Lab Assessment - Working with text

Welcome to our first lab!!

Complete the tasks described in bold below by typing the code in the cells.

Working with f-Strings

1. Print an f-string that displays `We are learning about Natural Language Processing in Business at MT SAC` using the variables provided.

In [3]:
```
school = 'MT SAC'
full_text = 'Natural Language Processing in Business'

# Enter your code here:
print(f'We are learning about {full_text} at {school}')
```

```
We are learning about Natural Language Processing in Business at MT SAC
```

In [ ]:
```

```

Create a variable called name and assing your name to it. Then, print and f-string that displays "My name is ("your name)"

Example: My name is Angel.

In [5]:
```
name = 'William Yeh'

# Print the f-string
print(f'My name is {name}.')
```

```
My name is William Yeh.
```

Now print your name around single quotes to get the string representation.

Example: My name is 'Angel'

In [7]:
```
name = 'William Yeh'

print(f"My name is '{name}'")
```

```
My name is 'William Yeh'
```

Note: A date in Python is not a data type of its own, but we can import a module named datetime to work with dates as date objects.

Using the datetime module, print today's date as follows:

February 24, 2022

In [8]:
```
import datetime

# Get today's date
today = datetime.date(2024, 8, 31)

# Format the date as "Month day, Year"
formatted_date = today.strftime("%B %d, %Y")

# Print the formatted date
print(formatted_date)
```

```
August 31, 2024
```

Working with Files

Create a file in the current working directory called contacts.txt and enter the following text in the first row:

First_Name Last_Name, Title, Extension, Email

hint: use %%writefile

In [9]:
```
%%writefile contacts.txt
First_Name Last_Name, Title, Extension, Email
```

```
Writing contacts.txt
```

Open the file and use .read() to save the contents of the file to a string called fields. Make sure the file is closed at the end.

In [10]:
```
# Write your code here:
# hint: use with open
with open('contacts.txt', 'r') as file:
    fields = file.read()



# Run fields to see the contents of contacts.txt:
print(fields)
```

```
First_Name Last_Name, Title, Extension, Email
```

Working with PDF Files

In [13]:
```
# Install the Library pyPDF2 - if needed
# !pip install pyPDF2
!pip install PyPDF2
```

```
Collecting PyPDF2
  Downloading pypdf2-3.0.1-py3-none-any.whl.metadata (6.8 kB)
Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
   ---------------------------------------- 0.0/232.6 kB ? eta -:--:--
   - -------------------------------------- 10.2/232.6 kB ? eta -:--:--
   ------------------------ ------------- 153.6/232.6 kB 2.3 MB/s eta 0:00:01
   ---------------------------------------- 232.6/232.6 kB 2.9 MB/s eta 0:00:00
Installing collected packages: PyPDF2
Successfully installed PyPDF2-3.0.1
```

4. Use PyPDF2 to open the file `Business_Proposal.pdf`. Extract the text of page 2.

In [14]:
```python
# Perform import
import PyPDF2

# Open the file as a binary object
with open('Business_Proposal.pdf', 'rb') as file:
    # Use PyPDF2 to read the text of the file
    pdf_reader = PyPDF2.PdfReader(file)

    # Get the text from page 2 (CHALLENGE: Do this in one step!)
    page_two_text = pdf_reader.pages[1].extract_text()

    # Close the file is handled by 'with' statement

# Print the contents of page_two_text
print(page_two_text)
```

```
AUTHORS:
Amy Baker, Finance Chair, x345, abaker@ourcompany.com
Chris Donaldson, Accounting Dir., x621, cdonaldson@ourcompany.com
Erin Freeman, Sr. VP, x879, efreeman@ourcompany.com
```

Open the file `contacts.txt` in append mode. Add the text of page 2 from above to `contacts.txt`.

In [16]:
```python
# Hint: use with open
#Enter your code here:
import PyPDF2

# Step 1: Extract the text from page 2 of Business_Proposal.pdf
with open('Business_Proposal.pdf', 'rb') as file:
    pdf_reader = PyPDF2.PdfReader(file)
    page_two_text = pdf_reader.pages[1].extract_text()

# Step 2: Open contacts.txt in append mode and add the extracted text
with open('contacts.txt', 'a') as file:
    file.write("\n" + page_two_text)
```

In [ ]:
```python

```

Pluralization and Singularization using textblob

Import Word from textblob

In [19]:
```
!pip install textblob

from textblob import Word

# Example word
word = Word('apple')

# Pluralize the word
plural_word = word.pluralize()
print(f'Plural: {plural_word}')

# Singularize the word
singular_word = Word('apples').singularize()
print(f'Singular: {singular_word}')
```

```
Requirement already satisfied: textblob in c:\users\william\anaconda3\lib\site-packa
ges (0.18.0.post0)
Requirement already satisfied: nltk>=3.8 in c:\users\william\anaconda3\lib\site-pack
ages (from textblob) (3.8.1)
Requirement already satisfied: click in c:\users\william\anaconda3\lib\site-packages
(from nltk>=3.8->textblob) (8.1.7)
Requirement already satisfied: joblib in c:\users\william\anaconda3\lib\site-package
s (from nltk>=3.8->textblob) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\william\anaconda3\lib\sit
e-packages (from nltk>=3.8->textblob) (2023.10.3)
Requirement already satisfied: tqdm in c:\users\william\anaconda3\lib\site-packages
(from nltk>=3.8->textblob) (4.65.0)
Requirement already satisfied: colorama in c:\users\william\anaconda3\lib\site-packa
ges (from click->nltk>=3.8->textblob) (0.4.6)
Plural: apples
Singular: apple
```

create a variable for the word "mouse" and call it mouse

In [20]:
```
# Create a variable for the word "mouse"
mouse = Word('mouse')

# Optionally, you can perform actions with the variable
print(f'Original word: {mouse}')
print(f'Pluralized: {mouse.pluralize()}')
print(f'Singularized: {mouse.singularize()}')
```

```
Original word: mouse
Pluralized: mice
Singularized: mouse
```

Pluralize the word "mouse"

In [21]:
```
# Create a variable for the word "mouse"
mouse = Word('mouse')

# Pluralize the word
plural_mouse = mouse.pluralize()
```

```
print(f'Plural of "mouse": {plural_mouse}')
```

```
Plural of "mouse": mice
```

create a variable for the word "tooth" and call it tooth

In [22]:
```
# Create a variable for the word "tooth"
tooth = Word('tooth')

# Optionally, you can perform actions with the variable
print(f'Original word: {tooth}')
print(f'Pluralized: {tooth.pluralize()}')
print(f'Singularized: {tooth.singularize()}')
```

```
Original word: tooth
Pluralized: teeth
Singularized: tooth
```

Pluralize the word "tooth"

In [23]:
```
# Create a variable for the word "tooth"
tooth = Word('tooth')

# Pluralize the word
plural_tooth = tooth.pluralize()

print(f'Plural of "tooth": {plural_tooth}')
```

```
Plural of "tooth": teeth
```

create a variable for the word "mice" and call it mice

In [24]:
```
# Create a variable for the word "mice"
mice = Word('mice')

# Optionally, you can perform actions with the variable
print(f'Original word: {mice}')
print(f'Singularized: {mice.singularize()}')
```

```
Original word: mice
Singularized: mouse
```

Singularize the word "mice"

In [25]:
```
# Create a variable for the word "mice"
mice = Word('mice')

# Singularize the word
singular_mice = mice.singularize()

print(f'Singular of "mice": {singular_mice}')
```

```
Singular of "mice": mouse
```

Import TextBlob

In [26]:
```
from textblob import TextBlob
```

Using textblob - pluralize the words 'lion tiger fish mouse'

In [27]:
```
from textblob import Word

# List of words to pluralize
words = ['lion', 'tiger', 'fish', 'mouse']

# Create Word objects and pluralize them
plural_words = [Word(word).pluralize() for word in words]

# Print the pluralized words
print('Pluralized words:', plural_words)
```

```
Pluralized words: ['lions', 'tigers', 'fish', 'mice']
```

