

CptS 111 Introduction to Algorithmic Problem Solving **(<http://piazza.com/wsu/fall2016/cpts111/home>)**

Washington State University (<https://wsu.edu>)

Gina Sprint (<http://eecs.wsu.edu/~gsprint/>)

PA8 Dictionaries (60 pts)

Due Friday, December 9th at Midnight.

Learner Objectives

At the conclusion of this programming assignment, participants should be able to:

- Declare and use dictionaries

Prerequisites

Before starting this programming assignment, participants should be able to:

- Declare and use lists
- Declare and use strings
- Perform File I/O
- Apply loops to perform iteration

Acknowledgments

Content used in this assignment is based upon information in the following sources:

- Original ASCII art downloaded from Ask Apache Online Tools (<http://www.askapache.com/online-tools/figlet-ascii/>)

Overview and Requirements

Write a program (`ascii_art_text.py`) to generate files containing words represented by ASCII art (https://en.wikipedia.org/wiki/ASCII_art). I have used this website (<http://www.askapache.com/online-tools/figlet-ascii/>) to generate 26 files each containing ASCII art for a capital letter of the alphabet (there are 26 letters in the alphabet). For this assignment, you will need to download a zip file containing these 26 files: stars.zip (<https://raw.githubusercontent.com/gspint23/cpts111/master/progassignments/files/stars.zip>). Unzip this `stars.zip` into a folder called `stars`. Take a look at `A.txt`, pretty cool huh? Check out the different fonts available from the website I used, they are awesome!

Program Details

The program should perform the following steps:

1. Populate a list of file names for each letter text file. To do this, use the `os` module's function called `listdir(<string path to directory>)` that returns a list of the files in a particular directory.
2. Generate a dictionary, called `letter_art`, of key:value pairs where each key is a letter string and each value is a list of strings representing the lines of the corresponding letter's ASCII art. For example, the key "A" maps to the list of strings representing the lines in `A.txt`. Recall: `<file object>.readlines()` returns a list of string lines in a `<file object>`
3. Update each ASCII art value in `letter_art` such that the ASCII art is composed of the same letter the ASCII art represents. For example, `letter_art["A"]` should store the ASCII art for A made of "A"s and "a"s. For an example, please see the output below.
4. Prompt the user for a word and store the user-entered string in a variable named `word`.
 - A. Convert `word` to all uppercase.
5. For each letter in `word`, write the corresponding ASCII art letter (from the `letter_art` dictionary) *horizontally* to an output file `word_art.txt`. Place a space line between ASCII art letters.

Note: Your python source file, `ascii_art_text.py`, should be in the same folder (e.g. PA8) containing the `stars` folder. So the directory structure looks like the following:

```
+--PA8
|  |  ascii_art_text.py
|  +--stars
|  |  |  A.txt
|  |  |  B.txt
|  |  |  ...
|  |  |  Z.txt
```

Example Output

Example output file if the user enters "abc":

```

AAA          BBBBbBBBBBBBBBBBBBB          CCCCCCCCCCCCCC
AaaaA       BbbbbbbbbbbbbbbbbbbB       CCCccccccccccccC
AaaaaaA     BbbbbbbBBBBBBBbbbbbbB     CCccccccccccccccC
AaaaaaaaA   BBbbbbbbB   BbbbbbbB   CccccCCCCCCCCCccccC
AaaaaaaaaA  BbbbbB     BbbbbbbB   CccccC     CCCCC
AaaaaaAaaaaA BbbbbB     BbbbbbbB   CccccC
AaaaaaA AaaaaA BbbbbBBBBBBBbbbbbbB CccccC
AaaaaaA AaaaaA BbbbbbbbbbbbbbbbbBB CccccC
AaaaaaA AaaaaA BbbbbBBBBBBBbbbbbbB CccccC
AaaaaaAaaaaA BbbbbB     BbbbbbbB   CccccC
AaaaaaAaaaaA BbbbbB     BbbbbbbB   CccccC     CCCCC
AaaaaaA          AaaaaA   BBbbbbbbBBBBBBBbbbbbbB   CccccCCCCCCCCCccccC
AaaaaA          AaaaaA   BbbbbbbbbbbbbbbbbbbB       CCccccccccccccccccC
AaaaaA          AaaaaA   BbbbbbbbbbbbbbbbbbbB       CCCccccccccccccccC
AAAAAA          AAAAAA   BBBBbBBBBBBBBBBBBBB          CCCCCCCCCCCCCC

```

Submitting Assignments

1. Use the Blackboard tool <https://learn.wsu.edu> (<https://learn.wsu.edu>) to submit your assignment to your TA. You will submit your code to the corresponding programming assignment under the "Content" tab. You must upload your solutions as `<your last name>_pa8.zip` by the due date and time.
2. Your .zip file should contain your .py file and the stars folder containing all ASCII art .txt files.

Grading Guidelines

This assignment is worth 50 bonus. Your assignment will be evaluated based on a successful compilation and adherence to the program requirements. We will grade according to the following criteria:

- 10 pts for proper use of stars directory file list
- 15 pts for proper use of the `letter_art` dictionary.
- 10 pts for updating the stars in the ASCII art to the same letter the ASCII art represents.
- 5 pts for prompting and receiving input from the user.
- 15 pts for writing the correct output file `word_art.txt`
- 5 pts for adherence to proper programming style and comments established for the class.