This page links to resources for learning about concurrency in Go. The items are presented in order, from beginner material to advanced topics.

Beginner

- Read Effective Go: Concurrency
- Watch Simulating a real-world system in Go
- Study The Go Programming Language Specification, especially
  - Go statements
  - Channel types
  - Send statements
  - Receive operator
  - Select statements
- Code A Tour of Go: Concurrency
- Read the Frequently Asked Questions (FAQ), especially
  - Why build concurrency on the ideas of CSP?
  - Why goroutines instead of threads?
  - Why are map operations not defined to be atomic?
  - What operations are atomic? What about mutexes?
  - Why doesn't my program run faster with more CPUs?
  - How can I control the number of CPUs?
  - What happens with closures running as goroutines?

Intermediate

- Study Go by Example from goroutines through stateful goroutines
- Watch Go Concurrency Patterns
- Watch A Practical Guide to Preventing Deadlocks and Leaks in Go
- Read Share Memory By Communicating and do the codewalk
- Read Go Concurrency Patterns: Timing out, moving on
- Watch Concurrency is not Parallelism
- Read Go Concurrency Patterns: Pipelines and Cancellation
- Read Rethinking Classical Concurrency Patterns
- Study Package sync
- Read Introducing the Go Race Detector
- Watch Go: code that grows with grace
- Read Mutexes and Semaphores Demystified

Advanced

- Watch Advanced Go Concurrency Patterns
- Read Advanced Go Concurrency Patterns
- Read Go Concurrency Patterns: Context
- Study The Go Memory Model
- Study Package atomic
- Read Principles of Designing Go APIs with Channels

- Read Advanced Go Concurrency Primitives
- Watch The Scheduler Saga
- Read The Scheduler Saga
- Watch Understanding Channels
- Read Understanding Channels