# Redis 命令



Redis 命令用于在 redis 服务上执行操作。

要在 redis 服务上执行命令需要一个 redis 客户端。Redis 客户端在我们之前下载的的 redis 的安装包中。

### **语法**

Redis 客户端的基本语法为：

$ redis-cli

### **实例**

以下实例讲解了如何启动 redis 客户端：

启动 redis 客户端，打开终端并输入命令 **redis-cli**。该命令会连接本地的 redis 服务。

$redis-cli

redis 127.0.0.1:6379>

redis 127.0.0.1:6379> PING

PONG

在以上实例中我们连接到本地的 redis 服务并执行 **PING** 命令，该命令用于检测 redis 服务是否启动。

## 在远程服务上执行命令

如果需要在远程 redis 服务上执行命令，同样我们使用的也是 **redis-cli** 命令。

### **语法**

$ redis-cli -h host -p port -a password

### **实例**

以下实例演示了如何连接到主机为 127.0.0.1，端口为 6379 ，密码为 mypass 的 redis 服务上。

$redis-cli -h 127.0.0.1 -p 6379 -a "mypass"

redis 127.0.0.1:6379>

redis 127.0.0.1:6379> PING

PONG

# Redis 键(key)

Redis 键命令用于管理 redis 的键。

### **语法**

Redis 键命令的基本语法如下：

redis 127.0.0.1:6379> COMMAND KEY\_NAME

### **实例**

redis 127.0.0.1:6379> SET w3ckey redis

OK

redis 127.0.0.1:6379> DEL w3ckey(integer) 1

在以上实例中 **DEL** 是一个命令， **w3ckey** 是一个键。 如果键被删除成功，命令执行后输出 **(integer) 1**，否则将输出 **(integer) 0**

## Redis keys 命令

下表给出了与 Redis 键相关的基本命令：

|  |  |
| --- | --- |
| **序号** | **命令及描述** |
| 1 | [DEL key](http://www.runoob.com/redis/keys-del.html) 该命令用于在 key 存在是删除 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 的过期时间亿以毫秒计。 |
| 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。 |

# Redis 字符串(String)

Redis 字符串数据类型的相关命令用于管理 redis 字符串值，基本语法如下：

### **语法**

redis 127.0.0.1:6379> COMMAND KEY\_NAME

### **实例**

redis 127.0.0.1:6379> SET w3ckey redis OK redis 127.0.0.1:6379> GET w3ckey "redis"

