**1.补充上篇中的os，sys模块的几个小功能**

import os,sys

res1=os.system('ipconfig')#os.system(commad)执行操作系统命令，不返回执行结果

res2=os.popen('ipconfig').read()#os.popen(commad)执行操作系统命令，返回执行结果，通过read方法可以读取内容

inpu=sys.argv#argv获取执行该python文件时的参数为list，如 ls -a 这里的a就是参数

if len(inpu)>1:

if inpu[1]=='-install':

print('安装成功')

elif inpu[1]=='-help':

print('靠人不如靠己，别老help')

else:

print('错了')

**2.time,datetime模块**

import time,datetime

print(time.time())#获取当前时间点的时间戳

print(time.localtime())#获取当前时间点的时间元祖

print(time.strftime('%Y-%m-%d %H:%M:%S'))#获取当前时间点的格式化时间

print(datetime.datetime.now())#获取当前时间点的格式化时间

print(datetime.datetime.now()+datetime.timedelta(+3))#获取当前时间点三天后的时间

print(datetime.datetime.now()+datetime.timedelta(-3))#获取当前时间点三天前的时间

print(time.gmtime(1234556489))#时间戳转换成时间元祖

print(time.mktime(time.gmtime(1234556489)))#时间元祖转换成时间戳

print(time.strftime('%Y',time.gmtime(1234556489)))#时间元祖转换成格式化时间

print(time.strptime('2009','%Y'))#格式化时间转换成时间元祖

def transfer(timestamp,format='%Y%m%d'):#时间戳格式化输出

timetuple=time.gmtime(timestamp)

print(time.strftime(format,timetuple))

def transfer2(timesformat,format='%Y%m%d'):#格式化时间转换成时间戳输出

timetuple=time.strptime(timesformat,format)

print(time.mktime(timetuple))

transfer(231568465)

transfer2('19770504')

**3.hashlib模块**

import hashlib

md=hashlib.md5()

md.update('123456'.encode())#一定要造待加密的字符串后面加上“.encode”编码，才能加密，否则报错

print(md.hexdigest())

def md\_sault(str,sault='123456'):#加盐,sault是盐值，默认是123456

import hashlib

str=str+sault

md=hashlib.md5()

md.update(str.encode())

print(md.hexdigest())

md\_sault('adsfad')

**4.redis模块**

import redis

coon=redis.Redis(host='192.168.217.128',password='123456',port=6379,db=1)

print(coon.hget('Idcards','花校纳').decode())#decode转码，这里get到的数据是bytes格式的，需要转码

coon.set('葛怕愿',123456789)

print(coon.get('戴铁视').decode())

coon.hset('Idcards','牛逼',123456789)

print(coon.hget('Idcards','牛逼').decode())

print(coon.keys())#数据库上所有的key

res=coon.hgetall('Idcards')

for i in res:

print(i.decode(),':',res[i].decode())

coon.delete('yxg')#删除数据，yxg是key

# 需求是把一个redis库里面所有的数据，导入到另一个redis里面

# 1、r1 和r2 连接上r1和r2两个数据库

# 2、 获取到r1上面所有的key keys

# 3、判断key的类型，r.type(k)，get hgetall .hset() .set()

r1=redis.Redis(host='211.149.218.16',port=6378,password='123456',db=2)#获取数据

r2=redis.Redis(host='211.149.218.16',port=6378,password='123456',db=3)#写数据

keys = r1.keys()#r1数据库上所有的key

for k in keys:

if r1.type(k)==b'hash':#判断是否为哈希类型，因为redis里面返回的数据都是bytes类型的，

#所以在hash前面加上b

hash\_data = r1.hgetall(k)#获取哈希的类型的数据

for k2,v in hash\_data.items():#循环刚才获取到的字典

r2.hset(k,k2,v)#set哈希类型的值

else:

v = r1.get(k)#从r1里面获取值，

r2.set(k,v)#set进去

**5.json模块**

import json

names={"yxg":123456,"yyy":654321,"zn":666}

res=json.dumps(names)#字典转json，dumps在mysql里有这个概念，mysqldump导出数据，这里字典是python自带的，json是外来的，所以dumps是把字典导出为json

new\_names=json.loads(res)#json转字典，与dumps相反，那边是导出，这里就是导入了，将外来的json导入为dict

print(new\_names)

print(type(new\_names))

print(res)

print(type(res))

f=open('a.txt','w',encoding='utf-8')

json.dump(names,f,ensure\_ascii=False)#将字典写入文件为json格式，不需要用write，直接json.dump(dict\_name,file\_name,ensure\_ascii=False)即可

注意ensure\_ascii=False是必须要的，json.dump()默认的是ascii编码，并不认识中文，需要关闭这个ascii编码

f=open('a.txt',encoding='utf-8')

res=json.load(f)#将json文件内容转换成字典格式，不需要用read，直接json.load(file\_object)即可

print(res)

print(type(res))

**6.mysql模块**

import pymysql

from pymysql.cursors import DictCursor

coon=pymysql.connect(host='localhost',port=3306,user='root',password='root',db='test',charset='utf8')

cur=coon.cursor(DictCursor)

cur.execute('select \* from stu')

res=cur.fetchall()

print(res)

for i in res:

print(i)

cur.close()

coon.close()