Intro to Computer Science   Project - Address Book 2

ASSIGNMENT OVERVIEW

In this assignment, you'll be creating a program called `addressbook2.py` which allows the user to manage a list of contact information. The data used by the main program will be stored in a separate file, `contacts.json`. The main program will use a `list` of *dictionaries* that will allow the user to enter, edit, view, and delete the records in the virtual address book, which should include a person's names, email address, phone numbers, etc.

This assignment is worth 75 points and is due on the crashwhite.polytechnic.org server is due on the crashwhite.polytechnic.org server at 23:59:59 on the date given in class.

BACKGROUND

One strategy for keeping track of data, even when a program isn't running, is by storing it in a database, an external file. In this assignment we'll be storing this information in a JSON file (*JavaScript Object Notation*), a common text format for storing information in a dictionary-style structure..

In the example `contacts.json` file shown below, each of the items enclosed in curly braces is a *dictionary*, a "record" of one person, and each data field in the dictionary is identified by its *key* (label) and a corresponding *value:* entries in a dictionary are often referred to as *key-value pairs*. The information in this JSON file will be loaded in to the program `addressbook2.py` when that program is run, and a menu system will manipulate contact information as the user desires.

```
[ { “firstname”: “Richard”,
    “lastname”: “White”,
    “email”: “rwhite@polytechnic.org”,
    “phone”: “626-396-6688” },
  { “firstname”: “Jill”,
    “lastname”: “Bush”,
    “email”: “jbush@polytecnic.org”,
    “phone”: “626-867-5309” } ]
```

PROGRAM SPECIFICATION

Create a Python program that:

a. imports the `json` library for importing and exporting JSON files
b. opens the `contacts.json` data file and brings in information into the list of dictionaries (the `initialize` function)
c. lets the user add an entry into a "list of dictionaries" called `contacts` (`add_contact` function)
d. writes the contacts data to the `contacts.json` data file before quitting the program
e. lets the user view all the information in contacts (`view_all_contacts` function)
f. lets the user to search for a name in contacts, and then prints out that person's information (`find_contact` function)
g. lets the user delete an entry (`delete_contact` function)
h. lets the user edit the information for a specific contact (`edit_contact` function)

The formatting of the JSON file will be handled by the `json` library.

The program will track at least three of the following pieces of information for an entry:

a. first name
b. last name
c. email address
d. home phone
e. cell phone
f. street address
g. birthdate

DELIVERABLES

`addressbook2.zip`

A zipped file (`addressbook2.zip`, see below) that contains a directory (folder) called `addressbook2`, that in turn contains your `addressbook2.py` and `contacts.json` files. To submit your assignment for grading, upload the zipped file to your directory in `/home/studentID/forInstructor` at `crashwhite.polytechnic.org` before the deadline.

Please be sure to include records for the specified names (Bush, White) from the example above in your database. You should also have at least one or two additional names in your `contacts2.json` file.

ASSIGNMENT NOTES

- This program is very similar to the original `addressbook.py` program, with two exceptions:
  - The internal representation of the data (used when the program is running) is a "list of dictionaries" rather than a "list of lists."
  - When the program is not running, contact information will be stored in JSON format in a text file called `contacts.json`. The formatting of that file will be handled by the `json` module.

- Dictionaries consist of *unordered* key-value pairs. You can find out which *keys* are in a dictionary using the `.keys()` method (see example below).

- To run through all the **fields** in each contact, we could use nested loops like this:

```
for contact in myContacts:
    for key in contact.keys():
         print(key, "=", contact[key])
```

  Because dictionaries are unordered, though, you can't be sure what order the *keys* will come out in: it might be *firstname, lastname, email, phone,* or it might be *email, firstname, phone, lastname*. For this reason, if you want to print out the results in a specific order, you'll have to specify the keys explicitly:

```
for contact in myContacts:
    print("First name:", contact["firstname"])
    print("Last name:", contact["lastname"])
    print("Email:", contact["email"])
    print("Phone:", contact["phone"])
```

- Use the previous version of your program as a jumping off point for writing this one. Because we’re using a different format for the external text files, you’ll need to use methods from the `json` module to get that information out of and into your running program.

```
#!/usr/bin/env python3

import json

def initialize():
    """Uses the json module to import an infile Stream from
    an external text file that stores contacts information
    in a dictionary.
    """
    try:
        infile = open("contacts.json","r")
        contacts = json.load(infile)    # load gets from infile Stream
        return contacts
    except:
        contacts = []
        return contacts


def write_file(contacts):
    “””Takes the contacts file and dumps it to the external text file.
    “””
    outfile = open("contacts.json", "w")
    outfile.write(json.dumps(contacts))
    outfile.close()  # close the external file
```

- When it comes time to upload your files, you’re going to be “zipping” your directory into a single file called `addressbook2.zip`.
    - In OS X or Ubuntu, find the directory containing your `addressbook2` folder, ctrl-click or right-click on the folder, and select “Compress” to create a zipped copy of that directory.
    - If you prefer working in the Terminal, you can use the zip command:

      2

      ```
      $ zip -r addressbook2.zip addressbook2
      ```

    - In Windows, right-click on the `addressbook` folder and select ***Send to > Compressed (zipped) folder***, then name the new zipped folder `addressbook2.zip`

- Once you’ve created the compressed file—the original directory is left intact—you can upload that file, `addressbook2.zip`, to the Dropbox on the server.

GETTING STARTED

1. Create a directory, `addressbook2`, that will be used to store all files related to your program.
2. Copy `addressbook.py` from your original version of this project so you can use that as a starting

point for this assignment.

3. Use a text editor to edit the `addressbook2.py` file, and begin working your way through the functions. Remember that you can `pass` functions or comment out code in one part of your program with triple-quotes while you focus on debugging other functions.

4. Consider writing and debugging functions in this order:
   `initialize()`
   `add_contact()`
   `view_contacts()`
   `write_file()`
   Once you've completed these and they appear to be working, then you can focus on `find()`, `delete()`, and `edit()` functions.

5. Use one of the zip strategies specified above to compress all your work into a single archive file.

6. Use `scp` to upload your file `addressbook2.zip` to the server.

QUESTIONS FOR YOU TO CONSIDER (NOT HAND IN)

1. What makes a JSON text file better than the text file we used in the first version of this program?

2. Does this program (using dictionaries) run noticeably faster than the program we wrote using lists?

SAMPLE INTERACTIONS

```
+---------------------+
|    ADDRESS BOOK     |
|                     |
| 1. Enter contact    |
| 2. View contacts    |
| 3. Find contact     |
| 4. Delete contact   |
| 5. Edit contact     |
| 6. Exit             |
+---------------------+
Enter your choice: 5
Enter Contact to edit: Bush
{'lastname': 'Bush', 'firstname': 'Jill', 'phone': 'asdf', 'email': 'asdf'}
Edit this contact? (y/N)y
First Name: Jill
Enter new First Name, or [Enter] to keep:
Last Name: Bush
Enter new Last Name, or [Enter] to keep:
Email: asdf
Enter new Email, or [Enter] to keep: jbush@windwardschool.org
Phone: asdf
Enter new Last Name, or [Enter] to keep:
+---------------------+
|    ADDRESS BOOK     |
|                     |
| 1. Enter contact    |
| 2. View contacts    |
| 3. Find contact     |
| 4. Delete contact   |
| 5. Edit contact     |
| 6. Exit             |
+---------------------+
Enter your choice: 2
First Name: Richard
Last Name: White
Email: adsf
Phone Number: asdf
-----
First Name: Jill
Last Name: Bush
Email: jbush@windwardschool.org
Phone Number: asdf
-----
+---------------------+
|    ADDRESS BOOK     |
|                     |
| 1. Enter contact    |
| 2. View contacts    |
| 3. Find contact     |
| 4. Delete contact   |
| 5. Edit contact     |
| 6. Exit             |
+---------------------+
Enter your choice: 6
Program closing
rwhite@MotteRouge$ cat contacts.json
[{"lastname": "White", "firstname": "Richard", "phone": "asdf", "email": "adsf"},
{"lastname": "Bush", "firstname": "Jill", "phone": "asdf", "email":
"jbush@windwardschool.org"}]
```