

**BINUS UNIVERSITY**

**BINUS INTERNATIONAL**

**Assignment Cover Letter**

**(Individual Work)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | |  | |  | |
| **Student Information**: **Surname** | | | | | **Given Name**  **Sreeya** | | **Student ID Number**  **2201816165** | |
| 1. | | **Reddy** |  | |
|  |  |
| **Course Code** | **: COMP6502** |  |  | | **Course Name** | | **: Introduction to Programming** | |
| **Class** | **: L1BC** |  |  | | **Name of Lecturer(s)** | | : Monica Hidajat | |
|  |  |  |  | |  | |  | |
| **Major** | **: CS** |  |  | |  | |  | |
| **Title of Assignment**  (if any) | : MadLib | |  |  | |  | |  | |
| **Type of Assignment**    **Submission Pattern** | **: Final Project** |  |  | |  | |  | |
| **Due Date** | **: 20-11-2018** |  |  | | **Submission Date** | | **: 19-11-2018** | |

# Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

# Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)

1. Kotrakona Harinatha Sreeya Reddy

**“MadLib”**

Name : Sreeya

ID : 2201816165

1. **Description**

**Concept:**

This program is designed to enable users to have a fun time by filling in words to complete a certain story. There are multiple stories provided in the program and the program will randomly select a certain story for the user to fill out. Furthermore, as a special feature, the program also allows the user to get a word randomly generated by the program itself by pressing the ‘r’ key.

What is a “MadLib” exactly? The MadLib was first invented by Leonard Stern and Roger Price back in 1953. They designed this activity because they had a vision, a vision to bring people closer together through this activity, and I truly believe that MadLibs are able to bridge the gap between very different people. Which is why I chose to make a MadLib for my final project.

1. **Design**

Start

Print welcome message

Input name

End

Input are you up for it y/n

n

y

Input song and rhymes or stories s/t

s

t

If input is

Anything except e

e

Input pointers y/n

y

n

Print hints

Input words or ‘r’