在以上实例中我们使用了 **SET** 和 **GET** 命令，键为 w3ckey。

## Redis 字符串命令

下表列出了常用的 redis 字符串命令：

|  |  |
| --- | --- |
| **序号** | **命令及描述** |
| 1 | [SET key value](http://www.runoob.com/redis/strings-set.html)  设置指定 key 的值 |
| 2 | \*[GET key](http://www.runoob.com/redis/strings-get.html)  获取指定 key 的值。 |
| 3 | \*[GETRANGE key start end](http://www.runoob.com/redis/strings-getrange.html)  返回 key 中字符串值的子字符 |
| 4 | \*[GETSET key value](http://www.runoob.com/redis/strings-getset.html) 将给定 key 的值设为 value ，并返回 key 的旧值(old value)。 |
| 5 | [GETBIT key offset](http://www.runoob.com/redis/strings-getbit.html) 对 key 所储存的字符串值，获取指定偏移量上的位(bit)。 |
| 6 | [MGET key1 [key2..]](http://www.runoob.com/redis/strings-mget.html) 获取所有(一个或多个)给定 key 的值。 |
| 7 | [SETBIT key offset value](http://www.runoob.com/redis/strings-setbit.html) 对 key 所储存的字符串值，设置或清除指定偏移量上的位(bit)。 |
| 8 | \*[SETEX key seconds value](http://www.runoob.com/redis/strings-setex.html) 将值 value 关联到 key ，并将 key 的过期时间设为 seconds (以秒为单位)。  这个是 set name 10 shuaige 意思是 10秒内 name的值是 shuaige 10秒过后就会释放掉。 |
| 9 | \*[SETNX key value](http://www.runoob.com/redis/strings-setnx.html) 只有在 key 不存在时设置 key 的值。Nx not exist |
| 10 | \*[SETRANGE key offset value](http://www.runoob.com/redis/strings-setrange.html) 用 value 参数覆写给定 key 所储存的字符串值，从偏移量 offset（下标） 开始。  Setrange name 2 liming  原值 name:wangwu 改完 waliming |
| 11 | [STRLEN key](http://www.runoob.com/redis/strings-strlen.html) 返回 key 所储存的字符串值的长度。 |
| 12 | [MSET key value [key value ...]](http://www.runoob.com/redis/strings-mset.html) 同时设置一个或多个 key-value 对。 |
| 13 | [MSETNX key value [key value ...]](http://www.runoob.com/redis/strings-msetnx.html)  同时设置一个或多个 key-value 对，当且仅当所有给定 key 都不存在。 |
| 14 | [PSETEX key milliseconds value](http://www.runoob.com/redis/strings-psetex.html) 这个命令和 SETEX 命令相似，但它以毫秒为单位设置 key 的生存时间，而不是像 SETEX 命令那样，以秒为单位。 |
| 15 | \*[INCR key](http://www.runoob.com/redis/strings-incr.html) 将 key 中储存的数字值增一。 |
| 16 | \*[INCRBY key increment](http://www.runoob.com/redis/strings-incrby.html) 将 key 所储存的值加上给定的增量值（increment） 。自增 |
| 17 | [INCRBYFLOAT key increment](http://www.runoob.com/redis/strings-incrbyfloat.html) 将 key 所储存的值加上给定的浮点增量值（increment） 。 |
| 18 | \*[DECR key](http://www.runoob.com/redis/strings-decr.html) 将 key 中储存的数字值减一。 |
| 19 | [DECRBY key decrement](http://www.runoob.com/redis/strings-decrby.html) key 所储存的值减去给定的减量值（decrement） 。 |
| 20 | \*[APPEND key value](http://www.runoob.com/redis/strings-append.html) 如果 key 已经存在并且是一个字符串， APPEND 命令将 value 追加到 key 原来的值的末尾。 |

更多命令请参考：[http://redis.readthedocs.org/en/latest/index.html](http://redis.readthedocs.org/en/latest/index.html" \t "http://www.runoob.com/redis/_blank)

# 2.哈希



# Redis 哈希(Hash)

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。

Redis 中每个 hash 可以存储 232 - 1 键值对（40多亿）。

### **实例**

redis 127.0.0.1:6379> HMSET w3ckey name "redis tutorial" description "redis basic commands for caching" likes 20 visitors 23000

OK

redis 127.0.0.1:6379> HGETALL w3ckey

1) "name"

2) "redis tutorial"

3) "description"

4) "redis basic commands for caching"

5) "likes"

6) "20"

7) "visitors"

8) "23000"

在以上实例中，我们设置了 redis 的一些描述信息(name, description, likes, visitors) 到哈希表的 w3ckey 中。

## Redis hash 命令

下表列出了 redis hash 基本的相关命令：

|  |  |
| --- | --- |
| **序号** | **命令及描述** |
| 1 | [HDEL key field2 [field2]](http://www.runoob.com/redis/hashes-hdel.html)  删除一个或多个哈希表字段 field表示字段的意思。 |
| 2 | [HEXISTS key field](http://www.runoob.com/redis/hashes-hexists.html)  查看哈希表 key 中，指定的字段是否存在。 |
| 3 | [HGET key field](http://www.runoob.com/redis/hashes-hget.html)  获取存储在哈希表中指定字段的值。 |
| 4 | [HGETALL key](http://www.runoob.com/redis/hashes-hgetall.html)  获取在哈希表中指定 key 的所有字段和值 |
| 5 | [HINCRBY key field increment](http://www.runoob.com/redis/hashes-hincrby.html)  为哈希表 key 中的指定字段的整数值加上增量 increment 。 |
| 6 | [HINCRBYFLOAT key field increment](http://www.runoob.com/redis/hashes-hincrbyfloat.html)  为哈希表 key 中的指定字段的浮点数值加上增量 increment 。 |
| 7 | [HKEYS key](http://www.runoob.com/redis/hashes-hkeys.html)  获取所有哈希表中的字段 |
| 8 | [HLEN key](http://www.runoob.com/redis/hashes-hlen.html)  获取哈希表中字段的数量 hlen user:001 |
| 9 | [HMGET key field1 [field2]](http://www.runoob.com/redis/hashes-hmget.html)  获取所有给定字段的值 |
| 10 | \*[HMSET key field1 value1 [field2 value2 ]](http://www.runoob.com/redis/hashes-hmset.html)  同时将多个 field-value (域-值)对设置到哈希表 key 中。 |
| 11 | \*[HSET key field value](http://www.runoob.com/redis/hashes-hset.html)  将哈希表 key 中的字段 field 的值设为 value 。 |
| 12 | \*[HSETNX key field value](http://www.runoob.com/redis/hashes-hsetnx.html)  只有在字段 field 不存在时，设置哈希表字段的值。 |
| 13 | [HVALS key](http://www.runoob.com/redis/hashes-hvals.html)  获取哈希表中所有值 |
| 14 | HSCAN key cursor [MATCH pattern] [COUNT count]  迭代哈希表中的键值对。 |

