**一、key pattern 查询相应的key**

　　（1）redis允许模糊查询key　　有3个通配符  \*、?、[]

　　（2）randomkey：返回随机key

　　（3）type key：返回key存储的类型

　　（4）exists key：判断某个key是否存在

　　（5）del key：删除key

　　（6）rename key newkey：改名

　　（7）renamenx key newkey：如果newkey不存在则修改成功

　　（8）move key 1：将key移动到1数据库

　　（9）ttl key：查询key的生命周期（秒）

　　（10）expire key 整数值：设置key的生命周期以秒为单位

　　（11）pexpire key 整数值：设置key的生命周期以毫秒为单位

　　（12）pttl key：查询key 的生命周期（毫秒）

　　（13）perisist key：把指定key设置为永久有效

**二、字符串类型的操作**

　　（1）set key value [ex 秒数] [px 毫秒数] [nx/xx]

　　　　　　如果ex和px同时写，则以后面的有效期为准

　　　　　　nx：如果key不存在则建立

　　　　　　xx：如果key存在则修改其值

　　（2）get key：取值

　　（3）mset key1 value1 key2 value2 一次设置多个值

　　（4）mget key1 key2 ：一次获取多个值

　　（5）setrange key offset value：把字符串的offset偏移字节改成value

　　　　　　　　　　　　　　　　　　如果偏移量 > 字符串长度，该字符自动补0x00

　　（6）append key value ：把value追加到key 的原值上

　　（7）getrange key start stop：获取字符串中[start, stop]范围的值

　　　　　　　　　　　　　　　　　　对于字符串的下标，左数从0开始，右数从-1开始

　　　　　　　　　　　　　　　　　　注意：当start>length，则返回空字符串

　　　　　　　　　　　　　　　　　　　　　当stop>=length，则截取至字符串尾

　　　　　　　　　　　　　　　　　　　　　如果start所处位置在stop右边，则返回空字符串

　　（8）getset key nrevalue：获取并返回旧值，在设置新值

　　（9）incr key：自增，返回新值，如果incr一个不是int的value则返回错误，incr一个不存在的key，则设置key为1

　　（10）incrby key 2：跳2自增

　　（11）incrbyfloat by 0.7： 自增浮点数

　　（12）setbit key offset value：设置offset对应二进制上的值，返回该位上的旧值

　　　　　　　　　　　　　　　　　注意：如果offset过大，则会在中间填充0

　　　　　　　　　　　　　　　　　　　　offset最大到多少

　　　　　　　　　　　　　　　　　　　　　2^32-1，即可推出最大的字符串为512M

　　（13）bitop operation destkey key1 [key2..]    对key1 key2做opecation并将结果保存在destkey上

　　　　　　　　　　　　　　　　　　　　　　　　　　opecation可以是AND OR NOT XOR

　　（14）strlen key：取指定key的value值的长度

　  （15）setex key time value：设置key对应的值value，并设置有效期为time秒

**三、链表操作**

　　Redis的list类型其实就是一个每个子元素都是string类型的双向链表，链表的最大长度是2^32。list既可以用做栈，也可以用做队列。

　　list的pop操作还有阻塞版本，主要是为了避免轮询

　　（1）lpush key value：把值插入到链表头部

　　（2）rpush key value：把值插入到链表尾部

　　（3）lpop key ：返回并删除链表头部元素

　　（4）rpop key： 返回并删除链表尾部元素

　　（5）lrange key start stop：返回链表中[start, stop]中的元素

　　（6）lrem key count value：从链表中删除value值，删除count的绝对值个value后结束

　　　　　　　　　　　　　　　　count > 0 从表头删除　　count < 0 从表尾删除　　count=0 全部删除

　　（7）ltrim key start stop：剪切key对应的链接，切[start, stop]一段并把改制重新赋给key

　　（8）lindex key index：返回index索引上的值

　　（9）llen key：计算链表的元素个数

　　（10）linsert key after|before search value：在key 链表中寻找search，并在search值之前|之后插入value

　　（11）rpoplpush source dest：把source 的末尾拿出，放到dest头部，并返回单元值

　　　　应用场景： task + bak 双链表完成安全队列



　　　　业务逻辑： rpoplpush task bak

　　　　　　　　　接收返回值并做业务处理

　　　　　　　　　如果成功则rpop bak清除任务，如果不成功，下次从bak表取任务

（12）brpoplpush source dest timeout：把source 的末尾拿出，放到dest头部，并返回单元值,如果source为空，那么在timeout秒之内阻塞并等待可弹出的元素出现。

　　（13）brpop，blpop key timeout：等待弹出key的尾/头元素

　　　　　　　　　　　　　　　　timeout为等待超时时间，如果timeout为0则一直等待下去

　　　　　　应用场景：长轮询ajax，在线聊天时能用到

**四、hashes类型及操作**

　　Redis hash 是一个string类型的field和value的映射表，它的添加、删除操作都是O(1)（平均）。hash特别适用于存储对象，将一个对象存储在hash类型中会占用更少的内存，并且可以方便的存取整个对象。

　　配置： hash\_max\_zipmap\_entries 64 #配置字段最多64个

　　　　　 hash\_max\_zipmap\_value 512 #配置value最大为512字节

　　（1）hset myhash field value：设置myhash的field为value

　　（2）hsetnx myhash field value：不存在的情况下设置myhash的field为value

　　（3）hmset myhash field1 value1 field2 value2：同时设置多个field

　　（4）hget myhash field：获取指定的hash field

　　（5）hmget myhash field1 field2：一次获取多个field

　　（6）hincrby myhash field 5：指定的hash field加上给定的值

　　（7）hexists myhash field：测试指定的field是否存在

　　（8）hlen myhash：返回hash的field数量

　　（9）hdel myhash field：删除指定的field

　　（10）hkeys myhash：返回hash所有的field

　　（11）hvals myhash：返回hash所有的value

（12）hgetall myhash：获取某个hash中全部的field及value

（13）hincrbyfloat key-name key increment：将key存储的值加上浮点数increment

**五、集合结构操作**

　　特点：无序性、确定性、唯一性

　　（1）sadd key value1 value2：往集合里面添加元素

　　（2）smembers key：获取集合所有的元素

　　（3）srem key value：删除集合某个元素

　　（4）spop key：返回并删除集合中1个随机元素（可以坐抽奖，不会重复抽到某人）

　　（5）srandmember key：随机取一个元素

　　（6）sismember key value：判断集合是否有某个值

　　（7）scard key：返回集合元素的个数

　　（8）smove source dest value：把source的value移动到dest集合中

　　（9）sinter key1 key2 key3：求key1 key2 key3的交集

　　（10）sunion key1 key2：求key1 key2 的并集

　　（11）sdiff key1 key2：求key1 key2的差集

（12）sinterstore res key1 key2：求key1 key2的交集并存在res里

（13）sdiffstore dest key-name1 key-name2：将那些存在第一个集合但不存在于其他集合中的元素（数学上的差集运算）存储到dest里面。

（14）sunionstore dest key-name1 key-name2 将那些至少存在于一个集合中的元素（数学上的并集运算）存储到dest里面。

**六、有序集合**

　　概念：它是在set的基础上增加了一个顺序属性，这一属性在添加修改元素的时候可以指定，每次指定后，zset会自动按新的值调整顺序。可以理解为有两列的mysql表，一列存储value，一列存储顺序，操作中key理解为zset的名字。

　　和set一样sorted，sets也是string类型元素的集合，不同的是每个元素都会关联一个double型的score。sorted set的实现是skip list和hash table的混合体。

