Day 6 文件处理与编码转换

一、Python文件操作

对文件的操作有2种，文本文件、二进制文件（视频、图片等）

1.1 open方法基本使用

```
open(file, mode='r', encoding=None):
```

几种打开模式

```
'r'       open for reading (default)
'w'       open for writing, truncating the file first (写模式，如果文件 在，先清空【危险】)
'x'       create a new file and open it for writing (创建模式：如果文件在，会报错)
'a'       open for writing, appending to the end of the file if it exists (日志)
'b'       binary mode (2进制模式)
't'       text mode (default)  (文本)
'+'       open a disk file for updating (reading and writing)
```

**The default mode is 'rt' (open for reading text).*

一个文件对象被open方法创建后，这个对象可用的有下面这些，我们先大体过一下，后边还会细讲每个方法：

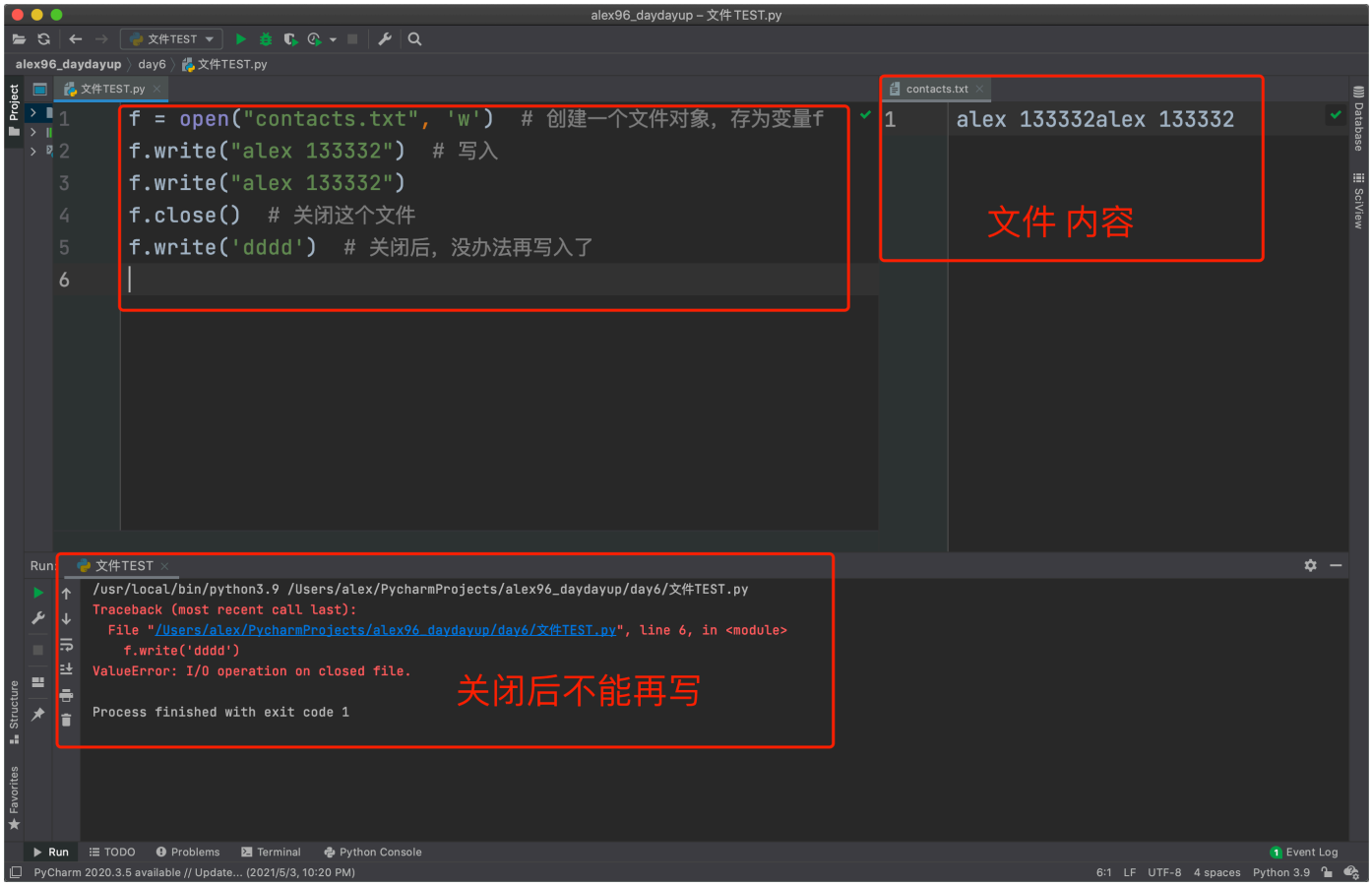
```
close    关闭文件
closed   查看文件是否已关闭
encoding 返回文件的编码
flush    把缓存里的写入的数据强制刷新硬盘
isatty   返回文件是否是'interactive'数据流,比如是个命令行终端，（在unix系统，一切皆文件）
mode     返回当前文件模式
name     返回文件名
read     读指定长度的内容，f.read(1024) 读1024字节，不指定参数的话，就读所有内容
readable 文件是否可读
readline 读一行
readlines 读所有，每行列表形式返回
seek     把光标移到指定位置
seekable 该文件光标是否可移动
tell     返回当前光标位置
truncate 截断文件，f.truncate(100)，从文件 开头截断100个字符，后边的都扔掉
writable 是否可写
```

```
write       写内容
writelines  把一个列表写入，每个元素是一行
```

先学会增删改查，再学它的其它各种用法

1、创建模式：创建文件

```
f = open("contacts.txt", 'w')  # 创建一个文件对象（文件句柄），存为变量f
f.write("alex 133332")  # 写入
f.write("alex 133332")
f.close()  # 关闭这个文件
f.write('dddd')  # 关闭后，没办法再写入了
```



注意：在`w`模式是创建一个文件，但若该文件 已存在，则会清空原文件，如果不行清空原文件，安全起见，用`x`模式，若原文件存在，会报错，不会直接清空它

```
Traceback (most recent call last):
  File "/Users/alex/PycharmProjects/alex96_daydayup/day6/文件TEST.py", line 1, in <module>
    f = open("contacts.txt", 'x')  # 创建一个文件对象，存为变量f
FileExistsError: [Errno 17] File exists: 'contacts.txt'
```

写入多行

生成一个随机密码文件

```
import random
import string

names = ["alex",'jack','rain','black_girl','peiqi']

f = open("password.txt", 'w')  # 创建一个文件对象，存为变量f
for i in names:
    passwd = random.sample(string.ascii_letters + string.digits, 8) # 生成8位随机字符
    line = f"{i}:{''.join(passwd)}\n" # 一定要加上\n换行符号才会换行
    f.write(line)
f.close()
```

password.txt

```
alex:amZsGdeM
jack:dGlMNgAB
rain:70E2crty
black_girl:SlH85rjp
peiqi:ZeCtVI4g
```

1.2 读模式：循环读取文件&查找

对`model_contacts.txt`进行循环

```
姓名          地区     身高     体重     电话
况咏蜜       北京     171     48     13651054608
王心颜       上海     169     46     13813234424
马纤羽       深圳     173     50     13744234523
乔亦菲       广州     172     52     15823423525
罗梦竹       北京     175     49     18623423421
刘诺涵       北京     170     48     18623423765
岳妮妮       深圳     177     54     18835324553
贺婉萱       深圳     174     52     08933434452
叶梓萱      上海      171     49      18042432324
杜姗姗     北京       167     49       13324523342
```

按行读取&循环

```
f = open("model_contacts.txt") # 默认`rt`模式
print(f.readline()) # 读第1行
print(f.readline()) # 读第2行
print(f.readline()) # 读第3行

print('----循环读后面的-----')

for line in f:
    print(f.readline())
```

打开文件后，光标位置在文件开头，每读一行，光标向下移动一行，因此可以一行行的往后读，已读了的内容不会重复被读取

读取指定字符个数

```
f = open("model_contacts.txt",) # 默认`rt`模式
print(f.read(3)) # 读取3个字符
```

注意：在文本模式下，这个3是代表3个字符，在2进制模式，这个3是指3个字节

文件里查找内容

要想在文件内找某个词，并打印出所在行，只能循环一遍文件，每行依次判断一下

```
f = open("model_contacts.txt")

for line in f:
    if "梦竹" in line:
        print(line)
```

1.3 追加模式

只会往文件最后追加，一般用于写日志的场景

```
f = open("model_contacts.txt", "a")

f.write("黑姑娘    北京    165    50      18834252322\n")
f.write("黑姑娘2   北京    165    50      18834252322\n")
```

注意： 追加模式，即使通过f.seek()把光标移到其它位置 ， 再f.write()的时候 ， 依然是写到最后的、

但是f.see(10),然后再f.truncate()，会实现文件截断，只保留10个字符。

1.4 修改

r+是修改模式

直接调用f.write(),会从开头开始写, 然后会往后覆盖... 如果只覆盖了后边某个文字的一半，就会出现 乱码

seek是移动字节

tell返回字节数

想把数据源里的`伍小仪`换成`刘翠花TracyLiu`, 怎么搞?

```
马纤羽      深圳    173    50    13744234523
乔亦菲      广州    172    52    15823423525
伍小仪      深圳    175    49    18623423421
刘诺涵      北京    170    48    18623423765
岳妮妮      深圳    177    54    18835324553
```

如果能移动到伍小仪的位置，直接从那个位置替换整行是可以的。

但问题就在于，你不知道她所在的位置， 注意， `f.seek(100)` 移动到第100个字节的位置， 但是你的文件 是utf8的编码，每个中文占3个字符，里面还掺杂了数字、空格、换行，你不可能一个个的字符换算成字节的长度的。

你说我通过循环到这一行，然后 `f.tell()` 不就可以知道她的位置 了么?

理论上可以，但实际也不行，循环时， `f.tell()` 被禁用啦,(因为文件循环是用的迭代器写的,这个后边会学)

所以怎么办呢?

你只能，把文件先读到内存，在内存里改掉这个数据，然后重写回文件 。

```
f = open("model_contacts.txt", "r+")

data = f.read() # 先把文件内容全读到内存
data_new = data.replace("伍小仪","刘翠花TracyLiu") # 替换
f.seek(0) # 光标移到0
f.truncate()  # 清空
f.write(data_new) # 写入新内容

f.close()
```

1.5 删除

```
import os
os.remove("model_contacts.txt")
```

1.6 处理不同编码的文件

我的 电脑 是Mac, 系统 默认编码是utf-8, open方法会默认用utf-8格式打开所有文件 。

但windows是gbk, 若收到别人从windwos上发一个文件，是gbk编码的，我直接打开会出现什么情况?

```
f = open("from_win/file_gbk2.txt",'r')
print(f.read())
```

```
# /usr/local/bin/python3.9 /read_windows_file.py
Traceback (most recent call last):
  File "/Users/alex/PycharmProjects/alex96_daydayup/day6/read_windows_file.py", line 3, in
<module>
    print(f.read())
  File "/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/codecs.py", line 322, in
decode
    (result, consumed) = self._buffer_decode(data, self.errors, final)
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xb7 in position 2: invalid start byte

Process finished with exit code 1
```

提示说用utf-8解析不了这个文件内容, why ?

因为open方法默认用系统的编码来解析文件 ， 我的mac又是utf-8, 所以它尝试用utf-8来解码这个gbk的文件内容， 肯定就报错了呀。

```
encoding is the name of the encoding used to decode or encode the
file. This should only be used in text mode. The default encoding is
platform dependent, but any encoding supported by Python can be
passed.  See the codecs module for the list of supported encodings.
```

那可怎么办呢？ 有同学说， 只需要在文件头声明改成gbk不就行了么?

```
# -*- encoding:gbk -*-
f = open("from_win/file_gbk2.txt",'r')
print(f.read())
```

发现还是报同样的错呀， 为何？ 并且 为何报的还是`'utf-8' codec can't decode byte 0xb7 in position 2: invalid start byte` ? 这不都gbk了么?

注意，**这里小白极容易混淆**，要划重点。。。

代码文件开头的编码声明，只是告诉py解释器，要用什么编码来解释这个代码文件 。

它跟open方法没关系 ， open是读另外一个文本文件，如果被读的文件，编码与系统编码不一致，就会处理不了。

处非，你给open方法也指定 `encoding="gbk"` ,这样，就会按gbk来读文件内容

```
f = open("from_win/file_gbk2.txt",'r',encoding="gbk")
print(f.read())
```

1.6 二进制模式操作文件

遇到图片、视频等非文本文件， 该如何处理呢?

直接用文本模式操作的话，会报错

二进制模式操作文件.py

```
9
10  f = open('qrcode.png','r')
11  for line in f:
12      print(line)
```

Run: 二进制模式操作文件

```
/usr/local/bin/python3.9 /Users/alex/PycharmProjects/alex96_daydayup/day6/二进制模式操作文件.py
Traceback (most recent call last):
  File "/Users/alex/PycharmProjects/alex96_daydayup/day6/二进制模式操作文件.py", line 11, in <module>
    for line in f:
  File "/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/codecs.py", line 322, in decode
    (result, consumed) = self._buffer_decode(data, self.errors, final)
UnicodeDecodeError: 'utf-8' codec can't decode byte 0x89 in position 0: invalid start byte

Process finished with exit code 1
```

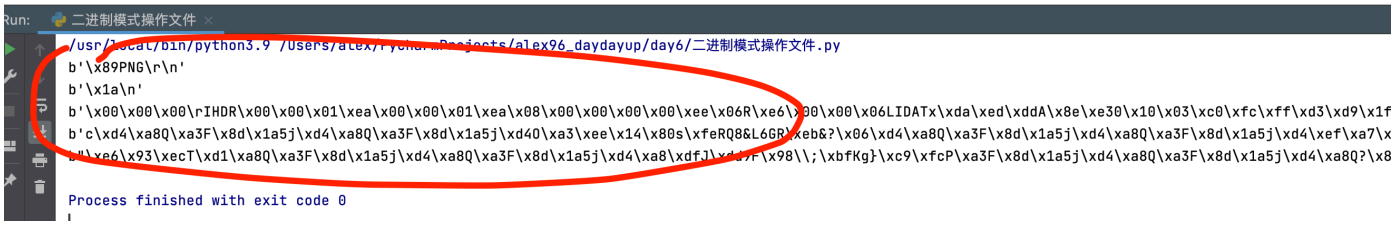
因为什么utf-8、gbk呀，只是文本的编码，肯定不适用于图片。

二进制读文件

所以处理非文本文件，就可直接用二进制模式，以2进制制形式读取数据，不进行解码...

```python
f = open('qrcode.png','rb')
for line in f:
    print(line)
```

for line in f



打印这个图片的内容看到， 为什么这么多以 `b` 开头的字符串是什么东西?

字节类型

这个就是python专门为显示2进制数据设计的一个专门数据类型，叫 `字节类型` ，直接给你打印一堆0101010你啥也看不出来。。 所以就用16进制+杂交显示啦。

二进制写文件

当以2进制模式创建一个文件时，那你只能写二进制数据，不能写文本啦

```python
f = open("file_gbk.txt", mode="wb")
s = "路飞学城"
f.write(s)
```

```
Run:  二进制模式操作文件
/usr/local/bin/python3.9 /Users/alex/PycharmProjects/alex96_daydayup/day6/二进制模式操作文件.py
Traceback (most recent call last):
  File "/Users/alex/PycharmProjects/alex96_daydayup/day6/二进制模式操作文件.py", line 17, in <module>
    f.write(s)
TypeError: a bytes-like object is required, not 'str'
```

那文本字符，怎么转成bytes类型呢? 进行一下编码就可以了，

目前在内存里是unicode格式， 存到文件里，就要转成utf-8或gbk...

```python
s = "路飞学城"
s_utf8 = s.encode("utf-8")  # 编码成utf-8
s_gbk = s.encode("gbk")  # 编码成gbk
print("utf8:", s_utf8)
print("gbk :", s_gbk)

print("由gbk解码回unicode:", s_gbk.decode("gbk")) # 必须告诉解释器，文本当前编码是什么，才能按相应的编码转回来

```

```
Run:  二进制模式操作文件
  /usr/local/bin/python3.9 /Users/alex/PycharmProjects/alex96_daydayup/day6/二进制模式操作文件.py
  utf8: b'\xe8\xb7\xaf\xe9\xa3\x9e\xe5\xad\xa6\xe5\x9f\x8e'
  gbk : b'\xc2\xb7\xb7\xc9\xd1\xa7\xb3\xc7'
  由gbk解码回unicode: 路飞学城
```

编码转换

在什么情况下会用到编码转换呢?

以后我们在开发网络程序时，如果远程一台电脑发送过来的消息是GBK的， 你的电脑是utf-8的，你想把对方的消息正确显示，就要按gbk来 `decode` 解码.