

# Redis-day01-note

---

## Redis介绍

### • 特点及优点

- 1、开源的，使用C编写，基于内存且支持持久化
- 2、高性能的Key-Value的NoSQL数据库
- 3、支持数据类型丰富，字符串strings，散列hashes，列表lists，集合sets，有序集合sorted sets 等等
- 4、支持多种编程语言（C C++ Python Java PHP ... ）
- 5、单进程单线程

### • 与其他数据库对比

- 1、MySQL ： 关系型数据库，表格，基于磁盘，慢
- 2、MongoDB: 键值对文档型数据库，值为类似JSON文档，数据结构相对单一
- 3、Redis的诞生是为了解决什么问题？  
# 解决硬盘IO带来的性能瓶颈

### • 应用场景

1、使用Redis来缓存一些经常被用到、或者需要耗费大量资源的内容，通过这些内容放到redis里面，程序可以快速读取这些内容

2、一个网站，如果某个页面经常会被访问到，或者创建页面时消耗的资源比较多，比如需要多次访问数据库、生成时间比较长等，我们可以使用redis将这个页面缓存起来，减轻网站负担，降低网站的延迟，比如说网站首页等

# redis的诞生是为了解决负载问题

## • redis版本

1、最新版本：5.0

2、常用版本：2.4、2.6、2.8、3.0(里程碑)、3.2、3.4、4.0(教学环境版本)、5.0

## • Redis附加功能

1、持久化

将内存中数据保存到磁盘中，保证数据安全，方便进行数据备份和恢复

2、过期键功能

为键设置一个过期时间，让它在指定时间内自动删除

<节省内存空间>

# 音乐播放器，日播放排名，过期自动删除

3、事务功能

原子的执行多个操作

4、主从复制

5、Sentinel哨兵

# 安装

- Ubuntu

#### # 安装

```
sudo apt-get install redis-server
```

#### # 服务端启动

```
sudo /etc/init.d/redis-server status | start  
| stop | restart
```

#### # 客户端连接

```
redis-cli -h IP地址 -p 6379 -a 密码
```

## 配置文件详解

- 配置文件所在路径

```
/etc/redis/redis.conf
```

mysql的配置文件在哪里? :

```
/etc/mysql/mysql.conf.d/mysqld.cnf
```

- 设置连接密码

1、requirepass 密码

2、重启服务

```
sudo /etc/init.d/redis-server restart
```

3、客户端连接

```
redis-cli -h 127.0.0.1 -p 6379 -a 123456  
127.0.0.1:6379>ping
```

- 允许远程连接

1、注释掉本地IP地址绑定

69行: `# bind 127.0.0.1 ::1`

2、关闭保护模式(把yes改为no)

88行: `protected-mode no`

3、重启服务

`sudo /etc/init.d/redis-server restart`

## 数据类型

- 通用命令 ==适用于所有数据类型==

# 切换库(number的值在0-15之间,db0 ~ db15)

`select number`

# 查看键

`keys 表达式 # keys *`

# 数据类型

`TYPE key`

# 键是否存在

`exists key`

# 删除键

`del key`

# 键重命名

`rename key newkey`

# 清除当前库中所有数据(慎用)

`flushdb`

# 清除所有库中所有数据(慎用)

`flushall`

## 字符串类型(string)

## • 特点

- 1、字符串、数字，都会转为字符串来存储
- 2、以二进制的方式存储在内存中

## 字符串常用命令-==必须掌握==

```
# 1. 设置一个key-value
set key value
# 2. 获取key的值
get key
# 3. key不存在时再进行设置(nx)
set key value nx # not exists
# 4. 设置过期时间(ex)
set key value ex seconds

# 5. 同时设置多个key-value
mset key1 value1 key2 value2 key3 value3
# 6. 同时获取多个key-value
mget key1 key2 key3
```

## 字符串常用命令-==作为了解==

```
# 1. 获取长度
strlen key
# 2. 获取指定范围切片内容
getrange key start stop
# 3. 从索引值开始，value替换原内容
setrange key index value
# 4. 追加拼接value的值
append key value
```

## 数值操作-==字符串类型数字(必须掌握)==

### # 整数操作

INCRBY key 步长

DECRBY key 步长

INCR key : +1操作

DECR key : -1操作

# 应用场景：抖音上有人关注你了，是不是可以用**INCR**呢，如果取消关注了是不是可以用**DECR**

# 浮点数操作：自动先转为数字类型，然后再进行相加减，不能使用**append**

incrbyfloat key step

## 键的命名规范

mset wang:email [wangweichao@tedu.cn](mailto:wangweichao@tedu.cn)

```
127.0.0.1:6379> mset wang:email
```

```
wangweichao@tedu.cn guo:email
```

```
guoxiaonao@tedu.cn
```

```
OK
```

```
127.0.0.1:6379> mget wang:email guo:email
```

```
1) "wangweichao@tedu.cn"
```

```
2) "guoxiaonao@tedu.cn"
```

```
127.0.0.1:6379>
```

