

## Bài 3: ĐỌC- GHI DỮ LIỆU KIỂU TỆP TRONG PYTHON

Ở bài học trước dữ liệu được đưa vào từ bàn phím, sau khi xử lý sẽ đưa kết quả ra màn hình. Ở bài học này, ta sẽ đề cập đến dữ liệu được đọc từ tệp (tập tin – file) để xử lý và kết quả cũng được ghi ra tệp. Nội dung đọc từ tệp và kết quả ghi ra tệp được biểu diễn dưới dạng các kí tự. Các kí tự được tổ chức theo từng dòng, mỗi dòng có kí tự hết dòng (EOF). Hệ điều hành không cho các chương trình làm việc trực tiếp với các tệp dữ liệu mà thông qua biến tệp. Điều này giúp cho an toàn về dữ liệu trong tệp.

Khi muốn đọc và ghi tệp, chúng ta cần phải mở tệp trước. Khi hoàn thành, tệp cần phải được đóng lại để các tài nguyên được gắn với tệp được giải phóng.

File input (tệp vào): chứa dữ liệu để bài cho sẵn (ví dụ: BAI1.INP)

File output (tệp ra): chứa kết quả chương trình tính được (ví dụ: BAI1.OUT)

### I. CÁCH MỞ - ĐÓNG TỆP BẰNG OPEN() – CLOSE()

Đây là cách viết truyền thống, vẫn thường gặp trong các bài thi HSG.

#### 1. Cú pháp mở/đóng tệp:

*<biến tệp> = open(<tên tệp>, [access\_mode])*

# Các lệnh đọc/ghi dữ liệu

*<biến tệp>.close()*

Trong đó:

tên tệp: là một giá trị xâu chứa tên tệp muốn truy cập

access\_mode: xác định chế độ của tệp được mở ra như đọc, ghi, thêm, ... Đây là thông số tùy chọn, chế độ mặc định là đọc.

[access\_mode] = 'r' nghĩa là mở file ở chế độ đọc.

[access\_mode] = 'w' nghĩa là mở file ở chế độ ghi đè

[access\_mode] = 'a' nghĩa là mở file ở chế độ ghi tiếp

Sau khi đọc/ghi xong phải đóng file để tránh lỗi

\* Mở/đóng file input

fi = open("input.inp", "r")

# các lệnh đọc dữ liệu

fi.close()

\* Mở/đóng file output

fo = open("output.out", "w")

# các lệnh ghi dữ liệu

fo.close()

#### 2. Các cách đọc dữ liệu từ file input.

##### 2.1. Đọc toàn bộ dữ liệu từ tệp.

- Kể cả kí tự xuống dòng (\n), ta dùng thủ tục : *<biến tệp>.read()*

- Để loại bỏ kí tự xuống dòng (\n), ta dùng thủ tục: *<biến tệp>.read().strip()*

Dữ liệu đọc được là một xâu kí tự.

\* Ví dụ:

input.inp	Đọc toàn bộ dữ liệu từ tệp	Giải thích
abcde 12345	f1 = open("input.inp", "r") s = f1.read() f1.close()	s = "abcde\n12345\n"
	f1 = open("input.inp", "r") s = f1.read().strip() f1.close()	s = "abcde\n12345"

**2.2. Đọc từng dòng trong tệp.**

- Kể cả kí tự xuống dòng, ta dùng thủ tục : <biến tệp>.readline()
  - Để loại bỏ kí tự xuống dòng, ta dùng thủ tục: <biến tệp>.readline().strip()
- Dữ liệu mỗi dòng đọc được là một xâu kí tự.

input.inp	Đọc từng dòng dữ liệu	Giải thích
abcde 12345	f1 = open("input.inp", "r") s = f1.readline() s' = f1.readline() f1.close()	s = "abcde\n" s' = "12345\n"
	f1 = open("input.inp", "r") s = f1.readline().strip() s' = f1.readline().strip() f1.close()	s = "abcde" s' = "12345"

