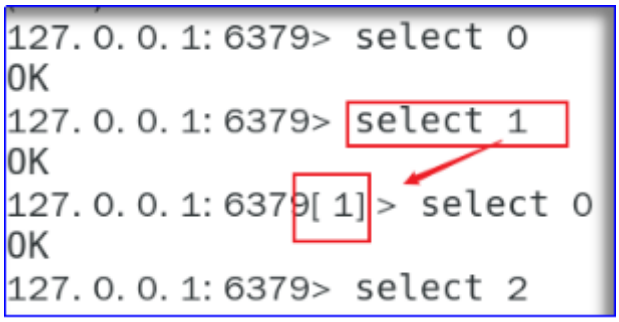
Redis数据库数据类型操作

Redis的数据库

Redis是一个字典结构的存储服务器，而实际上一个Redis实例提供了多个用来存储数据的字典，客户端可以指定将数据存储在哪个字典中。这与我们熟知的在一个关系数据库实例中可以创建多个数据库类似，所以可以将其中的每个字典都理解成一个独立的数据库。

每个数据库对外都是一个从0开始的递增数字命名，Redis默认支持16个数据库（可以通过配置文件支持更多，无上限），可以通过配置databases来修改这一数字。客户端与Redis建立连接后会自动选择0号数据库，不过可以随时使用SELECT命令更换数据库，如要选择1号数据库：



然而这些以数字命名的数据库又与我们理解的数据库有所区别。首先Redis不支持自定义数据库的名字，每个数据库都以编号命名，开发者必须自己记录哪些数据库存储了哪些数据。另外Redis也不支持为每个数据库设置不同的访问密码，所以一个客户端要么可以访问全部数据库，要么连一个数据库也没有权限访问。最重要的一点是多个数据库之间并不是完全隔离的，比如FLUSHALL命令可以清空一个Redis实例中所有数据库中的数据。综上所述，这些数据库更像是一种命名空间，而不适宜存储不同应用程序的数据。比如可以使用0号数据库存储某个应用生产环境中的数据，使用1号数据库存储测试环境中的数据，但不适宜使用0号数据库存储A应用的数据而使用1号数据库B应用的数据，不同的应用应该使用不同的Redis实例存储数据。由于Redis非常轻量级，一个空Redis实例占用的内存只有1M左右，所以不用担心多个Redis实例会额外占用很多内存。

