Python Programming

Intermediate Python: Scripting and Code Abstraction

Overview

Let's put your skills to the test!

You will practice these programming concepts we've covered in class: * Scripting * File I/O * Code abstraction with `itertools` * List comprehension

---

Deliverables

For the challenges below, you will create a new `.py` file and write code to solve the problem. Run the file from the command line to check your work. Detailed directions are given below.

Reminder: On your laptop, you can run the file from your command line with the following:

```
python problem1.py
```

> **Hint:** Make sure you are printing something out with the `print` statement. Otherwise, you won't see any output from running your program!

Requirements:

By the end of this, you should have: - Two `.py` files for the solutions. - A generated `csv` file called `tps_report.csv`.

Problem 1: TPS Report

Skill you're practicing: Using modules, reading docs, scripting, file I/O, and using dictionaries.

Your boss, Bill, asks you to come in on Saturday to finish your TPS report. You promise to finish the report, but as Bill walks away you smile knowing you have a trick up your sleeve: scripting! You plan on quickly mocking up a bit of code to finish the report and change your performance reviews in the

system, all without spending a minute of your Saturday!

You know of a Python module called `csv` that can be used to read and write `csv` files. CSV stands for "comma-separated values" and is a file type commonly associated with spreadsheets. You can check out the `csv` docs for examples of how this module works. We'll be using the `DictWriter` method, which maps a dictionary to output rows.

Example of DictWriter

```
import csv

# This line opens or creates a `names.csv` file.
with open('names.csv', 'w', newline='') as csvfile:

    # These are the header row values at the top.
    fieldnames = ['first_name', 'last_name']

    # This opens the `DictWriter`.
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

    # Write out the header row (this only needs to be done once!).
    writer.writeheader()

    # Write as many rows as you want!
    writer.writerow({'first_name': 'Baked', 'last_name': 'Beans'})
    writer.writerow({'first_name': 'Lovely', 'last_name': 'Spam'})
    writer.writerow({'first_name': 'Wonderful', 'last_name': 'Spam'})
```

Output of `DictWriter` Example

If you run the previous example locally, you'll end up with a spreadsheet that looks like this:

first_name	last_name
Baked	Beans
Lovely	Spam
Wonderful	Spam

If you view it in a text editor like Sublime Text, you'll just see the values separated by commas:

```
first_name,last_name
Baked,Beans
Lovely,Spam
Wonderful,Spam
```

> **Note:** If you double click on the file that is generated, it will open in whatever spreadsheet program you have on your computer. On a Mac, this is probably Numbers. On Windows, this is probably Excel. If you don't have a spreadsheet program on your computer, upload the file to Google Sheets and view it there.

Your Job!

Now that you have a good idea how to wrangle a `csv` file, let's think about what you'd like to do with this newfound power. You have a list of dictionaries called `employees`, which contains information about each employee including `first_name`, `last_name`, `hire_date`, `job_title`, and `performance_review`. You must open a new file called `tps_report.csv` and make a loop to write out every employee in the `employees` to `tps_report.csv`.

Your Starter Code

You can start with the starter code here. Please copy it into your own file called `problem1.py`.

```
# Import 'csv' module.
import csv

# Report is a list of objects.
# The keys in these objects will be the header fields of your spreadsheet.
report = [{...}, {...}, {...}]
```

Expected Output

```
first_name,last_name,job_title,hire_date,performance_review
Bill,Lumbergh,Vice President,1985,excellent
Michael,Bolton,Programmer,1995,poor
Peter,Gibbons,Programmer,1989,poor
Samir,Nagheenanajar,Programmer,1974,fair
Milton,Waddams,Collator,1974,does he even work here?
Bob,Porter,Consultant,1999,excellent
Bob,Slydell,Consultant,1999,excellent
```

Or, if you're viewing it in a spreadsheet program:

first_name	last_name	job_title	hire_date	performance_review
Bill	Lumbergh	Vice President	1985	excellent
Michael	Bolton	Programmer	1995	poor
Peter	Gibbons	Programmer	1989	poor
Samir	Nagheenanajar	Programmer	1974	fair
Milton	Waddams	Collator	1974	does he even work here?
Bob	Porter	Consultant	1999	excellent
Bob	Slydell	Consultant	1999	excellent

Hint: Instead of writing `writerow` many times in a row, try looping through the `employees` list.

Problem 2: Have You Seen My Stapler?

Skills you're practicing: Using modules, reading docs, scripting, file I/O, and using dictionaries.

Now we have a TPS report, but it isn't finished — there are still a couple of things left to do!

```
1. Add a field called `finished_review` to each employee. The value for every row
should be `yes`. You figure you can just add this field to each dictionary inside
the loop.

2. Change everyone's `performance_review` to `excellent`, unless it's your boss
`Bill` or someone with the `job_title` of `Consultant`. In that case, make their
`performance_review` value `poor`.

3. Enjoy your Saturday! □ □ □
```

Starter Code

Copy all of your code from `problem1.py` and use that as your base code for `problem2.py`!

Expected Output

```
first_name,last_name,job_title,hire_date,performance_review,finished_review
Bill,Lumbergh,Vice President,1985,poor,yes
Michael,Bolton,Programmer,1995,excellent,yes
Peter,Gibbons,Programmer,1989,excellent,yes
Samir,Nagheenanajar,Programmer,1974,excellent,yes
Milton,Waddams,Collator,1974,excellent,yes
Bob,Porter,Consultant,1999,poor,yes
Bob,Slydell,Consultant,1999,poor,yes
```

Or, in table form:

first_name	last_name	job_title	hire_date	performance_review	finished_review
Bill	Lumbergh	Vice President	1985	poor	yes
Michael	Bolton	Programmer	1995	excellent	yes
Peter	Gibbons	Programmer	1989	excellent	yes
Samir	Nagheenanajar	Programmer	1974	excellent	yes
Milton	Waddams	Collator	1974	excellent	yes
Bob	Porter	Consultant	1999	poor	yes
Bob	Slydell	Consultant	1999	poor	yes

Hint: If you get a `ValueError` saying that `finished_review` is not a field name, it means that you forgot to add it to the `fieldnames` array.

Look at You, Being a Boss!

What do you call a snake that's 3.14 meters long?

A `Pi-Thon`! :D