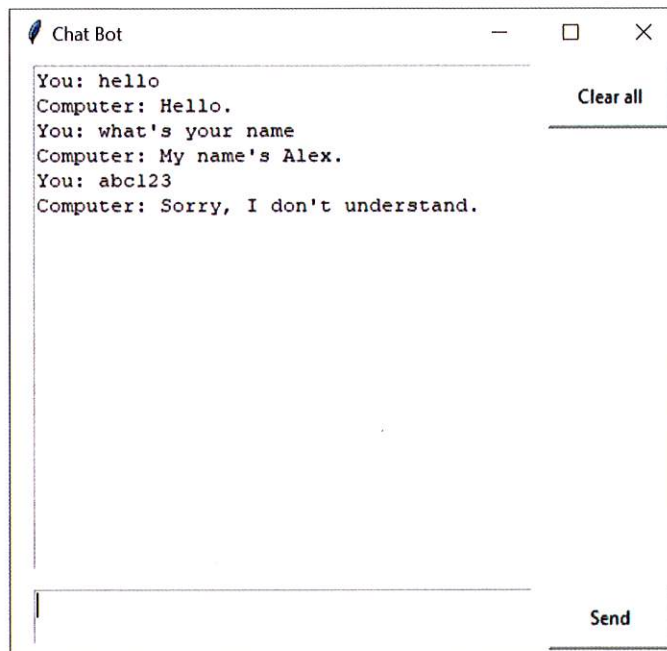


## BÀI 3 LẬP TRÌNH CHỨC NĂNG GỬI VÀ NHẬN TIN NHẮN

Trong bài này, chúng ta sẽ lập trình chức năng người dùng gửi tin nhắn và máy tính sẽ phản hồi nội dung phù hợp.

```
def send_text():
    user_text = text_box.get('1.0', tk.END)
    text_area.insert(tk.END, 'You: ' + user_text)
    text_box.delete('1.0', tk.END)
    computer_response(user_text)

def computer_response(text):
    computer_text = "Sorry, I don't understand.\n"
    if "hello" in text:
        computer_text = "Hello.\n"
    elif "name" in text:
        computer_text = "My name's Alex.\n"
    text_area.insert(tk.END, 'Computer: ' + computer_text)
```



Kết quả của bài học

### 3.1. Lập trình gửi tin nhắn

Khi chúng ta gõ văn bản vào ô nhập nội dung trò chuyện **text\_box**, nội dung này sẽ được hiển thị ở ô hội thoại **text\_area**. Việc lập trình nút được nhấn các bạn đã



được học ở Chương 3, chúng ta sẽ tạo hàm (ví dụ: `send_text`) và thiết lập để nút Send thực hiện việc gửi tin khi được nhấn.

```
3 wd.title("ChatBot")

# hàm thực hiện việc gửi tin
4 def send_text():
5     pass

...
9 button_send = tk.Button(width=10, height=2, text="Send",
                           command=send_text)
```

Để thực hiện gửi tin, chúng ta cần lấy nội dung của đối tượng `text_box` và sau đó thêm vào đối tượng `text_area`. Đối tượng Text cung cấp các phương thức để làm việc với văn bản:

Phương thức	Cú pháp	Ý nghĩa
<code>get()</code>	<code>get(&lt;vị trí đầu&gt;, &lt;vị trí cuối&gt;)</code>	Lấy các ký tự từ vị trí đầu đến vị trí cuối.
<code>insert()</code>	<code>insert(&lt;vị trí&gt;, &lt;nội dung&gt;)</code>	Chèn một nội dung văn bản bất kỳ vào một vị trí.
<code>delete()</code>	<code>delete(&lt;vị trí đầu&gt;, &lt;vị trí cuối&gt;)</code>	Xóa các ký tự từ vị trí đầu đến vị trí cuối.

Vị trí của ký tự trong Text được biểu diễn bằng một số thập phân với phần nguyên là số thứ tự dòng (bắt đầu từ 1) và phần thập phân là số thứ tự của ký tự trên dòng đó. Ngoài ra, chúng ta đã được biết thư viện **tkinter** cung cấp hằng số END là vị trí cuối cùng trong toàn bộ nội dung đối tượng. Dưới đây là một số ví dụ về vị trí:

Vị trí và câu lệnh	Ý nghĩa
<code>'1.0'</code>	Vị trí trước ký tự đầu tiên của dòng 1.
<code>'2.4'</code>	Vị trí sau ký tự thứ 4 của dòng 2.
<code>'3.end'</code>	Vị trí sau ký tự cuối cùng của dòng 3.
<code>get('1.2', '1.4')</code>	Lấy các ký tự từ vị trí sau ký tự thứ 2 (lấy từ ký tự thứ 3) đến vị trí sau ký tự thứ 4 (tương đương lấy 2 ký tự ở vị trí thứ 3 và 4) của dòng 1
<code>delete('2.0', '4.end')</code>	Xóa các ký tự từ vị trí trước ký tự đầu tiên của dòng 2 đến sau vị trí cuối cùng dòng 4, tương đương xóa ba dòng 2, 3 và 4.





```
delete('2.0', tkinter.END)
```

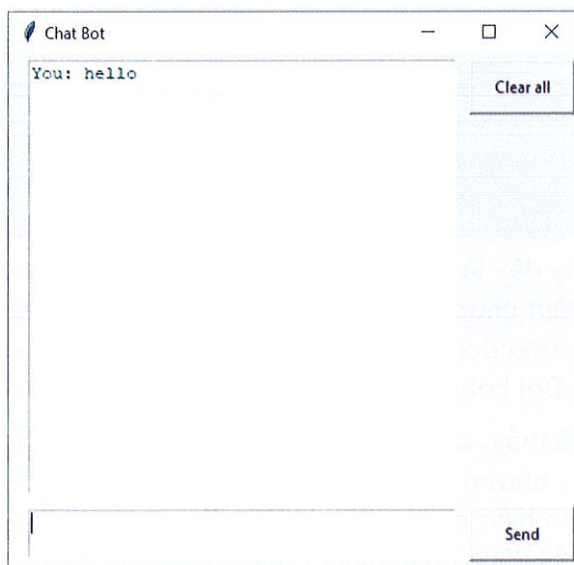
Xóa các ký tự từ vị trí trước ký tự đầu tiên của dòng 2 đến hết.

Chúng ta có thể sử dụng một biến để lưu nội dung mà người dùng nhập (ví dụ: **user\_text**) và lấy nội dung từ vị trí đầu tiên đến vị trí cuối cùng (dòng 5). Tiếp theo, chúng ta chèn nội dung này vào vị trí cuối cùng của ô hội thoại (dòng 6) và xóa toàn bộ nội dung của ô nhập nội dung trò chuyện (dòng 7).

```
4 def send_text():
5     user_text = text_box.get('1.0', tk.END)
6     text_area.insert(tk.END, user_text)
7     text_box.delete('1.0', tk.END)
```

Các bạn chạy thử sẽ thấy khi nhấn nút Send thì nội dung trò chuyện đã được thêm vào ô hội thoại. Các bạn có thể thêm chữ "You: " vào trước nội dung gửi để phân biệt câu hội thoại là của mình gửi hay máy tính trả lời.

```
6     text_area.insert(tk.END, 'You: ' + user_text)
```



### 3.2. Lập trình máy tính phản hồi

Trong hàm **send\_text()**, sau khi hiển thị nội dung trò chuyện của người dùng, chúng ta cần lập trình để máy tính phản hồi. Chúng ta có thể tiếp tục lập trình các công việc này trong hàm **send\_text()**, nhưng về mặt quản lý chương trình thì ta thấy việc hiển thị nội dung trò chuyện của người dùng và máy tính phản hồi là 2 công việc cùng cấp với nhau có thứ tự trước sau, không việc nào nằm bên trong việc nào, vì vậy ta nên tạo một hàm khác để xử lý (ví dụ: **computer\_response**). Nội dung trả lời của máy tính dựa trên thông tin nội dung trò chuyện mà người dùng gửi, vậy nên ta cần truyền nội dung này vào trong hàm như mô tả trong dòng 8.



Hàm **computer\_response** này sẽ thực hiện việc quyết định xem sẽ phản hồi với nội dung gì. Trong dự án này, chúng ta sẽ sử dụng phương pháp đơn giản nhất để lập trình chức năng phản hồi: Đưa ra phản hồi dựa trên các từ khóa có trong câu nói của người dùng. Ví dụ nếu câu nói của người dùng có chứa "hello" hoặc "hi", máy tính sẽ trả lời là "Hello", nếu câu nói chứa "name", máy tính sẽ trả lời một cái tên bất kỳ mà các bạn đặt... Để kiểm tra một xâu có chứa một xâu khác không, chúng ta sử dụng từ khóa **in**. Cú pháp **<xâu 1> in <xâu 2>** sẽ trả về giá trị **true** nếu trong **<xâu 2>** có xuất hiện **<xâu 1>**, nếu **<xâu 2>** không chứa **<xâu 1>** sẽ trả về **false**. Biến **computer\_text** được tạo để lưu nội dung máy tính phản hồi, nội dung này sẽ được chèn thêm vào ô hội thoại ở dòng 14.

```
4 def send_text():
5     user_text = text_box.get('1.0', tk.END)
6     text_area.insert(tk.END, 'You: ' + user_text)
7     text_box.delete('1.0', tk.END)
8     computer_response(user_text)

9 def computer_response(text):
10     if "hello" in text:
11         computer_text = "Hello."
12     elif "name" in text:
13         computer_text = "My name's Alex."
14     text_area.insert(tk.END, 'Computer: ' + computer_text)
```

