

# Redis Riferimento Rapido

Stringhe, liste, set, hash, pub/sub, persistenza

## Connessione

### CLI

```
redis-cli
redis-cli -h 127.0.0.1 -p 6379
redis-cli -a password -n 2
redis-cli --tls -u rediss://user:pass@host:6380
```

### Connessione con Driver (Python)

```
import redis
r = redis.Redis(host='localhost', port=6379, db=0)
r.set('key', 'value')
print(r.get('key'))
```

## Informazioni Server

```
PING -- restituisce PONG
INFO server -- dettagli server
INFO memory -- utilizzo memoria
DBSIZE -- numero di chiavi nel db corrente
```

## Stringhe

### Operazioni di Base

```
SET name "Alice"
GET name
SET counter 100
MSET a 1 b 2 c 3
MGET a b c
```

### Operazioni Numeriche

```
INCR counter -- 101
INCRBY counter 10 -- 111
DECR counter -- 110
DECRBY counter 5 -- 105
INCRBYFLOAT price 2.5
```

### Comandi sulle Stringhe

|                          |                                      |
|--------------------------|--------------------------------------|
| <b>SET key val</b>       | Imposta valore stringa               |
| <b>GET key</b>           | Ottieni valore stringa               |
| <b>SETNX key val</b>     | Imposta solo se la chiave non esiste |
| <b>SETEX key sec val</b> | Imposta con scadenza in secondi      |
| <b>APPEND key val</b>    | Aggiunge al valore esistente         |
| <b>STRLEN key</b>        | Lunghezza del valore stringa         |

## Liste

### Operazioni sulle Liste

```
LPUSH queue "first"
RPUSH queue "last"
LRANGE queue 0 -1 -- tutti gli elementi
LPOP queue
RPOP queue
```

### Comandi sulle Liste

|                              |   |
|------------------------------|---|
| <b>LPUSH / RPUSH</b>         | Inserisce a sinistra / destra della lista |
| <b>LPOP / RPOP</b>           | Estrae da sinistra / destra               |
| <b>LRANGE key start stop</b> | Ottieni intervallo di elementi            |
| <b>LLEN key</b>              | Lunghezza della lista                     |
| <b>LINDEX key idx</b>        | Elemento all'indice                       |
| <b>LREM key count val</b>    | Rimuovi count occorrenze di val           |
| <b>BLPOP key timeout</b>     | Pop bloccante (per code)                  |

## Set e Set Ordinati

### Operazioni sui Set

```
SADD tags "python" "redis" "docker"
SMEMBERS tags
SISMEMBER tags "python" -- 1 (vero)
SREM tags "docker"
SCARD tags -- conteggio
```

### Operazioni sui Set

```
SUNION set1 set2 -- unione
SINTER set1 set2 -- intersezione
SDIFF set1 set2 -- differenza
```

### Operazioni sui Set Ordinati

```
ZADD leaderboard 100 "Alice" 85 "Bob"
ZRANGE leaderboard 0 -1 WITHSCORES
ZREVRANGE leaderboard 0 2
ZSCORE leaderboard "Alice"
ZRANK leaderboard "Alice" -- rank in base 0
```

### Comandi sui Set Ordinati

|                                  |                                     |
|----------------------------------|-------------------------------------|
| <b>ZADD key score member</b>     | Aggiunge membro con punteggio       |
| <b>ZRANGE key start stop</b>     | Intervallo per rank (dal basso)     |
| <b>ZREVRANGE key start stop</b>  | Intervallo per rank (dall'alto)     |
| <b>ZINCRBY key incr member</b>   | Incrementa punteggio membro         |
| <b>ZRANGEBYSCORE key min max</b> | Intervallo per valore del punteggio |
| <b>ZCARD key</b>                 | Numero di membri                    |

## Hash

### Operazioni sugli Hash

```
HSET user:1 name "Alice" age 30
HGET user:1 name
HGETALL user:1
HMSET user:2 name "Bob" age 25
HMGET user:1 name age
```

### Comandi sugli Hash

|                            |                                |
|----------------------------|--------------------------------|
| <b>HSET key field val</b>  | Imposta campo hash             |
| <b>HGET key field</b>      | Ottieni campo hash             |
| <b>HGETALL key</b>         | Ottieni tutti i campi e valori |
| <b>HDEL key field</b>      | Elimina campo hash             |
| <b>HEXISTS key field</b>   | Verifica esistenza campo       |
| <b>HINCRBY key field n</b> | Incrementa valore campo        |
| <b>HKEYS key</b>           | Tutti i nomi dei campi         |
| <b>HLEN key</b>            | Numero di campi                |

