a Blazor component?

17. What is the purpose of the ̀async̀ and ̀await̀ keywords in C# and how are they used in Blazor?

18. How do you handle errors in a Blazor application?

19. Can you describe a scenario where you would choose Blazor WebAssembly over Blazor Server, and vice versa?

20. What are some advantages of using Blazor for web development?

21. How can you debug a Blazor application?

22. What is Blazor, in simple terms?

23. Can you explain how Blazor lets us use C# in the browser?

24. What's the difference between Blazor Server and Blazor WebAssembly?

25. If the user clicks a button, how does Blazor know what to do?

26. What's a component in Blazor? Think of it like Lego bricks.

27. How do you show some text on the screen using Blazor?

28. What is data binding in Blazor? Why is it useful?

29. How do you pass information from one Blazor component to another?

30. What are event handlers in Blazor? Give a simple example.

31. How can you make your Blazor app look pretty with CSS?

32. What is Razor syntax? Why do we use it in Blazor?

33. How can you debug your Blazor code if something goes wrong?

34. What are some common Blazor project file types, such as .razor and .cs?

35. What is the purpose of the @page directive in a Blazor component?

36. How would you handle user input in a Blazor form?

37. Explain how to use dependency injection in a basic Blazor application.

38. What is a layout in Blazor and how does it improve application structure?

39. Describe a scenario where you would choose Blazor WebAssembly over Blazor Server.

40. What is the role of the _Imports.razor file in a Blazor project?

41. Explain the lifecycle of a Blazor component. What happens when it's created and rendered?

42. How can you call a JavaScript function from Blazor code?

43. What tools do you use to build and run a Blazor application?

44. How do you handle errors or exceptions in Blazor?

45. Can you describe how routing works in Blazor?

46. How would you fetch data from an API in a Blazor component?

47. What are parameters in Blazor components used for?

48. Describe a situation where you might need to use state management in Blazor.

49. What is the purpose of using NuGet packages in a Blazor project?

50. How does Blazor's component lifecycle work, and what are the key lifecycle methods you'd use to manage component state?

51. Explain the difference between ̀RenderMode.Server̀ and ̀RenderMode.WebAssemblỳ, and when would you choose each one?

52. How can you implement authentication and authorization in a Blazor application, considering both server-side and client-side scenarios?

53. Describe how to handle exceptions globally in a Blazor application to provide a user-friendly experience.

54. What are some strategies for optimizing the performance of a Blazor application, especially when dealing with large datasets or complex UI?

55. How would you implement cascading parameters in Blazor, and what are they useful for?

56. Explain how to use JavaScript interop in Blazor, and what are some common use cases for it?

57. How can you implement and use custom events in Blazor components?

58. Describe the process of creating and using a custom component library in Blazor.

59. Explain how to implement routing and navigation in a Blazor application, including handling route parameters and query strings.

60. How would you implement a two-way data binding in Blazor using ̀EventCallback̀?

61. Explain how you would handle form validation in Blazor, including using data annotations and custom validation logic.

62. What are the different ways to manage application state in Blazor, and what are the pros and cons of each approach?

63. How can you implement localization and globalization in a Blazor application to support multiple languages and cultures?

64. Describe how to implement background tasks or long-running processes in a Blazor application.

65. How can you create and use dependency injection in Blazor components and services?

66. Explain the purpose and usage of ÌJSRuntimè in Blazor.

67. How do you handle different environments (e.g., development, staging, production) in Blazor, and how do you manage configuration settings for each environment?

68. Explain the different approaches to testing Blazor components, including unit testing and integration testing.

69. How can you implement lazy loading of components or modules in a Blazor application to improve initial load time?

70. What are the benefits of using the new component virtualization feature of Blazor?

71. How does Blazor's component model differ from traditional MVC or MVVM patterns, and what advantages does it offer?

72. Explain the concept of Razor components and how they contribute to building modular and reusable UI elements in Blazor.

73. Describe the different hosting models available in Blazor (Server, WebAssembly) and their respective trade-offs regarding performance, security, and deployment.

74. How does Blazor handle state management, and what are some strategies for managing application state effectively, especially in complex applications?

75. Explain the role of dependency injection in Blazor applications and how it promotes loose coupling and testability.

76. Describe how you would implement authentication and authorization in a Blazor application, including handling user roles and permissions.

77. How can you optimize Blazor applications for performance, considering factors like minimizing component re-rendering and reducing payload size?

78. Explain the process of integrating Blazor components with JavaScript libraries or existing JavaScript code.

79. Describe the different ways to handle events in Blazor and how to pass data between components using event callbacks.

80. How can you implement error handling and exception management in a Blazor application to provide a robust and user-friendly experience?

81. Explain the purpose of the ÌJSRuntimè interface in Blazor and how it enables interaction with the browser's JavaScript environment.

82. Describe how you would implement client-side routing in a Blazor application, including handling navigation and route parameters.

83. How does Blazor handle form validation, and what are some strategies for implementing custom validation logic?

84. Explain the concept of Blazor's render tree and how it influences the rendering process of components.

85. Describe how you would implement globalization and localization in a Blazor application to support multiple languages and cultures.

86. How can you use Blazor to build Single Page Applications (SPAs), and what are the benefits and challenges of this approach?

87. Explain the process of debugging Blazor applications, including using browser developer tools and debugging in Visual Studio.

88. Describe how you would implement unit testing and integration testing for Blazor components and applications.

89. How can you use Blazor to create reusable component libraries, and what are some best practices for designing and documenting components?

90. Explain the process of deploying Blazor applications to different environments, such as Azure, AWS, or self-hosted servers.

91. Describe how you would handle data binding in Blazor, including one-way and two-way binding scenarios.

92. How can you use Blazor to build Progressive Web Apps (PWAs), and what are the key features and benefits of PWAs?

93. Explain the concept of Blazor's Virtual DOM and how it improves performance by minimizing DOM updates.

94. Describe how you would implement caching in a Blazor application to improve performance and reduce server load.

95. How can you use Blazor to build real-time applications using technologies like SignalR or WebSockets?