

Exercises

There are solutions to these exercises in the next section. You should at least attempt each one before you read the solutions.

Exercise 2

In 1939 Ernest Vincent Wright published a 50,000 word novel called Gadsby that does not contain the letter “e.” Since “e” is the most common letter in English, that’s not easy to do.

In fact, it is difficult to construct a solitary thought without using that most common symbol. It is slow going at first, but with caution and hours of training you can gradually gain facility.

All right, I’ll stop now.

Write a function called `has_no_e` that returns `True` if the given word doesn’t have the letter “e” in it.

Modify your program from the previous section to print only the words that have no “e” and compute the percentage of the words in the list have no “e.”

Exercise 3

Write a function named `avoids` that takes a word and a string of forbidden letters, and that returns `True` if the word doesn’t use any of the forbidden letters.

Modify your program to prompt the user to enter a string of forbidden letters and then print the number of words that don’t contain any of them. Can you find a combination of 5 forbidden letters that excludes the smallest number of words?

Exercise 4

Write a function named `uses_only` that takes a word and a string of letters, and that returns `True` if the word contains only letters in the list. Can you make a sentence using only the letters `acefhlo`? Other than “Hoe alfalfa?”

Exercise 5

Write a function named `uses_all` that takes a word and a string of required letters, and that returns `True` if the word uses all the required letters at least once. How many words are there that use all the vowels `aeiou`? How about `aeiouy`?

Exercise 6

Write a function called `is_abecedarian` that returns `True` if the letters in a word appear in alphabetical order (double letters are ok). How many abecedarian words are there?

Hints:

The `in` operator

The word `in` is a boolean operator that takes two strings and returns `True` if the first appears as a substring in the second:

```
>>> 'a' in 'banana'
True
>>> 'seed' in 'banana'
False
```

For example, the following function prints all the letters from `word1` that also appear in `word2`:

```
def in_both(word1, word2):
    for letter in word1:
        if letter in word2:
            print letter
```