## Chiavi e Scadenza

### Comandi sulle Chiavi

|                              |  |
|------------------------------|--|
| <b>KEYS pattern</b>          | Trova chiavi corrispondenti al pattern (lento) |
| <b>SCAN cursor MATCH pat</b> | Itera le chiavi in modo incrementale (sicuro)  |
| <b>EXISTS key</b>            | Verifica se la chiave esiste                   |
| <b>DEL key</b>               | Elimina chiave                                 |
| <b>TYPE key</b>              | Tipo di dato della chiave                      |
| <b>RENAME key newkey</b>     | Rinomina una chiave                            |

### Comandi di Scadenza

```
EXPIRE key 3600 -- scade in 1 ora
PEXPIRE key 5000 -- scade in 5000 ms
TTL key -- secondi alla scadenza
PTTL key -- ms alla scadenza
PERSIST key -- rimuovi scadenza
```

## Pattern di Chiave

```
SET session:abc123 "data" EX 1800
-- EX = secondi, PX = millisecondi
-- NX = solo se non esiste
-- XX = solo se esiste
SET lock:order42 "owner" NX EX 10
```

## Pub/Sub

### Pub/Sub di Base

```
-- Subscriber (terminale 1)
SUBSCRIBE news alerts

-- Publisher (terminale 2)
PUBLISH news "Breaking: Redis 8 released"
```

### Sottoscrizione per Pattern

```
PSUBSCRIBE news.*
-- corrisponde a news.tech, news.sports, ecc.
```

### Comandi Pub/Sub

|                            |                             |
|----------------------------|-----------------------------|
| <b>SUBSCRIBE channel</b>   | Ascolta messaggi sul canale |
| <b>PUBLISH channel msg</b> | Invia messaggio al canale   |
| <b>PSUBSCRIBE pattern</b>  | Sottoscrive a un pattern    |
| <b>UNSUBSCRIBE channel</b> | Smetti di ascoltare         |
| <b>PUBSUB CHANNELS</b>     | Elenca i canali attivi      |

## Transazioni

### MULTI / EXEC

```
MULTI
SET balance:1 900
SET balance:2 1100
EXEC -- esegue atomicamente
```

### Lock Ottimistico

```
WATCH balance:1
val = GET balance:1 -- legge valore corrente
MULTI
SET balance:1 (val - 100)
EXEC
-- EXEC restituisce nil se balance:1 è cambiato
```

### Comandi di Transazione

|                  |   |
|------------------|---|
| <b>MULTI</b>     | Inizia il blocco transazione                          |
| <b>EXEC</b>      | Esegue i comandi in coda                              |
| <b>DISCARD</b>   | Scarta i comandi in coda                              |
| <b>WATCH key</b> | Monitora la chiave per cambiamenti (lock ottimistico) |
| <b>UNWATCH</b>   | Dimentica tutte le chiavi monitorate                  |

## Persistenza

### Snapshot RDB

|                 |                                      |
|-----------------|--------------------------------------|
| <b>SAVE</b>     | -- snapshot sincrono                 |
| <b>BGSAVE</b>   | -- snapshot in background            |
| <b>LASTSAVE</b> | -- timestamp dell'ultimo salvataggio |

### AOF (Append Only File)

|                             |  |
|-----------------------------|--|
| <b>appendonly yes</b>       | Abilita AOF in redis.conf                          |
| <b>appendfsync always</b>   | Fsync ad ogni scrittura (più sicuro, più lento)    |
| <b>appendfsync everysec</b> | Fsync ogni secondo (raccomandato)                  |
| <b>appendfsync no</b>       | Lascia decidere all'OS (più veloce, più rischioso) |

# Redis Riferimento Rapido

---

## Comandi di Persistenza

```
CONFIG GET save
CONFIG SET save "900 1 300 10"
-- snapshot se 1 modifica in 900s o 10 in 300s
BGREWRITEAOF -- riscrive AOF in background
```

## Pattern Comuni

### Lock Distribuito

```
SET lock:resource "owner-id" NX EX 30
-- NX = acquisisci solo se non in possesso
-- EX 30 = rilascio automatico dopo 30s
DEL lock:resource -- rilascio esplicito
```

### Rate Limiter

```
key = "rate:user:42"
INCR key
EXPIRE key 60 -- finestra di 60 secondi
-- rifiuta se GET key > max_requests
```

### Pattern di Cache

```
val = GET "cache:user:1"
if val is nil:
  val = fetch_from_db(1)
  SET "cache:user:1" val EX 300
```

### Storage di Sessione

```
HSET sess:abc uid 42 role "admin"
EXPIRE sess:abc 1800 -- TTL 30 min
HGETALL sess:abc
```