Lab 5: file IO (09.30.19) CS103 Fall 2019

author john k johnstone jkj at uab dot edu
course CS103 Fall 2019

version Fall 2019

materials

- lab05_19fa103.pdf (this document)
- lab05_19fa103.py (with functions you will implement)
- lab05_tests_19fa103.py (with test calls of your functions)
- many text files:
  - *short.txt*
  - *limerick.txt*
  - *sonnet_18_shakespeare.txt* (a Shakespeare sonnet)
  - *le_petit_prince_utf8.txt* (Antoine St. Exupery's Le Petit Prince)
  - *canto1_commedia_dante_edited.txt*
    (Canto 1 of Book 1 of Dante `La Divina Commedia`, with trailing punctuation removed)
  - *canto1_commedia_dante.txt* (with punctuation)

lab partner

Please work with the lab partner that you chose or were assigned in Lab02.

purpose

Today you will learn how to

- practice reading from a file
- practice iteration with for loops on strings

preparing for Lab04

- file IO material in lecture10 will be useful
- Python documentation **here**

in-class example

Try this first, then sometime later in lab, one of the TA's will solve for the class at the board.

- (*readAndPrint*) read a Shakespeare sonnet into a string, and print the string;
  challenge: can you print it without the extra newline character?

exercises (with your lab partner)

For each of these exercises, write a function. The function docstrings are provided in *lab05_19fa103.py*. There may be more exercises here than you can complete today in lab: that is purposeful to keep you pleasantly busy, and give you more exercises outside lab.

- (*splitText*) read a text file into a string, and split it into tokens
  (essentially words, but perhaps with some punctuation noise, perhaps `Hail!` rather than `Hail`)
- (*countWord*) count the words in a text file
- (*lastChar*) build a string composed of the last character on each line;
  write the output of 'sonnet_18_shakespeare.txt' to the file 'sonnet_last.txt'
  and the output of 'canto1_commedia_dante_edited.txt' to the file 'canto_last.txt'
- (*lastWord*) build a list composed of the last word on each line

challenges (optional for A+)

- generate n random points inside the square with bottom left corner (-200,-200), width 400, and height 400; write the point cloud to the file *cloud.pt*; count the points inside the circle of radius 200 centered at the origin, and return *4*f*, where f is the fraction of points that are inside the circle (does this value look familiar? why?)
- write a function that discovers whether a string is text in terza rima (all of Dante's Divine Comedy follows this constraint); in terza rima, lines are grouped in triplets (called canticles) that have the rhyming scheme ABA, BCB, CDC, and so on. To simplify things, you may assume that two lines rhyme if they share the last letter (which is almost true in Italian) and that punctuation has been removed from the end of a line (as in the edited form of Canto 1). So this challenge becomes a refinement of the last letter exercise. For example, the Dante extract should satisfy this function: since the canticles end
  aaa aea eie ioi oeo eae ... Remarkable that Dante could keep this up through the entire Commedia!
  For an additional challenge, use text with punctuation.

deliverables

B attendance (full participation throughout lab) and successful completion of 1) the in-class problem

A attendance and successful completion of 1) the in-class problem,
2) *lastChar* (and the output files 'sonnet_last.txt', 'canto_last.txt'), and 3) report the number of words in 'le_petit_prince_utf8.txt', in the designated comment at the top of lab05_19fa103.py

A+ attendance and successful completion of 1) the solved in-class problem,
2) lastChar, 3) # words in 'le_petit_prince_utf8.txt', and 4) *one of the challenges* (including the random point cloud file 'cloud.pt' if you do the first challenge)

Project Gutenberg

Project Gutenberg is an excellent data resource for natural language processing. Today's Dante is extracted from ebook 1012. You will always be interested in the plain-text (utf8) file.