Lab 8 Manual

Lab 8 Outline

Part 0: Evaluations!

You don't have to do it now in the lab, but please take a few minutes to fill out evaluations for Paul and tutors/TAs you've interacted throughout the quarter!

Part 1: Experimenting with Python [35 mins]

- Looping over list of dictionaries
- Working with csv files

Part 2: Quiz [15 mins]

You can take the quiz any time through the day before 11:59pm. This will still be a **timed (15 mins)**, multiple choice quiz on the lab material. If you completed the lab, you will be able to answer all of the questions on the quiz. We strongly recommend you finish the quiz during the lab section in case you forget to do it.

Looping over list of dictionaries

A dictionary is a mutable collection of elements. Each element in a dictionary is a key-value pair. For example, we can store information about a person in a dictionary below:

```
person = {
    'first_name': 'Joe',
    'last_name': 'Smith',
    'age': 51,
    'spouse': 'Edna'
    }
```

To get the value associated with a key, give the name of the dictionary and then place the key inside a set of square brackets, as shown here:

```
print(person['age'])
```

This prints the value associated with the key `'age'` from the dictionary `person`:

```
51
```

Now, try this in the terminal. Open interactive python with `python -i` and copy and paste the code into python. Next, try printing out the first name and last name of the person.

We can also store a set of dictionaries in a list. The `person` dictionary contains information about one person, but it has no room to store information about another person. One way we can store information about many people is to make a list of dictionaries, where each dictionary stores information about one person. For example, the following is a list of dictionaries that stores information about many people:

```
people = [{'first_name': 'Joe', 'last_name': 'Smith', 'age': 51, 'spouse': 'Edna'},
{'first_name': 'Edna', 'last_name': 'Smith', 'age': 48, 'spouse': 'Joe'},
{'first_name': 'Jacob', 'last_name': 'Jones', 'age': 33, 'spouse': 'Lily'},
{'first_name': 'Lily', 'last_name': 'White', 'age': 29, 'spouse': 'Jacob'},
{'first_name': 'Tiffany', 'last_name': 'Poe', 'age': 37, 'spouse': 'Sean'},
{'first_name': 'Emily', 'last_name': 'Miller', 'age': 24, 'spouse': 'Michael'},
{'first_name': 'Sean', 'last_name': 'Miller', 'age': 40, 'spouse': 'Tiffany'},
{'first_name': 'Michael', 'last_name': 'Poe', 'age': 25, 'spouse': 'Emily'}]
```

Now in `print_names.py`, write a function that loops over the `people` list, and print out the first and last name of each person in a line separated by one space.

The output should look like this:

```
Joe Smith
Edna Smith
```

```
Jacob Jones
Lily White
Tiffany Poe
Emily Miller
Sean Miller
Michael Poe
```

Working with csv files

Part 1.2: Working with csv files

Step A: Overview of CSE8ACSV documentation

In `lab8_csv.py`, you will implement a function that calculates the total number of Black Lives Matter protests among a certain group of states. You are provided a few functions that help with loading data from CSV (Comma Separated Values) files in `CSE8ACSV.py`. Take a look at `blm_state.csv` to get an idea of the data and the format. Below are the functions provided.

`def get_blm_data(filename)`

You are given a function that will convert a CSV file (Comma Separated Values) containing data related to Black Lives Matter protests by state into a list of dictionaries, formatted as following:

[{State: String, TotalProtests: float, ...}, {State: String, TotalProtests: float, ...}, ... more states ...]

This function is already present in the starter code to load the data into the variable `blm_protest_data`. Each dictionary in the list contains the following information:

Field Name	Field Type	Description
State	String	This field is a String that denotes the name of one of the 50 states in the US.
BlackPop	float	This field is a float representing the total black population of the state
BlackPoverty	float	This field is a float representing the percentage of black people below the poverty line in the state.
AsianPop	float	This field is a float representing the total asian population of the state
AsianPoverty	float	This field is a float representing the percentage of asian people below the poverty line in the state.
HispanicPop	float	This field is a float representing the total hispanic population of the state
HispanicPoverty	float	This field is a float representing the percentage of hispanic people below the poverty line in the state.
WhitePop	float	This field is a float representing the total white population of the state
WhitePoverty	float	This field is a float representing the percentage of white people below the poverty line in the state.
TotalProtests	float	This field is a float representing the total protests in the state.
TotalAttendance	float	This field is a float representing the total attendance of the protests in the state.

Step B: Start Coding!

Total BLM Protests

We are interested in calculating the total number of Black Lives Matter protests among a certain group of states. The data is already present in the starter code as a list of dictionaries, named `blm_protest_data`. Write a function in `lab8_csv.py` that returns the total number of protests that took place in the list of states provided as the input parameter.

`def total_blm_protests(states)`

- Parameter:
  - `states` - A list of strings denoting state names
- Return:
  - Returns an int representing the total number of protests in the list of state names (note that all values in the dictionaries are **floats**)

- Description:
  - This function uses the list of dictionaries `blm_protest_data` to obtain the total number of protests in the list of states.
- Examples:

```
>>> states = ["California", "Texas"]
>>> total_blm_protests(states)
110

>>> states = []
>>> total_blm_protests(states)
0

>>> states = ["Alaska", "New Mexico", "Arkansas", "New York"]
>>> total_blm_protests(states)
72
```