

Redis Cheat Sheet

Connection & Admin (redis-cli)

Redis is an in-memory key-value data structure store used for caching, sessions, and message brokering.

- **CLI Login:** `redis-cli -h 127.0.0.1 -p 6379 -a "password"`
- **Ping:** `PING -> PONG`
- **Select Database:** `SELECT 0` (Databases 0-15)
- **Total Keys:** `DBSIZE`
- **Flush Data:** `FLUSHDB` (current database), `FLUSHALL` (all databases)
- **Server Stats:** `INFO` (memory, CPU, connections)

General Key Operations

- **KEYS pattern** : Matches keys by pattern (e.g. `KEYS user:*`) (Avoid in production)
- **SCAN cursor [MATCH pattern] [COUNT count]** : Non-blocking key cursor iteration
- **EXISTS key** : Check key existence (returns 1 or 0)
- **DEL key [key ...]** : Delete one or more keys
- **TYPE key** : Inspect data structure type
- **EXPIRE key seconds** : Set key time-to-live in seconds
- **TTL key** : Check remaining TTL (-1: infinite, -2: expired/missing)
- **PERSIST key** : Remove timeout expiration
- **RENAME key newkey** : Rename key

Core Data Structures & Commands

1. Strings

- **SET key value** : Set string value
- **GET key** : Retrieve value
- **SETEX key seconds value** : Set with expiration
- **SETNX key value** : Set only if not exists (Distributed lock primitive)
- **MSET k1 v1 k2 v2 / MGET k1 k2** : Multi-key operations
- **INCR key / DECR key** : Atomic increment/decrement by 1

- **INCRBY key increment** : Atomic increment by amount

2. Hashes (Object Fields)

- **HSET user:1 name "Alice" age "28"** : Set hash fields
- **HGET user:1 name** : Get single field
- **HMGET user:1 name age** : Get multiple fields
- **HGETALL user:1** : Retrieve all fields and values
- **HDEL user:1 age** : Delete field
- **HEXISTS user:1 email** : Verify field existence
- **HINCRBY user:1 visits 1** : Increment numeric field

3. Lists (Queues & Stacks)

- **LPUSH mylist val / RPUSH mylist val** : Prepend / append element
- **LPOP mylist / RPOP mylist** : Remove from left / right
- **LRANGE mylist 0 -1** : Get full range
- **LLEN mylist** : Return list length
- **BLPOP mylist 10** : Blocking left pop with timeout

4. Sets (Unique Elements)

- **SADD myset "a" "b" "c"** : Add members to set
- **SMEMBERS myset** : List all members
- **SISMEMBER myset "a"** : Check membership
- **SREM myset "a"** : Remove member
- **SCARD myset** : Get cardinality
- **SINTER set1 set2** : Set intersection
- **SUNION set1 set2** : Set union
- **SDIFF set1 set2** : Set difference

5. Sorted Sets (ZSet - Leaderboards & Priority)

- **ZADD leaderboard 100 "user1" 250 "user2"** : Add members with score
- **ZRANGE leaderboard 0 -1 [WITHSCORES]** : Ascending score range
- **ZREVRANGE leaderboard 0 9 WITHSCORES** : Descending TOP 10
- **ZRANK leaderboard "user1"** : Ascending rank index
- **ZREVRANK leaderboard "user1"** : Descending rank index

- **ZSCORE leaderboard "user1"** : Member score lookup
- **ZINCRBY leaderboard 50 "user1"** : Increment member score

Pub/Sub & Messaging

- **Publish:** `PUBLISH news "Urgent alert"`
- **Subscribe:** `SUBSCRIBE news sports`
- **Pattern Subscribe:** `PSUBSCRIBE news.*`

Transactions

- **MULTI**
- **SET account:1 500**
- **INCR transaction:count**
- **EXEC** (or **DISCARD**)
- **Optimistic Locking:** `WATCH account:1`

Persistence & Monitoring

- **Asynchronous Snapshot:** `BGSAVE`
- **Live Command Stream:** `MONITOR`
- **Slow Query Log:** `SLOWLOG GET 10`