



其他

IT READ

python 使用 with

首頁
科技
程式語言

其他 · 發表 2019-02-08

ads

讀檔案:

要以讀檔案的模式開啟一個檔案物件，使用Python內建的`open()`函式，傳入檔名和標示符：

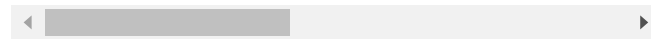
```
>>> f = open('E:\python\python\test.t
```



標示符'r'表示讀，這樣，我們就成功地打開了一個檔案。

如果檔案不存在，`open()`函式就會丟擲一個`IOError`的錯誤，並且給出錯誤碼和詳細的資訊告訴你檔案不存在：

```
f=open('E:\python\python\notfound.txt')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
FileNotFoundError: [Errno 2] No such
```



如果檔案開啟成功，接下來，呼叫`read()`方法可以一次讀取檔案的全部內容，Python把內容讀到記憶體，用一個`str`物件表示：

```
>>> f.read()
'Hello, python!'
```

最後一步是呼叫`close()`方法關閉檔案。檔案使用完畢後必須關閉，因為檔案物件會佔用作業系統的資源，並且作業系統同一時間能開啟的檔案數量也是有限的：

```
>>> f.close()
```

由於檔案讀寫時都有可能產生`IOError`，一旦出錯，後面的`f.close()`就不會呼叫。所以，為了保證無論是否

出錯都能正確地關閉檔案，我們可以使用**try ... finally**來實現：

```
try:
    f = open('/path/', 'r')
    print(f.read())
finally:
    if f:
        f.close()
```

每次都這麼寫實在太繁瑣，所以，**Python**引入了**with**語句來自動幫我們呼叫**close()**方法：

```
with open('/path/to/file', 'r') as f:
    print(f.read())
```




這和前面的**try ... finally**是一樣的，但是程式碼更佳簡潔，並且不必呼叫**f.close()**方法。

呼叫**read()**會一次性讀取檔案的全部內容，如果檔案有**20G**，記憶體就爆了，所以，要保險起見，可以反覆呼叫**read(size)**方法，每次最多讀取**size**個位元組的內容。另外，呼叫**readline()**可以每次讀取一行內容，呼叫**readlines()**一次讀取所有內容並按行

返回**list**。因此，要根據需要決定怎麼呼叫。

如果檔案很小，**read()**一次性讀取最方便；如果不能確定檔案大小，反覆呼叫**read(size)**比較保險；如果是配置檔案，呼叫**readlines()**最方便：


```
for line in f.readlines():  
    print(line.strip()) # 把末尾的'\n'
```



寫檔案

寫檔案和讀檔案是一樣的，唯一區別是呼叫**open()**函式時，傳入識別符號'**w**'或者'**wb**'表示寫文字檔案或寫二進位制檔案：

```
>>> f = open('E:\python\python\test.t  
>>> f.write('Hello, python!')  
>>> f.close()
```



可以反覆呼叫**write()**來寫入檔案，但是務必要呼叫**f.close()**來關閉檔案。當我們寫檔案時，作業系統往往不會立刻把資料寫入磁碟，而是放到記憶體快取起來，空閒的時候再慢慢寫入。只有呼叫

`close()`方法時，作業系統才保證把沒有寫入的資料全部寫入磁碟。忘記呼叫`close()`的後果是資料可能只寫了一部分到磁碟，剩下的丟失了。所以，還是用`with`語句來得保險：

```
with open('E:\python\python\test.txt')  
    f.write('Hello, python!')
```

要寫入特定編碼的文字檔案，請給`open()`函式傳入`encoding`引數，將字串自動轉換成指定編碼

字元編碼

要讀取非UTF-8編碼的文字檔案，需要給`open()`函式傳入`encoding`引數，例如，讀取GBK編碼的檔案：

```
>>> f = open('E:\python\python\gbk.txt')  
>>> f.read()  
'測試'
```

遇到有些編碼不規範的檔案，你可能會遇到`UnicodeDecodeError`，因為在文字檔案中可能夾雜了一些非法編碼的字元。遇到這種情況，`open()`函式還接收一個

