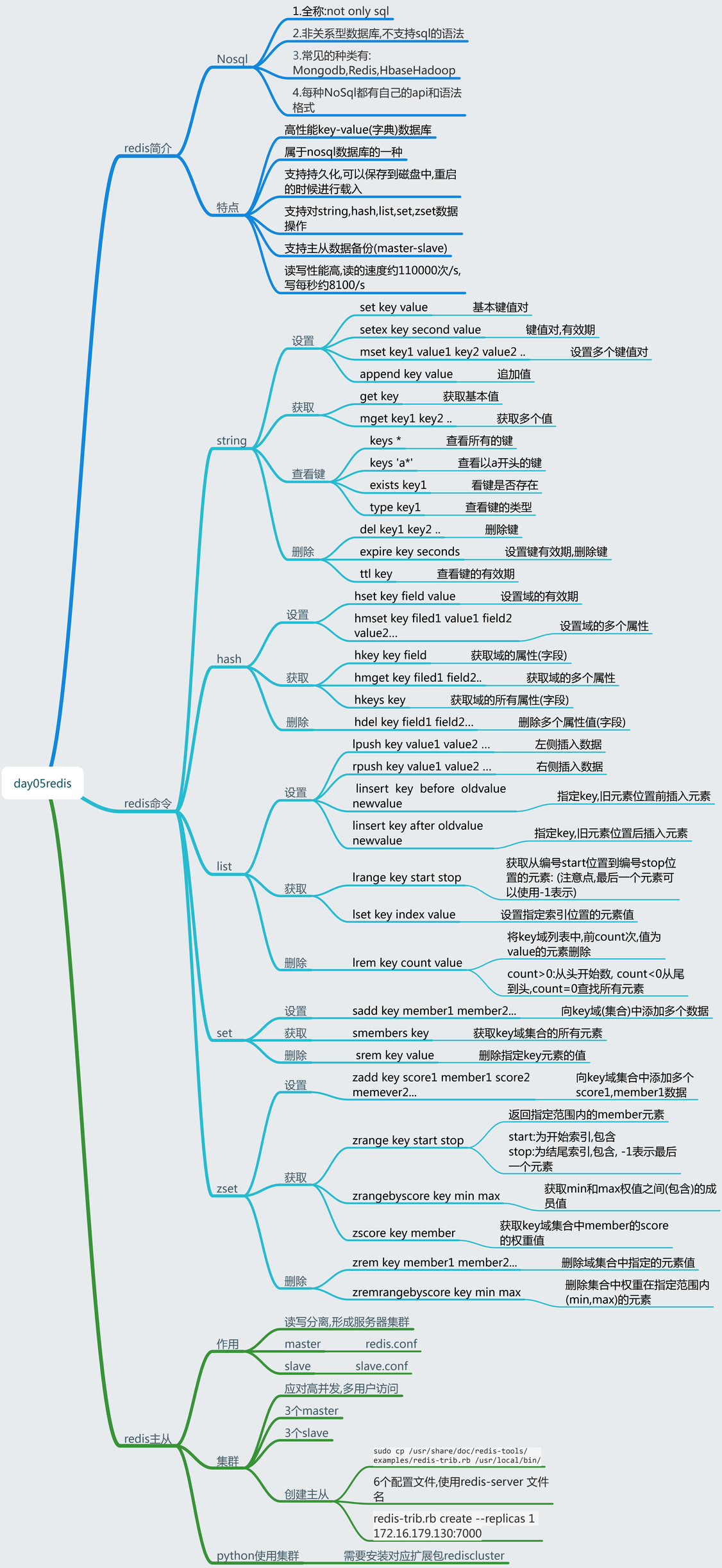

day05redis
redis简介
Nosql
1.全称:not only sql
2.非关系型数据库,不支持sql的语法
3.常见的种类有:Mongodb,Redis,HbaseHadoop
4.每种NoSql都有自己的api和语法格式
特点
高性能key-value(字典)数据库
属于nosql数据库的一种
支持持久化,可以保存到磁盘中,重启的时候进行载入
支持对string,hash,list,set,zset数据操作
支持主从数据备份(master-slave)
读写性能高,读的速度约110000次/s,写每秒约8100/s
redis命令
string
设置
set key value 基本键值对
setex key second value 键值对,有效期
mset key1 value1 key2 value2 .. 设置多个键值对
append key value 追加值
获取
get key 获取基本值
mget key1 key2 .. 获取多个值
查看键
keys * 查看所有的键
keys 'a*' 查看以a开头的键
exists key1 看键是否存在
type key1 查看键的类型
删除
del key1 key2 .. 删除键
expire key seconds 设置键有效期,删除键
ttl key 查看键的有效期
hash
设置
hset key field value 设置域的有效期
hmset key filed1 value1 field2 value2... 设置域的多个属性
获取
hkey key field 获取域的属性(字段)
hmget key filed1 field2.. 获取域的多个属性
hkeys key 获取域的所有属性(字段)
删除
hdel key field1 field2... 删除多个属性值(字段)
list
设置
lpush key value1 value2 ... 左侧插入数据
rpush key value1 value2 ... 右侧插入数据
linsert key before oldvalue newvalue 指定key,旧元素位置前插入元素
linsert key after oldvalue newvalue 指定key,旧元素位置后插入元素
获取
lrange key start stop 获取从编号start位置到编号stop位置的元素: (注意点,最后一个元素可以使用-1表示)
lset key index value 设置指定索引位置的元素值
删除
lrem key count value 将key域列表中,前count次,值为value的元素删除
count>0:从头开始数, count<0从尾到头,count=0查找所有元素
set
设置
sadd key member1 member2... 向key域(集合)中添加多个数据
获取
smembers key 获取key域集合的所有元素
删除
srem key value 删除指定key元素的值
zset
设置
zadd key score1 member1 score2 memever2... 向key域集合中添加多个score1,member1数据
获取
zrange key start stop
返回指定范围内的member元素
start:为开始索引,包含
stop:为结尾索引,包含, -1表示最后一个元素
zrangebyscore key min max 获取min和max权值之间(包含)的成员值
zscore key member 获取key域集合中member的score的权重值
删除
zrem key member1 member2... 删除域集合中指定的元素值
zremrangebyscore key min max 删除集合中权重在指定范围内(min,max)的元素
redis主从
作用
读写分离,形成服务器集群
master redis.conf
slave slave.conf
集群
应对高并发,多用户访问
3个master
3个slave
创建主从
sudo cp /usr/share/doc/redis-tools/examples/redis-trib.rb /usr/local/bin/
6个配置文件,使用redis-server 文件名
redis-trib.rb create --replicas 1 172.16.179.130:7000
python使用集群 需要安装对应扩展包rediscluster

