



链滴

Redis 命令小记

作者: [StefanYang](#)

原文链接: <https://ld246.com/article/1544526385921>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

字符串命令

字符串可以存储以下类型的值: 字符串; 整数; 浮点数

示例: 键名(key name) 值的类型(String)

```
GET  获取存储在给定键中的值  get hello
SET  设置存储在给定键中的值  set hello world
DEL  删除存储在给定键中的值  del hello
```

Redis中的自增命令和自减命令:

```
INCR  INCR key-name —— 将键存储的值加上1
DECR  DECR key-name —— 将键存储的值减去1
INCRBY INCRBY key-name amount —— 将键存储的值加上整数amount
DECRBY DECRBY key-name amount —— 将键存储的值减去整数amount
INCRBYFLOAT INCRBYFLOAT key-name amount —— 将键存储的值加上浮点数amount
```

处理字符串子串和二进制位的命令:

```
APPEND  APPEND key-name value —— 将值value追加到给定键key-name当前存储的值的末尾
GETRANGE  GETRANGE key-name start end —— 获取一个由偏移量start至偏移量end范围内所
字符组成的子串,包括start和end在内
SETRANGE  SETRANGE key-name start value —— 将从start偏移量开始的子串设置为定值
GETBIT  GETBIT key-name offset —— 将字符串看作是二进制位串(bit string), 并返回位串中偏移
为offset的二进制的值
SETBIT  SETBIT key-name offset value —— 将字符串看作是二进制位串, 并将串中偏移量为offse
的二进制位的值设置为value
BITCOUNT  BITCOUNT key-name [start end] —— 统计二进制位串里面为1的二进制的数量
BITOP  BITOP operation dest-key key-name [key-name ...] —— 对一个或多个二进制位串执行
括并(AND)、或(OR)、异或(XOR)、非(NOT)在内的任意一种按位运算操作,并将结果保存在dest-key
面
```

列表命令

示例: 键名(key name) 值的类型(list_有序列表)

```
LPUSH  将元素推入列表的左端(left end)  lpush list-key item
RPUSH  将元素推入列表的右端(right end)  rpush list-key item2
LPOP  从列表的左端弹出一个值,并返回被弹出的值  lpop list-key
RPOP  从列表的右端弹出一个值,并返回被弹出的值  rpop list-key
LINDEX  获取列表在给定位置上的单个元素  lrange list-key 1
LRANGE  获取列表在给定范围上的所有元素  lrange list-key 0 -1
```

一些常用的列表命令:

```
RPUSH  RPUSH key-name value [value ...] —— 将一个或多个值推入列表的右端
LPUSH  LPUSH key-name value [value ...] —— 将一个或多个值推入列表的左端
RPOP  RPOP key-name —— 移除并返回列表最右端的元素
LPOP  LPOP key-name —— 移除并返回列表最左端的元素
LINDEX  LINDEX key-name offset —— 返回列表中偏移量为offset的元素
LRANGE  LRANGE key-name start end —— 返回列表从start偏移量到end偏移量范围内的所有
素, 其中偏移量为start和end的元素也会包含在返回元素之内
```

LTRIM `LTRIM key-name start end` —— 对列表进行修剪, 只保留从start偏移量到end偏移量范围的元素, 其中偏移量为start和偏移量为end的元素也会被保留

阻塞式的列表弹出命令以及在列表之间移动元素的命令:

BLPOP `BLPOP key-name [key-name ...] timeout` —— 从第一个非空列表中弹出位于最左端的元素 或者在timeout秒之内阻塞并等待可弹出的元素出现

BRPOP `BRPOP key-name [key-name ...] timeout` —— 从第一个非空列表中弹出位于最右端的元素, 或者在timeout秒之内阻塞并等待可弹出的元素出现

RPOPLPUSH `RPOPLPUSH source-key dest-key` —— 从source-key列表中弹出位于最右端的元素 然后将这个元素推入dest-key列表的最左端, 并向用户返回这个元素

BRPOPLPUSH `BRPOPLPUSH source-key dest-key timeout` —— 从source-key列表中弹出位于右端的元素, 然后将这个元素推入dest-key列表的最左端, 并向用户返回这个元素; 如果source-key为空那么在timeout秒之内阻塞并等待可弹出的元素出现

集合命令

示例: 键名(key name) 值的类型(set_无序集合)

