**Redis 数据类型**

Redis支持五种数据类型：string（字符串），hash（哈希），list（列表），set（集合）及zset(sorted set：有序集合)

**Redis Key**

Redis是键值对数据库，如下列出对Key的操作命令

1. 基本命令语法

> COMMAND KEY\_NAME

示例：DEL runoobkey

1. 命令

- DEL key

该命令用于在 key 存在时删除 key。

- EXISTS key

检查给定 key 是否存在。

- PEXPIRE key milliseconds

设置 key 的过期时间以毫秒计。

- KEYS pattern

查找所有符合给定模式( pattern)的 key 。

- PERSIST key

移除 key 的过期时间，key 将持久保持。

1. 详细命令

<https://www.runoob.com/redis/redis-keys.html>

**String（字符串）**

String类型是二进制安全的，可以包含任何数据，如jpg图片或者序列化的对象

String 类型的值最大能存储 512MB。

1. 示例

redis 127.0.0.1:6379> SET name "runoob"

OK

redis 127.0.0.1:6379> GET name"runoob"

1. 命令

- SET key value

设置指定 key 的值

- GET key

获取指定 key 的值。

- GETSET key value

将给定 key 的值设为 value ，并返回 key 的旧值(old value)。

- MGET key1 [key2..]

获取所有(一个或多个)给定 key 的值。

- SETNX key value

只有在 key 不存在时设置 key 的值。

- STRLEN key

返回 key 所储存的字符串值的长度。

- MSET key value [key value ...]

同时设置一个或多个 key-value 对。

- INCR key

将 key 中储存的数字值增一。

- INCRBY key increment

将 key 所储存的值加上给定的增量值（increment） 。

- INCRBYFLOAT key increment

将 key 所储存的值加上给定的浮点增量值（increment） 。

- DECR key

将 key 中储存的数字值减一。

- DECRBY key decrement

key 所储存的值减去给定的减量值（decrement） 。

- APPEND key value

如果 key 已经存在并且是一个字符串， APPEND 命令将指定的 value 追加到该 key 原来值（value）的末尾。

**哈希(Hash)**

Redis hash 是一个String键值对集合，适合用于存储对象

1. 示例

> HMSET runoobkey name "redis tutorial" likes 20 visitors 23000

> HGETALL runoobkey

1. 命令

- HGETALL key

获取在哈希表中指定 key 的所有字段和值

- HMSET key field1 value1 [field2 value2 ]

同时将多个 field-value (域-值)对设置到哈希表 key 中。

- HSET key field value

将哈希表 key 中的字段 field 的值设为 value 。

1. 详细文档

<https://www.runoob.com/redis/redis-hashes.html>

**List（列表）**

Redis 列表是简单的字符串列表，按照插入顺序排序

1. 示例

> lpush runoob redis

> lpush runoob mongodb

> lrange runoob 0 10

1. 命令

- LINDEX key index

通过索引获取列表中的元素

- LLEN key

获取列表长度

- LPOP key

移出并获取列表的第一个元素

- LPUSH key value1 [value2]

将一个或多个值插入到列表头部

- LRANGE key start stop

获取列表指定范围内的元素

- LREM key count value

移除列表元素

- RPOP key

移除列表的最后一个元素，返回值为移除的元素。

- RPUSH key value1 [value2]

在列表中添加一个或多个值

**Set 集合**

Set 是 String 类型的无序集合，集合成员是唯一的，集合中不能出现重复的数据

1. 示例

> sadd runoob rabitmq

> smembers runoob

1. 命令

- SADD key member1 [member2]

向集合添加一个或多个成员

- SDIFFSTORE destination key1 [key2]

返回给定所有集合的差集并存储在 destination 中

- SINTER key1 [key2]

返回给定所有集合的交集

- SMEMBERS key

返回集合中的所有成员

- SREM key member1 [member2]

移除集合中一个或多个成员

**sorted set 有序集合**

与Set不相同的是，有序集合关联一个double类型的分数，通过分数来为集合中的成员进行从小到大的排序

1. 示例

> zadd runoob 0 rabitmq

> ZRANGEBYSCORE runoob 0 1000

1. 命令

- ZADD key score1 member1 [score2 member2]

向有序集合添加一个或多个成员，或者更新已存在成员的分数

- ZCARD key

获取有序集合的成员数

- ZCOUNT key min max

计算在有序集合中指定区间分数的成员数

- ZINCRBY key increment member

有序集合中对指定成员的分数加上增量 increment

- ZRANGEBYSCORE key min max [WITHSCORES] [LIMIT]

通过分数返回有序集合指定区间内的成员

- ZREM key member [member ...]

移除有序集合中的一个或多个成员

- ZREVRANK key member

返回有序集合中指定成员的排名，有序集成员按分数值递减(从大到小)排序

**事务**

Redis可以使用事务，以 MULTI 开始一个事务， 以EXEC 结束

示例：

redis 127.0.0.1:6379> MULTI

redis 127.0.0.1:6379> SET book-name "Mastering C++ in 21 days"

redis 127.0.0.1:6379> GET book-name

redis 127.0.0.1:6379> SADD tag "C++" "Programming" "Mastering Series"

redis 127.0.0.1:6379> SMEMBERS tag

redis 127.0.0.1:6379> EXEC