01_服务器

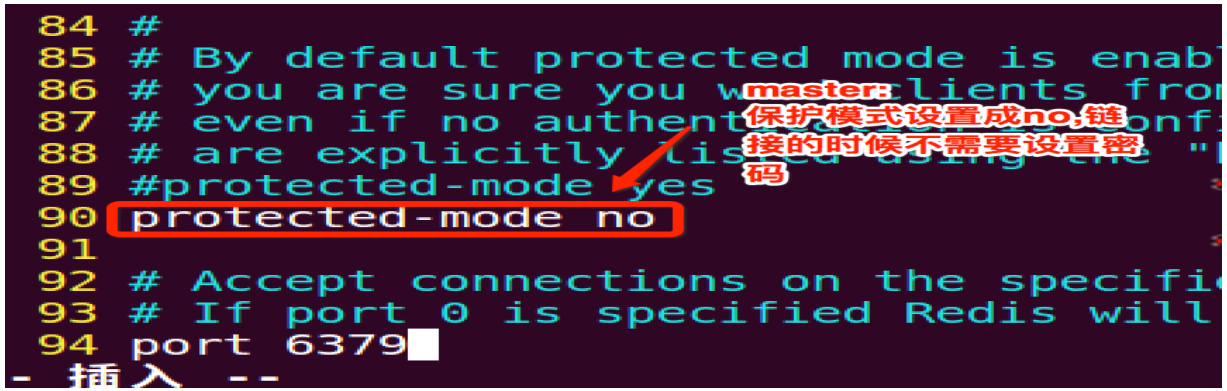
02_公众号

03.master

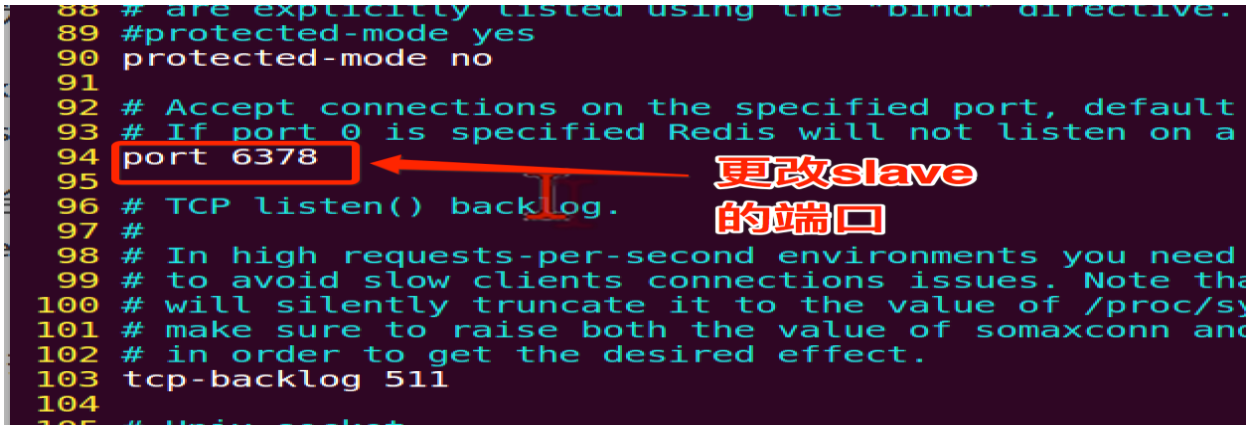
```
62 # the IPv4 lookback interface address
63 # accept connections only from client
64 # is running).
65 #
66 # IF YOU ARE SURE YOU WANT YOUR INSTA
67 # JUST COMMENT THE FOLLOWING LINE.
68 # ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
69 #bind 127.0.0.1
70 bind 192.168.87.52
71
72 # Protected mode is a layer of securi
73 # Redis instances left open on the in
-- 插入 --
```

修改为本地ip地址
(ifconfig)

04.master



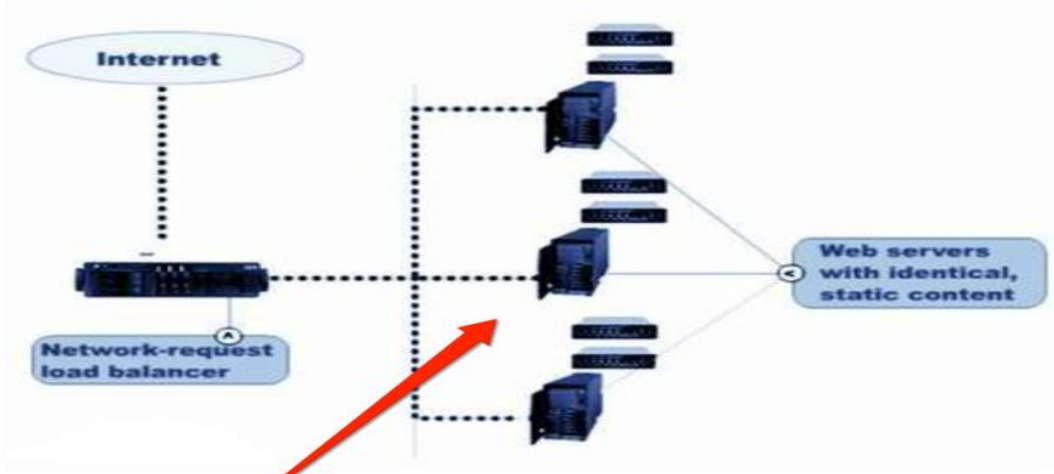
05.slave



06.slave

07.负载均衡

- 集群是一组相互独立的、通过高速网络互联的计算机，它们构成了一个组，并以单一系统的模式加以管理。一个客户与集群相互作用时，集群像是一个独立的服务器。集群配置是用于提高可用性和可缩放性。