**2.3. Đọc nhiều dòng và lưu thành một danh sách, mỗi dòng là một phần tử của danh sách.**

✓ **Cách 1:**

Dùng thủ tục <biến tệp>.readlines() # Danh sách các dòng gồm cả kí tự xuống dòng  
Hoặc <biến tệp>.readlines().strip() # Danh sách các dòng không chứa kí tự xuống dòng

input.inp	Đọc nhiều dòng và lưu thành danh sách các dòng	Giải thích
abcde 12345	f1 = open("input.inp", "r") a = f1.readlines() f1.close()	a = ['abcde\n', '12345\n']
	f1 = open("input.inp", "r") a = f1.readlines().strip() f1.close()	a = ['abcde', '12345']

✓ **Cách 2:** Dùng vòng lặp for để đọc từng dòng lưu vào biến ds:

ds = [int(line.strip()) for line in <biến tệp>] # Danh sách các dòng không chứa kí tự xuống dòng.

**3. Ghi kết quả ra file out**

Để ghi dữ liệu vào tệp, ta cần mở tệp ở chế độ ghi, access mode thường là dùng 'w' (ghi đè) hoặc 'a' (ghi tiếp). Ở cả hai chế độ đều sẽ tạo tệp mới nếu tệp mở chưa tồn tại. Sau đó dùng thủ tục: <biến tệp>.write(s)

Thủ tục trên sẽ ghi vào tệp có <bien tệp> giá trị của xâu s. Sử dụng các kí tự ‘\n’ để phân biệt các dòng với nhau. Mỗi lần sử dụng write con trả tệp sẽ được đặt ngay sau kí tự cuối cùng được ghi.

Dữ liệu khi ghi vào luôn là xâu nên nếu là số ta phải ép kiểu về xâu mới ghi vào tệp.

#### \* Ví dụ ghi ra file output các số từ 1 đến 5

✓ TH1: ghi trên nhiều dòng

```
fo = open("output.out", "w")
for i in range(1, 6):
    fo.write(str(i) + "\n")
fo.close()
```

✓ TH2: Ghi trên một dòng (các số cách nhau bởi khoảng trắng)

#### \* Cách 1:

```
for i in range(1, 6):
    fo.write(str(i) + " ")
fo.close()
```

#### \* Cách 2:

```
fo.write(" ".join(str(i) for i in range(1, 6)))
fo.close()
```

\* Nếu có danh sách các số, muốn in danh sách đó ra file output với các số trên cùng một dòng cách nhau bởi dấu cách:

```
fo.write(" ".join(map(str, ds)))
fo.close()
```

\* Lưu ý: - Để ghi hai số a và b trên cùng một dòng cách nhau một khoảng trắng ta dùng lệnh:

fo.write(str(a) + " " + str(b)) hoặc fo.write(f'{a} {b}')

- Để ghi hai số a và b mỗi số trên một dòng ta dùng lệnh:

fo.write(str(a) + "\n" + str(b)) hoặc fo.write(f'{a}\n{b}')

- Dùng f – string khi ghi cho các kiểu dữ liệu là số, xâu kí tự, ghi biểu thức hoặc phép tính, ghi nhiều biểu thức khác kiểu cùng lúc. Không nên dùng cho các kiểu dữ liệu khác.

