# 全局命令

|  |  |  |
| --- | --- | --- |
| 查看所有的键 | keys \* | 返回所有的键 |
| 键总数 | dbsize | 整数，键的总数 |
| 检查键是否存在 | exists key | 存在：1，不存在：0 |
| 删除键 | del key | 删除成功：1，删除失败：0 |
| 键过期 | expire key seconds |  |
| 观察键剩余过期时间 | ttl | 大于0是剩余时间,  等于-1是未设置,  等于-2键不存在 |
| 键的数据结构类型 | type key |  |

|  |  |
| --- | --- |
| 命令 | 描述 |
| del | 该命令用于在 key 存在是删除 key。 |
| dump | 序列化给定 key ，并返回被序列化的值。 |
| exists | 检查给定 key 是否存在。 |
| expire | seconds 为给定 key 设置过期时间。 |
| expireat | 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)。 |
| pexpireat | 设置 key 的过期时间亿以毫秒计。 |
| pexpireat | 设置 key 过期时间的时间戳(unix timestamp) 以毫秒计 |
| keys | 查找所有符合给定模式( pattern)的 key 。 |
| move | 将当前数据库的 key 移动到给定的数据库 db 当中。 |
| persist | 移除 key 的过期时间，key 将持久保持。 |
| pttl | 以毫秒为单位返回 key 的剩余的过期时间。 |
| ttl | 以秒为单位，返回给定 key 的剩余生存时间(ttl, time to live)。 |
| randomkey | 从当前数据库中随机返回一个 key 。 |
| rename | 修改 key 的名称 |
| renamenx | 仅当 newkey 不存在时，将 key 改名为 newkey 。 |
| type | 返回 key 所储存的值的类型。 |
| object encoding | 查询内部编码 |

# 字符串类型命令

**1.常用命令**

|  |  |  |
| --- | --- | --- |
| 设置值 | set key value  [ex seconds]  [px milliseconds]  [nx|xx] | **ex seconds**：  为键设置秒级过期时间。  **px milliseconds**：  为键设置毫秒级过期时间。  **nx**：  键必须不存在，用于添加。  **xx**：  与nx相反，键必须存在，用于更新。 |
| 获取值 | get key |  |
| 批量设置值 | mset key value [key value ...] |  |
| 批量获取值 | mget key [key ...] |  |
| 计数 | incr key | incr命令用于对值做自增操作，结果分三种：  **值不是整数：**返回错误。  **值是整数：**返回自增后的结果。  **键不存在：**按照值为0自增，返回结果为1。 |

**2.不常用命令**

|  |  |
| --- | --- |
| 追加值 | append key value |
| 字符串长度 | strlen key |
| 设置并返回原值 | getset key value |
| 设置指定位置的字符 | setrange key offeset value |
| 获取部分字符串 | getrange key start end |

# 哈希命令

|  |  |  |
| --- | --- | --- |
| 设置值 | hset key field value | hset user:Tom age 23  // key user:Tom,  // filed age, value 23 |
| 获取值 | hget key field | hset get user:Tom |
| 删除field | hdel key field [field ...] | hdel user:Tom age |
| 计算field个数 | hlen key |  |
| 批量设置或获取field-value | hmset key field value [field value ...]  hmget key field [field ...] | hmset key user:Tom age 23 city shaoguan gender man  hmget key user:Tom age city gender |
| 判断field是否存在 | hexists key field | hexists user:Tom |
| 获取所有field | hkeys key | hkeys user:Tom |
| 获取所有value | hvals keys | hvals user:Tom |
| 获取所有的field-value | hgetall key | hgetall user:Tom |

# 列表命令

**1.添加操作**

|  |  |
| --- | --- |
| 从右边插入元素 | rpush key value [value ...] |
| 添加操作 | lpush key value [value ...] |
| 向某个元素前或者后插入元素 | linsert key before|after pivot value |

**2.查找**

|  |  |
| --- | --- |
| 获取指定范围内的元素列表 | lrange key start end |
| 获取列表指定索引下标的元素 | lindex key index |
| 获取列表长度 | llen key |

**3.删除**

|  |  |
| --- | --- |
| 从列表左侧弹出元素 | lpop key |
| 从列表右侧弹出 | rpop key |
| 删除指定元素 | lrem key count value |
| 按照索引范围修剪列表 | ltrim key start end |

|  |  |
| --- | --- |
| 修改 | lset key index newValue |

**5.阻塞操作**

|  |  |
| --- | --- |
|  | blpop key [key ...] timeout |
|  | brpop key [key ...] timeout |

# 集合命令

**1.集合内操作**

|  |  |
| --- | --- |
| 添加元素 | sadd key element [element ...] |
| 删除元素 | srem key element [element ...] |
| 计算元素个数 | scard key |
| 判断元素是否在集合中 | sismember key element |
| 随机从集合返回指定个数元素 | srandmember key [count] |
| 从集合随机弹出元素 | spop key |
| 获取所有元素 | smembers key |

**2.集合间操作**

|  |  |
| --- | --- |
| 求多个集合的交集 | sinter key [key ...] |
| 求多个集合的并集 | suinon key [key ...] |
| 求多个集合的差集 | sdiff key [key ...] |
| 将交集、并集、差集的结果保存 | sinterstore destination key [key ...]  suionstore destination key [key ...]  sdiffstore destination key [key ...] |

# 有序集合

**1.集合内**

1.1添加成员

zadd key score member [score member ...]

1.2计算成员个数

zcard key

1.3计算某个成员的分数

zscore key member

1.4计算成员的排名

zrank key member

zrevrank key member

1.5删除成员

zrem key member [member ...]

1.6增加成员的分数

zincrby key increment member

1.7返回指定排名范围的成员

zrange key start end [withscores]

zrevrange key start end [withscores]

1.8返回指定分数范围的成员

zrangebyscore key min max [withscores] [limit offset count]

zrevrangebyscore key max min [withscores] [limit offset count]

1.9返回指定分数范围成员个数

zcount key min max

1.10删除指定排名内的升序元素

zremrangebyrank key start end

1.11删除指定分数范围的成员

zremrangebyscore key min max

**2.集合间的操作**

2.1交集

zinterstore destination numkeys key [key ...] [weights weight [weight ...]]

[aggregate sum|min|max]

2.2并集

zunionstore destination numkeys key [key ...] [weights weight [weight ...]]

[aggregate sum|min|max]

# 键管理

**1.单个键管理**

1.1键重命名

rename key newkey

1.2随机返回一个键

randomkey

1.3键过期

除了expire、ttl命令以外，Redis还提供了expireat、pexpire、pexpireat、pttl、persist等一系列命令，下面分别进行说明：

expire key seconds：键在seconds秒后过期。

expireat key timestamp：键在秒级时间戳timestamp后过期。

1.4迁移键

1.move

2.dump+restore

3.migrate

**2.遍历键**

1.全量遍历键

keys pattern

2.渐进式遍历

scan cursor [match pattern] [count number]

3.数据库管理

1.切换数据库

select dbIndex

2.flushdb/flushall

flushdb/flushall命令用于清除数据库

flushdb只清除当前数据库，

flushall会清除所有数据库。

