Redis基础

1. 位操作：getbit, setbit, bitcount

使用这些位操作可以非常紧凑地存储bool，100万bits=125KB

\*如果做进行位操作时，指定的偏移量位置超出了当前的长度，那么Redis会自动分配内存，并且将直到该偏移量位置中间的bit都设为0，所以如果把误偏移量设的很大，会造成内存浪费与服务器阻塞

2. 不支持数据类型嵌套

Redis所有的数据理性都不知道嵌套，比如hash类型存储的键值对必须是字符串；List类型中存放的对象也必须是String等等。

3. HSET命令不区分插入与更新

使用hset时不需要区分插入或是更新，如果想要更新的key不存在，redis会主动创建这个对象

4. List类型

List内部是使用双向链表来实现的，可以快速地向两端添加元素，所以访问的元素越接近两端，速度就越快。

可以处理一些关系型数据库难以处理的问题，比如社交网站的新鲜事，人们会对大量地最新的100个数据进行查询，这时候使用双向链表就很高效

List类型适合处理很少访问中间元素的情况

5. 事务

Redis能保证一个事务内的命令依次执行而不被其他命令插入，说明redis的事务更像是mysql事务隔离级别中的serializable，串行执行

如果事务中包含有语法错误的命令，redis会做出提示，并且会在用户发出EXEC执行事务时，直接返回错误，不执行事务中的任何一条命令

如果事务中包含有运行是错误的命令，redis会在事务执行时做出提示，但是redis不具备事务回滚的功能，事务会继续执行下去，其他正确地命令依然会得到执行

6. watch命令

WATCH命令会监控一个或多个值，一旦被监控的值，在执行了watch以后被修改，那么redis会阻止之后的一个事务的执行；该事务返回失败后，会取消所有变量的watch，后面的任何事务都可以正常执行

\*WATCH并不能保证其他客户端不修改这个键值

7.