

程式概述：

以python語言寫成，為了能實現chat的對話功能，利用select以處理同時輸入訊息跟接收socket資料的情況。將整個bot的功能及連線初始以class整合，以方便日後的修改。

特別定義：

★ 程式假設使用者的名稱只用冒號及驚嘆號以外的字元組成

以下分別說明各個功能的實作方法：

1. 星座運勢：

建立一個dictionary，key為星座名稱，value則是運勢內容，當從socket收到訊息時，若只有一個單字且能在dictionary找到對應的key，便將運勢內容以PRIVMSG傳送給發送訊息的人，若找不到對應的key則忽略訊息

2. 猜數字：（程式設定一次只能給一人玩）

當收到"!guess"的訊息時，先將user存在一個紀錄使用者跟其正在使用的功能的table，並隨機生成一個1~10之間的整數並存在class object裏，如果收到的訊息不是整數，或者整數大於十或小於一的話都會直接忽略。數字如果比答案小則傳送higher，太大則傳送lower，猜同一個數字會傳送訊息提醒避免猜一樣的數字，收到正確答案後便清除使用者在table的資料並送出遊戲結束的訊息。為了避免另一個使用者進入遊戲，在class裡會設一個guessing的布林值，True的話代表有人在遊戲中，其他人傳送"!guess"都會被自動忽略。

3. 找歌曲：

收到"!song"後將收到的歌名接在youtube搜尋引擎的URL後面(https://www.youtube.com/results?search_query=)，再利用beautifulsoup及requests等模組爬取搜尋結果的超連結，把連結傳回使用者。

4. 即時聊天：（設定只能和一個人聊天）

收到"!chat"後，將user存在一個紀錄使用者跟其正在使用的功能的table，之後收到該使用者訊息時，其所有傳送的内容會顯示在終端機上，其中内容如果包含其他功能的指令則不會觸發功能，在收到"!bye"之後便結束聊天。為了避免另一個使用者進入聊天，在class裡會設一個chatting的布林值，True的話代表有人在聊天中，其他人傳送"!chat"都會被自動忽略。如果在非聊天的情況下除了輸入"exit"會結束程式外，在終端機打字不會有作用。

遭遇的挑戰及解法：

剛開始在嘗試擷取傳送者的名稱時，原本想用冒號分割訊息，但過程中覺得作法很複雜，程式碼不容易理解。後來在做爬蟲時發現regular expression用來尋找片段文字很方便，做起來讓程式碼簡單許多，也比較直觀

在做chat時，發現如果在打字的過程中別人傳訊息來的話，還沒傳出的内容會被覆蓋，但是在按下enter後被覆蓋的自又會被傳出去。原本想用ANSI/VT100控制碼修改終端機的顯示方式以達到同時收發訊息的效果，但因為無法保留未鍵入的訊息而無法完成目標。

反思：

使用socket不難，但是要讓兩台電腦同步接收及發送訊息的話還需要研究網路以及作業系統方面的技巧，不然無法完成理想中的功能。而學會使用socket之後，未來能將自己做的小遊戲改成連線版本，提升程式的功能。