BÀI 5 LẬP TRÌNH CHỨC NĂNG LƯU LỊCH SỬ TRÒ CHUYỆN

Trong bài học này, chúng ta sẽ tìm hiểu cách làm việc với tập tin như tạo tập tin, đọc và ghi tập tin để thực hiện chức năng lưu lại lịch sử trò chuyện. Tất cả các công việc này sẽ được lập trình trên file **main.py**.

```
try:
    f = open("history.txt", "r")
except:
    f = open("history.txt", "x")
else:
    content = f.read()
    text_area.insert('1.0', content)
finally:
    f.close()
```

Kết quả của bài học

5.1. Các thao tác với tập tin

Python hỗ trợ hàm **open()** dùng để mở và làm việc với tập tin. Lệnh này có cú pháp **open(<tên tập tin>[, chế độ mở])**, trong đó, **<tên tập tin>** là tên hoặc đường dẫn đến tập tin muốn mở, **[chế độ mở]** quy định các thao tác có thể làm việc với tập tin được mở:

Chế độ	Ý nghĩa
"r"	Read – Mở file để đọc, nếu file không tồn tại sẽ báo lỗi.
"w"	Write – Mở file để ghi, nếu file đã tồn tại thì xóa nội dung cũ của file và thực hiện ghi từ đầu, nếu file không tồn tại thì tạo một file mới.
"a"	Append – Mở file để ghi bổ sung, nếu file đã tồn tại thì thực hiện ghi nối tiếp vào nội dung cũ, nếu file không tồn tại thì tạo một file mới.
"x"	Create – Tạo mới file, nếu file đã tồn tại sẽ báo lỗi.

Thông thường, chúng ta không chạy lệnh **open()** độc lập mà sử dụng để tạo đối tượng làm việc với file (có thể gọi là đối tượng file), dưới đây là một số ví dụ với tạo đối tượng file **f**:

Lệnh	Ý nghĩa
`f = open("danh_sach.txt", "r")`	Mở file **danh_sach.txt** cùng thư mục với file chương trình để đọc,

	nếu không tồn tại file này sẽ báo lỗi.
`f = open("lop_7A5/diem.txt", "w")`	Mở file **diem.txt** trong thư mục con **lop_7A5** để ghi, nếu file đã tồn tại sẽ bị xóa các nội dung cũ, nếu file chưa tồn tại sẽ được tạo mới.
`f = open("demo.txt", "x")`	Tạo mới một file có tên **demo.txt** cùng thư mục với file chương trình, nếu file đã tồn tại sẽ báo lỗi.
`f = open("test.txt")`	Mở tệp tin với chế độ "r" là chế độ mặc định.

Ngoài những chế độ trên, chúng ta còn các chế độ khác như đồng thời vừa đọc và ghi, chỉ định kiểu dữ liệu văn bản (text) hoặc nhị phân (binary)... Mặc định, chúng ta mở tập tin để thao tác với kiểu dữ liệu văn bản, các chế độ khác các bạn có thể tự tìm hiểu thêm.

Với đối tượng file, chúng ta có các phương thức phổ biến sau:

Phương thức	Vai trò
write()	Ghi văn bản vào file
read()	Đọc dữ liệu từ file
readline()	Đọc dữ liệu từ file theo từng dòng
readlines()	Đọc tất cả các dòng từ file, trả về một mảng chứa tất cả các dòng, mỗi phần tử là một dòng
close()	Đóng file

Các bạn lưu ý, những thao tác với file mặc định của Python trên không hỗ trợ với file có nội dung là tiếng Việt.

5.2. Lập trình chức năng lưu lịch sử trò chuyện

Để lập trình chức năng lưu lịch sử trò chuyện, chúng ta có ý tưởng lưu nội dung trò chuyện vào một tập tin văn bản (ví dụ: **history.txt,** các bạn có thể sử dụng tập tin có tên bất kỳ và có định dạng khác như **.dat**, **.inp**...), sau đó mỗi lần mở ứng dụng sẽ đọc lại tập tin đó và đưa vào nội dung của ô hội thoại.

Đầu tiên, chúng ta sẽ kiểm tra tập tin **history.txt** để lưu lịch sử có tồn tại hay không, nếu đã tồn tại thì đọc file này và đưa vào nội dung ô hội thoại, nếu không thì sẽ tạo một file **history.txt** mới. Chúng ta có nhiều cách để thực hiện việc này, ví dụ

như sử dụng thư viện **pathlib** hoặc **os.path**, hay sử dụng chế độ mở file "a+" hỗ trợ việc tạo mới file nếu file không tồn tại. Trong trường hợp này, chúng ta sẽ được hướng dẫn sử dụng nhóm lệnh bắt lỗi **try – except**. Nhóm lệnh này dùng để kiểm tra việc chạy các lệnh có gặp lỗi (error) hay các ngoại lệ (exception) hay không để thực hiện các công việc phù hợp. Cú pháp đầy đủ của **try – except** như sau:

```
try:
    <các lệnh cần kiểm tra>
except [tên lỗi]:
    <các lệnh thực hiện khi gặp lỗi>
[else:
    <các lệnh được thực hiện khi không gặp lỗi>]
[finally:
    <các lệnh được thực hiện dù có gặp lỗi hay không>]
```

Tên lỗi, từ khóa **else** và **finally** không bắt buộc phải có, các cấu trúc nâng cao hơn như bắt nhiều lỗi khác nhau hay danh sách tên lỗi các bạn có thể tự tìm hiểu thêm.