#### Ví dụ 1:

```
a, b = 3, 4
fo.write(f'{a} + {b} = {a + b}')
```

#### Ví dụ 2:

```
name = "An"
age = 14
score = 9.5
fo.write(f'Học sinh {name}, {age} tuổi, đạt {score} điểm.')
```

#### Ví dụ 3:

```
for i in range(1, 6):
    fo.write(str(i) + " ")
fo.close()
```

## II. CÁCH MỞ TỆP BẰNG WITH OPEN

#### \* Cú pháp mở file để đọc:

with open(<tên tệp>, 'r') as <bien tệp>:

# Thực hiện các thao tác với tệp

#### \* Cú pháp mở file để ghi:

```
with open(<tên tệp>, 'w') as <biến tệp>:  
    <biến tệp>.write()
```

➤ **Lưu ý:**

- ✓ Với cách mở dùng này, ta không cần đóng tệp vì chương trình sẽ tự động đóng tệp sau các thao tác với tệp.
- ✓ Các cách đọc/ghi dữ liệu tệp giống như đã nêu ở phần open() – close().
- ✓ Dữ liệu ghi ra file output phải đảm bảo là xâu kí tự (string)
- ✓ Hạn chế các thao tác với biến khi mở file. Sau khi mở file đọc dữ liệu gán vào biến thì thoát khỏi with. Các thao tác với biến hãy thực hiện thẳng hàng với từ with không thực hiện các thao tác sau từ with.

➤ **BẢNG TÓM TẮT**

Mục đích	Lệnh chính	Đặc điểm	Ghi nhớ
Đọc toàn bộ	f.read()	Dữ liệu nhỏ	Dễ dùng
Đọc 1 dòng	f.readline()	Có \n	Nên .strip()
Đọc tất cả dòng	f.readlines()	Danh sách chuỗi	Cần .strip()
Đọc nhiều dòng	for line in f	Tiết kiệm bộ nhớ	Tốt cho file lớn
Ghi file	f.write()	Ghi chuỗi	Dùng str() khi cần

➤ **SO SÁNH NHANH HAI CÁCH MỞ FILE**

Cách mở file	Ưu điểm	Nhược điểm
with open(...)	Tự động đóng file, an toàn, gọn	Viết hơi lả với người mới học
open()/close()	DỄ hiểu, truyền thống	DỄ quên close(), có thể lỗi ghi/đọc

➤ **Trường hợp đọc – ghi file có dấu Tiếng Việt.**

Ta chỉ cần thêm tham số **encoding="utf-8"** khi mở file.

a. **Khi dùng open() – close()**

\* **Đọc file**

```
fi = open("input.inp", "r", encoding="utf-8")
```

```
data = f.read()
```

```
fi.close()
```

\* **Ghi file**

```
fo = open("output.out", "w", encoding="utf-8")
```

```
fo.write("Xin chào, đây là dữ liệu có dấu tiếng Việt!")
```

```
fo.close()
```

b. **Khi dùng with open() (cách khuyến khích)**

\* **Đọc file**

```
with open("input.inp", "r", encoding="utf-8") as f:
```

```
    data = f.read()
```

\* **Ghi file**

```
with open("output.out", "w", encoding="utf-8") as f:
```

```
    f.write("Học sinh đạt giải Nhất môn Tin học!")
```

**Ví dụ 1:** Cho tệp văn bản HAISO.INP gồm hai dòng, mỗi dòng ghi một số nguyên. Ghi ra tệp HAISO.OUT gồm một dòng ghi hai số đó, mỗi số cách nhau một kí tự trắng.

Giải

- **Cách 1: Dùng read():** Đọc cả tệp và lưu chúng thành một xâu kí tự (bao gồm cả kí tự xuống dòng).

Sử dụng open() – close()	Sử dụng with open
<pre># Đọc toàn bộ nội dung file thành 1 chuỗi fi = open("HAISO.INP ", "r") data = fi.read() fi.close()  # Tách các dòng, loại bỏ ký tự xuống dòng numbers = data.splitlines() '''hoặc numbers = fi.read().splitlines() fi.close()'''  # Ghi các số ra file output trên cùng 1 dòng fo = open("HAISO.OUT ", "w") fo.write(" ".join(numbers)) fo.close()</pre>	<pre>with open("HAISO.INP ", "r") as fi:     data = fi.read().splitlines() with open("HAISO.OUT ", "w") as fo:     fo.write(" ".join(data))</pre>

