# Assisted Practice: Create AIML Patterns for QnA on Mental Wellness

In this demo, we will show you how to create a conversation flow for your chatbot using AIML and NLP that can be used in healthcare to create QnA for mental wellness.

AIML was developed by Richard Wallace. He made a bot called A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) which won several Artificial Intelligence awards. Interestingly, one of Turing tests to look for Artificial Intelligence is to have a human chat with a bot through a text interface for several minutes and see if they thought it was a human. AIML is a form of XML that defines rules for matching patterns and determining responses.

Let’s look at the steps that needs to be performed in detail:

**Step 1:** **Load AIML Files**

Data folder contains all the AIML files.

Each AIML file contains the conversation pattern which the kernel will load for chatting.

**Note:** Kernel object is the public interface for the AIML interpreter. "learn" method loads the contents of an AIML file into the kernel. While "respond" method is used to get the response from the learned AIML file, "LEARN AIML" is the pattern that k.respond from conversation.py calls. The tag loads the AIML file for a response.

**Step 2:** **Load the files to train**

Code to load the files to train is saved as learningFilesList.aiml

<aiml version="1.0">

<category>

<pattern>LOAD AIML</pattern>

<template>

<!-- Load standard AIML set -->

<learn>data/\*.aiml</learn>

</template>

</category>

</aiml>

**Step 3:** **Python Program**

Now we will write a code for loading and running the bot.

import os

import aiml

from autocorrect import spell

BRAIN\_FILE="D:/Bot/aiml\_pretrained\_model.dump"

k = aiml.Kernel()

if os.path.exists(BRAIN\_FILE):

print("Loading from brain file: " + BRAIN\_FILE)

k.loadBrain(BRAIN\_FILE)

else:

print("Parsing aiml files")

k.bootstrap(learnFiles="D:/Bot/learningFileList.aiml", commands="load aiml")

print("Saving brain file: " + BRAIN\_FILE)

k.saveBrain(BRAIN\_FILE)

while True:

query = input("User > ")

query = [spell(w) for w in (query.split())]

question = " ".join(query)

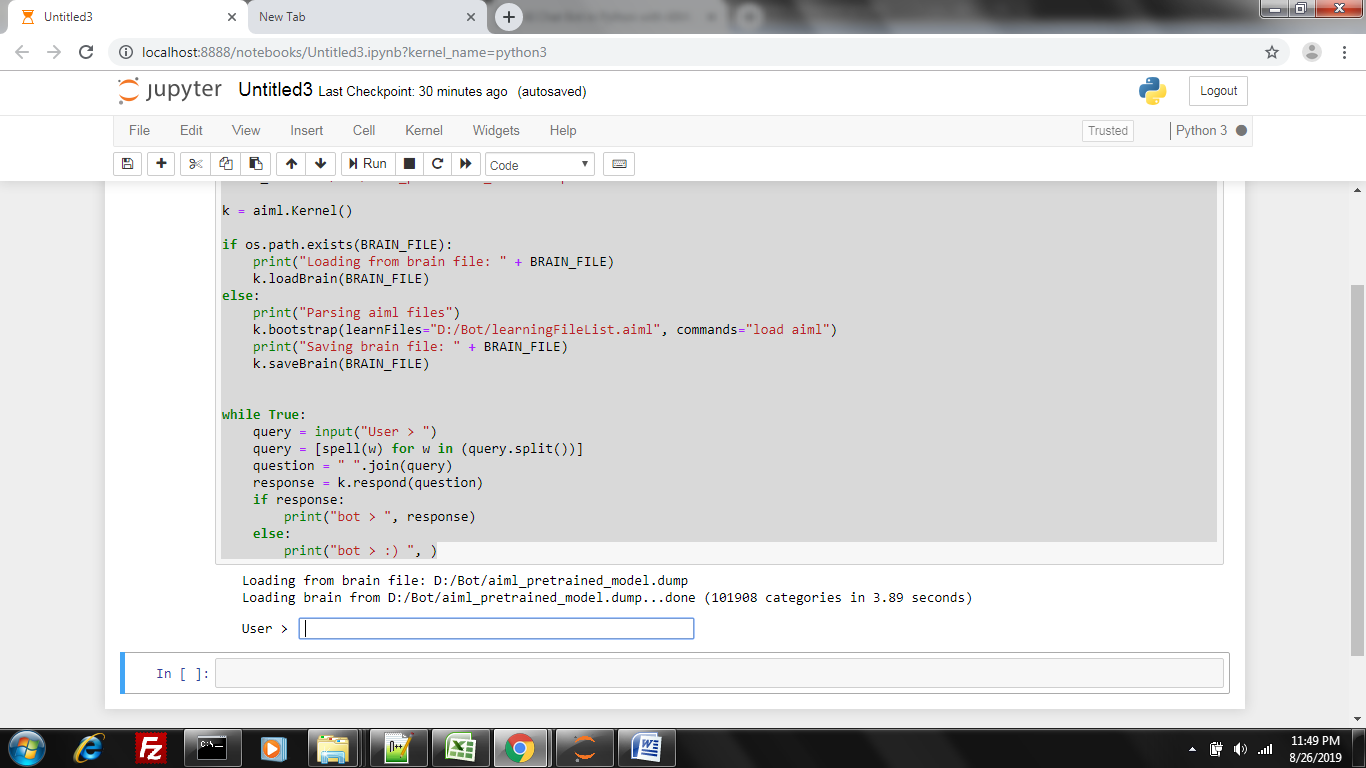
response = k.respond(question)

if response:

print("bot > ", response)

else:

print("bot > :) ", )

****

**Step 4: Check the sample output**

Enter some message and check the auto reply as shown below:

