

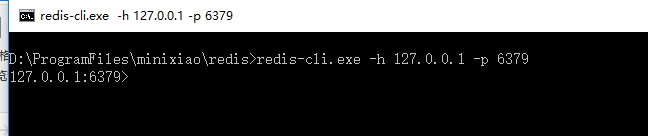
# 1.安装（以下命令都可以没有exe）

## D:\ProgramFiles\minixiao\redis> redis-server.exe redis.windows.conf



# 2.开始使用，重新打开cmd窗口

## D:\ProgramFiles\minixiao\redis>redis-cli.exe -h 127.0.0.1 -p 6379 -a 123456



# 3. 启动 Redis

$ redis-server

### 查看 redis 是否启动？

$ redis-cli

以上命令将打开以下终端：

redis 127.0.0.1:6379>

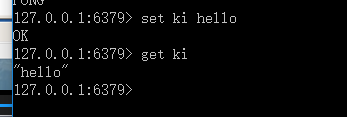
127.0.0.1 是本机 IP ，6379 是 redis 服务端口。现在我们输入 PING 命令。

redis 127.0.0.1:6379> ping

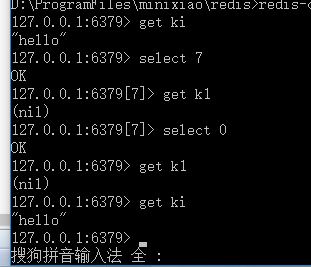
PONG

以上说明我们已经成功安装了redis。

# 4.helloworld



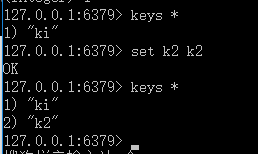
# 5.redis一共，16个库，现在切换到第8个库，查找ki，发现数据不在里面，说明我们的默认数据一开始是在第一个库中的



