Programming Assessment

for Chapel Hill Open Data

Instructions

Your goal for this assessment is to create a script that can perform a GET request with the given API, transform that data into a human readable format, and then write it to a CSV. The purpose of this exercise is to test your familiarity with the Extract & Transform operations. Read more about Extract, Transform, and Load (ETL) processes here.

The test files are initially in Python format, but you may use any programming language of your choosing to accomplish this task. You may convert the files to your desired format, but please keep the names the same.

If you are struggling: Google and StackOverflow are good resources. You can also email David Green at dgreen@townofchapelhill.org and we will offer some assistance. We won't give you the answers, but we can help you get pointed in the right direction.

Extract

The API you will communicate with is connected to the weather sensor that is located at the Chapel Hill Public Library. The final CSV should include these data points:

Temperature (F), Humidity (%), Wind Speed (mph), Wind Gust (mph), Daily Rainfall Totals (in), Monthly Rainfall Totals (in), Yearly Rainfall Totals (in), UV Radiation Index, and Date/Time

Read up on the API documentation here.

Please write your code in the **test.py** file you were given. **Do not move the file to a different folder**.

To help get you started, this is the URL schema that will pull in the correct data:

```
"https://api.ambientweather.net/v1/devices/" + str(test-
secrets.macAddress) + "?apiKey=" + str(test-secrets.apiKey) +
"&applicationKey=" + str(test-secrets.appKey) + "&limit=21"
```

You can copy/paste this URL into your request function in your Development Environment.

Optional: *Note the concatenation of the variables in the URL string. What is this syntax doing? Why do you think we format URLs this way? You can include your answer to this question as a comment above the URL in your code.*

Transform

Oftentimes, transforming data is the most time consuming process of ETL. To test your comfort with this, we ask for an additional feature in your script.

For the **Temperature (F)** column:

If the temperature is below 60 (F) write **"Too cold"**
If the temperature is above 85 (F) write **"Too hot"**
Else write **"Goldilocks"**

Don't worry if your finished CSV doesn't show all three transformations. We're just interested in seeing if your code has the logic to handle this.

If you are unable to complete the assignment, or you run out of time before you finish, don't despair. Please include your next steps as comments in the test file. We are primarily looking to find out if you can think through this assignment from a programming standpoint.

Completing the Assignment

Please ensure that your code is commented and well organized.

Upload it to your GitHub so we can review it, please provide us with the link. If you don't have a GitHub, please make one. If you don't know how to push commits to GitHub, check out the "Steps to Upload to GitHub" PDF included in the folder.

PLEASE DO NOT INCLUDE THE test-secrets.py FILE IN THE UPLOAD

You can either delete the file from the folder prior to the upload or you can use a .gitignore (bonus points for using the .gitignore).