Meg McCauley

Homework #3

Project Overview:

My goal with this project was to create a program that would turn any text into a Mad Lib. A Mad Lib is something where the user inputs nouns, verbs, adjectives, etc. and that replace previous parts of speech.

My approach started with first being able to identify parts of speech in a text document. I started off with the full text books from the Project Gutenberg website. However, it was then determined that this assignment would be better received with a short story or poem as the base.

After extensive internet searching, I finally happened upon three plain text documents containing short poems. These documents were originally intended for testing a Pig Latin Translator coding assignment as a part of Drexel University's Computer Science 171 course.
(Source: https://www.cs.drexel.edu/~introcs/F2K/placement/PracticeProg171.html)

Implementation:

My first step was to gather the data from the three documents. I utilized the pickling function so that I could save my data and not have to pull it from the web each time I ran my program.
Here are the steps I took to create my code, including my unit testing.

1. Pickle and print each of the three snippets
2. Create a small hard coded snippet for testing
3. Classify the part of speech of each word
4. Set user input to replace each noun
5. Print final string
6. Use full snippet as input
7. Repeat steps 2-4
8. Add functionality to replace verbs

The data structures that I used the most were list and pattern.

Here's how my code works. First, I took the string of text that I got online and used the translate() function to remove any characters that were not letters or spaces. For example, I removed any commas, periods, exclamation points, quotation marks, colons, or semicolons. This was done because the functions I use later on cannot work with non-alpha characters.

Next, I ran this string through split() to change each word in the string and into an element in an array. From here, I was able to access each individual word. My next step was to use the parse() function from pattern. Parse() outputs a lot of information but I was focusing only on part of speech. In order to get only this information, I used a for loop to extract the first two letters after the /, which were coded as the part of speech. I then compared this to the strings ‘NN’ or ‘VB’, to determine if they were a noun or verb, respectively. I then asked the user to input either a noun or verb, as the conditions determined.

The final step was outputting the user’s new story in the form of the edited array of words. I did this cleanly by imploying the join() function of array, which allows you to print out each element of an array joined by something. In this example, I simply joined then using a space. Then, as a finishing touch, I included the original text of the story so the user could see how their new story compared to the original.

Example Output:

When the functionality allowed for just noun replacement, here is the result of the interaction between the user and the script.

```
Welcome to Meg's Mad Libs!:
Enter a noun: cat
Enter a noun: mouse
Enter a noun: house
Enter a noun: chocolate
Enter a noun: computer
Enter a noun: suitcase
Enter a noun: phone
Enter a noun: dinosaur
Enter a noun: pen
Enter a noun: dolphin
Enter a noun: bookcase
Enter a noun: bracelet
Enter a noun: cookie
Enter a noun: glass
Enter a noun: hat

Thanks for playing! Here's your story:
There was a young cat named mouse Who traveled much faster than house She started one
chocolate In a relative computer And returned the previous suitcase To her phone said the
```

```
dinosaur one in pen I have learned dolphin new about bookcase As my bracelet was so great
cookie increased was my glass hat I failed to become any fatter

Original Story:
There was a young lady named Bright
Who traveled much faster than light
She started one day
In a relative way
And returned the previous night

To her friends said the Bright one in chatter
I have learned something new about matter
As my speed was so great
Much increased was my weight
Yet I failed to become any fatter
```

When the functionality was increased to the final script that provided support for replacing both nouns and verbs, here is the result of the user-script interaction.

```
Welcome to Meg's Mad Libs!:
Enter a verb: jump
Enter a noun: fence
Enter a verb: kick
Enter a noun: cat
Enter a verb: read
Enter a noun: book
Enter a verb: listen
Enter a noun: phone
Enter a noun: friend
Enter a verb: smite
Enter a noun: dragon
Enter a noun: pig
Enter a verb: scream
Enter a noun: spider
Enter a noun: trex
Enter a verb: eat
Enter a verb: drink
Enter a noun: water
Enter a noun: bracelet
Enter a noun: stapler
Enter a verb: lick
Enter a noun: lollipop
Enter a verb: bite
Enter a verb: chomp
Enter a noun: lettuce
Enter a noun: tomato
Enter a verb: smell
Enter a verb: whisper

Thanks for playing! Here's your story:
There jump a young fence kick cat Who read much faster than book She listen one phone In a relative friend And smite the previous dragon To her pig scream the spider one in trex I eat drink water new about bracelet As my stapler lick so great lollipop bite chomp my lettuce tomato I smell to whisper any fatter

Original Story:
There was a young lady named Bright
Who traveled much faster than light
She started one day
In a relative way
And returned the previous night

To her friends said the Bright one in chatter
I have learned something new about matter
As my speed was so great
Much increased was my weight
Yet I failed to become any fatter
```

Reflection:

My project was appropriately scoped and contained a lot of unit testing. I started off small, which I think really helped me be able to make it to my final product. My unit testing, as outlined above in the implementation section, is what made this code managable. My starting with a simple string and only replacing one part of speech, I was able to make sure that my code worked without adding in too many additional debuging steps.

To improve this project, I could ask the user to input a list of a certain number of nouns and verbs instead of having them prompted for each noun and verb. I could also pick only some of the nouns and verbs to change out, as the story is getting pretty full of the user’s words. At this point, the final output of the user’s story doesn’t always make a lot of sense. I could also have something to check for the singularity or plurality to make sure the verbs agree.