# 6.简单命令

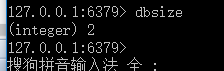
## 1.查看当前库的所有数据

### keys \*



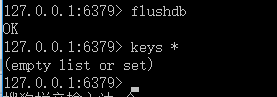
## 2.查看几个数据

## dbsize



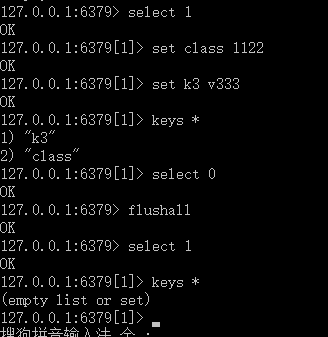
## 3.将数据清除

### flushdb



## 4.清除所有库的信息（移动到第2仓库，建立数据以后返回1号库进行命令，结果发现数据全部没有了）

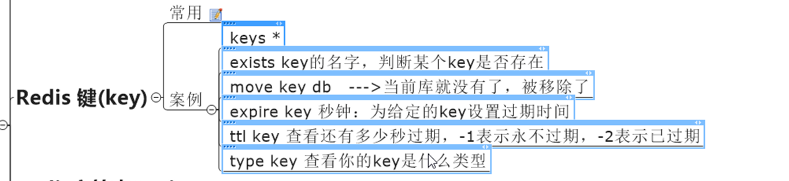
### flushall



# 5.redis五大数据类型

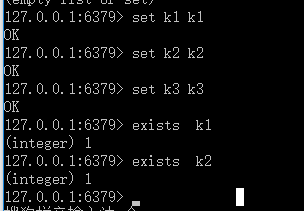


# 6.key



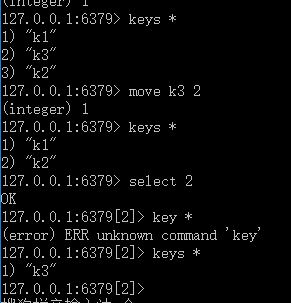
## 1.看看是否存在k1

### exists k1



## 2.将数据移动到其他库中(将数据移动到2号库)

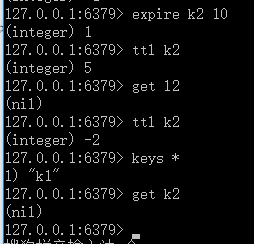
### move k3 2



## 3.设置过期时间，查看过期时间 (k2 设置10秒过期)

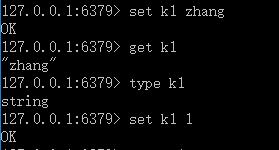
### expire k2 10

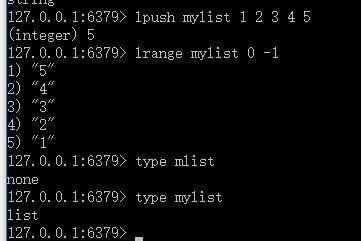
### ttl k2 查看过期时间



## 4.查看数据类型

### type k1





# 7.Stirng

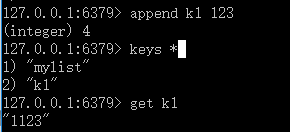
## 1.删除数据

### del mylist



## 2.追加数据

### append k1 123



## 3.查看数据的长度

### strlen k1



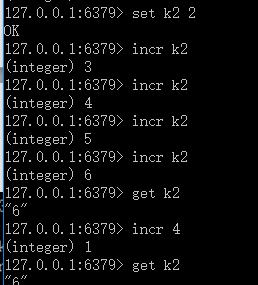
## 4.数据递增递减（只能是数字，不能是字符串）

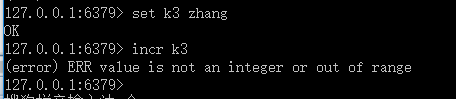
### incr k2 (k2 增加1)