当请求到来首先由负载均衡服务器处理，把请求转发到另外的一台服务器上。

Ngnix服务器

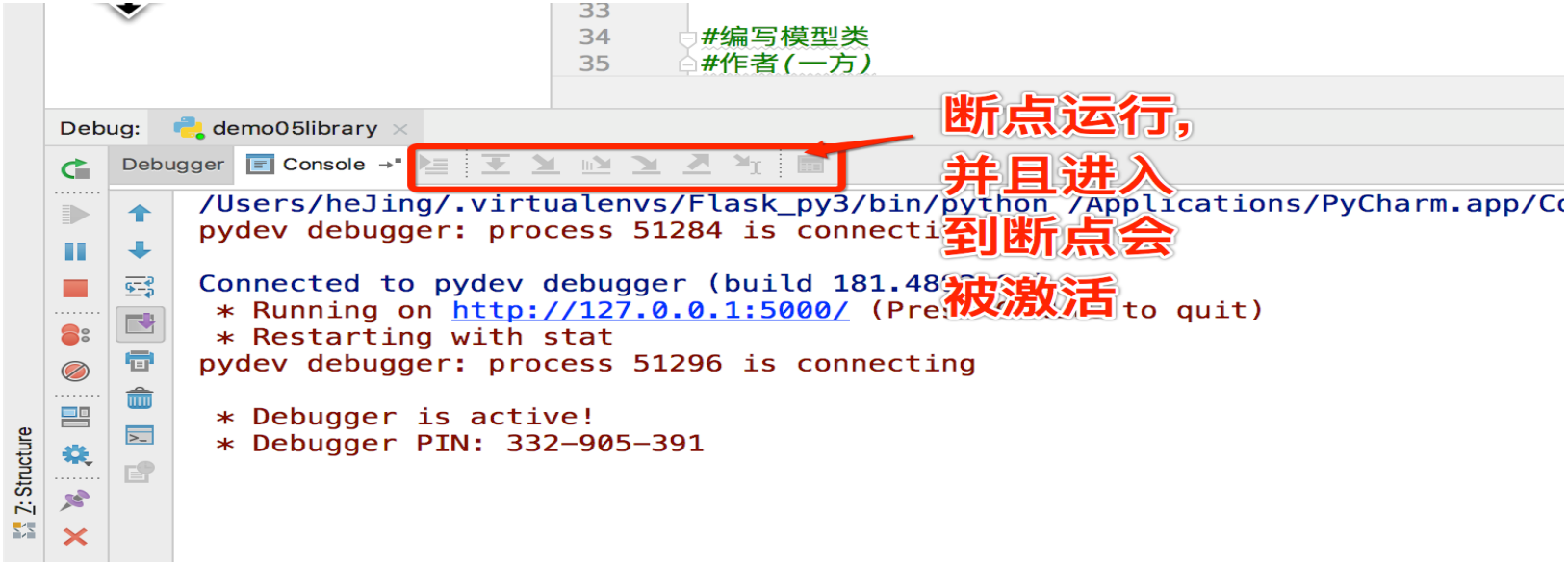
redis集群

- 分类
  - 软件层面

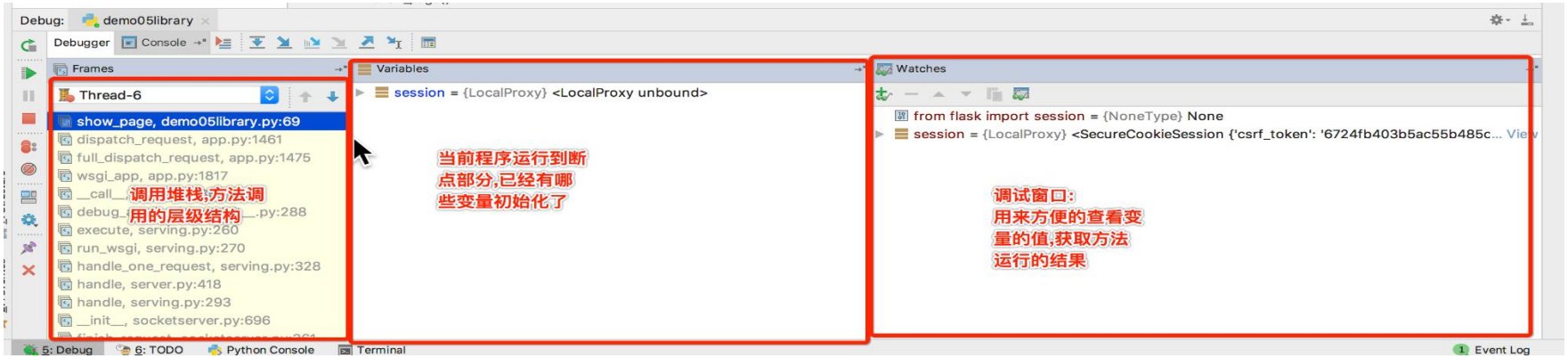
08.主从

```
192.168.87.52:7002
Adding replica 192.168.87.52:7003 to 192.168.87.52:7000
Adding replica 192.168.87.52:7004 to 192.168.87.52:7001
Adding replica 192.168.87.52:7005 to 192.168.87.52:7002
M: 6a18fefe978e04a3ef584ef04457bad593b231c6 192.168.87.52:7000
   slots:0-5460 (5461 slots) master
M: 373c49efdb92566efa90fe9795486e79715ffae8 192.168.87.52:7001
   slots:5461-10922 (5462 slots) master
M: ae23d87e3db545033690bdd6b9231a383281c346 192.168.87.52:7002
   slots:10923-16383 (5461 slots) master
S: 8fadc17b4452e8cbacd7b2e062eca72cccf48c6e 192.168.87.52:7003
   replicates 6a18fefe978e04a3ef584ef04457bad593b231c6
S: c902d9d1f501233e2455b3d23f7903b8333144f5 192.168.87.52:7004
   replicates 373c49efdb92566efa90fe9795486e79715ffae8
S: 334f98f416db0fa2967e8564da48ef8982989313 192.168.87.52:7005
   replicates ae23d87e3db545033690bdd6b9231a383281c346
Can I set the above configuration? (type 'yes' to accept):
```

