

48 Golang Interview Questions to Ask Talented Interviewees

Questions

1. What are Goroutines, and how are they different from threads?
2. Can you explain the purpose of the Go Playground?
3. Describe the use of the 'defer' statement in Go with an example.
4. How do you handle errors in Go?
5. What is a Go interface, and how do you implement it?
6. Explain the difference between a slice and an array in Golang.
7. What is the significance of the 'init' function in Go?
8. How does Go handle memory management and garbage collection?
9. What are some common use cases for Go channels?
10. How do you perform unit testing in Go?
11. Explain the concept of 'embedding' in Go structs.
12. What tools or libraries do you use for dependency management in Go?
13. Describe the use of context package in Go.
14. How would you handle concurrency in a Go program?
15. What are the advantages of using Go over other programming languages?
16. Can you explain the concept of 'zero values' in Go?
17. How does Go handle method overloading?
18. What is the purpose of the 'blank identifier' in Go?
19. Explain the difference between 'make' and 'new' functions in Go.
20. How does Go support concurrency at the language level?
21. What are empty structs in Go and when might you use them?
22. How does type assertion work in Go?
23. Explain the concept of method sets in Go.
24. What is the difference between := and = in Go?
25. How do you declare and use constants in Go?
26. Can you explain the concept of pointers in Go?
27. How do you create a new struct and access its fields?
28. Describe how Go handles variable shadowing.
29. What are the different types of loops in Go and how do they work?
30. How do you declare and use a function with multiple return values?
31. What is a type switch in Go?
32. How do you handle panics in Go?
33. Explain how to use Go's built-in 'recover' function.
34. How does Go support anonymous functions and closures?
35. What are the rules for naming variables and functions in Go?
36. Can you explain the concept of race conditions in Go and how to prevent them?
37. How does the select statement work in Go, and what are its use cases?
38. What is the purpose of WaitGroups in Go, and how do you use them?
39. Explain the difference between buffered and unbuffered channels in Go.
40. How does Go's garbage collector handle concurrent programs?
41. What is a mutex in Go, and how does it differ from channels for synchronization?
42. How do you implement graceful shutdown of a concurrent Go program?
43. Describe a situation where you had to optimize a Go application for performance. What approach did you take?
44. Can you describe a time when you had to debug a complex issue in a Go program? How did you go about it?
45. How do you handle dependencies in Go projects? Can you give an example of how you've managed package dependencies in a past project?
46. Tell me about a time when you had to work with a legacy codebase in Go. What challenges did you face and how did you overcome them?
47. Describe a project where you had to integrate a third-party service or API in a Go application. What steps did you take?
48. Have you ever had to make a significant change to a Go application's architecture? What was the reason and how did you approach it?