### incrby k2 4 (k2增加4)

### decr k2 (k2减少1)

### decrby k2 4 （k2减少4）

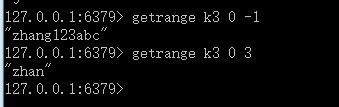




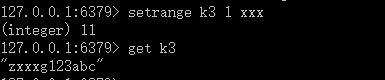
## 5.截取或者添加一段字符串

### getrange k3 0 -1 显示全部

### getrange k3 0 3 (包头不包尾)

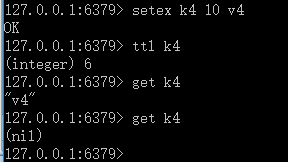


### setrange k3 1 xxx在序列1插入字符串xxx



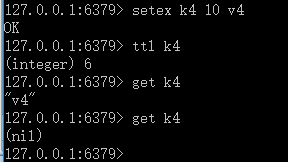
## 6.同时设置过期时间和值

### setex k4 10 v4



## 7.检查是否有值，不替换。并且添加数据

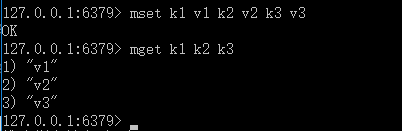
### setex k11 v11



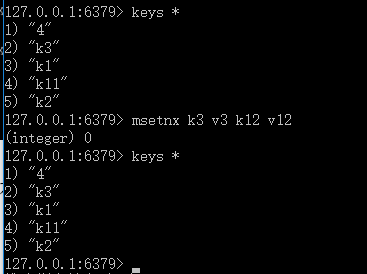
## 8.一次添加多条数据

### mset

### mget



## msetnx 如果里面有存在数据，则设置不能成功

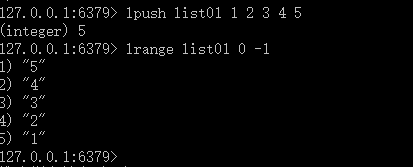


# 8.List



## 1.添加数据

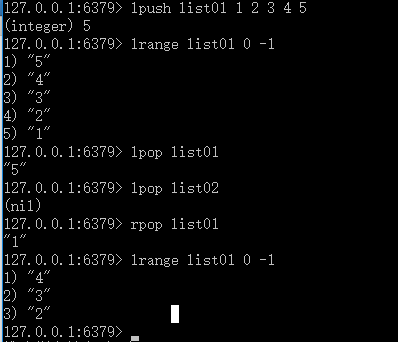
### lpush mylist0 1 2 4 5 6



## 2.出栈

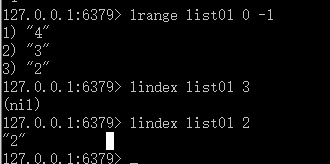
### lpop从栈顶出

### rpop从栈底出



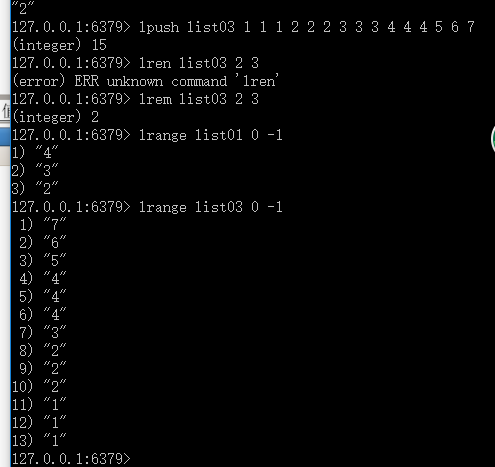
## 3.根据索引查找数据

### lindex list01 3



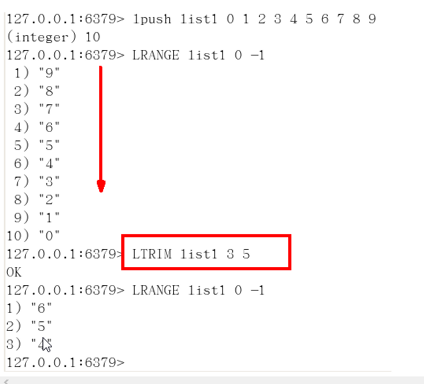
## 4.连续删除 2个3

### lrem list 2 3



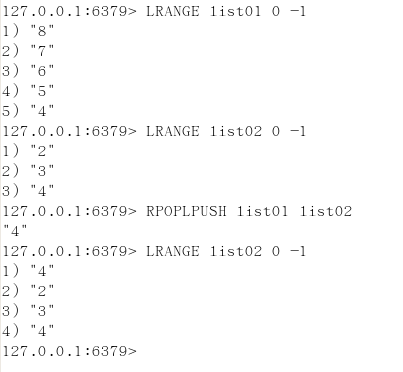