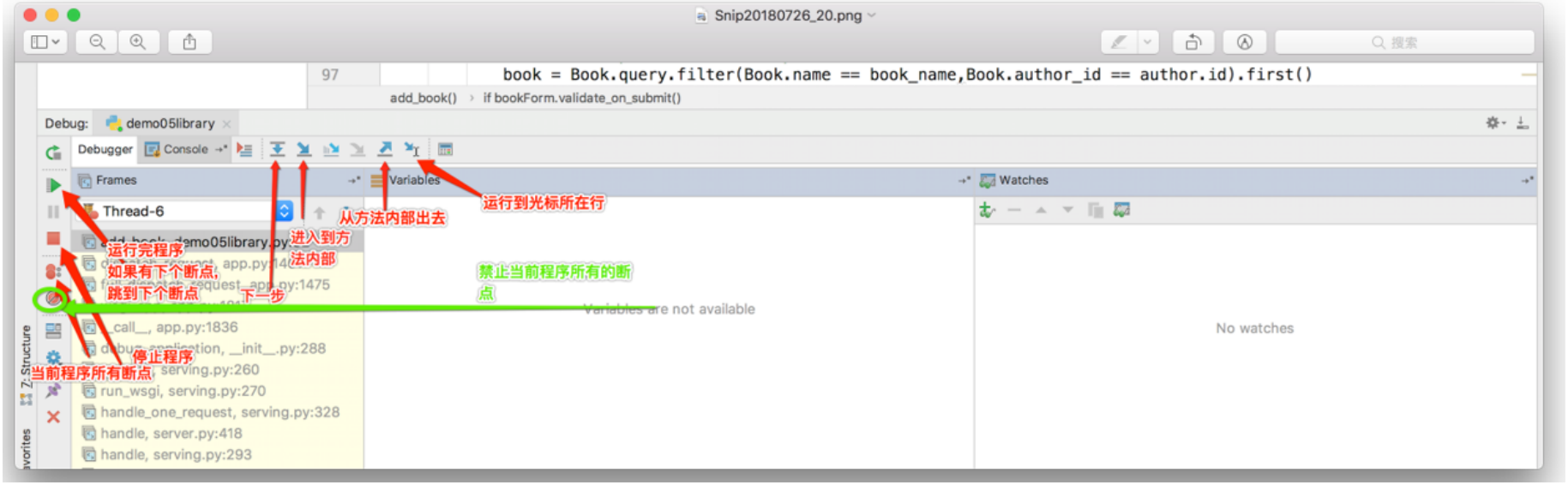
09.断点调试



10.断点调试



11.断点调试



debug 说明

debug(排除错误):

位置:断点一般打在看不懂,或者有疑问的地方,如果不知道怎么打,就打在有效代码第一行(视图函数内部)

不需要打在: 方法名称定义部分, 或者是类名定义部分

redis 操作命令

在 Ubuntu 下面操作:

redis 的操作是以 key - value 的形式存储的

key 为字符串

Value: 字符串,hash 表,list 集合,set 集合,zset 集合 五种类型

字符串类型:

1. 基本键值对: set key value
2. 键值对,有效期: setex key second value
3. 设置多个键值对: mset key1 value1 key2 value2 ..
4. 追加值: append key value

获取值:

1.获取基本值: get key
2.获取多个值: mget key1 key2 ..

查看键:

1.查看所有的键: keys *
2.查看以 a 开头的键: keys a*
3.看键是否存在: exists key
4.查看键的类型: type key1

删除键值对:

1.删除键: del key1 key2 ..
2.设置键有效期,删除键:expire key seconds
3.查看键的有效期: ttl key

hash 类型

增加数据

1.增加域的 key 和值: hset key field value
2.设置域的多个属性: hmset key filed1 value1 field2 value2…

获取数据:

1.获取域的属性(字段): hget key field
2.获取域的多个属性: hmget key filed1 field2..
3.获取域的所有属性(字段): hkeys key

删除数据:

1.删除多个属性值(字段): hdel key field1 field2…

list 类型:
插入数据:
1.左侧插入数据: lpush key value1 value2 …
2.右侧插入数据: rpush key value1 value2 …
3.指定 key,旧元素位置前插入元素:
格式: linsert key before oldvalue newvalue

4.指定 key,旧元素位置后插入元素:
格式: linsert key after oldvalue newvalue

获取数据:
1.获取从编号 start 位置到编号 stop 位置的元素:
格式: lrange key start stop
(注意点,最后一个元素可以使用-1 表示)
2.设置指定索引位置的元素值: lset key index value

删除数据:
1.将 key 域列表中,前 count 次,值为 value 的元素删除:
格式: lrem key count value
count:需要删除的个数
value: 需要删除的值
count>0:从头开始数, count<0 从尾到头,count=0 查找所有元素

set 类型
特点:无序(顺序)集合, 大小关系有序, 不能存储重复元素
1 6 3 9 2

添加元素
1.向 key 域(集合)中添加多个数据, sadd key member1 member2...

获取元素:
2.获取 key 域集合的所有元素: smembers key

删除元素:
1.删除指定 key 元素的值: srem key value

Zset 类型
特点: 有序集合, score 为权重值
序:自然顺序
增加数据
1.向 key 域集合中添加多个 score1,member1 数据
格式: zadd key score1 member1 score2 memever2...

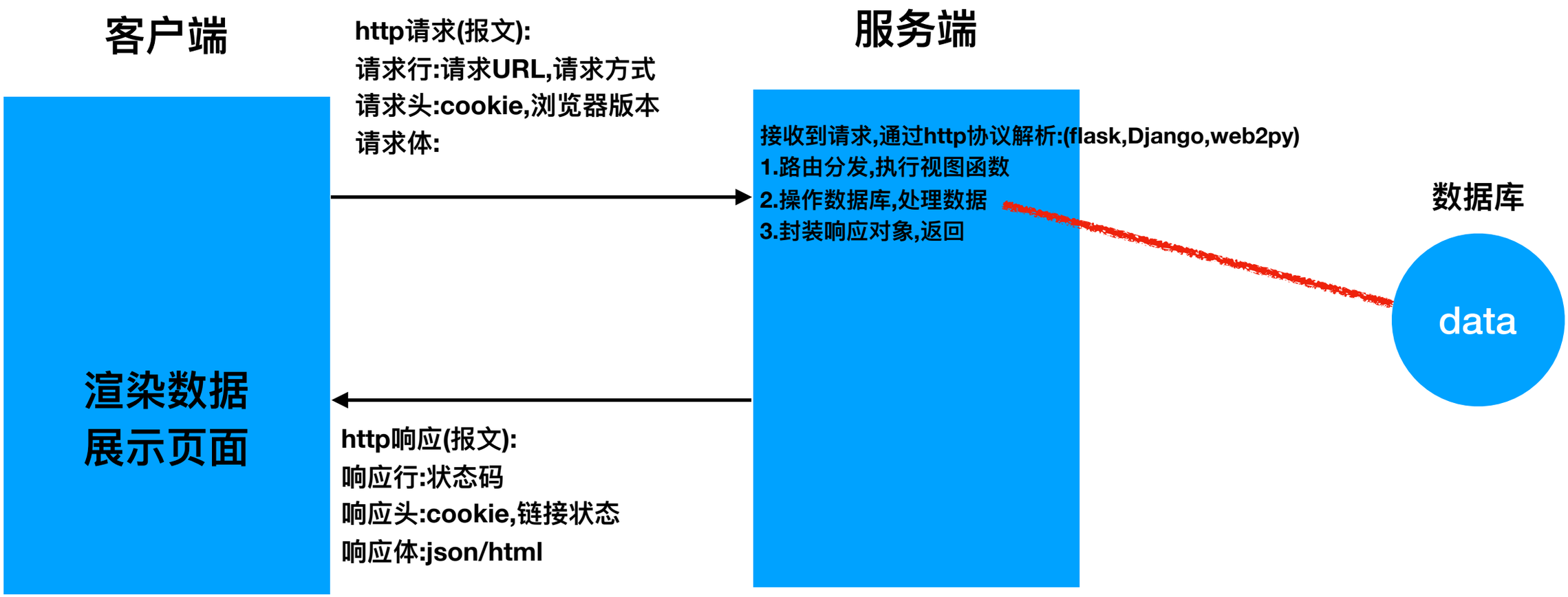
获取数据:
1.返回指定范围内的 member 元素:
格式:zrange key start stop
    start:为开始索引,包含
    stop:为结尾索引,包含, -1 表示最后一个元素

