Students:

This content is controlled by your instructor, and is not zyBooks content. Direct questions or concerns about this content to your instructor. If you have any technical issues with the zyLab submission system, use the **Trouble with lab** button at the bottom of the lab.

10.9 LAB CHECKPOINT: Save to CSV

LAB: Save Temperature Statistics to CSV

CSW8: Learning Goals

In this lab we will -

- Create a dictionary from its provided description
- Process the dictionary and store it in a CSV (comma separated values) file
- reference Figure 10.7.4: Writing rows to a csv module.

Introduction

Your success with Anita's task now has Chen asking for your help: can you store the data from a Python dictionary into a csv file?

Instructions

Given a dictionary with the month as a key and the temperatures stored in a list, you need to write this dictionary into the csv file `fahrenheit_data.csv` which will look like

```
January,67,47
February,67,47
...
```

Steps

1. Create a function `save_temperature_statistics(input_dict, filename)` that will accept a dictionary with the month-temperature mappings and a string with the filename to which to save the data.

   1.1 Use the `with` construct to open the file with the required filename. You will need open the file for **writing** (the `w` flag) exactly like this:

   ```
   with open(filename, 'w', newline='') as csvfile
   ```

Please notice the `newline` parameter here - it will prevent the csv library from writing extra empty lines in the file. Check the example in your IDE with and without the `newline=""` to see the difference.

1.2 After opening the file, create the `csv.writer` object as `csv_writer = csv.writer(csvfile)`.

1.3 After defining `csv_writer`, loop over all items in the dictionary using a `for` loop.

1.4.1. For each key and value pair in the dictionary (`input_dict`), create a new list `temperature_info_list` that contains 3 elements. The first element is the key of the current pair. The next two elements are the two elements in the value list. For instance, `temperature_info_list` for the item with key as `"January"` and value as `[67, 47]` will be `temperature_info_list = ["January", 67, 47]`.

1.4.2. Save this list as a line in the file using the `csv_writer.writerow(temperature_info_list)`. The function `writerow()` writes this information into the file by separating the list elements by commas.

2. In your main program, i.e, in the `if __name__ == "__main__"` block, you should create `temperatures_dict` and store the previous weather information provided to Anita. You should then call the function `save_temperature_statistics(temperature_dict, 'fahrenheit_data.csv')` to write the dictionary to the csv file.

398092.2571084.qx3zqy7

LAB ACTIVITY | 10.9.1: LAB CHECKPOINT: Save to CSV | 0 / 12

main.py — Load default template...

```
1 import csv
2
3 def save_to_csv(my_list, filename):
4
```

Develop mode | **Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run**

program and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → **main.py** (Your program)

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin working on this zyLab.
```

Trouble with lab?