

# 文件操作

## 引言

- 到目前为止，我们做的一切操作，都是在内存里进行的，这样会有什么问题吗？如果一旦断电或发生意外关机了，那么你辛勤的工作成果将瞬间消失。是不是感觉事还挺大的呢？现在你是否感觉你的编程技巧还缺了点什么？是的，我们还缺少将数据在本地文件系统进行持久化的能力，白话讲就是文件的读写能力。

## 文件打开

- Python内置了一个open()方法，用于对文件进行读写操作。使用open()方法操作文件就像把大象塞进冰箱一样，可以分三步走，一是打开文件，二是操作文件，三是关闭文件。
- 文件句柄/文件描述符
  - open()方法的返回值是一个file对象，可以将它赋值给一个变量，这个变量就是所谓的文件句柄。
  - file对象：
    - 可以调用read()和write()方法，对打开的文件进行读写操作。

- open方法的语法

- `f = open(filename, mode)`

- filename:

- 一个包含了你要访问的文件名称的字符串值，通常是一个文件路径。

- 文件路径作用：定位到指定文件

- mode:

- 打开文件的模式，有很多种，默认是只读方式r。

- 文件打开的模式：

- 

模式	操作	说明
r	只读	默认模式，如果文件不存在就报错，存在就正常读取。
w	只写	如果文件不存在，新建文件然后写入；如果存在，先清空文件内容，再写入。
a	追加	如果文件不存在，新建文件，然后写入；如果存在，在文件的最后追加写入。
x	新建	如果文件存在则报错，如果不存在就新建文件，然后写入内容，比w模式更安全。
b	二进制模式	比如rb、wb、ab，以bytes类型操作数据
+	读写模式	比如r+、w+、a+

- 常规文件打开模式操作演示

- b模式：

- 二进制模式，通常用来读取图片、视频等二进制文件。注意，它在读写的时候是以bytes类型读写的，因此获得的是一个bytes对象而不是字符串。在这个读写

过程中，需要自己指定编码格式。在使用带b的模式时一定要注意传入的数据类型，确保为bytes类型。

- +模式：

- 对于w+模式，在读写之前都会清空文件的内容，建议不要使用！
- 对于a+模式，永远只能在文件的末尾写入，(常用)！
- 对于r+模式，也就是读写模式，配合seek()和tell()方法，可以实现更多操作。

- 编码问题

- 要读取非UTF-8编码的文件，需要给open()函数传入encoding参数，例如，读取GBK编码的文件
- 遇到有些编码不规范的文件，可能会抛出UnicodeDecodeError异常，这表示在文件中可能夹杂了一些非法编码的字符。遇到这种情况，可以提供errors='ignore'参数，表示如果遇到编码错误后如何处理。

## 文件对象操作

- 每当我们用open方法打开一个文件时，将返回一个文件对象。这个对象内置了很多操作方法。
- f.read(size) #size读取数据的个数

- 读取一定大小的数据, 然后作为字符串或字节对象返回。  
size是一个可选的数字类型的参数, 用于指定读取的数据量。当size被忽略了或者为负值, 那么该文件的所有内容都将被读取并且返回。
- 注意:
  - 如果文件体积较大, 请不要使用read()方法一次性读入内存, 而是read(512)这种一点一点的读。
- f.readline()
  - 从文件中读取一行n内容。换行符为'\n'。如果返回一个空字符串, 说明已经已经读取到最后一行。这种方法, 通常是读一行, 处理一行, 并且不能回头, 只能前进, 读过的行不能再读了。
- f.readlines()
  - 将文件中所有的行, 一行一行全部读入一个列表内, 按顺序一个一个作为列表的元素, 并返回这个列表。  
readlines方法会一次性将文件全部读入内存, 所以也存在一定的风险。但是它有个好处, 每行都保存在列表里, 可以随意存取。

#需求：读取文件中的数据

```
fp = open( './test.txt', 'r' )
text = fp.read(10) #读取指定字节的数据
text_line = fp.readline() #一次读取一行数据
text_lines = fp.readlines() #读取多行数据，返回一个列表
print(text_lines)
fp.close()
```

- 总结：
  - 几种不同的读取和遍历文件的方法比较：如果文件很小，read()一次性读取最方便；如果不能确定文件大小，反复调用read(size)比较保险；如果是配置文件，调用readlines()最方便。普通情况，使用for循环更好，速度更快。
- f.write()
  - 将字符串或bytes类型的数据写入文件内。write()动作可以多次重复进行，其实都是在内存中的操作，并不会立刻写回硬盘，直到执行close()方法后，才会将所有的写入操作反映到硬盘上。在这过程中，如果想将内存中的修改，立刻保存到硬盘上，可以使用f.flush()方法。

```
fp = open( './test123.txt', 'w' )  
fp.write( 'hello bobo' )  
fp.close() #将文件内容清空，在写入新数据
```

```
fp = open( './test123.txt', 'a' )  
fp.write( 'hello bobo' )  
fp.close() #在文件数据末尾追加数据
```

- f.close()
  - 关闭文件对象。当处理完一个文件后，调用f.close()来关闭文件并释放系统的资源。文件关闭后，如果尝试再次调用该文件对象，则会抛出异常。忘记调用close()的后果是数据可能只写了一部分到磁盘，剩下的丢失了，或者更糟糕的结果。也就是说大象塞进冰箱后，一定不要忘记关上冰箱的门。

## with关键字

with关键字用于Python的上下文管理器机制。为了防止诸如open这一类文件打开方法在操作过程出现异常或错误，或者最后忘了执行close方法，文件非正常关闭等可能导致文件泄露、破坏的问题。Python提供了with这个上下文管理器机制，保证文件会被正常关闭。在它的管理下，不需要再写close语句。注意缩进。

```
with open( './test123.txt', 'r') as fp: #fp =  
open()
```

```
    text = fp.read(5)  
print(text)
```

#上下两组代码功效一样

```
fp = open( './test123.txt', 'r')  
text = fp.read(5)  
print(text)  
fp.close()
```

- 对图片，音频，视频，压缩包等二进制的数据进行文件读写操作
  - 实现一个图片文件的拷贝
    - 1.打开一个图片文件，读取其二进制的数据
    - 2.将读取到的数据写入到另一个路径下

```
fp = open('bobo.jpg', 'rb') #r是读取文本数据，rb是读取二进制数据
img_data = fp.read() #读取到了图片的二进制数据

#将图片的二进制数据写到另一个路径下
fp1 =
open('/Users/zhangxiaobo/Desktop/bobo123.jpg', 'wb')
fp1.write(img_data)

fp.close()
fp1.close()
```

## 练习：登录注册

'''

需求：注册功能

1. 让用户录入用户名，密码，重复密码，手机号基本信息
2. 检测两次输入的密码是否一致，一致则注册成功，否则注册失败
3. 注册成功后需要将用户信息存储到文件中进行保存

'''



```
def regist():
    #让用户录入用户名, 密码, 重复密码, 手机号基本信息
    print('-----欢迎来到注册页面-----')
    username = input('输入注册的账号: ')
    password = input('输入注册的密码: ')
    repeatpwd = input('重复密码:')
    phonenumber = input('手机号: ')
    #检测两次输入的密码是否一致, 一致则注册成功, 否则注册失败

    if password == repeatpwd: #==可以判断两个字符串是否一样:
        #两次密码一致: 注册成功, 将用户信息写入到文件保存

        with open('userData.txt', 'a+') as fp:
            #\n表示回车换行的意思
            fp.write(username+'-'+password+'-'+
'+phonenumber+'\n')
            print('恭喜您注册成功, 账号为: ', username)

    else: #两次密码不一致, 注册失败
        print('两次密码不一致, 注册失败, 请重新注册! ')
```

```
'''
```

登录功能：

1. 让用户录入用户名，密码
2. 验证登录状态

```
'''
```

```
def login():
```

```
    #1. 让用户录入用户名，密码
```

```
    print('-----欢迎来到登录页面-----')
```

```
    username = input('请输入登录的账号: ')
```

```
    password = input('请输入登录的密码: ')
```

```
    #2, 验证登录状态
```

```
    #读取文件中的用户信息的数据
```

```
    with open('userData.txt', 'r') as fp:
```

```
        #[bobo-123-222, jay-456-666]
```

```
        user_data_list = fp.readlines() #读取每一  
行数据，返回的是一个列表
```

```
        for data in user_data_list:
```

```
            data = data.strip() #取出字符串中的回车
```

```
            #切分字符串，取出用户名和密码即可
```

```
            #s_data = [bobo, 123, 222]
```

```
            s_data = data.split('-')
```

```
            #从文件中获取的用户名和密码
```

```
            f_username = s_data[0]
```

```
            f_password = s_data[1]
```

```
            #判断登录状态
```

```

        flat = 0 #如果flat=0表示登录失败, 为1表示登录成功

        if username == f_username and password == f_password:
            flat = 1 #表示登录成功
            break
        if flat == 1:
            print('登录成功')
        else:
            print('登录失败')

login()
# regist()#调用该函数完成注册功能

```

## 管理系统完整实现(自己拓展:了解):

```

import time
def regist():
    print('\n\n欢迎来到注册页面')
    print('-----')
    username = input('enter username:')
    password = input('enter password:')
    repeatpwd = input('repeat enter password:')

```

```

email = input('enter email:')
#判定用户的注册状态
if password == repeatpwd:
    isHave = False#用户名是否重复的状态显示,
isHave==True表示用户名重复

    with open('./userData.txt','r+') as fp:
        #bobo-123-123@qq.com
        #jay-456-456@qq.com
        #读取文件中所有注册用户的用户名
        user_data_list = fp.readlines()
        for user_data in user_data_list:
            userName = user_data.split('-')

[0]

            if username == userName:#用户名重
复

                isHave = True
            if isHave == False:
                fp.write(username+'-
'+password+'-'+email+'\n')
                print('注册成功,三秒后自动跳转到登录
页面.....')
                time.sleep(3) #暂停三秒
                login()
            else:
                print('注册失败, %s用户名已被占
用! '%username)
                print('是否重新注册? y/n')

```

```
        c = input('enter your choose:')
        if c == 'y' or c == 'Y':
            regist()
        else:
            exit()
    else:
        print('两次密码不一致，注册失败！')
def insertBook():
    title = input('插入图书的名称:')
    publish_date = input('插入图书的出版时间:')
    author = input('插入图书的作者:')
    publish = input('插入图书的出版社:')
    with open('./bookData.txt', 'a') as fp:
        fp.write(title+'-'+publish_date+'-'
'+author+'-'+publish+'\n')
    print(title, '插入成功！')
    print('是否继续插入？ y/n')
    c = input('enter your choose:')
    if c == 'y' or c == 'Y':
        insertBook()
    else:
        manageView()
def showBooks():
    print('\n\n')
    with open('./bookData.txt', 'r') as fp:
        book_list = fp.readlines()
        for book in book_list:
```

```
        print(book)
c = input('返回主页面请按1:')
if c == '1':
    manageView()
def deleteBook():
    #指定一个删除的条件
    name = input('请输入即将要删除的图书名称:')
    with open('./bookData.txt', 'r') as fp:
        book_list = fp.readlines()
    for book in book_list:
        if name in book:
            book_list.remove(book)

    fp = open('./bookData.txt', 'w')
    for book in book_list:
        fp.write(book)
    fp.close()
    print(name, '被成功删除! 3秒后自动跳转会主页面.....')
    time.sleep(3)
    manageView()
def updateBook():
    name = input('请输入即将要修改图书的名字:')
    with open('./bookData.txt', 'r') as fp:
        book_list = fp.readlines()
    for book in book_list:
        if name in book:
```

```
        old_author = book.split('-')[2]
        new_author = input('请输入新的作者名
称:')
```

```
        new_book =
book.replace(old_author,new_author)
        book_list.append(new_book)
        book_list.remove(book)
        break
fp = open('./bookData.txt','w')
for book in book_list:
    fp.write(book)
print('图书修改成功! 等待3秒返回主页面.....')
fp.close()
time.sleep(3)
manageView()
```

```
def manageView():
    print('\n\n欢迎来到管理系统')
    print('1.查看所有图书\n2.插入图书\n3.删除图书
\n4.修改图书\n5.返回主页面\n')
    choose = input('enter your choose:')
```

```

    if choose == '1':
        showBooks()
    elif choose == '2':
        insertBook()
    elif choose == '3':
        deleteBook()
    elif choose == '4':
        updateBook()
    else:
        main_view()
def login():
    print('\n\n欢迎来到登录页面')
    print('-----')
    username = input('enter username:')
    password = input('enter password:')
    #判定登录状态
    login_state = False
    with open('./userData.txt','r') as fp:
        user_data_list = fp.readlines()
        for user_data in user_data_list:
            un = user_data.split('-')[0]
            pw = user_data.split('-')[1]
            if username == un and password ==
pw:
                login_state = True
                break

```



```

        if login_state == True:
            manageView()
        else:
            print('登录失败,是否返回主页面? y/n')
            c = input('enter your choose:')
            if c == 'y' or c == 'Y':
                main_view()
            else:
                exit()

def findPwd():
    un = input('enter your username:')
    em = input('enter your email')
    with open('./userData.txt', 'r') as fp:
        isFind = False
        user_data_list = fp.readlines()
        for user_data in user_data_list:
            username = user_data.split('-')[0]
            email = user_data.split('-')[2]
            #将email中的回车去掉
            email = email.strip()
            if username == un and email == em:
                password = user_data.split('-')

[1]
                print('找回的密码为: ', password)
                isFind = True
                break

        if isFind == False:

```

```
        print('密码找回失败，用户名和邮箱不匹配！')
```

```
def main_view():
```

```
    print('欢迎来到xxx管理系统')
```

```
    print('-----')
```

```
    print('1.登录\n2.注册\n3.找回密码\n4.退出')
```

```
    choose = input('请选择功能:')
```

```
    if choose == '1':
```

```
        login()
```

```
    elif choose == '2':
```

```
        regist()
```

```
    elif choose == '3':
```

```
        findPwd()
```

```
    elif choose == '4':
```

```
        return #退出整个程序
```

```
    else:
```

```
        print('输入有误，程序退出!')
```

```
main_view()
```