2.获取 min 和 max 权值之间(包含)的成员值:
格式:zrangebyscore key min max

3.获取 key 域集合中 member 的 score 的权重值:
格式:zscore key member

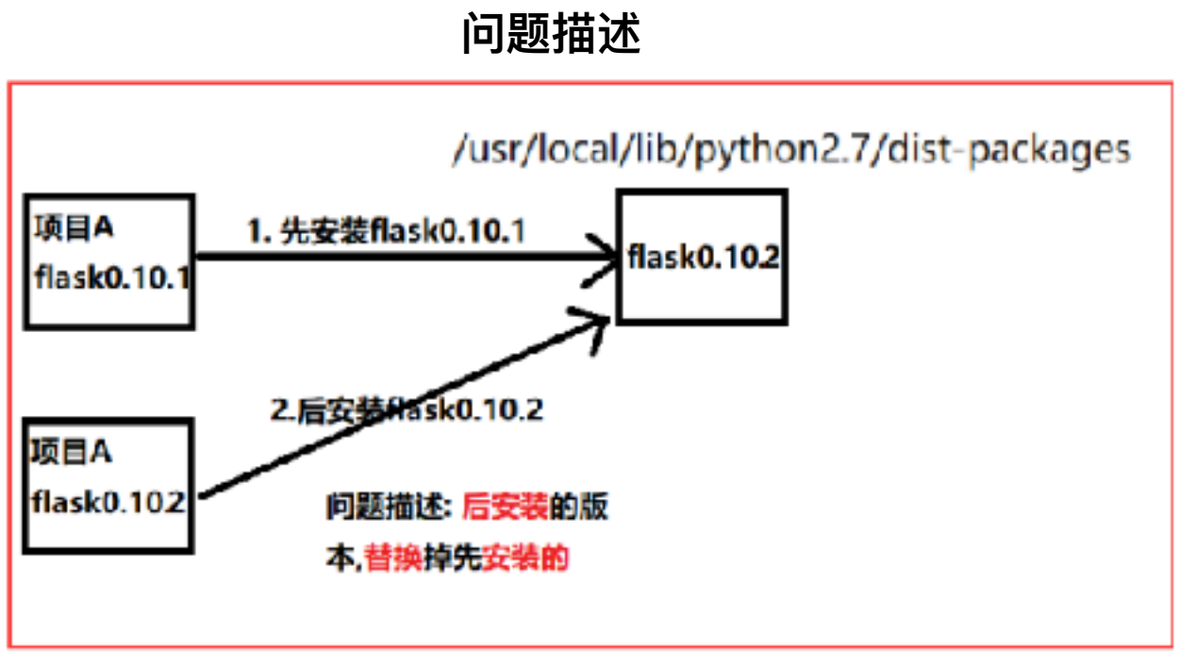
删除元素:
1.删除域集合中指定的元素值:
格式:zrem key member1 member2...
2.删除集合中权重在指定范围内(min,max)的元素:
格式:zremrangebyscore key min max

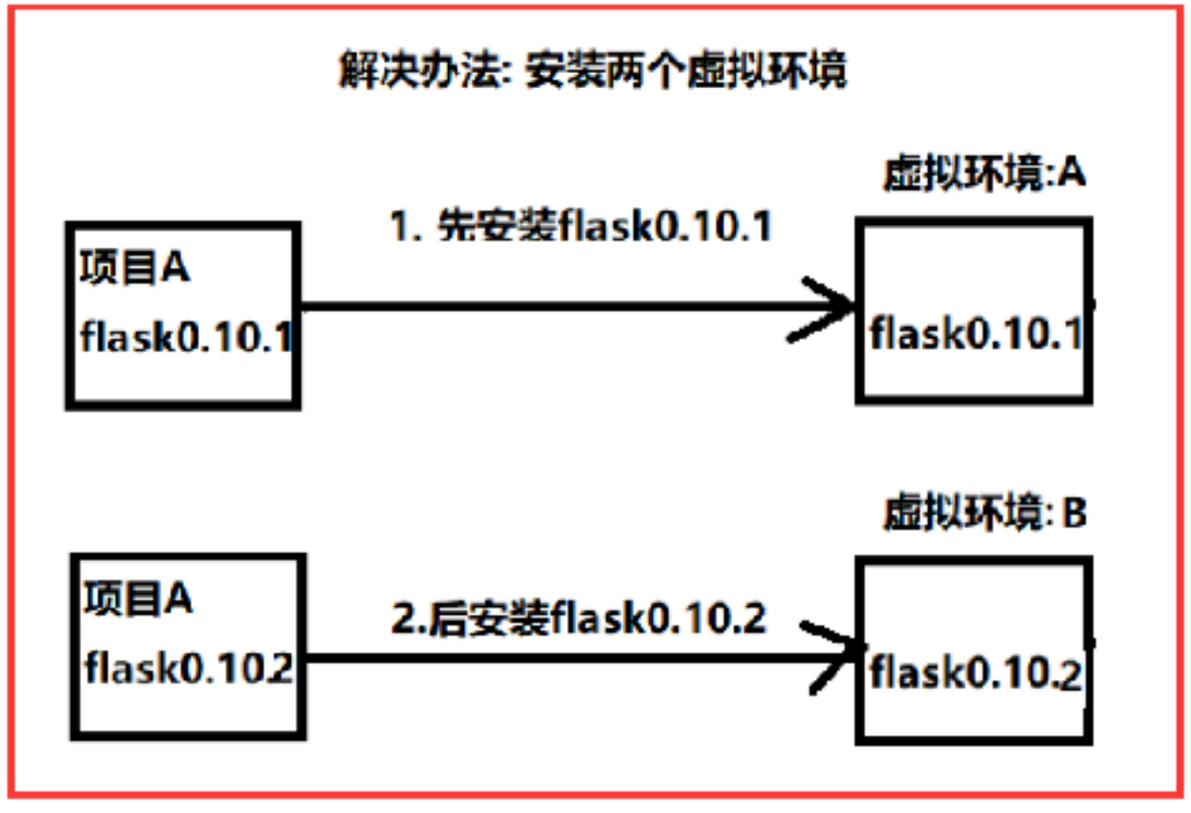
HTTP通讯流程


客户端
渲染数据
展示页面
http请求(报文):
请求行:请求URL,请求方式
请求头:cookie,浏览器版本
请求体:
http响应(报文):
响应行:状态码
响应头:cookie,链接状态
响应体:json/html
服务端
接收到请求,通过http协议解析:(flask,Django,web2py)
1.路由分发,执行视图函数
2.操作数据库,处理数据
3.封装响应对象,返回
数据库
data