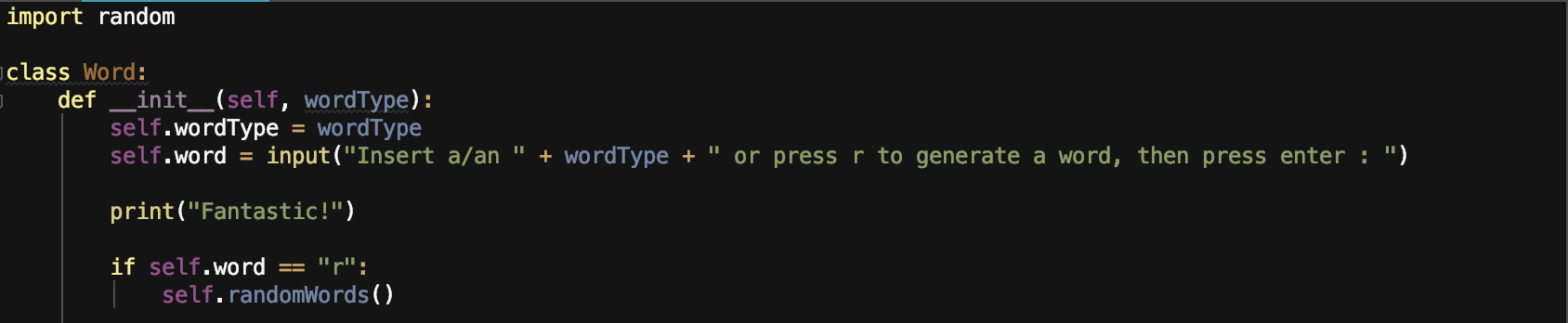
Print story

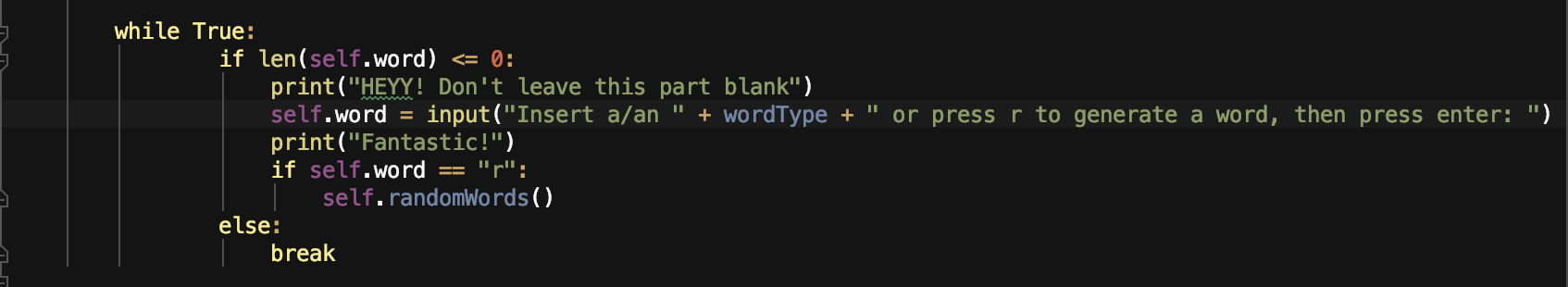
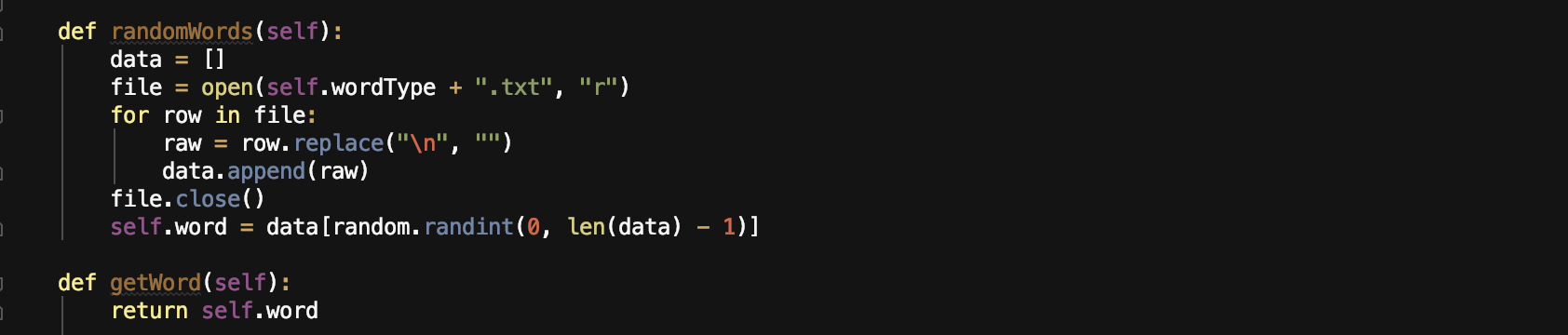
Input e to end or other key

If input is

1. **Discussion**

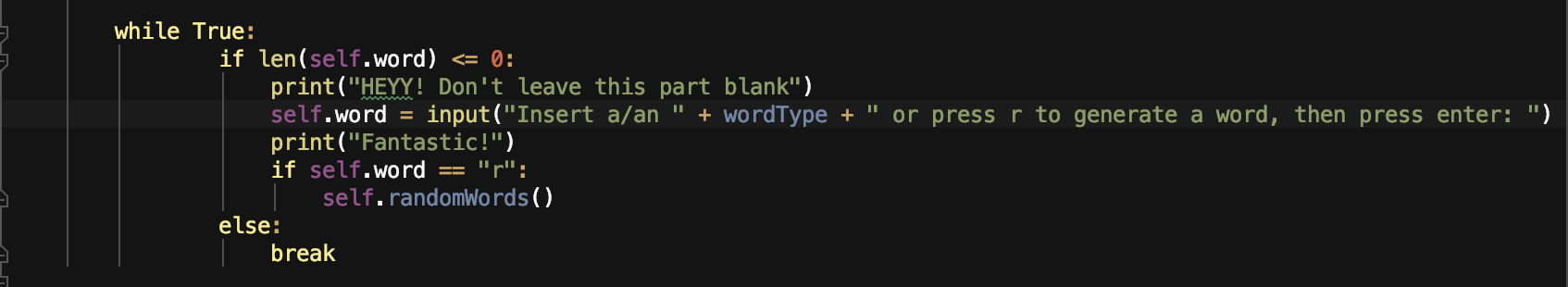
**Modules/Functions/Classes:**

One class function was used in this program. The class made the overall code much more simpler and prevented repetition. For example, with the class, there is no need to repeatedly type “input(“Insert a/an “ + wordType “ or press r to generate a word, then press enter : “) in the main code. 





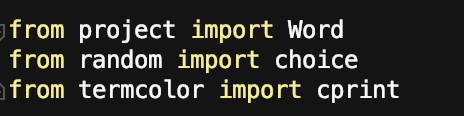
This part of the class function, randomly selects a word from a certain file -depending on which type of word the user wants- . The def randomWords(self) is where the various text files containing certain words such as “action words” is read. The new line is replaced with an empty space and that data is appended to an array which is named data.

