Go Kata Workbook — 1 to 100

A Zero → Hero path for learning Go through progressive katas. Each kata folder contains the authoritative README for that kata.

Kata Index

- **001** — FizzBuzz (focus: Basics: loops, conditionals, slices, strconv)
- **002** — Sum of Integers (focus: Basics: loops, function, edge cases)
- **003** — Reverse String (focus: Strings, runes vs bytes)
- **004** — Palindrome Check (focus: Strings, normalization basics)
- **005** — Count Vowels (focus: Loops, unicode, maps)
- **006** — Factorial (focus: Recursion vs loop, overflow checks)
- **007** — Fibonacci (focus: Iteration, slices)
- **008** — Filter Even (focus: Slices, order, immutability)
- **009** — Unique Strings (focus: Maps as sets, order preservation)
- **010** — Word Count (focus: Strings, fields, maps)
- **011** — Anagram Check (focus: Rune counts, normalization)
- **012** — Min/Max (focus: Errors, edge cases)
- **013** — Rotate Slice (focus: Generics, indexing)
- **014** — Binary Search (focus: Algorithms, generics)
- **015** — Merge Two Sorted Lists (focus: Two pointers)
- **016** — Run-Length Encoding (focus: Strings.Builder, runes)
- **017** — Run-Length Decoding (focus: Parsing, errors)
- **018** — Caesar Cipher (focus: ASCII letters shifting)
- **019** — ISBN-10 Validator (focus: Checksums)
- **020** — Roman Numerals (focus: Greedy mapping)
- **021** — Parse CSV Line (focus: Parsing quoted CSV)
- **022** — JSON Pretty Print (focus: encoding/json)
- **023** — HTTP Status Classifier (focus: Switch ranges)
- **024** — Time Window Check (focus: time package)
- **025** — LRU Cache (single-thread) (focus: Structs, maps, list)
- **026** — Stack (focus: Data structures, generics)
- **027** — Bracket Matcher (focus: Stacks, runes)

- **028** — Markdown Heading Extractor (focus: Line scanning)
- **029** — Longest Common Prefix (focus: Strings)
- **030** — Kebab/Snake → Camel (focus: String transforms)
- **031** — File Extension Counter (focus: os, filepath)
- **032** — Top N Words (focus: Sorting, ties)
- **033** — CSV to JSON (focus: I/O, encoding)
- **034** — Temperature Converter (focus: Parsing floats)
- **035** — Log Line Parser (focus: Key=value parsing)
- **036** — Rate Limiter (token bucket) (focus: Time, structs)
- **037** — INI Parser (focus: Sections, maps)
- **038** — Mini Template Renderer (focus: Placeholders)
- **039** — UUID v4 Generator (focus: crypto/rand)
- **040** — Password Strength Scorer (focus: Rules)
- **041** — HTTP Query Builder (focus: net/url)
- **042** — Context Timeout Wrapper (focus: context)
- **043** — Retry with Backoff (focus: time)
- **044** — Concurrent MapReduce (focus: Goroutines, channels)
- **045** — Worker Pool (focus: Concurrency patterns)
- **046** — Safe Counter (focus: sync/atomic or mutex)
- **047** — Debounce (focus: Time, channels)
- **048** — Throttle (focus: Time, channels)
- **049** — JSON Patch (subset) (focus: encoding/json)
- **050** — Config Loader (focus: env + file)
- **051** — HTTP Client with Retries (focus: net/http)
- **052** — TCP Echo Server (focus: net)
- **053** — Line-Oriented Reader (focus: bufio)
- **054** — JSON Lines Filter (focus: Stream processing)
- **055** — Mini HTTP Router (focus: net/http)
- **056** — Middleware Chain (focus: net/http)
- **057** — In-Memory KV Store (focus: Maps, RWMutex)
- **058** — Simple Pub/Sub (focus: Fanout)
- **059** — Fixed-Window Metrics (focus: Time buckets)
- **060** — Trie Autocomplete (focus: Data structures)

- **061** — HTTP Server with Graceful Shutdown (focus: net/http, context)
- **062** — Context-aware Logger (focus: Advanced practice)
- **063** — Structured Errors (focus: Advanced practice)
- **064** — Circuit Breaker (focus: State machines, time)
- **065** — Concurrent File Downloader (focus: Concurrency, net/http)
- **066** — Semaphore (focus: Concurrency primitives)
- **067** — Barrier (focus: Concurrency coordination)
- **068** — Fan-in/Fan-out Pipeline (focus: Channels, composition)
- **069** — Batcher (focus: Time, channels)
- **070** — LRU Cache (thread-safe) (focus: Advanced practice)
- **071** — Token Bucket (thread-safe) (focus: Advanced practice)
- **072** — Bloom Filter (focus: Advanced practice)
- **073** — Merkle Tree (focus: Advanced practice)
- **074** — Binary Heap Priority Queue (focus: Advanced practice)
- **075** — Dijkstra Shortest Path (focus: Advanced practice)
- **076** — A* Grid Pathfinding (focus: Advanced practice)
- **077** — Event Sourcing Mini (focus: Advanced practice)
- **078** — SQLite-backed Repo (pure Go driver) (focus: Advanced practice)
- **079** — JSON Schema Validator (subset) (focus: Advanced practice)
- **080** — Rate-limited HTTP Scraper (focus: Advanced practice)
- **081** — Websocket Chat (basic) (focus: Advanced practice)
- **082** — SSE Stream (focus: Advanced practice)
- **083** — JWT Sign/Verify (focus: crypto, encoding)
- **084** — HMAC Request Signing (focus: crypto/hmac)
- **085** — Password Hashing (bcrypt/argon2) (focus: crypto, security)
- **086** — File Integrity Checker (focus: io, crypto/sha256)
- **087** — Tar/Gzip Archiver (focus: archive/tar, compress/gzip)
- **088** — Diff (line-based) (focus: Algorithms)
- **089** — Mini grep (focus: CLI, IO)
- **090** — Mini wc (focus: CLI, IO)
- **091** — Mini cut (focus: CLI, IO)
- **092** — Mini head/tail (focus: CLI, IO)
- **093** — Mini sort (focus: CLI, IO)

- **094** — Mini uniq (focus: CLI, IO)
- **095** — Concurrent Test Harness (focus: Advanced practice)
- **096** — Benchmarking Kata (focus: testing/benchmark)
- **097** — Generics: Set/Map Utilities (focus: Advanced practice)
- **098** — Generics: Optional/Result (focus: Advanced practice)
- **099** — Reflection: Struct Tag Parser (focus: reflect)
- **100** — Plugin-free DI Container (focus: Design)

How to work

Open a kata folder, read its README, implement kata.go, then replace the t.Skip in kata_test.go with real tests. Run go test ./.... Submit one kata at a time for marking and correction.