　　当元素被添加到集合中时，一个元素到score的映射被添加到hash table中，所以给定一个元素获取score的开销是O(1)。另一个score到元素的映射被添加的skip list，并按照score排序，所以就可以有序地获取集合中的元素。添加、删除操作开销都是O(logN)和skip list的开销一致，redis的skip list 实现是双向链表，这样就可以逆序从尾部去元素。sorted set最经常使用方式应该就是作为索引来使用，我们可以把要排序的字段作为score存储，对象的ID当元素存储。

　　（1）zadd key score1 value1：添加元素

　　（2）zrange key start stop [withscore]：把集合排序后,返回名次[start,stop]的元素  默认是升续排列  withscores 是把score也打印出来

　　（3）zrank key member：查询member的排名（升序0名开始）

　　（4）zrangebyscore key min max [withscores] limit offset N：集合（升序）排序后取score在[min, max]内的元素，并跳过offset个，取出N个

　　（5）zrevrank key member：查询member排名（降序 0名开始）

　　（6）zremrangebyscore key min max：按照score来删除元素，删除score在[min, max]之间

　　（7）zrem key value1 value2：删除集合中的元素

　　（8）zremrangebyrank key start end：按排名删除元素，删除名次在[start, end]之间的

　　（9）zcard key：返回集合元素的个数

　　（10）zcount key min max：返回[min, max]区间内元素数量

　　（11）zinterstore dest numkeys key1[key2..] [WEIGHTS weight1 [weight2...]] [AGGREGATE SUM|MIN|MAX]

　　　　　　求key1，key2的交集，key1，key2的权值分别是weight1，weight2

　　　　　　聚合方法用 sum|min|max

　　　　　　聚合结果 保存子dest集合内

　　　　　　注意：weights,aggregate如何理解？

　　　　　　　　　　答：如果有交集，交集元素又有score，score怎么处理？aggregate num->score相加，min最小score，max最大score，另外可以通过weights设置不同的key的权重，交集时  score\*weight

（12）zscore key member 获取对应key的对应member的分数值

（13）zincrby key increment member 对应key的对应member的分数值自增

（14）ZREVRANGE key start stop [WITHSCORES] 返回有序集中，指定区间内的成员。

（15）ZREVRANGEbyscore key-name min max 返回有序集中，指定区间内的成员，按照分支从大到小的顺序来返回。

（16）zunionstore dest-key key-count key1 key2 key3 [weights weight1 weight2] [aggregate sum|min|max] :对给定的有序集合执行类似于集合的并集运算。默认为sum

**七、服务器相关命令**

　　（1）ping：测定连接是否存活

　　（2）echo：在命令行打印一些内容

　　（3）select：选择数据库

　　（4）quit：退出连接

　　（5）dbsize：返回当前数据库中key的数目

　　（6）info：获取服务器的信息和统计

　　（7）monitor：实时转储收到的请求

　　（8）config get 配置项：获取服务器配置的信息

　　　　 config set 配置项  值：设置配置项信息

　　（9）flushdb：删除当前选择数据库中所有的key

　　（10）flushall：删除所有数据库中的所有的key

　　（11）time：显示服务器时间，时间戳（秒），微秒数

　　（12）bgrewriteaof：后台保存rdb快照

　　（13）bgsave：后台保存rdb快照

　　（14）save：保存rdb快照

　　（15）lastsave：上次保存时间

　　（16）shutdown [save/nosave]

　　　　　　注意：如果不小心运行了flushall，立即shutdown nosave，关闭服务器，然后手工编辑aof文件，去掉文件中的flushall相关行，然后开启服务器，就可以倒回原来是数据。如果flushall之后，系统恰好bgwriteaof了，那么aof就清空了，数据丢失。

　　（17）showlog：显示慢查询

　　　　　　问：多慢才叫慢？

　　　　　　答：由slowlog-log-slower-than 10000，来指定（单位为微秒）

　　　　　　问：服务器存储多少条慢查询记录

　　　　　　答：由slowlog-max-len 128，来做限制