```
from nltk.chat.util import Chat, reflections

pairs = [
        [
            r"my name is(.*)",
            ["Hello %1, how are you today?",]
        ],
        [
            r"what is your name?",
            ["My name is Jarvis and I will help you with your finacial queries today.",]
        ],
        [
            r"where to put(.*)money",
            ["Basically there are many options to invest-\n 1. Regional and \n 2. Stocks.\nIn which section would you like to invest?",]
        ],
        [
            r"1|Regional(.*)",
            ["There are many- SBI,HSBC,DB. Which bank would you like to go for?",]
        ],
        [
            r"SBI(.*)",
            ["SBI offers 10 percent Interest.",]
        ],
        [
            r"DB(.*)",
            ["DB offers 09 percent Interest.",]
        ],
        [
            r"HSBC(.*)",
            ["HSBC offers 11 percent Interest.",]
        ],
        [
            r"2|(.*)Stocks(.*)",
            ["We have 2 companies to offer: 1. AAA 2. BBB.\n choose any one to know more.\n",]
        ],
        [
            r"AAA(.*)",
            ["The company AAA has a ROI = 11 percent",]
        ],
        [
            r"BBB(.*)",
            ["The company BBB has a ROI = 15 percent",]
        ],
        [
            r"hi|hey|hello(.*)",
            ["Hello", "Hey there",]
        ],
        [
            r"exit|quit|q",
            ["Signing out, see you again ^_^",]
        ],
    ]

def chatbot():
    '''Chat is the class that contains all the logic to be used by the chatbot,
    reflections is a predefined dictionary containing a set of input values and it's corresponding output values
    '''
    print("Booting up.....\nHey there. I'm a simple ChatBot made without ML, only using NLTK library.")
    print("Please type in English language (lower case) what you want to ask me.\nPress enter Q to exit")
    chat = Chat(pairs, reflections)
    chat.converse()

if __name__ == "__main__":
    chatbot()
```

Executing (7s) Cell › chatbot() › converse() › raw_input() › _input_request() › recv() › recv_multipart()