

Scrabble challenge

Contents

Project

Goals

- Problem Statement
- Resources
- Breaking down the problem
 - Step 1: construct a word list
 - Step 2: get the rack
 - Step 3: find valid words
 - Step 4: scoring
- Checking your work
- Bonus challenge
- Congr!



Project

Write a Scrabble cheater from scratch.

Goals

- practice breaking down a problem and solving it in Python from scratch
- practice command line argument parsing
- practice reading from files
- practice working with dictionaries and for loops

Problem Statement

Write a Python script that takes a Scrabble rack as a command-line argument and prints all valid Scrabble words that can be constructed from that rack, along with their Scrabble scores, sorted by score. An example invocation and output:

```
[[Media:]]
$ python scrabble.py ZAEFIEE
17 feeze
17 feaze
16 faze
15 fiz
15 fez
12 zee
12 zea
11 za
6 fie
6 fee
6 fae
5 if
5 fe
5 fa
```

```
5 ef
2 ee
2 ea
2 ai
2 ae
```

Resources

- <http://courses.cms.caltech.edu/cs11/material/advjava/lab1/sowpods.zip> contains all words in the official [SOWPODS](http://en.wikipedia.org/wiki/SOWPODS) (<http://en.wikipedia.org/wiki/SOWPODS>) word list, one word per line.
- Here is a dictionary containing all letters and their Scrabble values:

```
scores = {"a": 1, "c": 3, "b": 3, "e": 1, "d": 2, "g": 2,
          "f": 4, "i": 1, "h": 4, "k": 5, "j": 8, "m": 3,
          "l": 1, "o": 1, "n": 1, "q": 10, "p": 3, "s": 1,
          "r": 1, "u": 1, "t": 1, "w": 4, "v": 4, "y": 4,
          "x": 8, "z": 10}
```

Breaking down the problem

Step 1: construct a word list

Write the code to open and read the sowpods word file. Create a list, where each element is a word in the sowpods word file. Note that each line in the file ends in a newline, which you'll need to remove from the word.

Step 1 resources:

- File input and output: <http://docs.python.org/tutorial/inputoutput.html#reading-and-writing-files>.
- Stripping characters (like whitespace and newlines) from a string: <http://docs.python.org/library/stdtypes.html#str.strip>.

Step 2: get the rack

Write the code to get the Scrabble rack (the letters available to make words) from the command line argument passed to your script. For example if your script were called ``scrabble_cheater.py``, if you ran `python scrabble_cheater.py RSTLNEI`, RSTLNEI would be the rack.

Handle the case where a user forgets to supply a rack; in this case, print an error message saying they need to supply some letters, and then exit the program using the `exit()` function. Make sure you are consistent about capitalization

Step 2 resources:

- Command line argument parsing: <http://docs.python.org/library/argparse.html#module-argparse>.
- Getting and checking the number of command line arguments: <http://docs.python.org/library/sys.html>.
- Converting letters to lower case: <http://docs.python.org/library/stdtypes.html#str.lower>

Step 3: find valid words

Write the code to find all words from the word list that are made of letters that are a subset of the rack letters. There are many ways to do this, but here's one way that is easy to reason about and is fast enough for our purposes: go through every word in the word list, and for every letter in that word, see if that letter is contained in the rack. If it is, save the word in a `valid_words` list. Make sure you handle repeat letters: once a letter from the rack has been used, it can't be used again.

Step 3 resources:

- List manipulation: <http://docs.python.org/tutorial/datastructures.html#more-on-lists>.
- for loops: <http://docs.python.org/tutorial/controlflow.html#for-statements>

Step 4: scoring

Write the code to determine the Scrabble scores for each valid word, using the scores dictionary from above.

Step 4 resources:

- Dictionary manipulation: <http://docs.python.org/tutorial/datastructures.html#dictionaries>.

Checking your work

What happens when you run your script on the following inputs?

```
$ python scrabble.py
Usage: scrabble.py [RACK]
```

```
$ python scrabble.py AAAaaaa
2 aa
```

```
$ python scrabble.py ZZAEEEI
22,zeze
21,ziz
12,zee
12,zea
11,za
3,aia
2,ee
2,ea
2,ai
2,aa
2,ae
```

Bonus challenge**Congr!**

You've implemented a substantial, useful script in Python from scratch that is perfect for cheating at Scrabble or Words with Friends. Keep practicing!



Retrieved from "https://wiki.openhatch.org/w/index.php?title=Scrabble_challenge&oldid=7092"

This page was last edited on 6 November 2017, at 03:23.

Content is available under [Creative Commons Attribution-ShareAlike 4.0 International \(CC BY-SA 4.0\)](#) unless otherwise noted.