If you took 6.00.1x and went through the problem sets, there has been helper code that you didn't need to understand. In this exercise, we would like to show you how some of that works and hope that this will be a good way to review some basic concepts.

You can download loadWords.zip to have this little piece of code and a sample wordlist file so you can run this in IDLE.

If you get an error like this:
`IOError: [Errno 2] No such file or directory: 'words.txt'`
when you try running `loadWords.py`, you will have to change the line
`PATH_TO_FILE = 'words.txt'`
to contain the entire path to where the file is located, making sure to use forward slashes.
For example: `PATH_TO_FILE = 'C:/Users/Ana/Documents/ps1/words.txt'`

Help

Now, the main part of the code looks like this:

```
def loadWords():
    inFile = open(PATH_TO_FILE, 'r', 0)
    line = inFile.readline()
    wordlist = string.split(line)
    print "  ", len(wordlist), "words loaded."
    return wordlist
```

This method could be modified to work with other types of files too! First, let's go through each line of code.

Line 2 uses the open method to read a file on your computer. The `'r'` argument means we only want to read from the file. The third argument says we want to read the entire file in at once. There are a few different ways to open a file (you can open it to write to it; or read it line-by-line, for example); check out the Python documenation to find out more.

Line 3 reads a single line from the file. A line is defined by a string ending with `'\n'`. This code assumes our file (named by the `PATH_TO_FILE` variable) has one line only.

Line 4 splits the line (see the documenation on string, specifically, the documentation on string.split for more) into words, so wordlist is an array of words in the file.

We would like to modify this code so it can't crash the entire program, and so it will work for a file with more than one line of words. See the following restructured code:

```
1. def loadWords2():
2.     try:
3.         inFile = open(PATH_TO_FILE, 'r', 0)
4.     #line of code to be added here#
5.         print "The wordlist doesn't exist; using some fruits for now"
6.         return ['apple', 'orange', 'pear', 'lime', 'lemon', 'grape', 'pineapple']
7.     line = inFile.readline()
8.     wordlist = string.split(line)
9.     print "  ", len(wordlist), "words loaded."
10.    return wordlist
```

First, we aren't always sure that the file defined by `PATH_TO_FILE` always exists, and we don't want our program to crash if it is missing. This is called defensive programming.

PROBLEM 1A (1 point possible)

What line of code should we add at Line 4 to solve this problem? Mark all that apply:

- ☐ `except:`
- ☐ `except IOError:`
- ☐ `except ValueError as e:`
- ☐ `except FileNotFound:`
- ☐ `except IOError as e:`

Check

PROBLEM 1B (1 point possible)

Secondly, we have decided that we want to split our words up by commas instead of spaces.

What line should we change to solve this problem?

Line Number:

Check

PROBLEM 1C (1 point possible)

From Problem 1B, what should the new line be?

Check

Show Discussion

New Post