CREATING CHATBOT USING PYTHON

Steps to solve the problem:

1. Prepare the Dependencies

The first step in creating a Chatbot in Python with the ChatterBot library is to install the library in your system. It is best if you create and use a new Python virtual environment for the installation. To do so, you have to write and execute this command in your Python terminal:

pip install chatterbot

pip installchatterbot_corpus

2. Import Classes

Importing classes is the second step in the Python Chatbot creation process. All you need to do is import two classes – ChatBot from chatterbot and ListTrainer from chatterbot.trainers. To do this, you can execute the following command:

from chatter bot import ChatBot

from chatterbot.trainers import ListTrainer

3. Create and Train the Chatbot

This is the third step on creating chatbot in python. The chatbot you are creating will be an instance of the class "ChatBot." After creating a new ChatterBot instance, you can train the bot to improve its performance. Training ensures that the bot has enough knowledge to get started with specific responses to specific inputs. You have to execute the following command now:

my_bot = ChatBot (name='PyBot', read_only=True, logic_adapters= ['chatterbot.logic.MathematicalEvaluation', 'chatterbot.logic.BestMatch'])

Here, the argument represents the name of your Python chatbot. If you wish to disable the bot's ability to learn after the training, you can include the "read_only=True" command. The command "logic_adapters" denotes the list of adapters used to train the chatbot. While the "chatterbot.logic.MathematicalEvaluation" helps the bot to solve math problems, the "chatterbot.logic.BestMatch" helps it to choose the best match from the list of responses already provided. Since you have to provide a list of responses, you can do it by specifying the lists of strings that can be later used to train your Python chatbot and find the best match for each query.

4. Communicate with the Python Chatbot

To interact with your Python chatbot, you can use the .get_response() function. This is how it should look while communicating:

>>>print(my_bot.get_response("hi"))

How do you do?

5. Train your Python Chatbot with a Corpus of Data

In this last step of how to make a Chatbot in Python, for training your python Chatbot even further, you can use an existing corpus of data. Here's an example of how to train your Python Chatbot with a corpus of data provided by the bot itself:

```
from chatterbot.trainers import ChatterBotCorpusTrainer

corpus_trainer = ChatterBotCorpusTrainer(my_bat).

corpus__trainer.train('chatterbot.corpus.english')
```