SADD 将给定元素添加到集合 `sadd set-key item`

SMEMBERS 返回集合包含的所有元素 `smembers set-key`

SISMEMBER 检查指定元素是否存在于集合中 `sismember set-key item`

SREM 如果给定的元素存在于集合中,那么移除这个元素 `srem set-key item`

一些常用的集合命令:

SADD `SADD key-name item [item ...]` —— 将一个或多个元素添加到集合里面, 并返回被添元素中原本并不存在于集合里面的元素数量

SREM `SREM key-name item [item ...]` —— 从一个集合里面移除一个或多个元素, 并返回被移的元素的数量

SISMEMBER `SISMEMBER key-name item` —— 检查元素item是否存在于集合key-name里

SCARD `SCARD key-name` —— 返回集合包含的元素的数量

SMEMBERS `SMEMBERS key-name` —— 返回集合包含的所有元素

SRANDMEMBER `SRANDMEMBER key-name [count]` —— 从集合里面随机低返回一个或多个元素。当count为正数时,命令返回的随即元素不会重复;当count为负数时, 命令返回的随机元素可能重复

SPOP `SPOP key-name` —— 随机地移除集合中的一个元素, 并返回被移除的元素

SMOVE `SMOVE source-key dest-key item` —— 如果集合source-key包含元素item, 那么集合source-key里面移除元素item, 并将元素item添加到集合dest-key中; 如果item被成功移除, 么返回1, 否则返回0

用户组合和处理多个集合的Redis命令:

SDIFF `SDIFF key-name [key-name ...]` —— 返回那些存在于第一个集合, 但不存在于其它集中的元素 (数学上的差集运算)

SDIFFSTORE `SDIFFSTORE dest-key key-name [key-name ...]` —— 将那些存在于第一个集但不存在于其它集中的元素 (数学上的差集运算) 存储到dest-key键里面

SINTER `SINTER key-name [key-name ...]` —— 返回那些同时存在于所有集合中的元素 (数学上的交集运算)

SINTERSTORE `SINTERSTORE dest-key key-name [key-name ...]` —— 将那些同时存在于所集合的元素 (数学上的交集运算) 存储到dest-key键里面

SUNION `SUNION key-name [key-name ...]` —— 返回那些至少存在于一个集合中的元素 (数学上的并集计算)

SUNIONSTORE `SUNIONSTORE dest-key key-name [key-name ...]` —— 将那些至少存在于

个集合中的元素（数学上的并集计算）存储到dest-key里面

散列命令

示例: 键名(key name) 值的类型(map_键值对集合)

HSET 在散列里面关联起给定的键值对 hset hash-key sub-key1 value1
HGET 获取指定散列键的值 hget hash-key sub-key1
HGETALL 获取散列中包含的所有键值对 hgetall hash-key
HDEL 如果给定的键存在于散列里面,那么移除这个键 hdel hash-key sub-key1

用于添加和删除键值对的散列操作:

HMGET HMGET key-name key [key ...] —— 从散列里获取一个或多个键的值
HMSET HMSET key-name key value [key value ...] —— 为散列里面一个或多个键设置值
HDEL HDEL key-name key [key ...] —— 删除散列里面的一个或多个键值对, 返回成功找到删除的键值对数量
HLEN HLEN key-name —— 返回散列包含的键值对数量

Redis散列的更高级特性:

HEXISTS HEXISTS key-name key —— 检查给定键是否存在于散列中
HKEYS HKEYS key-name —— 获取散列包含的所有键
HVALS HVALS key-name —— 获取散列包含的所有值
HGETALL HGETALL key-name —— 获取散列包含的所有键值对
HINCRBY HINCRBY key-name key increment —— 将键key存储的值加上整数increment
HINCRBYFLOAT HINCRBYFLOAT key-name key increment —— 将键key存储的值加上浮数increment

有序集合命令

示例: 键名(key name) 值的类型(map_键值对集合) (键值对中的键被称为成员member,值被称为值score)

ZADD 将一个带有给定分值的成员添加到有序集合里面 zadd zset-key 788 member1
ZRANGE 根据元素在有序排列中所处的位置,从有序集合里面获取多个元素 zrange zset-key 0 -1 withscores
ZRANGEBYSCORE 获取有序集合在给定分值范围内的所有元素 zrangebyscore zset-key 0 800 withscores
ZREM 如果给定成员在有序集合中,那么移除这个成员 zrem zset-key member1

