Building a Basic Chatbot with Python's NLTK Library Spardha Python in Plain English

In [1]:
```
import io
import random
import string
import numpy as np

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# import warnings
# warnings.filterwarnings("ignore")

import nltk
from nltk.stem import WordNetLemmatizer
from nltk.corpus import shakespeare, gutenberg, wordnet2021
```

In [7]:
```
training_corpus = wordnet2021.raw().casefold()

sent_tokens = nltk.sent_tokenize(training_corpus) # converts to list of sentenc
word_tokens = nltk.word_tokenize(training_corpus) # converts to list of words
```

In [8]:
```
lemmer = nltk.stem.WordNetLemmatizer()
#WordNet is a semantically-oriented dictionary of English included in NLTK.

def lem_tokens(tokens):
    return [lemmer.lemmatize(token) for token in tokens]

def lem_normalize(text):
    remove_punct_dict = dict((ord(punct), "\w+") for punct in string.punctuatio
    return lem_tokens(nltk.word_tokenize(text.casefold().translate(remove_punct
```

In [9]:
```
GREETING_INPUTS = ("hello", "hi", "greetings", "sup", "what's up", "hey", "yo")
GREETING_RESPONSES = ["hi", "hey", "*nods*", "hi there", "hello", "I am glad! Y

def greeting(sentence):
    for word in sentence.split():
        if word.lower() in GREETING_INPUTS:
            return random.choice(GREETING_RESPONSES)
```

In [10]:
```
def response(user_response):
    spar_response = ""
    sent_tokens.append(user_response)

    tfidf_vec = TfidfVectorizer(tokenizer=lem_normalize, stop_words="english")
    tfidf = tfidf_vec.fit_transform(sent_tokens)

    vals = cosine_similarity(tfidf[-1], tfidf)
    idx, flat = vals.argsort()[0][-2], vals.flatten()
    flat.sort()
    req_tfidf = flat[-2]

    if req_tfidf == 0:
        spar_response += "I don't understand you"
    else:
        spar_response += sent_tokens[idx]
```

```
    return spar_response
```

In [ ]:
```
flag = True
print("Spar: My name is Spar. I will answer your queries about Chatbots. If you

while flag:
    user_response = input()
    user_response = user_response.lower()

    if user_response == "bye":
        flag = False
        continue

    if user_response in["thanks", "thank you"]:
        flag = False
        print("Spar: You are welcome...")
        continue

    if greeting(user_response) != None:
        print("Spar: " + greeting(user_response))
        continue

    print("Spar: ", end="")
    print(response(user_response))
    sent_tokens.remove(user_response)
```

```
Spar: My name is Spar. I will answer your queries about Chatbots. If you want
to exit, type Bye!

Spar: the last day of operations) is given."
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```
dir(nltk.corpus)
```

In [ ]:

In [ ]: