【白话python连载（9.1）】使用python处理文件--听说读写谁与争锋

* python处理文件的一般方法

现代操作系统不允许普通的程序直接操作磁盘，所以，

程序是无法直接读写磁盘中的文件，而是请求操作系统提供数据操作接口，为我们生成一个文件对象，从而获取或写入需要的数据。Open打开文件的常规的语法为open(' **D:/try/merge\_result.csv'**, 'r', encoding='gbk', errors='ignore'），

**文件读取第一个参数：读取位置的设置（位置参数，必须指定）：**

可以使用绝对位置：以path = 'D:/try/merge\_result.csv'为例，特别需要注意的是python中对文件位置分隔符正斜杠’/和反斜杠’\’的使用。因为python中反斜杠‘\’有转义关键字符的功能。

|  |  |
| --- | --- |
| 文件位置 | 分割方式 |
| path1= 'D:/try/merge\_result.csv' | 直接使用正斜杠进行文件位置分割 |
| path2 = r'D:/try/merge\_result.csv' | 使用r，表示获取原声字符。忽略/的转义作用 |
| path3 = 'D://try//merge\_result.csv' | 使用双斜杠，进行转义表示单反斜杠 |
| path1= 'd:/try/merge\_result.csv' | D盘文件地址的大小写，不会影响定位到文件位置 |

**文件读取第二个参数：读取方式的设置（位置参数，必须指定）：**

|  |
| --- |
| **open打开文件方式的相关参数设置，以常规的utf-8格式进行读取** |
| r，表示以只读的方式打开获取文件的内容，此时若文件不存在则会报错 |
| w，表示以只写的方式打开获取文件的内容，用于写入数据，若文件不存在则会自动创建文件 |
| a，表示在写入文件时以尾部追加的方式进行，以实例中 |
| **open打开文件方式后附加内容（r，w，a）** |
| rb，wb，ab，以二进制的模式进行数据读写 |
| r+，w+，a+，都可以实现文件的读写操作，只是文件的指针位置不同，r+打开已经存在的文件进行读写，操作指针在文件头部；w+，创建或覆盖一个文件进行读写；a+，创建或追加内容在一个文件中，追加的话文件指针在文件末尾。 |
| rb+，wb+，ab+，同理这些是以二进制的模式进行读写数据。 |

**文件读取第三个参数和第四个参数：对文件编码方式的调整（命名参数，有默认值）：**

encoding指定读取文件的编码方式，errors='ignore'则是忽略文件中出现的编码不一致问题。

**对文件的常用操作：**

|  |  |
| --- | --- |
| 获取文件对象：f=open(' D:/try/merge\_result.csv', 'r'） | |
| f.read(size) | 默认以字符串形式返回文件的所有内容，可以指定读取的文件的内容大小 |
| f.readline() | 按行读取文件内容，返回文件的第一行内容 |
| f.readlines(size) | 返回文件的多行内容，默认**通过迭代器**返回所有行，使用循环获取。 |
| f.write(‘file\_data’) | 以字符串形式，在文件中写入数据 |
| f.writelines(‘file\_data’) | 以字符序列的格式，在文件中写入数据 |
| f.close() | 关闭打开的文件，在每次操作结束后都应调用 |

**为了简化操作文件方式：（可以同时实现文件的打开，读写，关闭操作）**

with open('/Users/michael/test.txt', 'w') as f:

f.write('Hello, world!')

这样操作也可以防止在实际使用中忘记使用调用close（）方法

读写文件就是请求操作系统打开一个文件对象（通常称为文件描述符），然后，通过操作系统提供的接口从这个文件对象中读取数据（读文件），或者把数据写入这个文件对象（写文件）。

\_\_author\_\_ **= 'ziyi'***#!/usr/bin/python  
# -\*- coding: utf-8 -\*-***import** os  
**import** pandas **as** pd  
**import** glob  
**import** csv  
  
**def merge\_csv**(*file\_address*)**:** csv\_list **=** glob.**glob**(*file\_address* **+ '\*.csv'**)  
 print(**'该文件下下一共有'**, len(csv\_list),**'个csv文件需要合并'**)  
 **for** one\_csv **in** csv\_list**:** print(one\_csv)  
 *# read方法是以字符串形式获取内容* one\_open **=** open(one\_csv,**'r'**).**read**()  
 **with** open(**'D:/try/merge\_result.csv'**,**'a'**) **as** f**:** f.**write**(one\_open)  
 print(**'合并{}个文件完成'**.**format**(len(csv\_list)))  
 **return 'D:/try/merge\_result.csv'  
  
def drop\_duplicate**(*file*)**:** df **=** pd.**read\_csv**(*file*)  
 datalist **=** df.**drop\_duplicates**()  
 datalist.**to\_csv**(*file*,header**=False**)  
 print(**'去重操作完成'**)  
  
**if** \_\_name\_\_ **== '\_\_main\_\_':** file\_address **= 'D:\\test\_accounts1\\'** merge\_result **= merge\_csv**(file\_address)  
 **drop\_duplicate**(merge\_result)

"D:\anaconda python\python3.6.exe" D:/vernacular/coding\_ziyi/merge\_csv.py

该文件下下一共有 6 个csv文件需要合并

D:\test\_accounts1\accounts.csv

D:\test\_accounts1\accounts\_all\_0.csv

D:\test\_accounts1\accounts\_test\_15.csv

D:\test\_accounts1\accounts\_test\_22.csv

D:\test\_accounts1\success\_accounts\_15.csv

D:\test\_accounts1\test\_one.csv

合并6个文件完成

去重操作完成

Process finished with exit code 0

* python的os模块详细介绍
* 从小入手-python小应用、