Hset user:001 name baotong设置 001是id

# 3.Redis 列表(List)

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导列表的头部（左边）或者尾部（右边）

一个列表最多可以包含 232 - 1 个元素 (4294967295, 每个列表超过40亿个元素)。

### **实例**

redis 127.0.0.1:6379> LPUSH w3ckey redis

(integer) 1

redis 127.0.0.1:6379> LPUSH w3ckey mongodb

(integer) 2

redis 127.0.0.1:6379> LPUSH w3ckey mysql

(integer) 3

redis 127.0.0.1:6379> LRANGE w3ckey 0 10

1) "mysql"

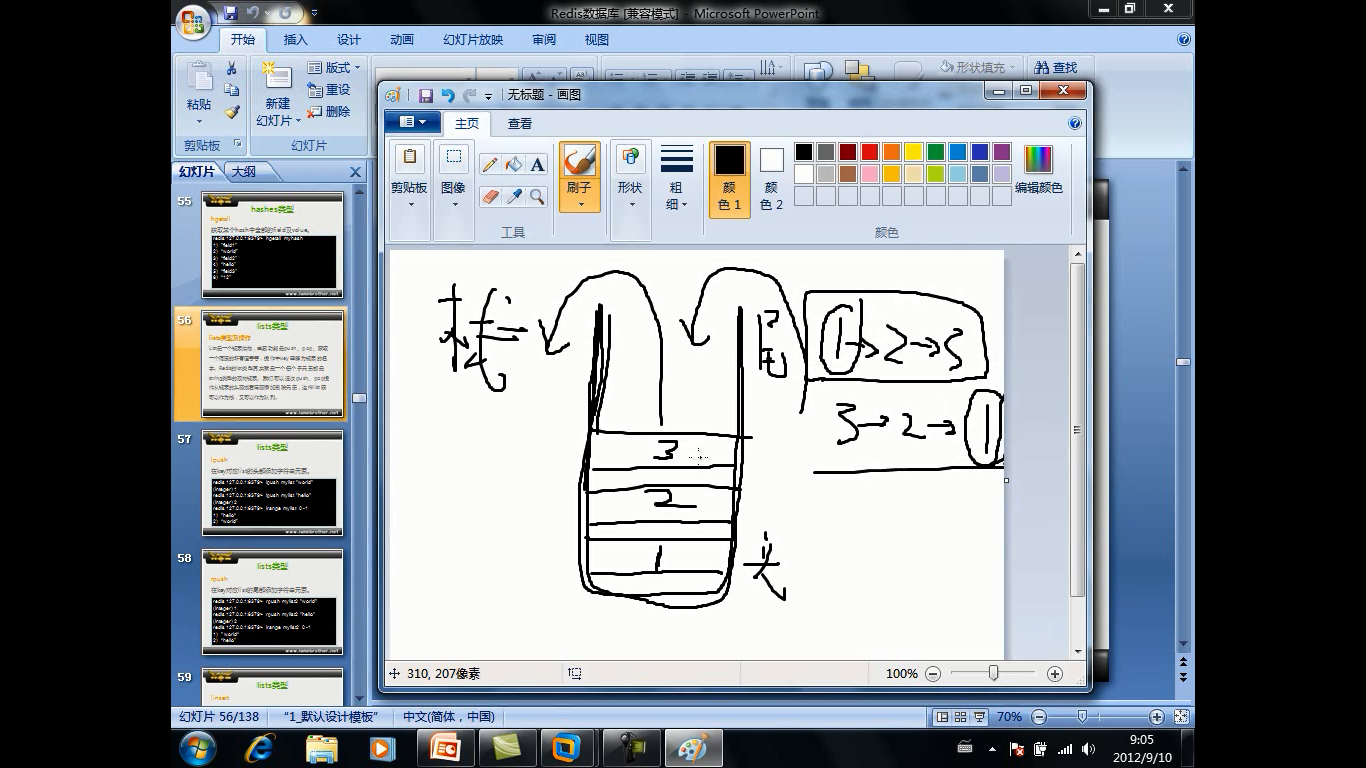
2) "mongodb"

