

# 61 Ruby interview questions to ask your applicants

## Questions

---

1. Can you explain the difference between a symbol and a string in Ruby?
2. What are Ruby blocks and how are they used?
3. How do you manage dependencies in a Ruby project?
4. What is the purpose of the `self` keyword in Ruby?
5. Can you explain the concept of mixins in Ruby?
6. What are some common uses of metaprogramming in Ruby?
7. How do you handle exceptions in Ruby?
8. What is the role of garbage collection in Ruby?
9. How do you ensure the quality and reliability of your Ruby code?
10. What is the difference between `map` and `each` methods in Ruby?
11. How do you define a class in Ruby?
12. What are instance variables and how do they differ from class variables?
13. Can you explain what a Ruby gem is and how it is used?
14. What is a module in Ruby and how is it different from a class?
15. How do you create getter and setter methods in Ruby?
16. What is the use of the `initialize` method in a Ruby class?
17. Can you explain the difference between `include` and `extend` in Ruby?
18. How do you iterate over a hash in Ruby?
19. What is the purpose of the `require` statement in Ruby?
20. How does the `respond_to?` method work?
21. What are the different types of variables in Ruby?
22. How can you create a new instance of a class in Ruby?
23. What is the purpose of the `attr_accessor` method?
24. How do you write a conditional statement in Ruby?
25. What is a Proc in Ruby and how does it differ from a lambda?
26. How do you check if an object is an instance of a certain class?
27. Can you explain what a mixin is used for in Ruby?
28. What does the `super` keyword do in Ruby?
29. How do you handle method overloading in Ruby?
30. Can you explain the concept of 'Duck Typing' in Ruby?
31. What is the significance of 'yield' in Ruby?
32. How do you handle thread safety in Ruby?
33. What is the purpose of the 'freeze' method in Ruby?
34. Can you explain the difference between 'require' and 'load' in Ruby?
35. What is the purpose of a 'singleton method' in Ruby?
36. How do you ensure your Ruby application is scalable?
37. What are Ruby's enumerable methods, and how do you use them?
38. What is the role of 'monkey patching' in Ruby?
39. How do you handle performance optimization in a Ruby application?
40. How do you optimize a Ruby on Rails application for performance?
41. Can you explain how memory management works in Ruby and how to optimize it?
42. What techniques do you use to profile and benchmark a Ruby application?
43. How would you optimize database queries in a Ruby application?
44. What are some best practices for caching in Ruby on Rails?
45. How can you reduce the memory footprint of a Ruby application?
46. What are the common performance bottlenecks in Ruby applications and how do you address them?
47. How do you handle slow tests in a Ruby test suite?
48. What tools do you use for profiling Ruby applications?
49. How can you minimize the load time of a Ruby on Rails application?
50. What strategies do you use for optimizing Ruby code for concurrency?
51. How do you ensure efficient garbage collection in a Ruby application?
52. What are the benefits and drawbacks of using just-in-time (JIT) compilation in Ruby?
53. Can you explain how to use memoization to optimize performance in Ruby?
54. How would you explain the concept of exception handling to a junior developer?
55. Can you describe a situation where you've used custom exceptions in Ruby?
56. How do you decide between using 'rescue' at the method level versus wrapping a specific line of code?
57. What's the difference between 'raise' and 'throw' in Ruby, and when would you use each?
58. How would you debug a rescued exception that doesn't provide enough information about the error?
59. Can you explain the concept of 'ensure' in Ruby's exception handling? When would you use it?
60. How do you handle errors in Ruby that occur in background jobs or separate threads?
61. What's your approach to balancing detailed error messages for debugging with security concerns in a production environment?