Chúng ta có thể thấy đây là cách làm thủ công, chưa phải ứng dụng trí tuệ nhân tạo và không thực sự làm chương trình trở nên "thông minh", các bạn có thể tham khảo thêm các thư viện ứng dụng trí tuệ nhân tạo để huấn luyện và hỗ trợ việc trả lời thông minh như ChatterBot hoặc dịch vụ Dialogflow của Google.

Các bạn chạy thử sẽ thấy, khi nhập nội dung chứa từ "hello" hoặc "name" thì máy tính phản hồi chính xác, nhưng nếu ta nhập một từ bất kỳ khác "hello" và "name" thì chương trình sẽ bị lỗi do biến **computer\_text** không được định nghĩa. Chúng ta có thể lập trình để nếu người dùng nhập một từ nằm ngoài các trường hợp trên, máy tính sẽ phản hồi "I don't understand". Có nhiều cách để làm việc này, ví dụ như thêm lệnh **else** hoặc đặt giá trị khởi tạo cho **computer\_text** như mô tả trong dòng 10. Ví dụ trong cuốn sách sẽ sử dụng tiếng Anh, còn các bạn có thể dùng bất kỳ ngôn ngữ nào.

```
9 def computer_response(text):
10     computer_text = "Sorry, I don't understand."
11     if "hello" in text:
12         computer_text = "Hello."
13     elif "name" in text:
14         computer_text = "My name's Alex."
15     text_area.insert(tk.END, 'Computer: ' + computer_text)
```





Các bạn chạy chương trình và thử nhập nhiều câu nói khác nhau sẽ thấy nội dung người dùng nhập hiển thị cùng một dòng với phần văn bản máy tính phản hồi trước đó. Để sửa lỗi này, chúng ta thêm vào cuối mỗi câu phản hồi của máy tính một ký tự xuống dòng '\n'.

```

9 def computer_response(text):
10     computer_text = "Sorry, I don't understand.\n"
11     if "hello" in text:
12         computer_text = "Hello.\n"
13     elif "name" in text:
14         computer_text = "My name's Alex.\n"
15     text_area.insert(tk.END, 'Computer: ' + computer_text)

```

Như vậy, ta đã hoàn thành chức năng phản hồi của máy tính mỗi khi người dùng gửi một câu nói. Các bạn có thể bổ sung thêm các câu phản hồi khác để chương trình trở nên thú vị hơn.



### Tóm tắt lý thuyết và bài tập thực hành

Trong bài này, chúng ta đã tìm hiểu một số phương thức của đối tượng Text như **insert()**, **delete()**... Chúng ta đã lập trình chức năng gửi nội dung trò chuyện khi nhấn nút Send và lập trình máy tính phản hồi.

**Bài tập 1.** Hãy tìm cách phân biệt sự khác nhau của vị trí cuối dòng trên và vị trí đầu dòng dưới, ví dụ vị trí '1.end' và '2.0'.

**Bài tập 2.** Đối tượng ô hội thoại **text\_area** dùng để hiển thị các dòng văn bản, nhưng thực tế cũng dùng để nhập văn bản từ bàn phím, khiến người dùng có thể trực tiếp gõ văn bản vào ô hội thoại. Hãy lập trình để người dùng không làm được điều đó, biết lệnh để vô hiệu hóa đối tượng Text **text\_area** là **text\_area.configure(state='disabled')** và lệnh để đối tượng hoạt động trở lại là **text\_area.configure(state='normal')**.

**Bài tập 3.** Bổ sung thêm nhiều lựa chọn trả lời cho máy tính khiến máy tính “thông minh” hơn, ví dụ nếu người dùng gửi nội dung trò chuyện có “bye” thì máy tính đáp “Goodbye!”, nếu có “good morning” thì cũng đáp lại “Good morning”...