3) "redis"

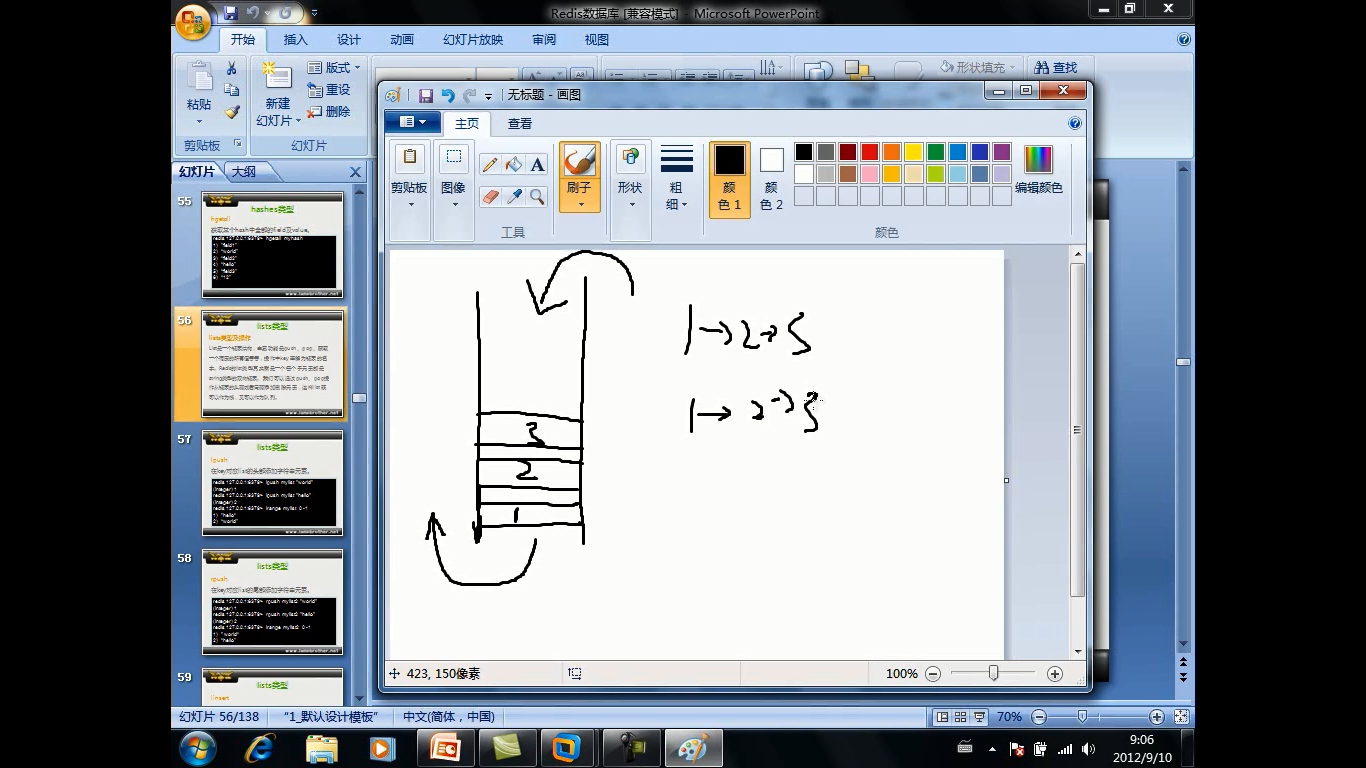
在以上实例中我们使用了 **LPUSH** 将三个值插入了名为 w3ckey 的列表当中。