Spell Checking and Correction

import Word from textblob

In [28]:
```
from textblob import TextBlob
```

create a variable called "word" and assing the word "confidense" usign the Word method

In [29]:
```
# hint: word = Word('confidense')

# Create a variable called "word" and assign it the word "confidense"
word = Word('confidense')

# Optionally, you can correct the spelling and print the result
corrected_word = word.correct()
print(f'Corrected word: {corrected_word}')
```

```
Corrected word: confidence
```

Check the spelling or the workd "confidense"

In [24]:
```
word.spellcheck()
```

Out[24]:
```
[('confidence', 1.0)]
```

Correct the spelling

In [30]:
```
# Correct the spelling

# Create a variable called "word" and assign it the word "confidense"
word = Word('confidense')

# Check the spelling
```

```
spellcheck_results = word.spellcheck()
print(f'Spellcheck results: {spellcheck_results}')

# Correct the spelling
corrected_word = word.correct()
print(f'Corrected word: {corrected_word}')
```

```
Spellcheck results: [('confidence', 1.0)]
Corrected word: confidence
```

Correct the spelling in this sentence `'I donot know hwo to spel sentense'`.

Assing the text: 'I donot know how to spel sentense' to a variable called "sentence"

In [32]:
```
from textblob import TextBlob

# Assign the text to a variable called "sentence"
sentence = 'I donot know hwo to spel sentense'

# Create a TextBlob object with the sentence
blob = TextBlob(sentence)

# Correct the spelling in the sentence
corrected_sentence = blob.correct()

# Print the corrected sentence
print(f'Corrected sentence: {corrected_sentence}')
```

```
Corrected sentence: I dont know who to spell sentence
```

Correct the spelling in the sentence

In [33]:
```
# Print the corrected sentence
print(f'Corrected sentence: {corrected_sentence}')
```

```
Corrected sentence: I dont know who to spell sentence
```

Get the Sentiment (Polarity and Subjectivity) of the phrase: "I love homework"

In [41]:
```
from textblob import TextBlob
TextBlob("I love homework").sentiment
```

Out[41]:
```
Sentiment(polarity=0.5, subjectivity=0.6)
```

Do you agree with the result?

Yes, the sentiment analysis result provided by TextBlob seems reasonable:

Polarity: 0.5 indicates a positive sentiment. In the context of the phrase "I love homework," a positive polarity is expected because the word "love" conveys a positive sentiment.
Subjectivity: 0.6 suggests that the statement is somewhat subjective. The phrase expresses a personal feeling or opinion about homework, which aligns with the subjectivity score.

So, yes, TextBlob got it correct.

Get the polarity scores

In [34]:
```
# install vaderSentiment
#!pip install vaderSentiment
```

In [38]:
```
# !pip install vaderSentiment
!pip install vaderSentiment

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# Initialize the VADER sentiment analyzer
analyser = SentimentIntensityAnalyzer()

# Get the polarity scores for the phrase
scores = analyser.polarity_scores("OMG! This class is amazing")

# Print the polarity scores
print(scores)
```

```
Requirement already satisfied: vaderSentiment in c:\users\william\anaconda3\lib\site
-packages (3.3.2)
Requirement already satisfied: requests in c:\users\william\anaconda3\lib\site-packa
ges (from vaderSentiment) (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\william\anaconda
3\lib\site-packages (from requests->vaderSentiment) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\william\anaconda3\lib\site-p
ackages (from requests->vaderSentiment) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\william\anaconda3\lib
\site-packages (from requests->vaderSentiment) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\william\anaconda3\lib
\site-packages (from requests->vaderSentiment) (2024.2.2)
{'neg': 0.0, 'neu': 0.494, 'pos': 0.506, 'compound': 0.6239}
```

Share the name of a song or a poem and translate it to two other languages

"Yesterday" by The Beatles. We'll translate it into Spanish and French.

Original Song Title: "Yesterday"

Translation to Spanish Title: "Ayer" Translation to French Title: "Hier"

In [ ]: