



File IO trong Python

Nội dung

1. Thực hành bài cũ
2. Mở file
3. Các phương thức làm việc với file
4. Đọc & Ghi file
5. Đóng file
6. Bài tập thực hành

Khái niệm về File

- ❑ **File** – tệp, tập tin. File là tập hợp của các thông tin được đặt tên được lưu trữ trên bộ nhớ máy tính như đĩa cứng, đĩa mềm, CD, DVD,...*Hiểu theo một cách khác thì File chính là một dãy bit có tên và được lưu trữ trên các thiết bị bộ nhớ của máy tính.*
- ❑ Có 3 loại file thông dụng là file văn bản, hình ảnh và âm thanh. Trong bài viết này đối tượng mà hướng đến chính là: *File văn bản.*

Mở file

❑ Để có thể đọc và ghi một file nào đó thì mở file là công việc tất yếu. Python cho phép chúng ta thực hiện điều đó với hàm `open`.

❑ Cú pháp:

```
file object = open(file_name [, access_mode][, buffering])
```

❑ Trong đó:

❑ **file_name**: là tập tin cần mở.

❑ **access_mode**: là một vài ký tự quy định cách thức mà tập tin sẽ được mở ra để làm gì. Khi mode bị bỏ qua thì mặc định sẽ là ký tự 'r'.

❑ **buffering**: quy định về bộ đệm sử dụng khi mở file [`<0, 0, 1, 1<`].

Access _mode

Mode	Mô tả
r	Mở file chỉ để đọc
r+	Mở file để đọc và ghi
rb	Mở file trong chế độ đọc cho định dạng nhị phân, đây là chế độ mặc định. Con trỏ tại phần bắt đầu của file
rb+	Mở file để đọc và ghi trong định dạng nhị phân. Con trỏ tại phần bắt đầu của file
w	Tạo một file mới để ghi, nếu file đã tồn tại thì sẽ bị ghi mới
w+	Tạo một file mới để đọc và ghi, nếu file tồn tại thì sẽ bị ghi mới
wb	Mở file trong chế độ ghi trong định dạng nhị phân. Nếu file đã tồn tại, thì ghi đè nội dung của file đó, nếu không thì tạo một file mới
wb+	Mở file để đọc và ghi trong định dạng nhị phân. Nếu file tồn tại thì ghi đè nội dung của nó, nếu file không tồn tại thì tạo một file mới để đọc và ghi
a	Mở file để ghi thêm vào cuối file, nếu không tìm thấy file sẽ tạo mới một file để ghi mới
a+	Mở file để đọc và ghi thêm vào cuối file, nếu không tìm thấy file sẽ tạo mới một file để đọc và ghi mới
ab	Mở file trong chế độ append trong chế độ nhị phân. Con trỏ là ở cuối file nếu file này đã tồn tại. Nếu file không tồn tại, thì tạo một file mới để ghi
ab+	Mở file trong chế độ đọc và append trong định dạng nhị phân. Con trỏ file tại cuối nếu file đã tồn tại. Nếu không tồn tại thì tạo một file mới để đọc và ghi

```
import os

# Open file:
# Get file path
dirpath = os.path.dirname(__file__)
file = os.path.join(dirpath, 'lesson8_fileIO.txt')

# Open file (mode default: r = Read only)
open(file)
print('Open file with default mode.')

# Open file (mode: r+ = Read and Write)
open(file, 'r+')
print('Open file with Read & Write mode.')
```

Read File

❑ Cú pháp:

fileObject.read([size])

❑ Phương thức này trả về một chuỗi có kích thước bằng size. Nếu không truyền size thì toàn bộ nội dung của file sẽ được đọc.

```
import os

# Open file:
# Get file path
dirpath = os.path.dirname(__file__)
# print(dirpath)
filepath = os.path.join(dirpath, 'lesson8_fileIO.txt')
# print(filepath)

# Open file
file = open(filepath, 'r+')
# print('Open file with Read mode.')

# Read file
str = file.read()
print('File content: ', str)
```



```
import os

# Open file:
# Get file path
dirpath = os.path.dirname(__file__)
# print(dirpath)
filepath = os.path.join(dirpath, 'lesson8_fileIO.txt')
# print(filepath)

# Open file
file = open(filepath, 'r+')
# print('Open file with Read mode.')

# Read file with size
s = file.read(16)
print('File content: ', s)
```

Đọc file với Readline

- ❑ **Readline** – một phương thức thường dùng trong đọc file. Phương thức này cho phép đọc một dòng trong file và trả về chuỗi.
- ❑ *Lưu ý*: Ký tự \n (xuống dòng) sẽ được đọc và trả về ở cuối chuỗi nếu đọc file theo phương thức này.
- ❑ Cú pháp:

fileObject.readline()

```
# Read file with size
s1 = file.readline()
s2 = file.readline()
s3 = file.readline()
s4 = file.readline()
s5 = file.readline()

print('Line 1: ', s1)
print('Line 2: ', s2)
print('Line 3: ', s3)
print('Line 4: ', s4)
print('Line 5: ', s5)
```

Write file

❑ Tương tự đọc file, để ghi một file ta cần mở file bằng cú pháp để ghi và sử dụng phương thức write để ghi vào.