Redis 一个实例默认有16个数据库

删除数据库中数据

flushdb 删除当前数据库中数据

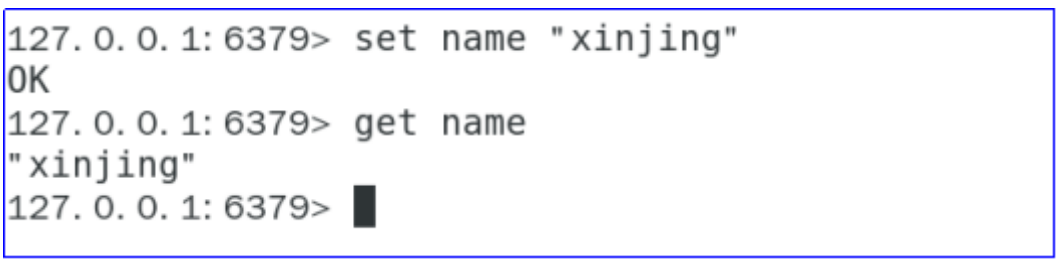
flushall 删除所有数据库中数据

Redis的数据类型及操作

字符串类型

字符串类型的reids操作命令

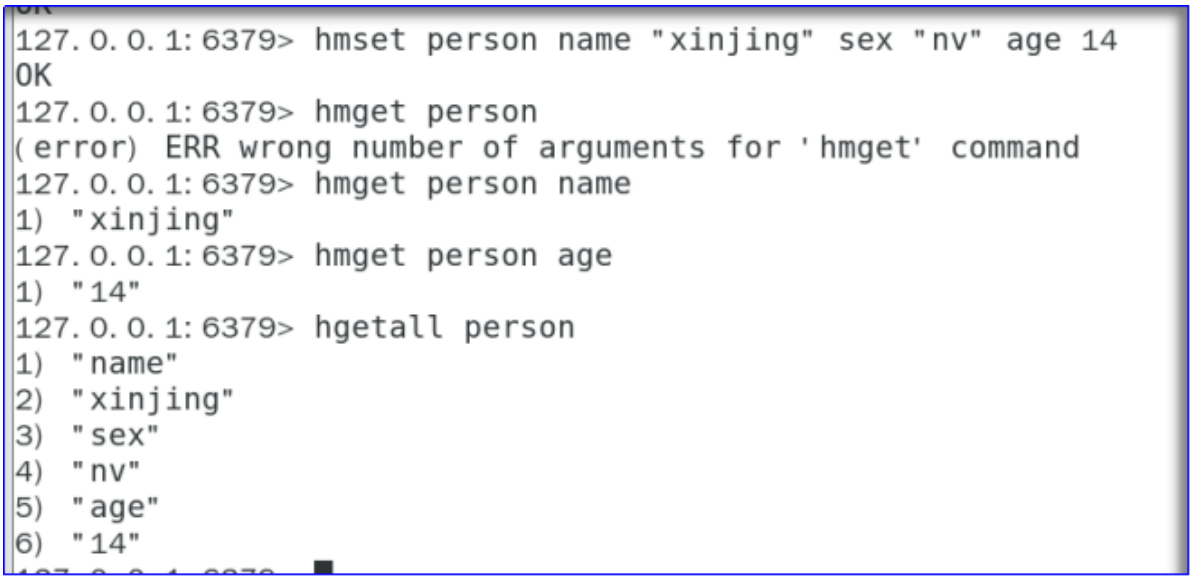
设置键值对，获得键值



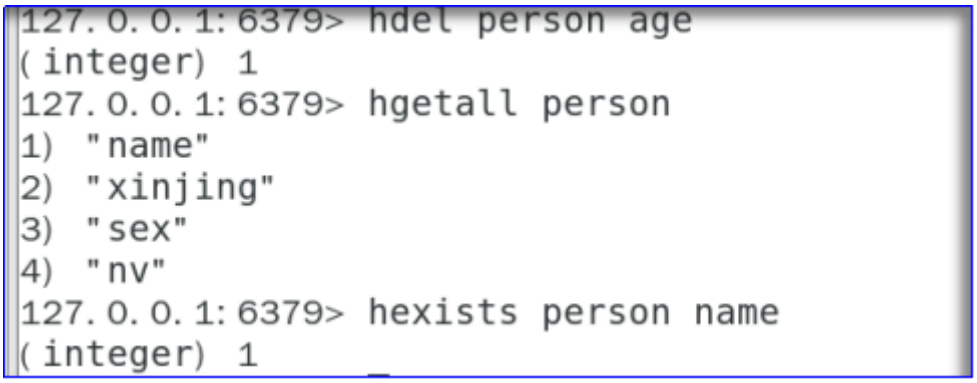
Hash类型

Redis hash 是一个 string 类型的 field（字段） 和 value（值） 的映射表，hash 特别适合用于存储对象。Redis 中每个 hash 可以存储 2^32 - 1 键值对（40多亿）。

键值对的设置与获取



键值对的删除与判定存在



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到列表的头部（左边）或者尾部（右边）一个列表最多可以包含 2^32 - 1 个元素 (4294967295, 每个列表超过40亿个元素)。

栈式操作命令 Lpush lpop lrange lindex llen 头部进出

```
127.0.0.1:6379> lpush country china
(integer) 1
127.0.0.1:6379> lpush country india
(integer) 2
127.0.0.1:6379> keys *
1) "country"
2) "person"
127.0.0.1:6379> get country
(error) WRONGTYPE Operation against a key holding the wrong kind of value
127.0.0.1:6379> lrange country 0 10
1) "india"
2) "china"
127.0.0.1:6379> lpush country USA
(integer) 3
127.0.0.1:6379> lrange country 0 10
1) "USA"
2) "india"
3) "china"
127.0.0.1:6379> lindex country 1
"india"
127.0.0.1:6379> lindex country 0
"USA"
127.0.0.1:6379> lpop country
"USA"
```

队列式操作命令 rpush rpop 尾部进出

```
127.0.0.1:6379> rpush subject math
(integer) 1
127.0.0.1:6379> rpush subject english computer
(integer) 3
127.0.0.1:6379> rlen subject
(error) ERR unknown command `rlen`, with args beginning with: `subject`,
127.0.0.1:6379> llen subject
(integer) 3
127.0.0.1:6379> rrange subject
(error) ERR unknown command `rrange`, with args beginning with: `subject`,
127.0.0.1:6379> lrange sbuject
(error) ERR wrong number of arguments for 'lrange' command
127.0.0.1:6379> lrange sbuject 0 10
(empty array)
127.0.0.1:6379> lrange subject 0 10
1) "math"
2) "english"
3) "computer"
127.0.0.1:6379> rpop subject
"computer"
127.0.0.1:6379> lrange subject  0 10
1) "math"
2) "english"
```

集合set

Redis 的 Set 是 String 类型的无序集合。集合成员是唯一的，这就意味着集合中不能出现重复的数据。Redis 中集合是通过哈希表实现的，所以添加，删除，查找的复杂度都是 O(1)。

集合中最大的成员数为 2^32 - 1 (4294967295，每个集合可存储40多亿个成员)。

集合增加元素

```
127.0.0.1:6379> sadd nameset "xinjing"
(integer) 1
127.0.0.1:6379> sadd nameset "wangwu"
(integer) 1
127.0.0.1:6379> sadd nameset "lisi"
(integer) 1
127.0.0.1:6379> sadd nameset "lisi"
(integer) 0
```

显示集合所有成员和判定集合成员

```
127.0.0.1:6379> smembers nameset
1) "xinjing"
2) "lisi"
3) "wangwu"
127.0.0.1:6379> sismember nameset xinjing
(integer) 1
```

获得集合中元素数量
scard nameset
随机获得集合的一个元素
spop nameset

Redis有序集合（sorted set）

Redis 有序集合和集合一样也是 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。
有序集合的成员是唯一的,但分数(score)却可以重复。
集合是通过哈希表实现的，所以添加，删除，查找的复杂度都是 O(1)。 集合中最大的成员数为 2^32 - 1 (4294967295, 每个集合可存储40多亿个成员)。
Zadd添加操作

```
127.0.0.1:6379> zadd namezset 1 xinjing
(integer) 1
127.0.0.1:6379> zadd namezset 2 zhangsan
(integer) 1
127.0.0.1:6379> zadd namezset 3 zhangsan
(integer) 0
127.0.0.1:6379> zadd namezset 3 wangwu
(integer) 1
127.0.0.1:6379> zrange namezset 0 10
1) "xinjing"
2) "wangwu"
3) "zhangsan"
```