## string命令汇总

### # 字符串操作

1、set key value

2、set key value nx

3、get key

```
3、mset key1 value1 key2 value2
4、mget key1 key2 key3
5、set key value nx ex seconds
6、strlen key
# 返回旧值并设置新值（如果键不存在，就创建并赋值）
7、getset key value
# 数字操作
7、incrby key 步长
8、decrby key 步长
9、incr key
10、decr key
11、incrbyfloat key number#(可为正数或负数)
# 设置过期时间的两种方式
# 方式一
1、set key value ex 3
# 方式二
1、set key value
2、expire key 5 # 秒
3、pexpire key 5 # 毫秒
# 查看存活时间
ttl key
# 删除过期
persist key
```

- **string数据类型注意**

### # key值取值原则

- 1、key值不宜过长，消耗内存，且在数据中查找这类键值的计算成本高
- 2、不宜过短，可读性较差

### # 值

- 1、一个字符串类型的值最多能存储512M内容

## 练习

- 1、查看 db0 库中所有的键
- 2、设置键 `trill:username` 对应的值为 `user001`，并查看
- 3、获取 `trill:username` 值的长度
- 4、一次性设置 `trill:password`、`trill:gender`、`trill:fansnumber` 并查看（值自定义）
- 5、查看键 `trill:score` 是否存在
- 6、增加10个粉丝
- 7、增加2个粉丝（一个一个加）
- 8、有3个粉丝取消关注你了
- 9、又有1个粉丝取消关注你了
- 10、思考、思考、思考...，清除当前库
- 11、一万个思考之后，清除所有库

## 列表数据类型（List）

- 特点



- 1、元素是字符串类型
- 2、列表头尾增删快，中间增删慢，增删元素是常态
- 3、元素可重复
- 4、最多可包含 $2^{32} - 1$ 个元素
- 5、索引同python列表

## • 列表常用命令

### # 增

- 1、从列表头部压入元素  
`LPUSH key value1 value2`
- 2、从列表尾部压入元素  
`RPUSH key value1 value2`
- 3、从列表src尾部弹出1个元素，压入到列表dst的头部  
`RPOPLPUSH src dst`
- 4、在列表指定元素后/前插入元素  
`LINSERT key after|before value newvalue`

### # 查

- 5、查看列表中元素  
`LRANGE key start stop`  
# 查看列表中所有元素：`LRANGE key 0 -1`
- 6、获取列表长度  
`LLEN key`

### # 删

- 7、从列表头部弹出1个元素  
`LPOP key`
- 8、从列表尾部弹出1个元素  
`RPOP key`
- 9、列表头部，阻塞弹出，列表为空时阻塞

BLPOP key timeout

10、列表尾部,阻塞弹出,列表为空时阻塞

BRPOP key timeout

# 关于BLPOP 和 BRPOP

1、如果弹出的列表不存在或者为空,就会阻塞

2、超时时间设置为0,就是永久阻塞,直到有数据可以弹出

3、如果多个客户端阻塞再同一个列表上,使用First In First Service原则,先到先服务

11、删除指定元素

LREM key count value

count>0: 表示从头部开始向表尾搜索,移除与value相等的元素,数量为count

count<0: 表示从尾部开始向表头搜索,移除与value相等的元素,数量为count

count=0: 移除表中所有与value相等的值

12、保留指定范围内的元素

LTRIM key start stop

LRTIM mylist1 0 2 # 只保留前3条

# 应用场景: 保存微博评论最后500条

LTRIM weibo:comments 0 499

# 改

13、LSET key index newvalue

## 练习

- 1、查看所有的键
- 2、向列表 `spider:urls` 中以RPUSH放入如下几个元素：  
`01_baidu.com`、`02_taobao.com`、`03_sina.com`、  
`04_jd.com`、`05_xxx.com`
- 3、查看列表中所有元素
- 4、查看列表长度
- 5、将列表中`01_baidu.com` 改为 `01_tmall.com`
- 6、在列表中`04_jd.com`之后再加1个元素 `02_taobao.com`
- 7、弹出列表中的最后一个元素
- 8、删除列表中所有的 `02_taobao.com`
- 9、剔除列表中的其他元素，只剩前3条

## 与python交互

- 模块(redis)

Ubuntu

```
sudo pip3 install redis
```

- 使用流程

```
import redis
# 创建数据库连接对象
r =
redis.Redis(host='127.0.0.1',port=6379,db=0,password='123456')
```

- 通用命令代码示例

- **python操作list**

**list案例: 一个进程负责生产任务, 一个进程负责消费任务**

进程1: 生产者

进程2: 消费者

- **python操作string**