一些常用的有序集合命令:

ZADD ZADD key-name score member [score member ...] —— 将带有给定分值的成员添加到有序集合里面
ZREM ZREM key-name member [member ...] —— 从有序集合里面移除给定的成员, 并返回被移除成员的数量
ZCARD ZCARD key-name —— 返回有序集合包含的成员数量
ZINCRBY ZINCRBY key-name increment member —— 将member成员的分值加上increment
ZCOUNT ZCOUNT key-name min max —— 返回分值介于min和max之间的成员数量
ZRANK ZRANK key-name member —— 返回成员member在有序集合中的排名
ZSCORE ZSCORE key-name member —— 返回成员member的分值
ZRANGE ZRANGE key-name start stop [WITHSCORES] —— 返回有序集合中排名介于start

和stop之间的成员，如果给定了可选的WITHSCORES选项，那么命令会将成员的分值也一并返回

有序集合的范围型数据获取命令和范围型数据删除命令，以及并集命令和交集命令：

ZREVRANK ZREVRANK key-name member —— 返回有序集合里成员member的排名，成按照分值从大到小排列
ZREVRANGE ZREVRANGE key-name start stop [WITHSCORES] —— 返回有序集合给定范围内的成员，成员按照分值从大到小排列
ZRANGEBYSCORE ZRANGEBYSCORE key min max [WITHSCORES] [LIMIT offset count] —— 返回有序集合中，分值介于min和max之间的所有成员
ZREVRANGEBYSCORE ZREVRANGEBYSCORE key max min [WITHSCORES] [LIMIT offset count] —— 获取有序集合中分值介于min和max之间的所有成员，并按照分值从大到小的顺序来返回它们
ZREMRANGEBYRANK ZREMRANGEBYRANK key-name start stop —— 移除有序集合中排介于start和stop之间的所有成员
ZREMRANGEBYSCORE ZREMRANGEBYSCORE key-name min max —— 移除有序集合中值介于min和max之间的所有成员
ZINTERSTORE ZINTERSTORE dest-key key-count key [key ...] [WEIGHTS weight [weight ...]] [AGGREGATE SUM|MIN|MAX] —— 对给定的有序集合执行类似于集合的交集运算
ZUNIONSTORE ZUNIONSTORE dest-key key-count key [key ...] [WEIGHTS weight [weight ...]] [AGGREGATE SUM|MIN|MAX] —— 对给定的有序集合执行类似于集合的并集运算

Redis提供的发布与订阅命令

SUBSCRIBE SUBSCRIBE channel [channel ...] —— 订阅给定的一个或多个频道
UNSUBSCRIBE UNSUBSCRIBE [channel [channel ...]] —— 退订给定的一个或多个频道，如执行时没有给定任何频道，那么退订所有频道
PUBLISH PUBLISH channel message —— 向给定频道发送消息
PSUBSCRIBE PSUBSCRIBE pattern [pattern ...] —— 订阅与给定模式相匹配的所有频道
PUNSUBSCRIBE PUNSUBSCRIBE [pattern [pattern ...]] —— 退订给定的模式，如果执行时有给定任何模式，那么退订所有频道

排序：SORT命令的定义

SORT SORT source-key [BY pattern] [LIMIT offset count] [GET pattern [GET pattern ...]] [ASC DESC] [ALPHA] [STORE dest-key] —— 根据给定的选项，对输入列表、集合或有序集合进行排序，然后返回或者存储排序的结果

用于处理过期时间的Redis命令

PERSIST PERSIST key-name —— 移除键的过期时间
TTL TTL key-name —— 查看给定键距离过期还有多少秒
EXPIRE EXPIRE key-name seconds —— 让给定键在指定的秒数之后过期
EXPIREAT EXPIREAT key-name timestamp —— 将给定键的过期时间设置为给定的UNIX时间戳
PTTL PTTL key-name —— 查看给定键的过期时间还剩多少毫秒
PEXPIRE PEXPIRE key-name milliseconds —— 让给定键在指定的毫秒数之后过期
PEXPIREAT PEXPIREAT key-name timestamp-milliseconds —— 将一个毫秒级精度的UNIX时间戳设置为给定键的过期时间