

In [1]:

```
from textblob import TextBlob
import nltk
from textblob import Word
import sys
```

In [2]:

```
def parse(string):
    """
    Parse a paragraph. Devide it into sentences and try to generate quesstions f
    rom each sentences.
    """

    try:
        txt = TextBlob(string)
        # Each sentence is taken from the string input and passed to genQuestion
        () to generate questions.
        for sentence in txt.sentences:
            genQuestion(sentence)

    except Exception as e:
        raise e
```

In [14]:

```

def genQuestion(line):
    """
    outputs question from the given text
    """

    if type(line) is str:      # If the passed variable is of type string.
        line = TextBlob(line) # Create object of type textblob.blob.TextBlob

    bucket = {}                # Create an empty dictionary

    for i,j in enumerate(line.tags): # line.tags are the parts-of-speech in Eng
        if j[1] not in bucket:
            bucket[j[1]] = i # Add all tags to the dictionary or bucket variable

    question = ''              # Create an empty string

    # These are the english part-of-speech tags used in this demo program.
    #.....
    # NNS      Noun, plural
    # JJ       Adjective
    # NNP      Proper noun, singular
    # VBG      Verb, gerund or present participle
    # VBN      Verb, past participle
    # VBZ      Verb, 3rd person singular present
    # VBD      Verb, past tense
    # IN       Preposition or subordinating conjunction
    # PRP      Personal pronoun
    # NN       Noun, singular or mass
    #.....

    # Create a list of tag-combination

    l1 = ['NNP', 'VBG', 'VBZ', 'IN']
    l2 = ['NNP', 'VBG', 'VBZ']

    l3 = ['PRP', 'VBG', 'VBZ', 'IN']
    l4 = ['PRP', 'VBG', 'VBZ']
    l5 = ['PRP', 'VBG', 'VBD']
    l6 = ['NNP', 'VBG', 'VBD']
    l7 = ['NN', 'VBG', 'VBZ']

    l8 = ['NNP', 'VBZ', 'JJ']
    l9 = ['NNP', 'VBZ', 'NN']

    l10 = ['NNP', 'VBZ']
    l11 = ['PRP', 'VBZ']
    l12 = ['NNP', 'NN', 'IN']
    l13 = ['NN', 'VBZ']

    # With the use of conditional statements the dictionary is compared with the
    list created above

```

```

    if all(key in bucket for key in l1): #'NNP', 'VBG', 'VBZ', 'IN' in sentence.
        question = 'What' + ' ' + line.words[bucket['VBZ']] + ' ' + line.words[bucket['NNP']] + ' ' + line.words[bucket['VBG']] + '?'

    elif all(key in bucket for key in l2): #'NNP', 'VBG', 'VBZ' in sentence.
        question = 'What' + ' ' + line.words[bucket['VBZ']] + ' ' + line.words[bucket['NNP']] + ' ' + line.words[bucket['VBG']] + '?'

    elif all(key in bucket for key in l3): #'PRP', 'VBG', 'VBZ', 'IN' in sentence.
        question = 'What' + ' ' + line.words[bucket['VBZ']] + ' ' + line.words[bucket['PRP']] + ' ' + line.words[bucket['VBG']] + '?'

    elif all(key in bucket for key in l4): #'PRP', 'VBG', 'VBZ' in sentence.
        question = 'What' + ' ' + line.words[bucket['PRP']] + ' ' + ' does ' + line.words[bucket['VBG']] + ' ' + line.words[bucket['VBG']] + '?'

    elif all(key in bucket for key in l7): #'NN', 'VBG', 'VBZ' in sentence.
        question = 'What' + ' ' + line.words[bucket['VBZ']] + ' ' + line.words[bucket['NN']] + ' ' + line.words[bucket['VBG']] + '?'

    elif all(key in bucket for key in l8): #'NNP', 'VBZ', 'JJ' in sentence.
        question = 'What' + ' ' + line.words[bucket['VBZ']] + ' ' + line.words[bucket['NNP']] + '?'

    elif all(key in bucket for key in l9): #'NNP', 'VBZ', 'NN' in sentence
        question = 'What' + ' ' + line.words[bucket['VBZ']] + ' ' + line.words[bucket['NNP']] + '?'

    elif all(key in bucket for key in l11): #'PRP', 'VBZ' in sentence.
        if line.words[bucket['PRP']] in ['she', 'he']:
            question = 'What' + ' does ' + line.words[bucket['PRP']].lower() + ' ' + line.words[bucket['VBZ']].singularize() + '?'

    elif all(key in bucket for key in l10): #'NNP', 'VBZ' in sentence.
        question = 'What' + ' does ' + line.words[bucket['NNP']] + ' ' + line.words[bucket['VBZ']].singularize() + '?'

    elif all(key in bucket for key in l13): #'NN', 'VBZ' in sentence.
        question = 'What' + ' ' + line.words[bucket['VBZ']] + ' ' + line.words[bucket['NN']] + '?'

    # When the tags are generated 's is split to ' and s. To overcome this issue.
    if 'VBZ' in bucket and line.words[bucket['VBZ']] == "'":
        question = question.replace(" ' ", "'s ")

    # Print the generated questions as output.
    if question != '':
        print('\n', 'Question: ' + question )

```

In [16]:

```
def main():

    filehandle = open('file.txt', 'r')
    textinput = filehandle.read()
    print('\n-----INPUT TEXT-----\n')
    print(textinput, '\n')
    print('\n-----INPUT END-----\n')

    parse(textinput)

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

-----INPUT TEXT-----

Bansoori is an Indian classical instrument. Akhil plays Bansoori and Guitar. Puliogare is a South Indian dish made of rice and tamarind. Priya writes poems. Osmosis is the movement of a solvent across a semipermeable membrane toward a higher concentration of solute. In biological systems, the solvent is typically water, but osmosis can occur in other liquids, supercritical liquids, and even gases. When a cell is submerged in water, the water molecules pass through the cell membrane from an area of low solute concentration to high solute concentration. For example, if the cell is submerged in saltwater, water molecules move out of the cell. If a cell is submerged in freshwater, water molecules move into the cell. Raja-Yoga is divided into eight steps, the first is Yama -- non - killing, truthfulness, non - stealing, continence, and non - receiving of any gifts. Next is Niyama -- cleanliness, contentment, austerity, study, and self - surrender to God.

-----INPUT END-----

Question: What is Bansoori?

Question: What does Akhil play?

Question: What is Puliogare?

Question: What does Priya write?

Question: What is Osmosis?

Question: What is solvent?

Question: What is cell?

Question: What is example?

Question: What is cell?

Question: What is Raja-Yoga?

Question: What is Niyama?

In []:

In []: