# sort排序命令基本使用

在Redis中，存在sort命令，该命令用于获取键的值，并可以根据需要对其进行排序。命令如下：

**sort key [by pattern] [LIMIT offset count] [GET pattern [get pattern……]**

**[asc|desc] [alpha] [store destination]**

key表示键；

1）使用sort命令只能对列表键、集合键、有序集合键进行排序；

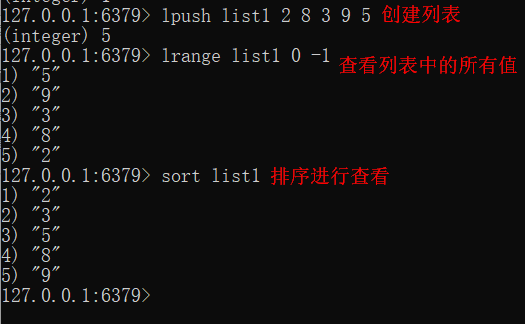
2）可以看出sort命令有许多的可选参数，如果没有设置可选参数，而只是用sort key命令对键进行排序，则默认为升序排序；

3）如果要排序的键对应的值不是能表示为数字，则使用sort key命令进行排序会报错。

## 对列表进行排序

1. 创建列表：lpush list1 2 8 3 9 5
2. 查看该列表所有值：lrange list1 0 -1
3. 使用sort获取值并排序：sort list1

效果如下：



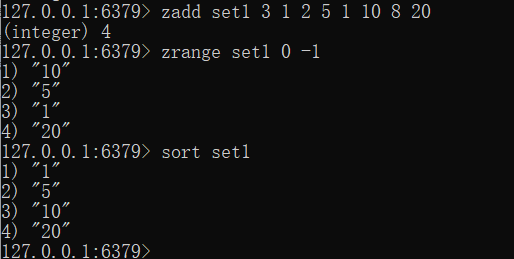
可以看出使用sort获取列表键值时对其进行了排序。

## 对有序集合进行排序

使用sort命令对有序集合进行排序时，会忽略掉有序集合中元素对应的小数，而只根据元素本身的值进行排序。

1. 创建有序集合：zadd set1 3 1 2 5 1 10 8 20（包括小数和元素）
2. 查看有序集合元素：zrange set1 0 -1
3. 使用sort命令：sort set1

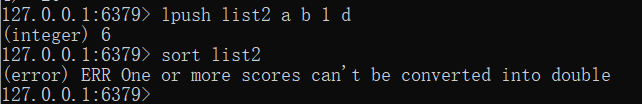
效果如下：



有序集合本身就是按照元素对应的小数进行了排序。但从执行sort命令来看，会根据元素值进行排序。

## 对字符串值进行排序

使用sort命令对列表、集合、有序集合排序时，如果存在某个或多个值不能表示为数字，则会返回一个错误。例如：



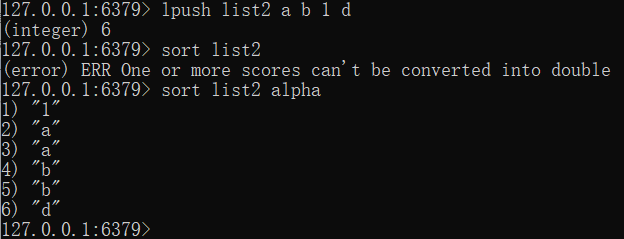
因为使用sort命令排序时会尝试将元素值转换为double值进行比较。那么如何对字符串类型的值进行排序呢。命令如下：

