

47 Scala interview questions to hire top developers

Questions

1. Can you explain the primary differences between Scala and Java?
2. What are case classes in Scala and why are they useful?
3. How does Scala handle concurrency, and what libraries or frameworks have you used for this purpose?
4. What is a trait in Scala, and how does it differ from a class?
5. Can you explain the concept of pattern matching in Scala?
6. How do you implement immutability in Scala, and why is it important?
7. Describe how implicit classes and methods work in Scala.
8. What are the key benefits of using Scala's functional programming features?
9. How do you manage dependencies in a Scala project?
10. Can you give an example of how you have used higher-order functions in Scala?
11. What are the benefits of using Scala for functional programming?
12. Can you describe the role of a companion object in Scala?
13. How does Scala handle exceptions, and what is a 'Try' in Scala?
14. What is a higher-order function in Scala and how is it useful?
15. Can you explain the concept of lazy evaluation in Scala?
16. What is an Option in Scala and how do you use it?
17. Why is immutability important in Scala?
18. What are implicit parameters in Scala, and how are they useful?
19. How does Scala's 'for comprehension' differ from a traditional for loop, and when would you use it?
20. Can you explain what a 'functor' is in Scala and provide an example?
21. What is the difference between 'foldLeft' and 'foldRight' in Scala collections?
22. How does Scala's type inference work, and what are its limitations?
23. Explain the concept of 'currying' in Scala and provide a practical use case.
24. What is the purpose of the 'apply' and 'unapply' methods in Scala?
25. How do you implement and use partial functions in Scala?
26. Can you explain what 'tail recursion' is and how Scala optimizes it?
27. What are 'type classes' in Scala, and how do they support ad-hoc polymorphism?
28. How does Scala's 'Future' work, and how does it differ from Java's CompletableFuture?
29. Explain the concept of 'monads' in Scala and provide an example.
30. What is the difference between 'val', 'lazy val', and 'def' in Scala?
31. How do you use 'Either' for error handling in Scala?
32. Can you explain what 'by-name parameters' are in Scala and when to use them?
33. Can you explain what a pure function is and why it's important in functional programming?
34. How does Scala support functional programming paradigms?
35. What is the difference between map and flatMap in Scala collections?
36. Can you explain what a closure is in Scala and provide an example of its use?
37. How does Scala's for comprehension relate to functional programming concepts?
38. What is function composition in Scala, and why is it useful?
39. How would you approach debugging a complex issue in a Scala application that's causing performance degradation?
40. Describe a situation where you had to integrate a Scala application with a legacy Java system. What challenges did you face, and how did you overcome them?
41. How do you ensure your Scala code is maintainable and scalable as the codebase grows?
42. Can you describe a challenging bug you encountered in a Scala project and how you resolved it?
43. Explain a situation where you had to optimize a Scala program for better performance. What steps did you take?
44. How do you handle dependencies and library management in a Scala project, especially when working in a team?
45. Describe a time when you had to refactor a large Scala codebase. What challenges did you face, and how did you handle them?
46. How do you handle data serialization and deserialization in Scala, especially when dealing with multiple data formats?
47. Can you describe a situation where you had to write a highly concurrent or parallel program in Scala? What were the key considerations?