博客地址：<http://www.cnblogs.com/znyyy/p/8067531.html>

文件操作其实和我们日常处理文件一样的，先打开文件，然后操作，最后保存关闭，在python里就是这三步骤：

　　1、打开文件获取文件的句柄，句柄就理解为这个文件  
　　2、通过文件句柄操作文件  
　　3、关闭文件

**文件操作有以下三个模式：**

　　r：读模式【可读； 不可写，不存在则报错】  
　　w：写模式【不可读；不存在则创建；存在则删除内容】　　//注意：只要写了w模式，不论后面跟的啥，文件不存在则创建，存在则清空文件内容，然后再写。  
　　a：追加模式【不可读； 不存在则创建；存在则只追加内容】//注意：追加内容是在最后的指针位置开始写，所以一般这样的话，要先f.seek(0),回到文件的第一行第一个字节的位置。

**在模式后面跟上'+'号，则表示同时可以读写文件：**

　　r+：【可读、可写；可追加，如果打开的文件不存在的话，会报错】  
　　w+：【写读模式，使用w+的话，已经存在的文件内容会被清空，可以读到已经写的文件内容】  
　　a+：【追加读写模式，不存在则创建；存在则只追加内容；】

**在模式后面跟上'b'，则表示处理二进制文件：rb,wb,ab**

**1.读取文件**

f=open('a.txt','r',encoding='utf-8')#mac下不需要设置字符类型，操作系统默认字符类型是utf-8；windows需要设置这个字符类型，系统默认的是gbk字符类型

print(f.read())#读取文件内容，返回的是一个字符串，当文件太大时，不建议这样操作，防止内存撑爆

print(f.readlines())#读取文件内容，返回的是一个list，list每个元素是文件的每一行

print(f.readline())#每次只读取一行

f.close()

#for循环文件句柄是通过行循环文件的内容，这样做的好处是当文件太大时，防止内存撑爆

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

print(f.read())

i=1

for content in f:

print('第{line}行:{content}'.format(line=i,content=content))

i+=1

f.close()

**2.写文件**

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

f.write('hahaha')#只能写字符串,不能是列表

f.writelines(['aaa','bbb','ccc'])#写的是list

f=open('a.txt','r+',encoding='utf-8')#r+读写模式,只要有r权限在，当文件不存在时都会报错

f.write('1111')#读写模式下，写数据的时候会从第一行开始写，替换之前文件里第一行的内容前n个字符(n为写入的字符长度)

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

f.write('asd')

f.seek(0)#seek(n),移动指针到n位置，0为首位

print(f.read())

**3.追加模式**

追加模式跟上面的差不多，要注意的地方是：追加内容是在最后的指针位置开始写，所以一般这样的话，要先f.seek(0),回到文件的第一行第一个字节的位置，要时刻关注指针的位置。

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

f.seek(0)

print(f.read())

i=1

for content in f:

print('第{line}行:{content}'.format(line=i,content=content))

i+=1

f.close()

**4.with方法操作文件**

with方法操作文件，可以少些close文件关闭方法，这个with会自动关闭文件，不需要我们自己再重复写了，其他用法一样。

with open('a.txt','r') as f:

for line in f:

print(line)

**5.文件修改**

有一个细节，日常修改文件的时候，打开文件的时候，会产生一个隐藏文件，其实我们修改的内容过程是这样的：打开文件->在隐藏的那个文件上增删改文件内容->保存隐藏文件->之前的文件删除->隐藏文件重命名开始的文件名称，是这样一个过程，那么我们要修改文件时，其实是一样的：

　　1.读模式打开待修改的文件A  
　　2.修改读取到的内容  
　　3.将修改后的内容写进新的文件B  
　　4.删除文件A  
　　5.将文件B重命名为A

import os

with open('阳光总在风雨后.txt','r',encoding='utf-8') as f1,open('a.txt','w',encoding='utf-8') as f2:

for line in f1:

new\_line=line.replace('阳光','彩虹')

f2.write(new\_line)

os.remove('阳光总在风雨后.txt')

os.rename('a.txt','阳光总在风雨后.txt')

　　当然还有另外一种方法，不需要删文件，直接清空文件即可

f1=open('阳光总在风雨后.txt','a+',encoding='utf-8')

f1.seek(0)

res=f1.read()

new\_content=res.replace('彩虹','阳光')

f1.seek(0)

f1.truncate()

f1.write(new\_content)

f1.close()

**6.集合**

　　集合也是数据类型，，一个类似列表东西，它的特点是无序的，不重复的，也就是说集合中是没有重复的数据，作用如下：

　　1、它可以把一个列表中重复的数据去掉，而不需要你再写判断  
　　2、可以做关系比较，比如数学里的交集、并集等

　　集合的操作和列表差不多：

lis=[1,2,3,4,4,5,5,6,7,8,9]

myset=set(lis) #这就定义了一个集合

print(myset) #去掉了重复数据，4,5，这里只显示一次

myset.add(888)#添加元素

myset.update([777,666,666]) #添加值

myset.remove(777)#删除元素，如果元素不存在会报错

myset.pop()#删除一个随机的元素，并返回删除的元素

myset.discard('dddd')#如果删除的元素存在，删除，不存在不做处理

　　集合操作方法：

set1 = {1, 2, 3, 4, 5, 6, 7,8,9}

set2 = {1, 2, 3, 4, 6}

set3 = {1, 2}

print(set1.intersection(set2)) # 取交集，也就是取set1和set2中都有的

print(set1 & set2)# 取交集

print(set1.union(set2)) # 取并集，也就是把set1和set2合并了，然后去除重复的

print(set1 | set2)# 取并集

print(set1.difference(set2)) #取差集 在list中存在，在set2中没有的

print(set1 - set2)

print(set3.issubset(set1))#判断set3是不是set1的子集

print(set1.issuperset(set3))#判断set1是不是set3的父集

print(set1.isdisjoint(set3))#判断set1和set3是否有交集

print(set1.symmetric\_difference(set2))#对称差集，输出两个列表中都没有的值，也就是把两个集合中相同的去掉

print(set1 ^ set2)

**7.写一个监控服务器的脚本，每分钟运行一次，这分钟内如果ip访问次数超过200，则计入黑名单。**

　　log日志文件格式如下：

　　58.19.57.99 - - [04/Jun/2017:05:23:30 +0800] "GET / HTTP/1.0" 302 161 "-" "Wget/1.12 (linux-gnu)" "-"  
　　58.19.57.99 - - [04/Jun/2017:05:23:30 +0800] "GET /blog HTTP/1.0" 301 233 "-" "Wget/1.12 (linux-gnu)" "-"

import time

pin=0

while True:

with open('access.log','rb') as f:

ip\_list=[]

f.seek(pin)

for line in f:

ip=line.decode().split()[0]

print(ip)

ip\_list.append(ip)

for ip in set(ip\_list):

if ip\_list.count(ip)>=200:

print('黑名单：%s，访问次数：%d'%(ip,ip\_list.count(ip)))

pin=f.tell()

print(pin)

time.sleep(5)