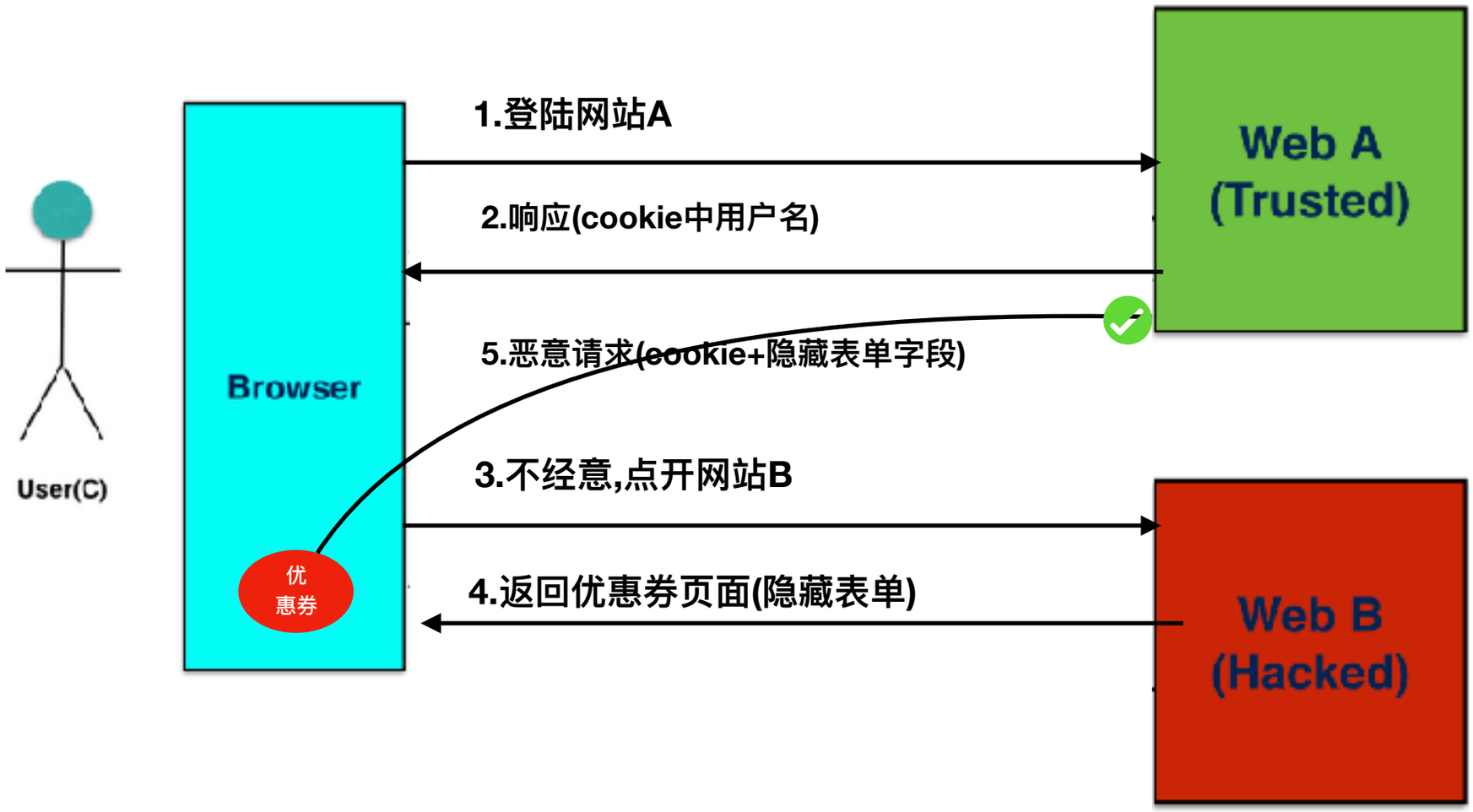
为什么要有虚拟环境

为了更加方便的调试各种版本的flask,python解释器版本

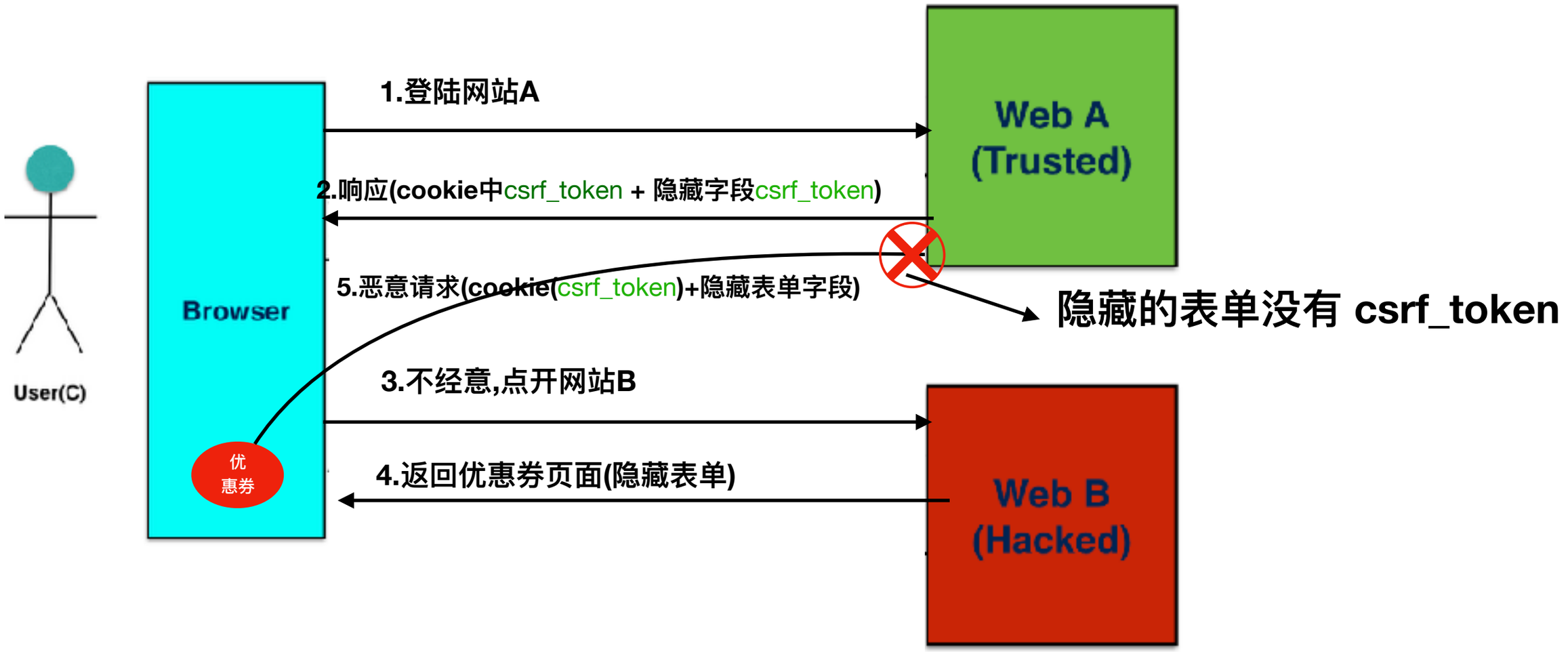




攻击过程


1.登陆网站A
2.响应(cookie中用户名)
5.恶意请求(cookie+隐藏表单字段)
3.不经意,点开网站B
4.返回优惠券页面(隐藏表单)
Web A
(Trusted)