## 5.截取并重新赋值

### ltrim list1 3 5



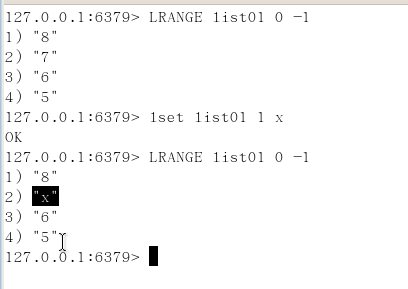
## 6.从栈底出一个到另一个栈的栈顶

### rpoplpush



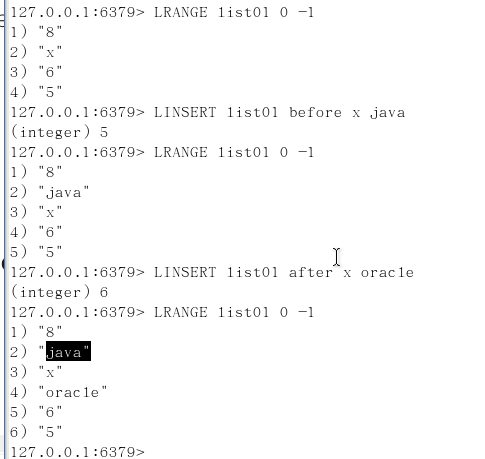
## 7.替换

### lset



## 8.在给定字符之前或者之后插入数据

### linsert before 或者after



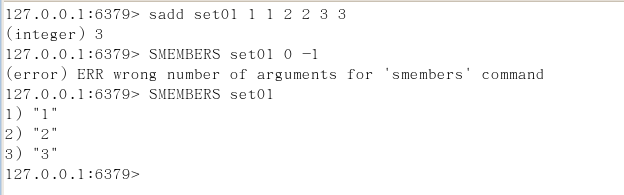
# 9.set 基本上和list差不多，只是重复的问题



## 1.添加数据,查看数据

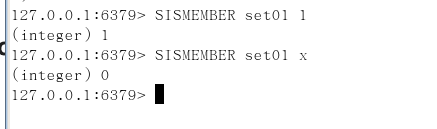
### sadd

### smembers



## 2.检测数据是否存在

### sismember set01 1 看看1 是否存在于set01



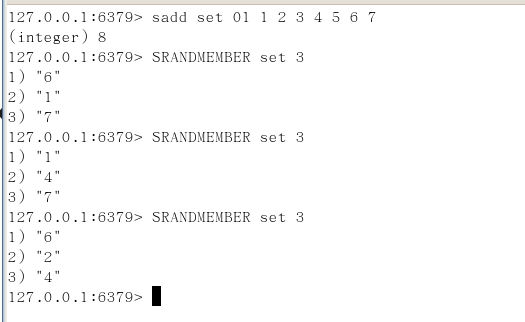
## 3.获取集合元素个数

### scard



## 4.随机出几个数

### srandmember



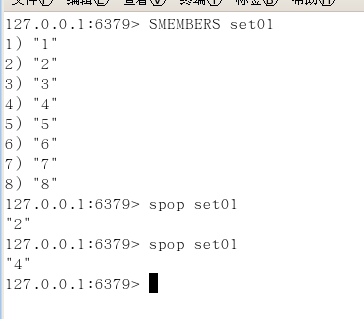
## 5.删除元素

### srem



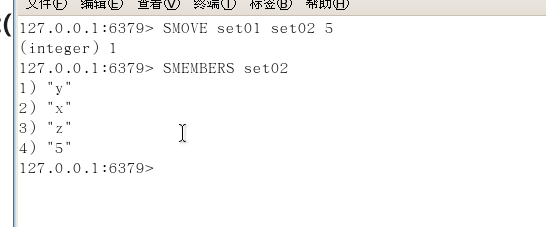
## 6.随机出栈

### spop



## 7.移动值到另一个集合中

### smove

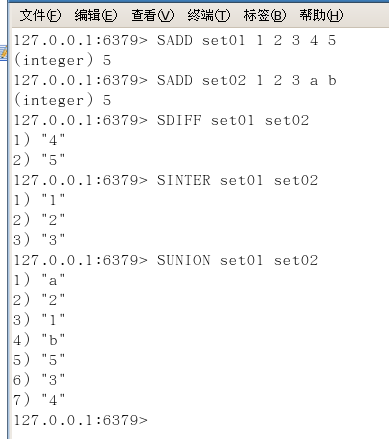


## 8.差集，交集，并集

### sdiff

### sinter

### sunin



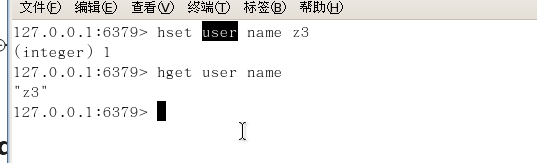
# 10.hash



## 1.设置值和取值

### hset

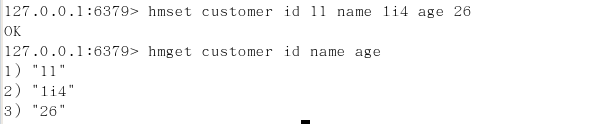
### hget



## 2.一次设置多个值

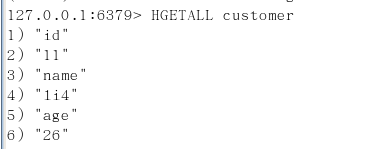
### hmset

### hmget



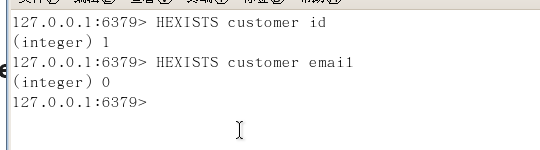
