Mini Projects: Lecture 2

Matthew Ellison
MIT/SPISE

July 9, 2019

1 15-Letter Words

Write a program which creates a text file with all the 15-letter words in the English language dictionary `words.txt`.

2 Palindrome Words —— sdroW emordilaP

Write a program to create a text file with all the words in words.txt which are palindromes (that is, they are the same forwards and backwards, like 'racecar')
Hint: *`'elephant'[::-1]` is the same as `'tnahpele'`*

3 Secret Messages

The Caesar Shift cipher encodes text by shifting each letter a fixed amount in the alphabet. For example, encoding "the quick brown fox jumped over the lazy dog" with a shift of 3 gives "wkh txlfn eurzq ira mxpshg ryhu wkh odcb grj" ('w' is 3 letters after 't', etc). Write a program which prompts the user to enter text and a shift, and prints the encrypted text.

3.1 Extra Challenge

Write a program to decode an encrypted message. You can do this by printing all 26 possible decryptions and visually deciding which is correct. Or, for an extra extra challenge, choosing the correct decryption based on whether the decrypted words are valid (i.e. are in `words.txt`).

4 Square Root V2

Write a program to take square roots using the method outlined in lecture 1's mini-project 3, Square Root. This time, though, use your knowledge of loops to repeat step 2 many times and generate a very accurate result.

5 Pascal's Triangle

Pascal's triangle begins as shown below, where each number is the sum of the two numbers above. Write a program to generate the rows of Pascal's triangle.

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

5.1 Extra Challenge

Print the rows of Pascal's triangle in a nice pattern, like the one above. You may find the tab character `'\t'` useful.
Hint: *It makes sense to represent each row as a list of numbers, so Pascal's triangle could then be represented as a list of lists, one for each row.*

6 Fortune Cookie Generator

Chinese restaurants in the US typically give diners a fortune cookies after their meal, which has a slip of paper with a message inside. Here are some typical examples:

```
"You must repect others before you can gain their respects"
"People say you have sharp sense and super intellect"
"Domestic conditions demand your attention"
"Every person is the creation of himself,
    the image of his own thinking and believing"
"Next time order the shrimp"
```

In this mini-project, you'll write a program to randomly produce grammatical (though likely nonsensical) fortunes. The idea is to use a template such as:

```
Every [noun1] is the [noun2] of himself,
    the [noun3] of his own thinking and believing.  or
People say you have a [adjective1] [noun1] and
    [adjective2] [noun2].
```

And then fill in the blanks with random examples of nouns, adjectives, or whatever other part of speech fits. To help you with this part, you've been provided with the files `nouns.txt`, `verbs.txt`, and `adjectives.txt` (and a few others), which have the same format as `words.txt`.

Hint: *You can choose a random element from a list if you import the `random` library and use the `random.choice()` command.*