➤ **Cách 2:** Dùng **readline()**: Đọc từng dòng

Sử dụng open() – close()	Sử dụng with open
<pre>* <b>Cách 1:</b> Kiểu dữ liệu mặc định là string fi = open("HAISO.INP ", "r") a = fi.readline().strip() b = fi.readline().strip() fi.close()  fo = open("output.out", "w") fo.write(a + " " + b) fo.close()  * <b>Cách 2:</b> Ép kiểu số nguyên f = open("HAISO.INP ", "r") a = int(f.readline()) b = int(f.readline()) f.close()  fo = open("HAISO.OUT ", "w") fo.write(str(a) + " " + str(b)) fo.close()</pre>	<pre>* <b>Cách 1:</b> Kiểu dữ liệu mặc định là string with open("HAISO.INP ", "r") as fi:     a = fi.readline().strip()     b = fi.readline().strip() with open("output.out", "w") as fo:     fo.write(str(a) + " " + str(b))  * <b>Cách 2:</b> Ép kiểu số nguyên with open("HAISO.INP ", "r") as fi:     a = int(fi.readline())     b = int(fi.readline()) with open("HAISO.OUT ", "w") as fo:     fo.write(str(a) + " " + str(b))</pre>

➤ **Cách 3:** Dùng **readlines()**: Đọc cả tệp và lưu thành một danh sách (mỗi phần tử của danh sách là một dòng bao gồm cả kí tự xuống dòng). Dùng **.strip()** để loại bỏ kí tự xuống dòng.

Sử dụng open() – close()	Sử dụng with open
<pre># Đọc toàn bộ các dòng trong file thành danh sách f = open("HAISO.INP ", "r") lines = f.readlines() f.close()  # Loại bỏ ký tự xuống dòng \n ở mỗi phần tử</pre>	<pre>with open("HAISO.INP ", "r") as f:     numbers = [x.strip() for x in f.readlines()] with open("HAISO.OUT ", "w") as fo:     fo.write(" ".join(numbers))</pre>

```

numbers = [x.strip() for x in lines]
# Ghi các số ra file output, cách nhau bởi
# khoảng trắng
fo = open("HAIISO.OUT ", "w")
fo.write(" ".join(numbers))
fo.close()

```

➤ **Tổng hợp và so sánh các cách:**

Cách đọc	Hàm sử dụng	Kiểu dữ liệu trả về	Khi nào dùng
read()	Chuỗi toàn bộ file	str	Khi muốn xử lý toàn bộ nội dung 1 lần
readline()	Một dòng mỗi lần gọi	str	Khi cần đọc từng dòng riêng biệt
readlines()	Danh sách các dòng	list	Khi cần xử lý từng dòng tuần tự

**Ví dụ 2:** Cho tệp văn bản HAIISO.INP gồm hai số nguyên cách nhau một khoảng trắng. Ghi ra tệp HAIISO.OUT gồm hai dòng, mỗi dòng ghi một số.

➤ **Cách 1: Dùng *read()*:** Đọc cả tệp và lưu chúng thành một xâu kí tự (bao gồm cả kí tự xuống dòng).

Sử dụng <i>open() – close()</i>	Sử dụng <i>with open</i>
<pre> * Cách 1: Kiểu mặc định là xâu fi = open("HAIISO.INP ", "r") a, b = fi.read().split() fi.close() fo = open("HAIISO.OUT ", "w") fo.write(a + "\n" + b) fo.close() * Cách 2: Ép kiểu số fi = open("HAIISO.INP ", "r") a, b = map(int, fi.read().split()) fi.close() fo = open("HAIISO.OUT ", "w") fo.write(str(a) + "\n" + str(b)) fo.close() </pre>	<pre> * Cách 1: Kiểu mặc định là xâu with open("HAIISO.INP ", "r") as f:     a, b = f.read().split() with open("HAIISO.OUT ", "w") as f:     f.write(f'{a}\n{b}') # hoặc f.write(a + "\n" + b)  * Cách 2: Ép kiểu số with open("HAIISO.INP ", "r") as f:     a, b = map(int, f.read().split()) with open("HAIISO.OUT ", "w") as f:     f.write(f'{a}\n{b}') # hoặc f.write(str(a) + "\n" + str(b)) </pre>

➤ **Cách 2: Dùng *readline()*:** Đọc từng dòng

Sử dụng <i>open() – close()</i>	Sử dụng <i>with open</i>
<pre> * Cách 1: Kiểu mặc định là xâu fi = open("HAIISO.INP ", "r") a, b = fi.readline().strip().split() # hoặc a, b = fi.readline().split() fi.close() fo = open("HAIISO.OUT ", "w") fo.write(a + "\n" + b) </pre>	<pre> * Cách 1: Kiểu mặc định là xâu with open("HAIISO.INP ", "r") as f:     a, b = f.readline().strip().split() with open("HAIISO.OUT ", "w") as f:     f.write(f'{a}\n{b}') </pre>

```

fo.close()
* Cách 2: Ép kiểu số nguyên
f = open("HAISO.INP ", "r")
a, b = map(int, f.readline().split())
f.close()
fo = open("HAISO.OUT ", "w")
fo.write(str(a) + "\n" + str(b))
fo.close()

```

```

* Cách 2: Ép kiểu số nguyên
with open("HAISO.INP ", "r") as f:
    a, b = map(int, f.readline().split())
with open("HAISO.OUT ", "w") as fo:
    fo.write(f"\n{a}\n{b}")

```

- **Cách 3:** Dùng `readlines()`: Đọc cả tệp và lưu thành một danh sách (mỗi phần tử của danh sách là một dòng bao gồm cả kí tự xuống dòng). Dùng `.strip()` để loại bỏ kí tự xuống dòng.

Sử dụng <code>open() – close()</code>	Sử dụng <code>with open</code>
<pre> * Cách 1: Kiểu mặc định là xâu f = open("HAISO.INP ", "r") lines = f.readlines()      # đọc tất cả các                            # dòng vào danh sách f.close() a, b = lines[0].strip().split() # xử lý dòng                            # đầu tiên # Ghi ra file output fo = open("HAISO.OUT ", "w") fo.write(a + "\n" + b) fo.close() * Cách 2: Ép kiểu số nguyên f = open("HAISO.INP ", "r") lines = f.readlines() f.close() a, b = map(int, lines[0].strip().split()) fo = open("HAISO.OUT ", "w") fo.write(str(a) + "\n" + str(b)) fo.close() </pre>	<pre> * Cách 1: Kiểu mặc định là xâu with open("HAISO.INP ", "r") as f:     a, b = " ".join(f.readlines()).split() with open("HAISO.OUT ", "w") as fo:     fo.write(f"\n{a}\n{b}")  * Cách 2: Ép kiểu số nguyên with open("HAISO.INP ", "r") as f:     data = " ".join(f.readlines()).split()     a, b = map(int, data) with open("HAISO.OUT ", "w") as fo:     fo.write(f"\n{a}\n{b}") </pre>

### BÀI TẬP THỰC HÀNH.

Cho tệp văn bản **TINH.INP** hai số nguyên a và b trên cùng một dòng cách nhau bởi một khoảng trắng. Ghi ra tệp văn bản **TINH.OUT** gồm:

- Dòng đầu tiên in ra tổng của hai số đó
- Dòng thứ hai in ra hiệu của hai số
- Dòng thứ ba in ra tích của hai số đó
- Dòng thứ tư in ra thương của hai số đó với kết quả làm tròn hai chữ số thập phân

**Ví dụ:**

<b>TINH.INP</b>	<b>TINH.OUT</b>
15 4	19 11 60 3.75