## 3.一次输出多个值

### hgetall



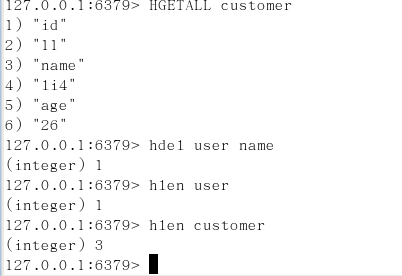
## 4.判断是否存在key

### hexists



## 5.删除相关字段

### hdel



## 6.取得hash长度

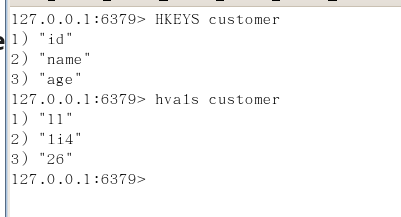
### hlen



## 7.单独得到key或者value

### hkeys

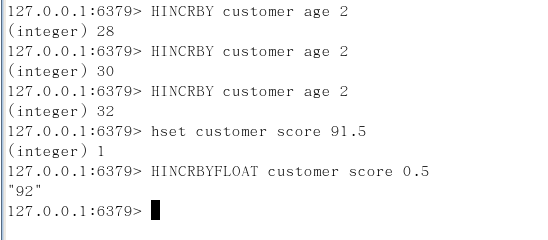
### hvals



## 8.整数增加

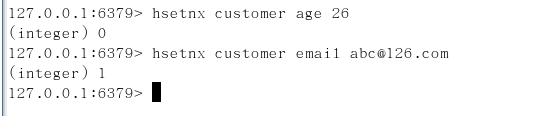
### hincrby

### hincrbyfloat

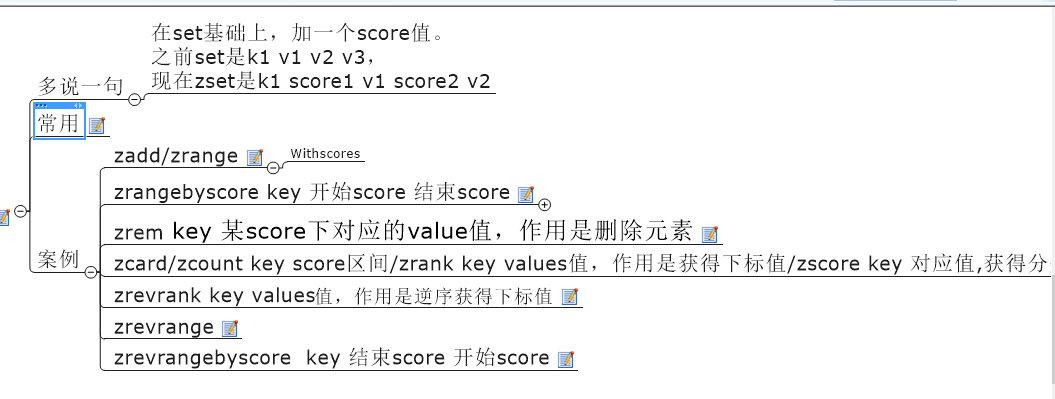


## 9.加值并进行判断

### hsetnx

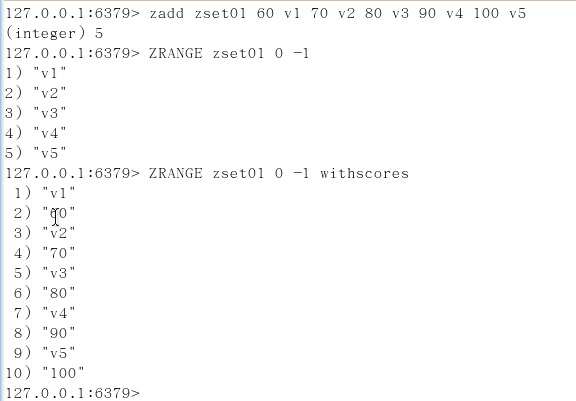


# 11.zset



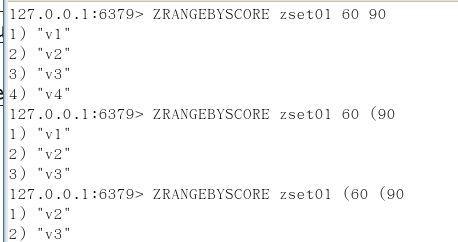
## 1.加值和取值

### zdd(withscores)



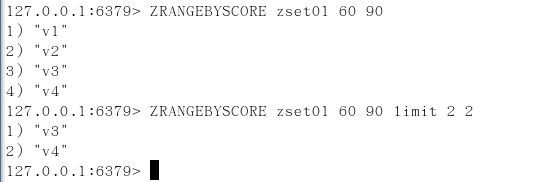
## 2.根据分数区间取值

### zrangebyscore



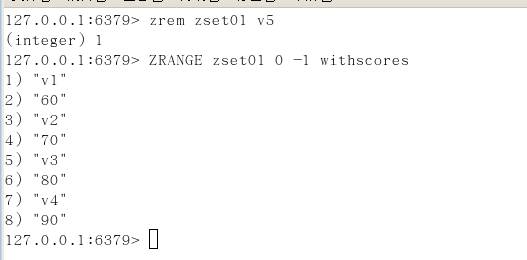
## 3.从结果集截取

### limit



## 4.删除元素

### zrem

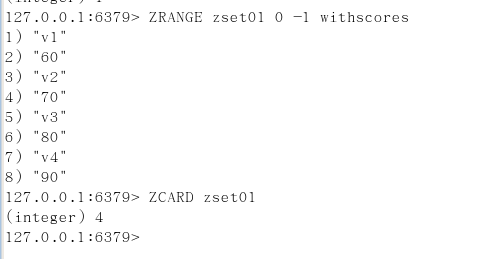


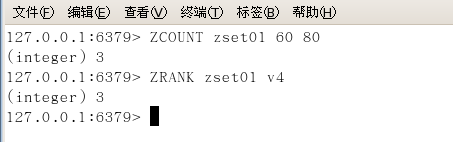
## 5.统计个数，取得值，和下标

### zcard

### zcount

### zrank

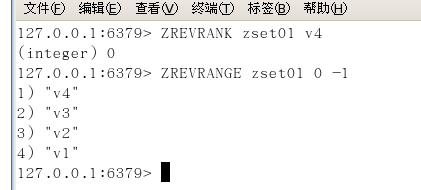




## 6.逆序取值和排序

### zrevrank

### zrevrange



## 7.反转排序输出

### zrevrangebyscore