This part of the class ensures that the user does not leave any blanks. It forces the user to input a word. The len(self.word) <= 0, checks if the length of the word is less than or equal to 0. If, the length is indeed less than or equal to 0, then the program instantly prompts the user to input the word over and over again until the user fulfils the requirements.

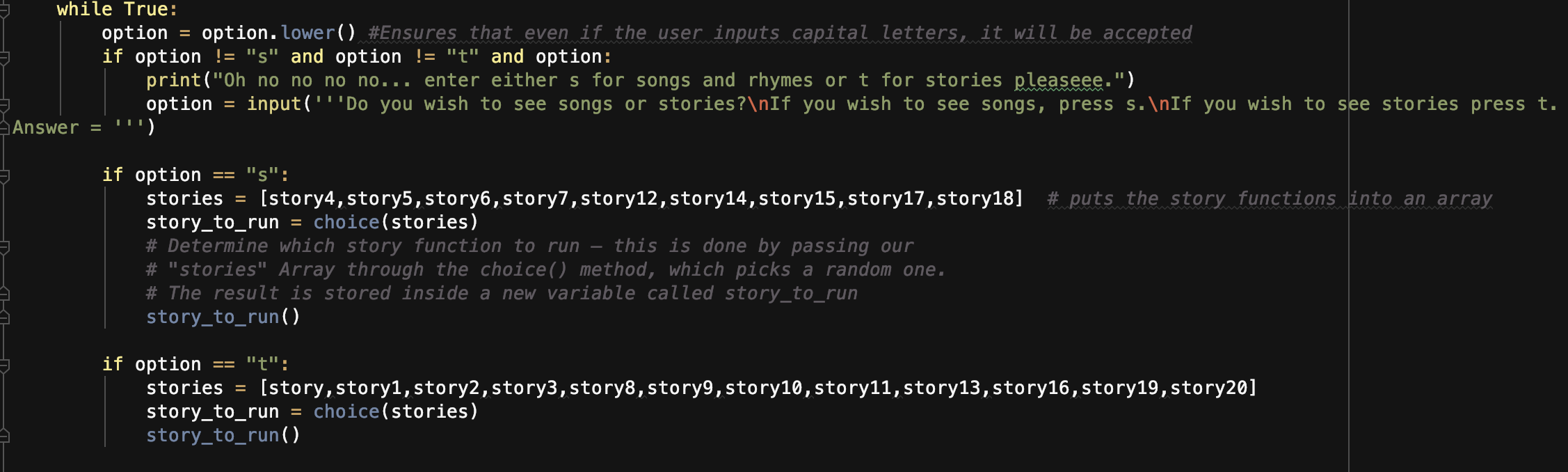
if self.word == "r":  
 self.randomWords()

This part of the class is what enables the random word to be generated if the user presses ‘r’. When self.word == “r”, then it automatically calls the randomWords functions and generates the random word.

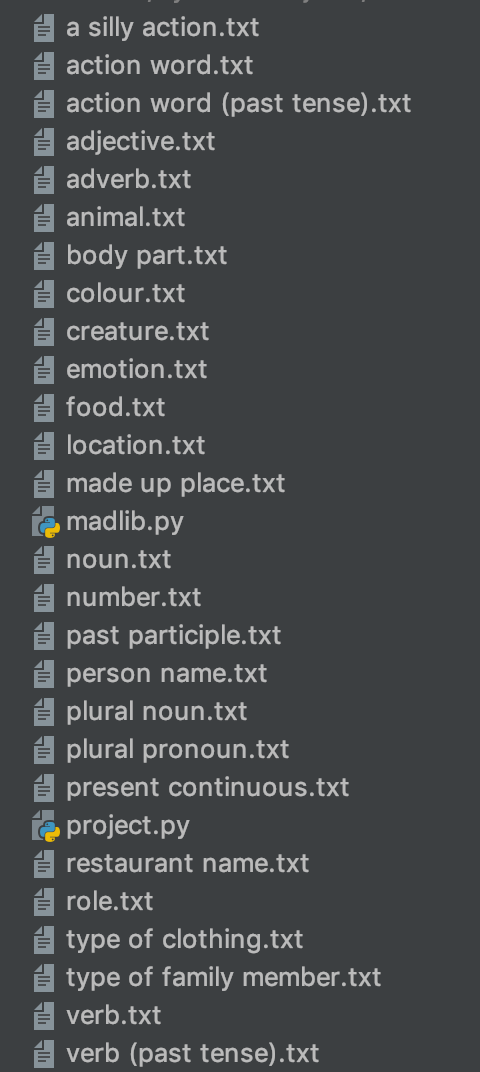
The modules that were implemented in the program are random and termcolor.