`errors`引數，表示如果遇到編碼錯誤後如何處理。最簡單的方式是直接忽略：

```
>>> f = open('E:\python\python\gbk.txt', 'r', errors='ignore')
```



二進位制檔案

前面講的預設都是讀取文字檔案，並且是UTF-8編碼的文字檔案。要讀取二進位制檔案，比如圖片、視訊等等，用'`rb`'模式開啟檔案即可：

```
>>> f = open('E:\python\python\test.jpg', 'rb')
>>> f.read()
b'\xff\xd8\xff\xe1\x00\x18Exif\x00\x00'
```



總結：以後讀寫檔案儘量使用**`with open`**語句，少使用**`f = open()`**語句

對於多個檔案的讀寫，可以寫成以下兩種方式：

1、

```
with open('C:\Desktop\text.txt', 'r') as f:
    with open('C:\Desktop\text1.txt', 'w') as f1:
        with open('C:\Desktop\text2.txt', 'w') as f2:
            # ...
```



2、

```
with open('C:\Desktop\text.txt','r')
.....
with open('C:\Desktop\text1.txt','r')
.....
with open('C:\Desktop\text2.txt','r')
.....
```

檔案的讀寫方式列表：

模 式	描 述
r	以只读方式打开文件。文件的指针将会放在文件的开头。这是默认模式。
rb	以二进制格式打开一个文件用于只读。文件指针将会放在文件的开头。这是默认模式。
r+	打开一个文件用于读写。文件指针将会放在文件的开头。
rb+	以二进制格式打开一个文件用于读写。文件指针将会放在文件的开头。
w	打开一个文件只用于写入。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。
wb	以二进制格式打开一个文件只用于写入。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。
w+	打开一个文件用于读写。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。
wb+	以二进制格式打开一个文件用于读写。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。
a	打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。也就是说，新的内容将会被写入到已有内容之后。如果该文件不存在，创建新文件进行写入。
ab	以二进制格式打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。也就是说，新的内容将会被写入到已有内容之后。如果该文件不存在，创建新文件进行写入。
a+	打开一个文件用于读写。如果该文件已存在，文件指针将会放在文件的结尾。文件打开时会是追加模式。如果该文件不存在，创建新文件用于读写。
ab+	以二进制格式打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。如果该文件不存在，创建新文件用于读写。 https://blog.csdn.net/xr100vip

File 物件屬性：

文件对象属性	描 述
file.closed	表示文件已经被关闭，否则为False
file.mode	Access文件打开时使用的访问模式
file.encoding	文件所使用的编码
file.name	文件名
file.newlines	未读取到行分隔符时为None。只有一种行分隔符时为一个字符串，当文件有多种类型的行分隔符时，则为一个包含所有当前所遇到的行结束的列表
file.softspace	为0表示在输出一数据后，要加上一个空格符，1表示不加。这个属性一般程序员用不着，由程序内部使用。 https://blog.csdn.net/xr100vip

Reference:

標籤：

 您可能也會喜歡...

- [python 使用 with open \(\) as 讀寫檔案](#)
- [Flash AIR檔案操作:使用 FileStream物件讀寫檔案](#)
- [python使用pandas模組實現檔案讀取、字串處理、去重排序、excel生成](#)
- [Python使用Pandas對大型 csv檔案進行資料處理](#)
- [Python：使用 os、shutil 模組對檔案與目錄操作](#)
- [i/o流中 使用字元流 改寫檔案特定部分內容](#)
- [python使用JPytype來執行 java檔案](#)
- [初學Java IO之使用](#)
- [Python--使用執行緒--批量檔案的移動 以及FTP共享](#)
- [FileInputStream和 FileReader讀取檔案 四十一](#)
- [自己資料夾](#)
- [C#常用IO流與讀寫檔案](#)
- [【轉載】python使用loggin和ConfigParser配置檔案中遇到的問題](#)
- [Python使用Ctypes與 C/C++ DLL檔案通訊過程介紹及例項分析](#)
- [python 讀寫檔案函式](#)
- [Python：python3類-安裝和使用庫-讀寫檔案](#)
- [open\(\)用法](#)

python 讀寫三菱PLC數據，使用以太網讀寫Q系列，L系列，Fx系列的PLC數據

首頁
Python教學



ITREAD01.COM © 2018. 版權所有。