❑ Cú pháp:

fileObject.write(string)

❑ Phương thức này cho phép ghi một chuỗi có nội dung là string vào vị trí của con trỏ trong tập tin.

```
info = \  
    'Name: ITPlus\n' + \  
    'Mail: itplus@gmail.com\n' + \  
    'Skype: itplus\n' + \  
    'Address: Ha noi'  
f = open('lesson8fileW.txt', 'w')  
  
f.write(info)
```

Phương thức làm việc với file

STT	Phương thức và Miêu tả
1	<u>Phương thức file.close()</u> Đóng một file. Bạn không thể đọc hoặc ghi một file đã được đóng
2	<u>Phương thức file.flush()</u> Flush (xóa sạch hoặc chuyển ra ngoài) bộ đệm nội tại.
3	<u>Phương thức file.fileno()</u> Trả về File descriptor (là một số nguyên) mà được sử dụng bởi trình triển khai cơ bản để yêu cầu các hoạt động I/O từ hệ điều hành
4	<u>Phương thức file.isatty()</u> Trả về true nếu file được kết nối tới một thiết bị tty(-like), nếu không là false
5	<u>Phương thức file.next()</u> Trả về dòng tiếp theo từ file mỗi khi phương thức này được gọi
6	<u>Phương thức file.read([size])</u> Đọc số byte là size đã cho từ file (có thể ít hơn nếu nó đã đọc đến EOF trước khi đọc đến lượng byte là size)

Phương thức làm việc với file (2)

STT	Phương thức và Miêu tả
7	<u>Phương thức file.readline([size])</u> Đọc cả một dòng từ file đó. Một ký tự newline (dòng mới) ở cuối được giữ trong chuỗi
8	<u>Phương thức file.readlines([sizehint])</u> Đọc tới EOF bởi sử dụng readline() và trả về một danh sách chứa các dòng. Nếu tham số sizehint tùy ý là có mặt thì thay vì đọc tới EOF, thì tất cả các dòng có tổng số byte xấp xỉ là sizehint sẽ được đọc
9	<u>Phương thức file.seek(offset[, whence])</u> Thiết lập vị trí hiện tại của file
10	<u>Phương thức file.tell()</u> Trả về vị trí hiện tại của con trỏ đọc/ghi bên trong file
11	<u>Phương thức file.truncate([size])</u> Cắt gọn kích cỡ file. Nếu tham số size tùy ý này có mặt, thì file được cắt gọn về kích cỡ đó
12	<u>Phương thức file.write(str)</u> Ghi một chuỗi tới file. Phương thức này không trả về giá trị nào
13	<u>Phương thức file.writelines(sequence)</u> Ghi một dãy các chuỗi tới file. Dãy sequence này có thể là bất cứ đối tượng mà tạo ra chuỗi nào, ví dụ như một List các chuỗi

Đóng file

- ❑ Khi bạn đã thực hiện xong các hoạt động trên File thì cuối cùng cần đóng File đó.
- ❑ Python tự động đóng một File khi đối tượng tham chiếu của một File đã được tái gán cho một file khác.
- ❑ Cú pháp:

file object = file.close()

Bài tập thực hành

- ❑ B1. Viết chương trình Python để đọc toàn bộ tập tin văn bản từ một file *.txt cho trước.
- ❑ B2. Viết một chương trình Python để đọc n dòng đầu tiên của một tập tin cho trước. Với n là số được nhập từ bàn phím.
- ❑ B3. Viết một chương trình Python để ghi đoạn văn bản vào một tập tin và hiển thị văn bản đó.
- ❑ B4: Tạo file '**demo_file1.txt**' trong thư mục source code. Với nội dung file như sau: 'Thuc \n hanh \n voi \n file\n IÓ\n'.
 - ❑ a) In ra màn hình nội dung file đó trên một dòng,
 - ❑ b) In ra màn hình nội dung file đó theo từng dòng.

❑ 5: Viết chương trình nhập từ bàn phím các thông tin cá nhân bao gồm: tên, tuổi, email, skype, địa chỉ, nơi làm việc.

❑a) Sau đó lưu các thông tin trên vào file 'setInfo.txt'.

❑b) Đọc thông tin từ file 'setInfo.txt' và hiển thị kết quả ra màn hình.

❑6: Cho một file 'demo_file2.txt' có nội dung: '*Dem so luong tu xuat hien abc abc abc 12 12 it it itplus*'. Đếm số lượng xuất hiện của các từ trong file đó.

→ Kết quả trả về: {'Dem': 1, 'so': 1, 'luong': 1, 'tu': 1, 'xuat': 1, 'hien': 1, 'abc': 3, '12': 2, 'it': 2, 'itplus': 1}

THANKS FOR ATTENDING!