### **Redis 列表命令**





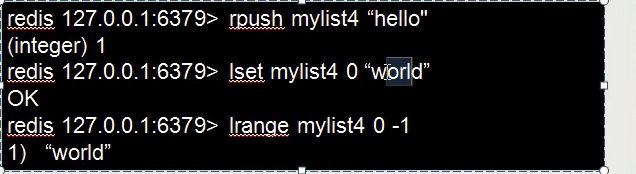
\*\*\*\*栈 先进后厨



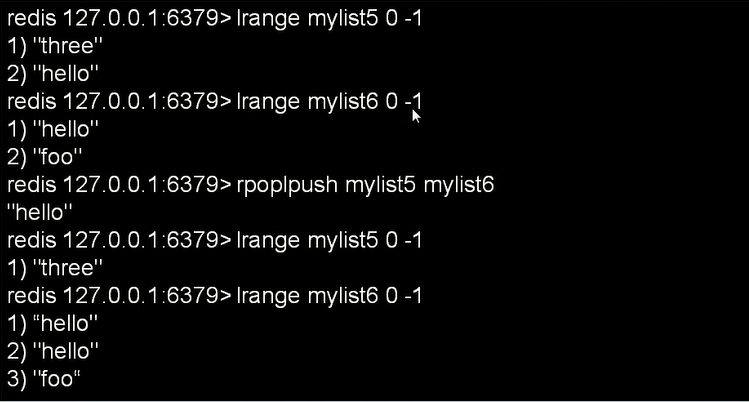
\*\*\*\*队列先进先出。

下表列出了列表相关的基本命令：

|  |  |
| --- | --- |
| **序号** | **命令及描述** |
| 1 | [BLPOP key1 [key2 ] timeout](http://www.runoob.com/redis/lists-blpop.html)  移出并获取列表的第一个元素， 如果列表没有元素会阻塞列表直到等待超时或发现可弹出元素为止。 |
| 2 | [BRPOP key1 [key2 ] timeout](http://www.runoob.com/redis/lists-brpop.html)  移出并获取列表的最后一个元素， 如果列表没有元素会阻塞列表直到等待超时或发现可弹出元素为止。 |
| 3 | [BRPOPLPUSH source destination timeout](http://www.runoob.com/redis/lists-brpoplpush.html)  从列表中弹出一个值，将弹出的元素插入到另外一个列表中并返回它； 如果列表没有元素会阻塞列表直到等待超时或发现可弹出元素为止。 |
| 4 | [LINDEX key index](http://www.runoob.com/redis/lists-lindex.html)  通过索引获取列表中的元素 |
| 5 | [LINSERT key BEFORE|AFTER pivot value](http://www.runoob.com/redis/lists-linsert.html)  在列表的元素前或者后插入元素 头的方向为前 |
| 6 | [LLEN key](http://www.runoob.com/redis/lists-llen.html)  获取列表长度 |
| 7 | [LPOP key](http://www.runoob.com/redis/lists-lpop.html)  移出并获取列表的第一个元素 |
| 8 | [LPUSH key value1 [value2]](http://www.runoob.com/redis/lists-lpush.html)  将一个或多个值从头部插入到列表 |
| 9 | [LPUSHX key value](http://www.runoob.com/redis/lists-lpushx.html)  将一个或多个值插入到已存在的列表头部 |
| 10 | [LRANGE key start stop](http://www.runoob.com/redis/lists-lrange.html)  获取列表指定范围内的元素lrange mytest 0 -1（从头部取出元素到结尾） |
| 11 | [LREM key count value](http://www.runoob.com/redis/lists-lrem.html)  移除列表元素 |
| 12 | [LSET key index value](http://www.runoob.com/redis/lists-lset.html)  通过索引设置列表元素的值 |
| 13 | [LTRIM key start stop](http://www.runoob.com/redis/lists-ltrim.html)  对一个列表进行修剪(trim)，就是说，让列表只保留指定区间内的元素，不在指定区间之内的元素都将被删除。 |
| 14 | [RPOP key](http://www.runoob.com/redis/lists-rpop.html)  移除并获取列表最后一个元素 |
| 15 | [RPOPLPUSH source destination](http://www.runoob.com/redis/lists-rpoplpush.html)  移除列表的最后一个元素，并将该元素添加到另一个列表并返回 |
| 16 | [RPUSH key value1 [value2]](http://www.runoob.com/redis/lists-rpush.html)  在列表尾部添加一个或多个值 |
| 17 | [RPUSHX key value](http://www.runoob.com/redis/lists-rpushx.html)  为已存在的列表添加值 |

Lset



rpoplpush 从左表尾部出来添加到右表头部。