速度快：所有数据都存储在内存中。

Redis后台启动: ./redis-server ../redis.conf

关闭：./redis-cli shutdown

配置文件：redis.conf

JedisConnectionException

<http://blog.csdn.net/u012661010/article/details/73822712>

# Jedis



# 五种数据类型

**字符串**（String）、字符串列表（list）、有序字符串集合（sorted set）、**哈希**（hash）、字符串集合（set）。

# Key定义

不要太长，最好不要超过1024，也不能太短，考虑可读性。

# string

Value最大为512M。

set key value：赋值

get key：取值

getset key value：先取值后赋值

del key：删除

incr key：将指定值递增1，不存在则初始为0在自增（结果为1），不能转为整型则出错

decr key：递减，没有初始0并减一（结果为-1），不能转型则出错

incrby key num：指定值增加num

decrby key num：指定值减少num

# hash

String key和String value的map容器

赋值、取值

hset key field value：

hmset key field value [field value …]：新增多个元素

hget key field：

hmget key key …：获取多个元素

hgetall key：获取所有元素

删除

hdel key …：

# list

按照插入顺序排序的链表

两端添加、查看链表、两端弹出、获取个数、扩展。

lpush key value …：头插入

rpush key value …：右部插入

lrange key start end

lpop key：左边弹出

rpop key：右边弹出

llen key：长度

lrem key count value：删除count个值为value的元素

lset key index value：下标index处设置值为value

linsert key before value insertValue：在值为value的元素之前插入insertVlaue

linsert key after value insertValue：

# set

没有排序的字符集合，不能出现重复元素，可在服务器端完成聚合操作，效率高。

交、差和并集运算。

sadd key value …：添加元素

srem key value … ：删除元素

smembers key：查看

sismember key value：判断元素是否存在

sdiff key1 key2：差集运算（仅第一个不同元素）

sinter key1 key2：交集运算

sunion key1 key2：并集运算

scard key：数量

srandmember key：一个随机元素

sdiff/inter/union**store** key1 key2 key3：将key2和key3的差/交/并集存到key1中

# sorted-set

成员在集合中的位置是有序的，排序依赖于每一个元素的“分数”

zadd key score value …：增加分数为score的元素value

zscore key value：获得指定元素value的分数

zcard key：成员个数

zrem key value …：删除元素

zrange key from to withscores limit offset rows：范围查询，包括分数

zrevrange key from to withscores：倒序范围查询

zremrangebyrank key from to：范围删除

zremrangebyscire key from to：分数范围删除

zcount key from to：按分数计数

游戏积分排行榜，构建索引数据。

# 键操作

keys \*：查看所有键

keys my?：匹配键为my开头的键

exists key：验证键是否存在

rename key newKey：重命名键

expire key 100：超时时间

ttl key：剩余超时时间

type key：获取key的数据类型

# Redis特性

## 多数据库

一个redis实例最多16个数据库（0-15）。

select index：选择数据库，默认选择0号数据库。

move key index：移动键到指定数据库

## 事务

mulit、exec、discard操纵事务。

mulit：开启事务，之后的命令被存到命令队列。

exec：提交事务

discard：回滚操作

# 持久化

## RDB

在指定的时间间隔内将内存中的数据写入到磁盘

优势：

1. 只包含一个文件，利于文件备份
2. 利于灾难恢复
3. 性能最大化
4. 启动快速

缺点：

宕机情况时数据完全丢失，数据量大时写入磁盘耗时

## AOF

以日志的形式记录每一个操作，启动时通过读取日志重建数据

优势：

1. 更高的数据安全性
2. 日志以追加的形式添加，已有数据不会丢失
3. 日志过大时，Redis能加以操作记录

效率低于RDB，磁盘占用大于RDB。

## 不使用持久化

通过修改配置文件以禁用

## 同时使用RDB和AOF