Cụ thể, chúng ta mong muốn thực hiện việc mở file **history.txt** để đọc, nếu gặp lỗi (vì file không tồn tại) thì sẽ tạo file mới, nếu không gặp lỗi sẽ thực hiện đọc nội dung và thêm vào ô hội thoại. Trong khối lệnh **else**, biến **content** dưới đây dùng để lưu toàn bộ nội dung của đối tượng file **f**, các bạn có thể không cần bước sử dụng biến để lưu nội dung file mà trực tiếp gọi lệnh `text_area.insert('1.0', f.read())`, nhưng việc dùng biến sẽ khiến chương trình rõ ràng hơn.

```
try:
    f = open("history.txt", "r")
except:
    f = open("history.txt", "x")
else:
    content = f.read()
    text_area.insert('1.0', content)
```

Khi làm việc với file, các bạn luôn luôn nhớ cần thực hiện đóng file sau khi làm việc, dù cho file được mở với bất kỳ chế độ nào. Trong trường hợp này ta sẽ sử dụng từ khóa **finally** để việc đóng file được thực hiện dù file được mở để đọc hay được tạo mới. Các bạn lưu ý, việc đọc và đưa nội dung vào ô hiển thị được thực hiện sau khi đối tượng ô hiển thị được tạo ra, vì vậy các bạn có thể lập trình các lệnh này ở vị trí sau khi khởi tạo và thiết lập các đối tượng.

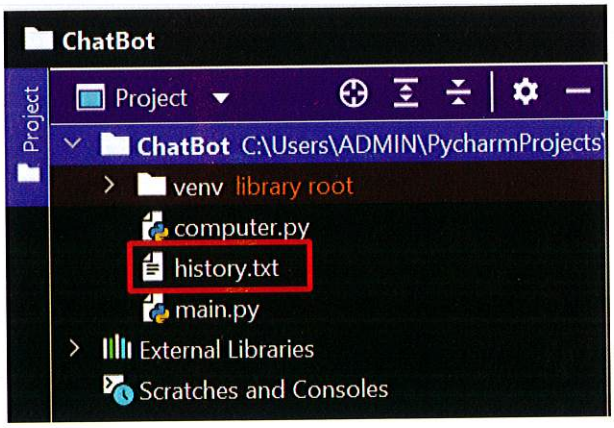
```
...   ...
22 button_send.grid(row=1, column=1, padx=5, pady=5)

23 try:
24     f = open("history.txt", "r")
```

```
except:
    f = open("history.txt", "x")
else:
    content = f.read()
    text_area.insert('1.0', content)
finally:
    f.close()

tk.mainloop()
```

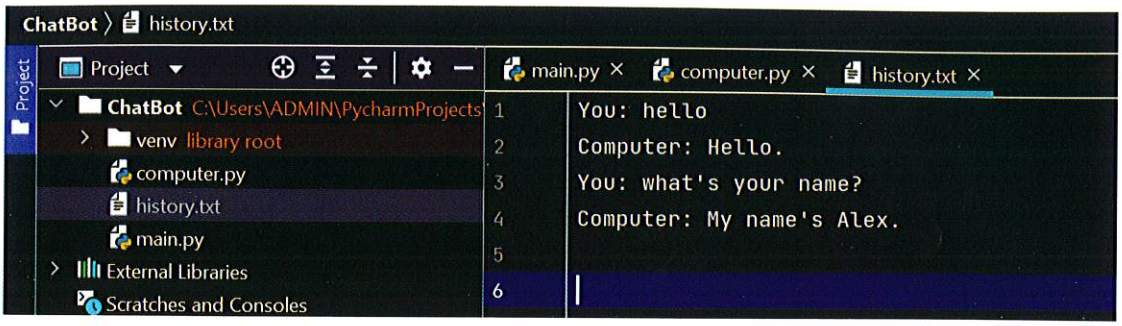
Các bạn chạy thử chương trình và sau đó tắt chương trình đi sẽ thấy file **history.txt** được tạo mới.



Tiếp theo chúng ta sẽ lập trình chức năng lưu lại nội dung trò chuyện vào file **history.txt**. Chúng ta sử dụng một cách đơn giản là sử dụng chế độ mở file "w", mỗi khi người dùng gửi nội dung và máy tính trả lời, chúng ta xóa nội dung cũ của file và lưu lại toàn bộ nội dung mới của ô hội thoại, công việc này thực hiện trong hàm **computer_response()**. Các cách khác như nối thêm từng đoạn hội thoại nhỏ hay thêm từng câu trò chuyện vào file cũ các bạn có thể tự mình suy nghĩ phát triển thêm.

```
def computer_response(text):
    ...
    f = open("history.txt", "w")
    f.write(text_area.get('1.0', tk.END))
    f.close()
```

Các bạn chạy thử sẽ thấy nội dung được lưu vào file **history.txt** sau mỗi lần máy tính trả lời, các lần chạy sau sẽ thấy nội dung được lấy từ file này, các bạn có thể sử dụng PyCharm trực tiếp mở để kiểm tra nội dung file **history.txt**. Như vậy, dự án ChatBot đến đây đã được hoàn thành.



Tóm tắt lý thuyết và bài tập thực hành

Trong bài học này, chúng ta đã tìm hiểu cấu trúc **try – except** để lập trình bắt lỗi khi đọc file, qua đó kiểm tra được file đã tồn tại hay không. Các bạn cũng đã nắm được các thao tác đọc và ghi file cơ bản.

Bài tập 1. Hãy thử tạo sẵn file **history.html**, điền một số nội dung tùy ý có cả các ký tự tiếng Việt, sau đó chạy chương trình, quan sát kết quả và giải thích.

Bài tập 2. Lập trình chức năng lưu cuộc trò chuyện theo từng nội dung trò chuyện được thêm vào bằng chế độ "a".