**sort key alpha**

key表示键；

只需对sort命令追加一个alpha参数即可，表示根据元素值对应的ASCII码值进行排序。

如下：

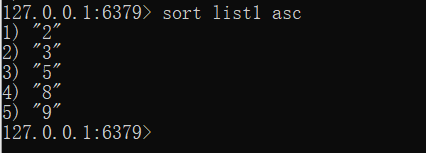


# sort命令的可选参数

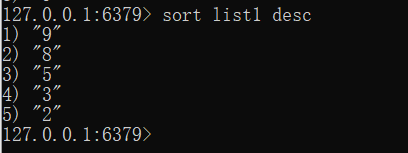
## asc与desc

使用sort获取排序时，默认是按照从小到大（升序）进行排序的。可以通过在sort命令后面追加可选参数asc或desc来改变排序规则。

1）asc表示升序排序，也是默认的排序规则。如下：



2）desc表示降序排序。如下：



## limit

limit可选参数用于限制查询结果。命令如下：

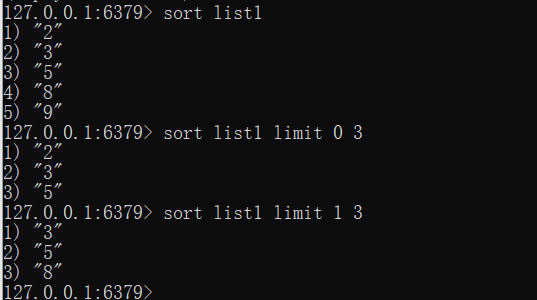
**sort key limit offset count**

key表示要排序的键；

offset表示偏移量，也就是要从哪开始将值进行返回；

count表示要查询的数量，也就是返回值的数量；

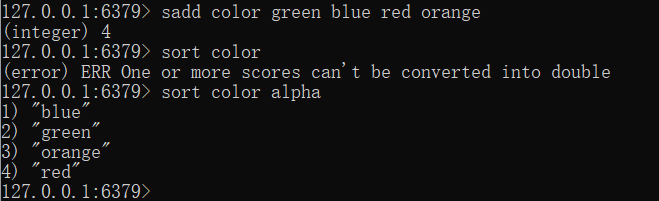
如下：



## by

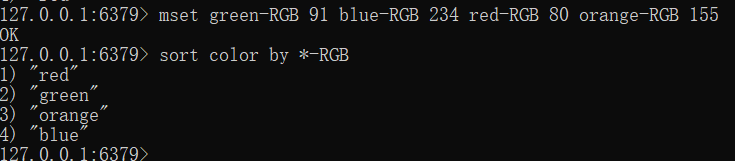
1. 使用sort命令进行排序，会按照元素本身的值进行排序，元素本身的值决定了该元素在排序后所处的位置。如果我们想要按照其他键来排序，则可以通过by参数来实现。
2. 执行sort命令可以指定某些字符串键或者某个哈希键所具有的某些域来作为排序依据，对这个键进行排序。

比如：



可以看出现在是根据元素值来进行排序的。

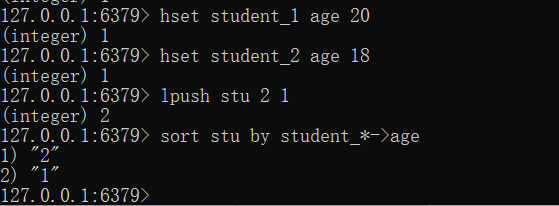
而我们现在可以定义多个字符串键来自定义排序规则，然后再通过sort命令加上by参数进行排序，如下：