Browser
优
惠券
User(C)
Web B
(Hacked)

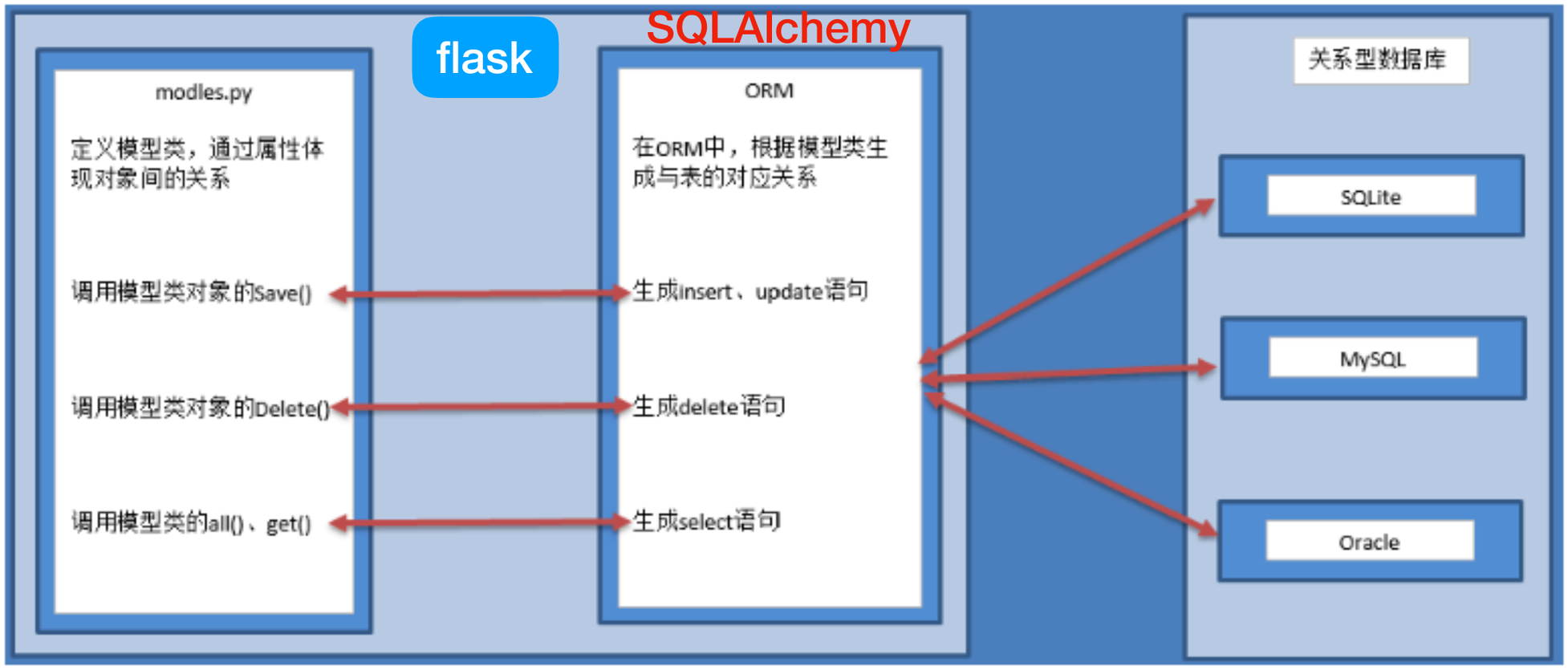
解决办法



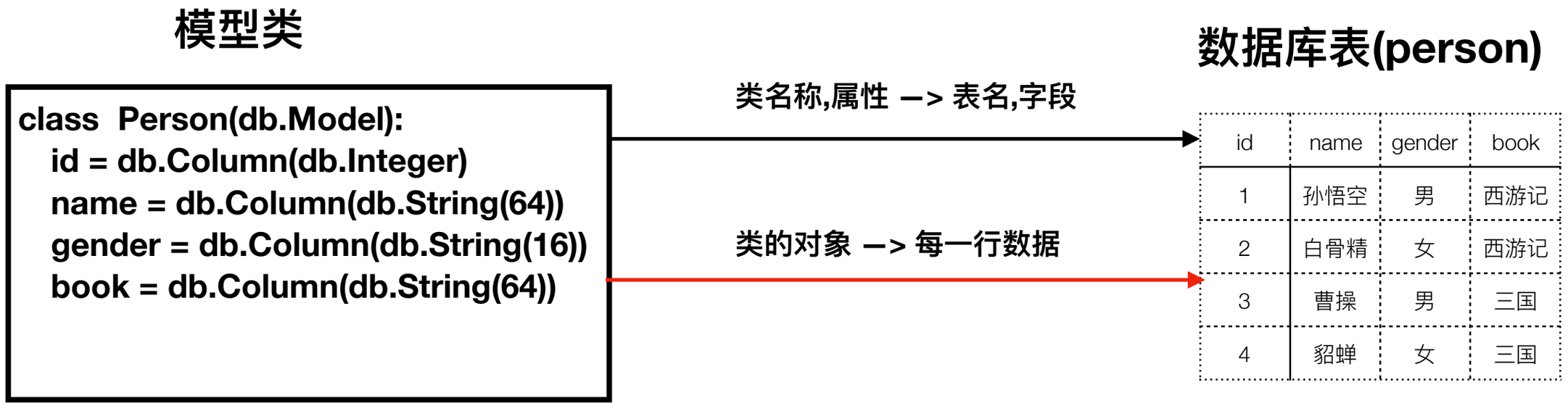
同源策略:

不同的网站之间的cookie是不能共享

ORM(object relationship mapping)
对象关系映射模型



模型类,数据库表



总结,通过SQLALchemy模型扩展包:

1. 将类名称,属性, 翻译成表的表名和字段值
2. 每一个对象都会翻译成表中的一行数据

作者: xx

书籍: xx

添加

作者: xx1

书籍: book1

书籍: book2

书籍: book3

作者: xx2

书籍: book1

书籍: book2

书籍: book3

多对多关系

学生表

主键(id)	学生名(name)
1	张三
2	李四
3	王五

课程表

主键(id)	课程名(name)
1	物理
2	化学
3	生物

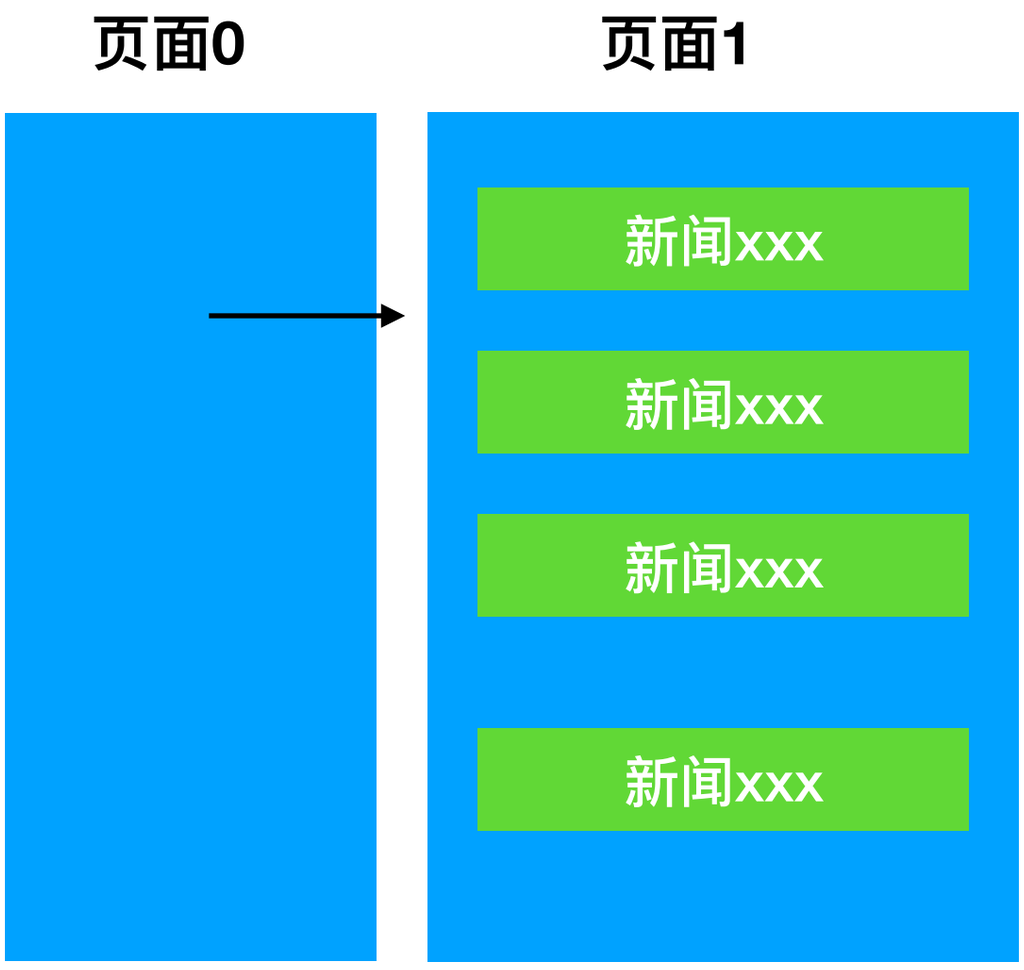
学生课程表(中间表)

外键(student.id)	外键(course.id)
1	2
1	3
2	2
3	1
3	2
3	3

- 查询某个学生选修了哪些课程，例如:
  查询王五选修了哪些课程
  - 取出王五的 id 去 Student_Course 表中查询 student.id 值为 3 的所有数据
  - 查询出来有3条数据，然后将这3条数据里面的 course.id 取值并查询 Course 表即可获得结果
- 查询某个课程都有哪些学生选择，例如：查询生物课程都有哪些学生选修
  - 取出生物课程的 id 去 Student_Course 表中查询 course.id 值为 3 的所有数据
  - 查询出来有2条数据，然后将这2条数据里面的 student.id 取值并查询 Student 表即可获得结果

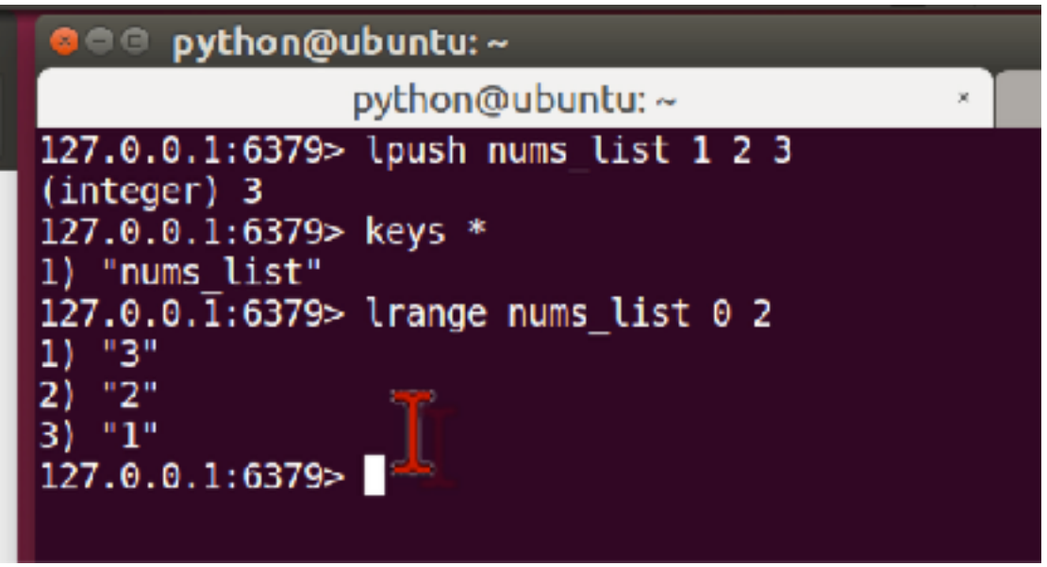
hash存储格式:
hset key field value

hash获取格式:
hget key filed



比如:
hset person name zhangsan
hset person age 13

应用场景:
hset data page1 四条数据
hset data page2 四条数据
分页存储!



zset集合存储内容图解: