

CSV and JSON Files

WV Hotdogs' Current Employees List: CSV and JSON Files

Introduction

Chapter 16 is all about CSV and JSON files. Let's apply them to a problem.

Assignment & Discussion

West Virginia Hot Dogs now has a consolidated employee list thanks to your efforts from PA 3. Now those employees need to be placed into the list of Current Employees, which is in CSV table format.

Your mission in this homework assignment is to create and manipulate CSV and JSON files. Remember to think what each task is trying to accomplish and break them down one at a time into the steps needed to achieve the tasks. Follow the Task list for best results.

Tasks

Write a Python 3 program to accomplish the following:

1. Remember from Chapter 9 (pp. 204-205) that given the pieces of a directory structure, Python can create the actual path name and help ensure the code runs on various machine types. Instead of a full hardcoded of the path name, practice letting Python create the Desktop directory path before changing to the Desktop. That is, ensure the program's current working directory is the Desktop by using the following code:

```
homeFolder = '/'    # For Mac. If you use Windows, your homeFolder is
                    # the Window's root directory.
desktopPath = os.path.join(homeFolder, 'Users', yourUserDirectory, 'Desktop')
os.chdir(desktopPath)
```
2. Create a new CSV file called **CurrentEmpList**. Assign the file name to a variable and then use the variable name thereafter.
3. Each row in this file should denote things: the employee's name and job title. Employees from both work locations are in a consolidated file, **WVDogsNewHires.txt**.
 - Read the list from the provided file, one line at a time.
 - Pull each employee's name and job title from the file. Split method may be useful.
 - Place each employee entry on a separate line in the CSV file.Note: the first line in the file contains the column names, not employee information.
4. To verify the data wrote correctly, read the data from **CurrentEmpList** and print it to the screen.

AIST 2120 Programming Assignment 5 (Ch. 16)

5. WVHotDogs likes to keep track of the dates that each new employee list is created. In a separate file, store the date that the newly created **CurrentEmpList** was written. This should be a JSON file that contains a dictionary .
 - Give the dictionary keys for Month, Day and Year.
 - Also, provide associated values for each dictionary keys; they will represent today, e.g., June, 28, and 2021. Note: Remember the correct data type for each value.
 - A little stretch: Consider using the *datetime* module (p. 395) to obtain today's date.

Notes

- Follow the Sample Output for screen output and file content. These examples give you a good idea of what I'll look for in your CSV and JSON files.

Requirements

- ☐ **Application**
 - Write your program in Python 3.
 - Ensure you have written your pseudocode plan, named **csv_json_opsPC_YourName**.
 - Ensure your file is named properly: **csv_json_ops_YourName.py**.
 - If you choose to use a different file editor, note that I will run and evaluate your submissions in IDLE. If you write your code using another IDE, remember to run your finished program in IDLE to ensure no unfortunate surprises. If it doesn't run in IDLE, it doesn't run.
- ☐ **Submission**
 - Submit your source code file via D2L. Neither email nor hardcopy submissions will be accepted.
 - Ensure you retain a complete copy of your program source code file(s).
- ☐ **Due Date:** Per D2L assignment instructions.
- ☐ **Late Penalty:** I will assess a penalty for late submissions, generally 10% per day for up to four days (96 hours from assignment due date and time). I will not accept submissions more than one week, 7 calendar days, after the original due date. Ensure your work is complete and on time!

STYLE GUIDE

- ☐ **File headers.** Include a header in the below format at the top of all .py source code files.

```
# Your Name
# AIST 2120: Programming Assignment 5
# Submission Date
```

AIST 2120 Programming Assignment 5 (Ch. 16)

- **Sample Output.** The below output is provided as a sample of how your output could look. Your output is not required to look exactly like my sample, but this sample does meet all assignment requirements for screen output save one: it does not include the date matches.
 - Keep in mind that the output is to inform the user and should be both clear and pleasing to those reading it.

Sample Screen Output

```
-----
---      AIST 2120      ---
--- Programming Assignment 5 ---
---      CSV and JSON Files      ---
-----

The current working directory is ... /Users/Mistony/Desktop

WVDogsNewHires.txt is ready for processing.
CurrentEmpList.csv is ready to recieve data.

Processing of CurrentEmpList.csv complete.
It now contains:
    ['NAME', 'JOB TITLE']
    ['Snoopy', 'Pilot']
    ['Charlie-Brown', 'Greeter']
    ['Lucy', 'Counselor']
    ['Linus', 'Musician']
    ['Franklin', 'BusinessOperations']|
    ['Luke', 'Security']
    ['Rey', 'Versatilist']
    ['Finn', 'FacilitiesManager']
    ['Julia-Child', 'Chef']

The JSON date file has been written.
It should contain 6/28/2021.

-----
---      AIST 2120      ---
--- Programming Assignment 5 ---
---      COMPLETE      ---
-----
```

CSV Output File: CurrentEmpList.csv

CurrentEmpList	
NAME	JOB TITLE
Snoopy	Pilot
Charlie-Brown	Greeter
Lucy	Counselor
Linus	Musician
Franklin	BusinessOperations
Luke	Security
Rey	Versatilist
Finn	FacilitiesManager
Julia-Child	Chef

AIST 2120 Programming Assignment 5 (Ch. 16)

Example JSON Output File: employee.JSON

```
employee.json x
1 {"Month": 6, "Day": 29, "Year": 2021}
```