可以看出，通过by参数可以根据字符串类型键对应的值来进行排序，但到底还是通过ASCII码来进行排序的。

也可以看出，这些字符串类型键对应的值是数字类型的，如果其中有一个键对应的值是字符串类型的，则sort命令还需要追加alpha参数。

当然也可以根据哈希表域的值来排序。如下：



按照道理本来排序后应该1，2；但由于根据哈希表域的值自定义了排序依据。

## store

使用sort命令可以获取键的值并将它们排序，但并不会保存这些排序结果。如果想把排序结果保存起来，则可以使用store参数。

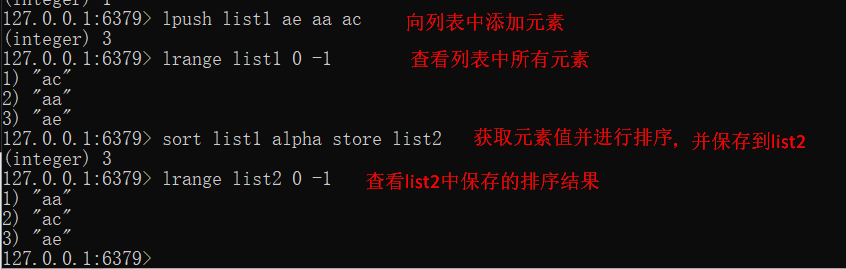
通过store参数可以把排序结果保存到指定的键中，在需要的时候从这个键中取出。命令如下：

**sort key store destination**

key表示键；

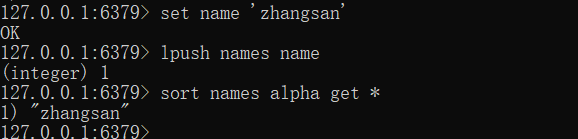
destination表示要保存排序结果的键；

如下：



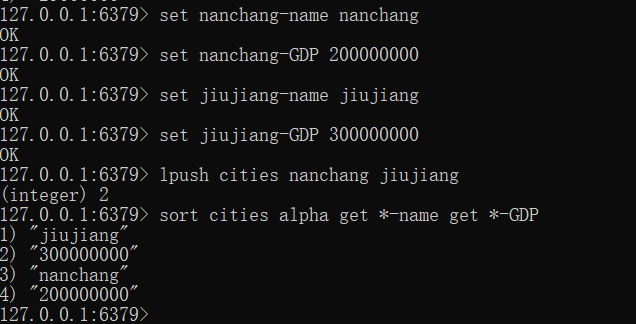
## get

1. sort命令的get参数就是get命令，可以将其理解为在sort命令中嵌套的get命令。
2. 通常情况下，我们通过set命令设置键值对，通过 get命令根据给定的键名来获取对应的值，如get name，表示获取name键对应的值；而在sort命令中，嵌套的get命令并不是通过给定的键名来获取键的值，而是根据sort命令的排序结果，将排序结果列表中的每一个元素值赋给get命令，当做get命令的key参数，也就是键名；然后嵌套的get命令再根据给定的元素值，将其作为键名参数，来获取该键对应的值。（简单理解就是：将sort命令的排序结果列表中的每一个值赋给get命令，get命令将其视为键名，根据键名获取对应的值）如下：

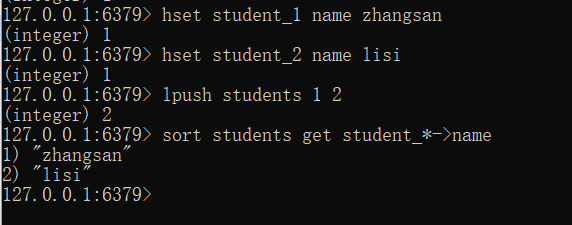


sort names alpha get \* 可以分为两步理解：

1. 执行sort names alpha命令
2. 执行get name命令（因为在执行sort names alpha命令时返回的结果列表中只有name）
3. 一条sort命令可以有多个get参数，这也就表示着一条sort命令中可以嵌套多个get命令。这时，会先执行排在前面的get命令，再执行后面的get命令。它们的键名都是sort命令的排序结果列表中的每一个值。如下：



1. get参数除了可以获取键名对应的值外，还可以获取哈希表中域的值。如下：



sort students get student\_\*->name命令可以拆分为：

（1）sort students：表示获取students列表中的值并进行排序；

（2）get student\_\*->name：将sort students命令返回的已排序列表中每一个值填充在\*中；

# 多参数的执行顺序

sort命令有很多的可选参数，如果执行sort命令时，设置了很多的可选参数，那么这些可选参数的执行顺序又是怎样的优先级呢？如下（由高到低）：

1. alpha
2. asc或desc
3. by
4. limit
5. get
6. store