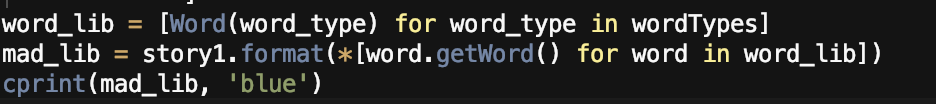
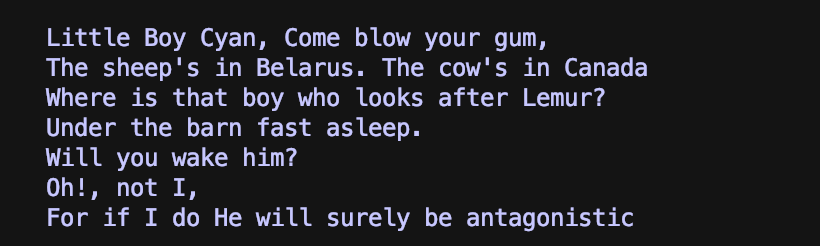


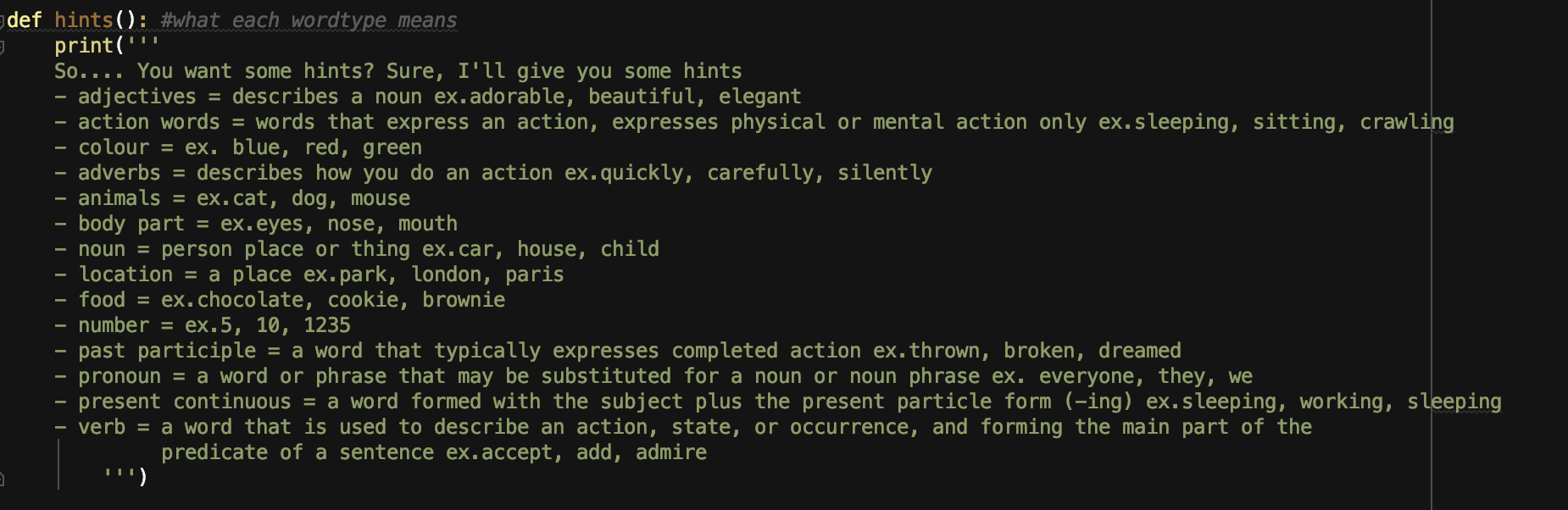
Random was the main feature used to make this project as the random module was the how the program would randomly generate a story. This is done by first putting all the story functions we have into an array. Then the program determines which story function to run by passing the “stories” array through the choice() method. The result is then stored inside a new variable “story\_to\_run”. The option.lower() allows the user to type capital letter “S”and “T” and still be considered correct.



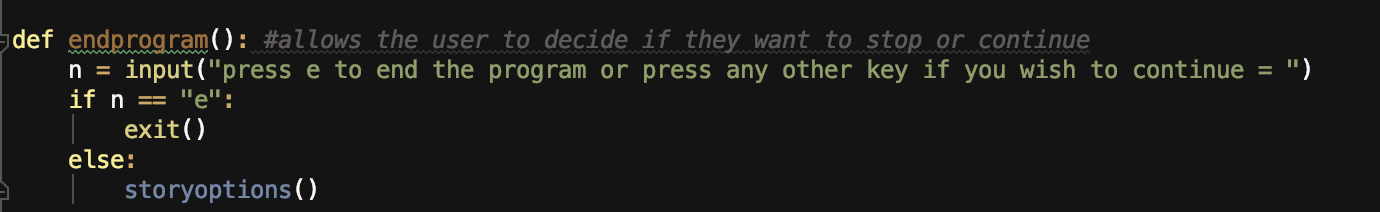
The random module was also used to generate the random word. The program has several text files such “action word” or “adjectives” and many each more, each text file containing lots of words.

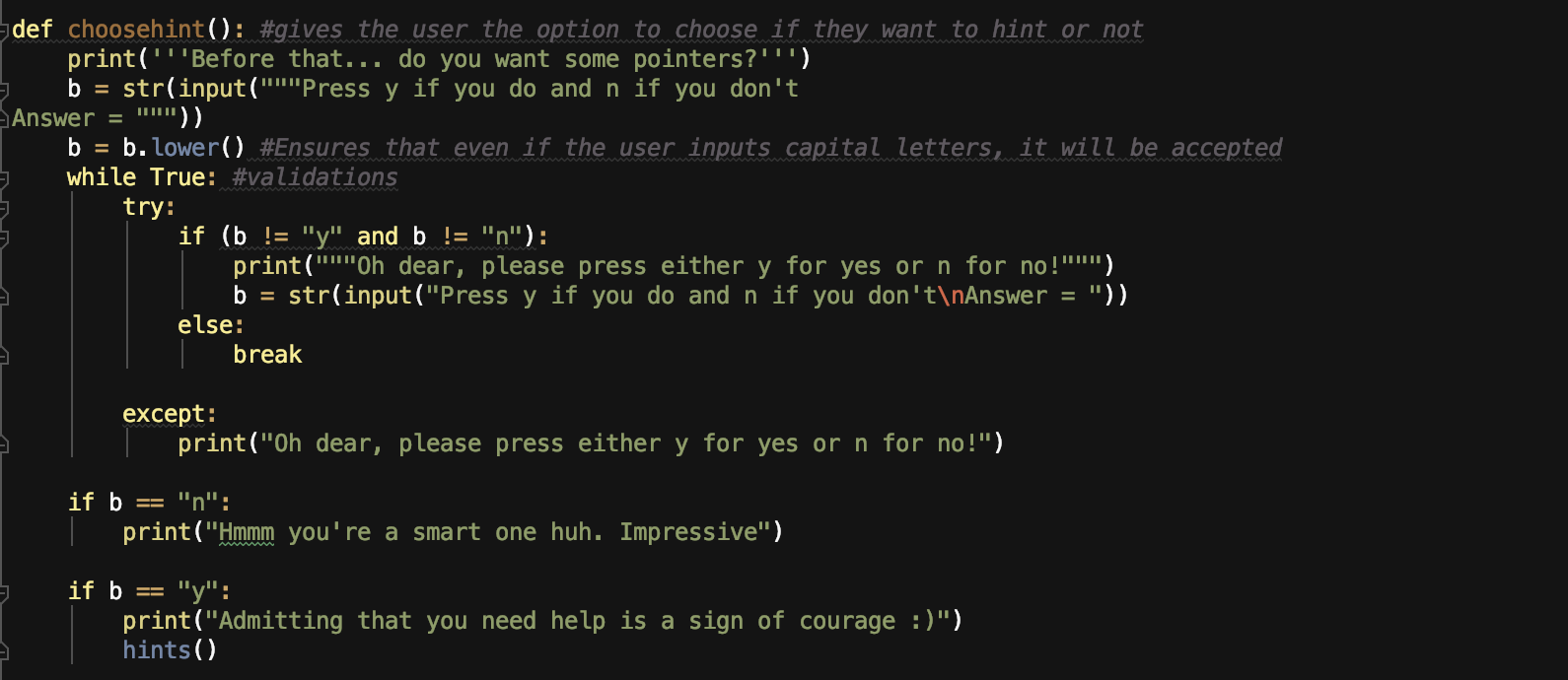


Termcolor was an additional module used to make the output of the program more aesthetically pleasing. From termcolor, cprint was imported. Cprint allows the output to be printed in color, in this case, the output is printed in blue.

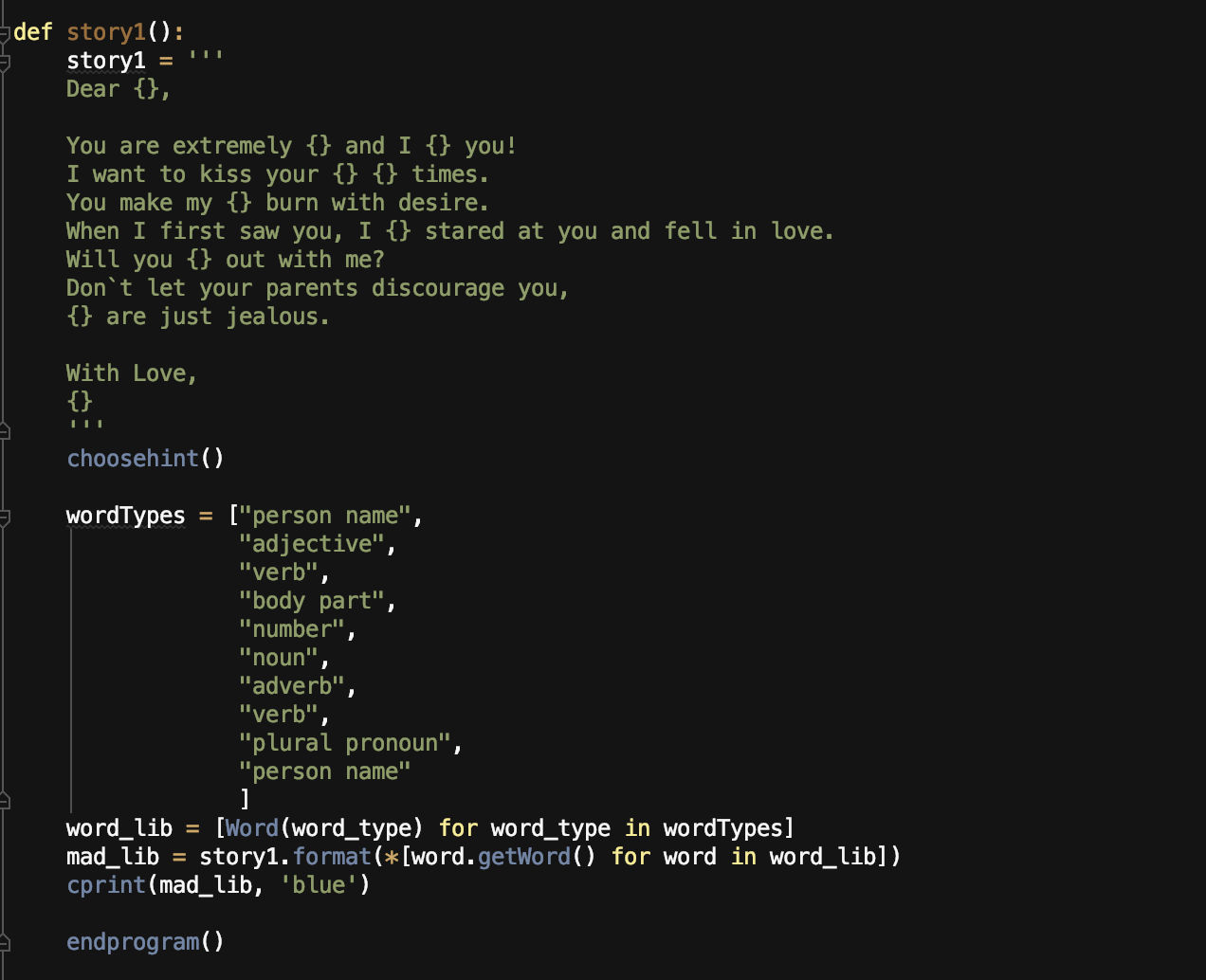
This program has lots of def functions. The def function is a code that will be reused to execute an action.







These are some of the def functions used in the program. In the end, these functions will be called when we are making the story template.



Since there are many story templates, it would be extremely tedious to type all those lines of code multiple times. Thus we put those lines of code in a def function and just call them when necessary.

In the story template,

word\_lib = [Word(word\_type) for word\_type in wordTypes]

that line of code makes a list of Word objects of each of the types listed in WordTypes. Then,

mad\_lib = story.format(\*[word.getWord() for word in word\_lib])

that line of code creates a list of strings of the Words we have in word\_lib. The \* unpacks the list and the [] signifies that output is a list.

**How it works:**

The way the programs works can be seen by the flow chart in the design section. Nevertheless, I shall explain the code in a more detailed manner to clear things up.

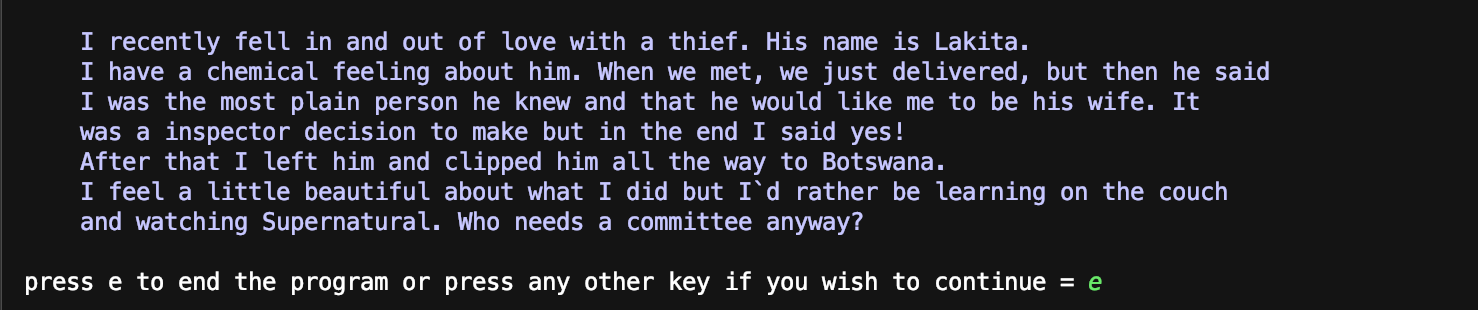
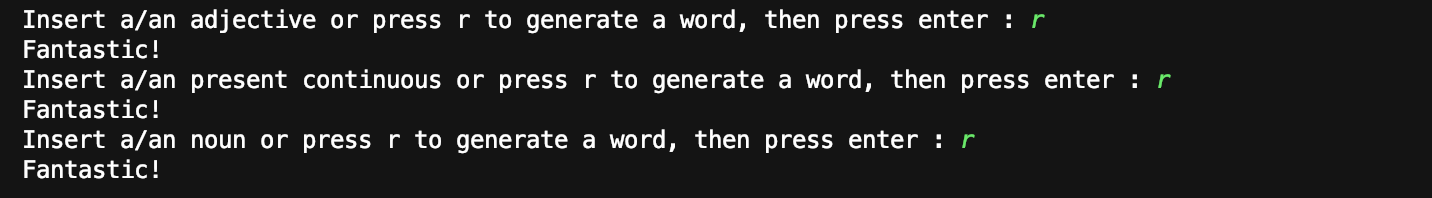
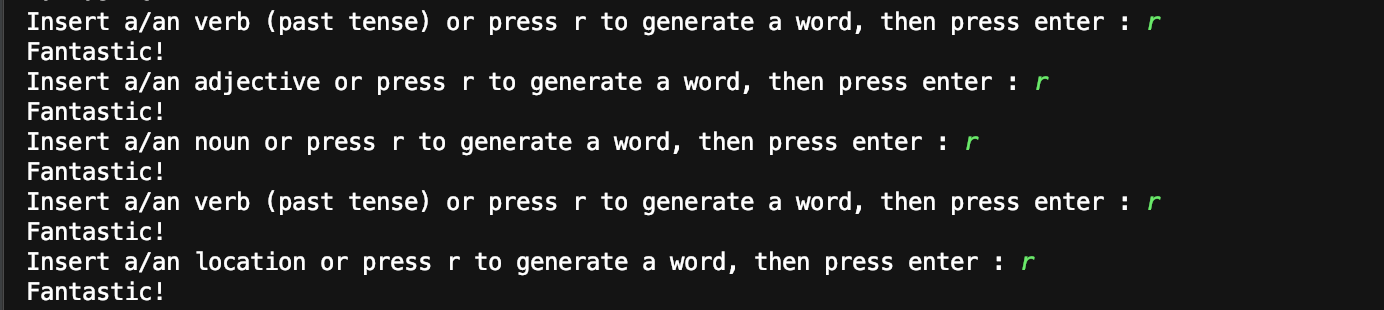
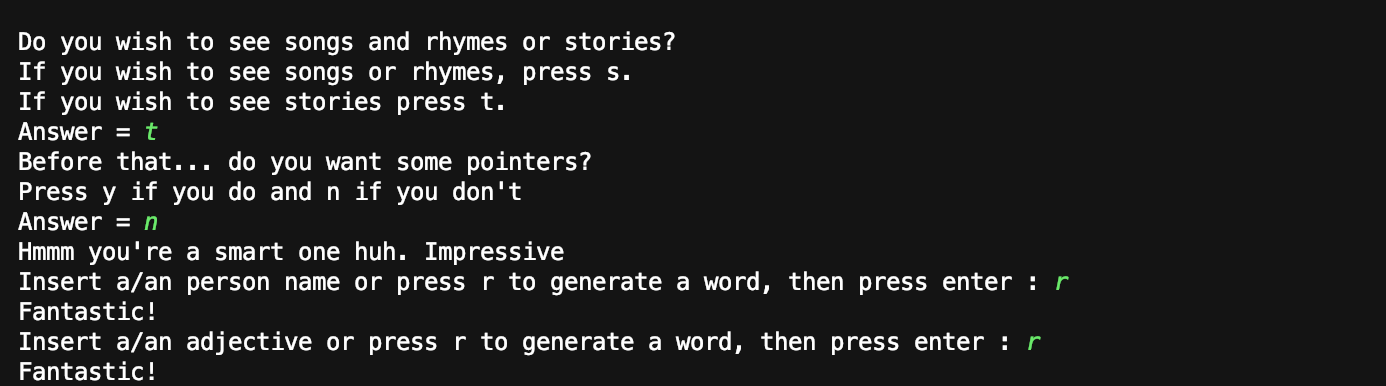
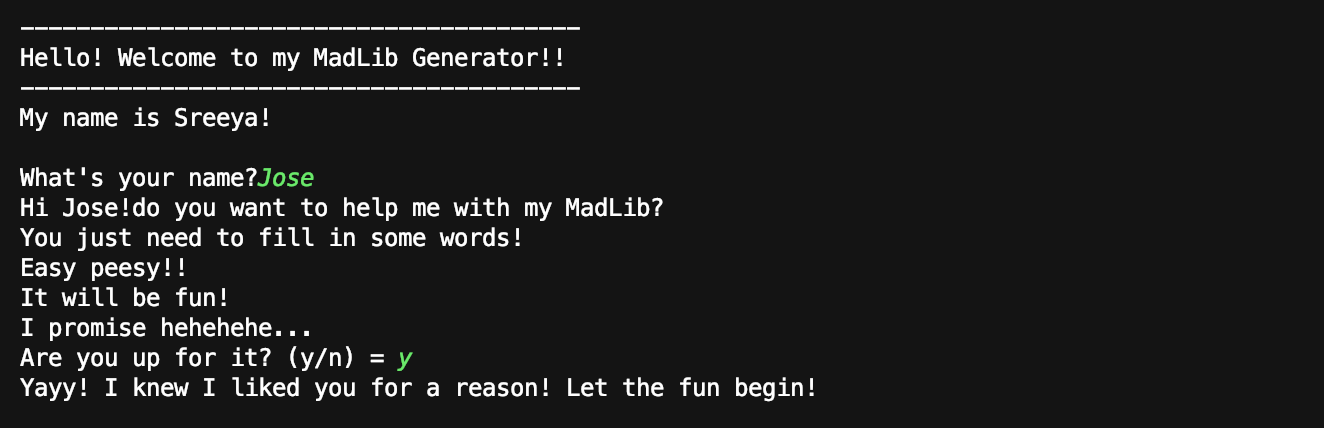
The moment the program starts, a welcome message will appear and the user will be asked to mention his or her name. The program only allows the user to input letters, if the user inputs anything besides letters, the question will keep repeating until the user only inputs letters. Afterwards, the user will be asked if they are “up for it”. The user will be asked to input either “y” for yes or “n” for no. The program accepts capital “Y” and capital “N” through the .upper() function. If the user presses “n” then the program ends, and if the user presses “y” the program continues. However, if the user presses any other key, the question will keep repeating.

Next, the def storyoptions() is called where the user is asked if they would prefer to see songs and rhymes or stories. If the user wishes to see songs and rhymes, they will input “s”. If the user wishes to see stories, they will input “t”. The program also accepts capital letter “S” and “T”. If the user inputs anything else besides “s” or “t” or “S” or “T”, the program will keep repeating the question until the user inputs the correct answer.

Afterwards, the def choosehints() is called where the user will be asked if they wish to see hints or not. If they do, then they should press the “y” or “Y” key. If they do not wish to see the hint, then they should press the “n” or “N” key. If they press any other key, the question will keep on repeating until they give the correct answer. If the user presses the “y” or “Y” key, then the def hints() will be called where the hints are printed.

Finally, the user has to input the type of words specified or they could simply press ‘r’ to generate a random word. Then, a random story will be printed out. Afterwards, the def endprogram() is called where the user is asked to decide if they wish to continue or stop. If they wish to stop, the user will have to press the “e” key, else, they can press any other key.

1. **Evidence**





1. **Resources**

<https://python3patternsidioms>test.readthedocs.io/en/latest/Comprehensionshtml

http://www.redkid.net/madlibs/