## 1.redis常用命令

String(字符串)：

|  |
| --- |
| redis 127.0.0.1:6379> SET name "runoob"  OK  redis 127.0.0.1:6379> GET name  "runoob"  redis 127.0.0.1:6379> DEL runoobkey //删除  (integer) 1 |

Hash(哈希)：

Redis hash 是一个键名对集合。

Redis hash是一个string类型的field和value的映射表，hash特别适合用于存储对象。

|  |
| --- |
| 127.0.0.1:6379> HMSET **user:1** username runoob password runoob points 200  OK  127.0.0.1:6379> HGETALL user:1  1) "username"  2) "runoob"  3) "password"  4) "runoob"  5) "points"  6) "200"  user:1为键值 |

List（列表）

Redis 列表是简单的字符串列表，按照插入顺序排序。你可以添加一个元素到列表的头部（左边）或者尾部（右边）。

|  |
| --- |
| redis 127.0.0.1:6379> lpush runoob redis  (integer) 1  redis 127.0.0.1:6379> lpush runoob mongodb  (integer) 2  redis 127.0.0.1:6379> lpush runoob rabitmq  (integer) 3  redis 127.0.0.1:6379> lrange runoob 0 10  1) "rabitmq"  2) "mongodb"  3) "redis"  redis 127.0.0.1:6379> |

Set（集合）

Redis的Set是string类型的无序集合。

sadd 命令

添加一个string元素到,key对应的set集合中，成功返回**1**,如果元素已经在集合中返回**0**,key对应的set不存在返回错误

|  |
| --- |
| redis 127.0.0.1:6379> sadd runoob redis  (integer) 1  redis 127.0.0.1:6379> sadd runoob mongodb  (integer) 1  redis 127.0.0.1:6379> sadd runoob rabitmq  (integer) 1  redis 127.0.0.1:6379> sadd runoob rabitmq  (integer) 0  redis 127.0.0.1:6379> smembers runoob  1) "rabitmq"  2) "mongodb"  3) "redis" |

zset(sorted set：有序集合)

Redis zset 和 set 一样也是string类型元素的集合,且不允许重复的成员。

不同的是每个元素都会关联一个double类型的分数。redis正是通过分数来为集合中的成员进行从小到大的排序。

zset的成员是唯一的,但分数(score)却可以重复。

zadd 命令

添加元素到集合，元素在集合中存在则更新对应score

zadd key score member

|  |
| --- |
| redis 127.0.0.1:6379> zadd runoob 0 redis  (integer) 1  redis 127.0.0.1:6379> zadd runoob 0 mongodb  (integer) 1  redis 127.0.0.1:6379> zadd runoob 0 rabitmq  (integer) 1  redis 127.0.0.1:6379> zadd runoob 0 rabitmq  (integer) 0  redis 127.0.0.1:6379> ZRANGEBYSCORE runoob 0 1000  1) "redis"  2) "mongodb"  3) "rabitmq" |

## 2.redis keys 命令

|  |  |
| --- | --- |
| 1 | [DEL key](http://www.runoob.com/redis/keys-del.html) 该命令用于在 key 存在时删除 key。 |
| 2 | [DUMP key](http://www.runoob.com/redis/keys-dump.html)  序列化给定 key ，并返回被序列化的值。 |
| 3 | [EXISTS key](http://www.runoob.com/redis/keys-exists.html)  检查给定 key 是否存在。 |
| 4 | [EXPIRE key](http://www.runoob.com/redis/keys-expire.html) seconds 为给定 key 设置过期时间。 |
| 5 | [EXPIREAT key timestamp](http://www.runoob.com/redis/keys-expireat.html)  EXPIREAT 的作用和 EXPIRE 类似，都用于为 key 设置过期时间。 不同在于 EXPIREAT 命令接受的时间参数是 UNIX 时间戳(unix timestamp)。 |
| 6 | [PEXPIRE key milliseconds](http://www.runoob.com/redis/keys-pexpire.html)  设置 key 的过期时间以毫秒计。 |
| 7 | [PEXPIREAT key milliseconds-timestamp](http://www.runoob.com/redis/keys-pexpireat.html)  设置 key 过期时间的时间戳(unix timestamp) 以毫秒计 |
| 8 | [KEYS pattern](http://www.runoob.com/redis/keys-keys.html)  查找所有符合给定模式( pattern)的 key 。 |
| 9 | [MOVE key db](http://www.runoob.com/redis/keys-move.html)  将当前数据库的 key 移动到给定的数据库 db 当中。 |
| 10 | [PERSIST key](http://www.runoob.com/redis/keys-persist.html)  移除 key 的过期时间，key 将持久保持。 |
| 11 | [PTTL key](http://www.runoob.com/redis/keys-pttl.html)  以毫秒为单位返回 key 的剩余的过期时间。 |
| 12 | [TTL key](http://www.runoob.com/redis/keys-ttl.html)  以秒为单位，返回给定 key 的剩余生存时间(TTL, time to live)。 |
| 13 | [RANDOMKEY](http://www.runoob.com/redis/keys-randomkey.html)  从当前数据库中随机返回一个 key 。 |
| 14 | [RENAME key newkey](http://www.runoob.com/redis/keys-rename.html)  修改 key 的名称 |
| 15 | [RENAMENX key newkey](http://www.runoob.com/redis/keys-renamenx.html)  仅当 newkey 不存在时，将 key 改名为 newkey 。 |
| 16 | [TYPE key](http://www.runoob.com/redis/keys-type.html)  返回 key 所储存的值的类型。 |

## 3.字符串命令