Module 12 Challenge

Start Assignment

Due Dec 19, 2022 by 11:59pm **Points** 100 **Submitting** a text entry box or a website url

Background

You're now ready to take on the full web-scraping and data analysis project for the mission to Mars. You've learned to identify HTML elements on a page, identify their `id` and `class` attributes, and use this knowledge to extract information via both automated browsing with Splinter and HTML parsing with Beautiful Soup. You've also learned to scrape various types of information. These include HTML tables and recurring elements, like multiple news articles on a webpage.

As you work on this Challenge, remember that you're strengthening the same core skills that you've been developing until now: collecting data, organizing and storing data, analysing data, and then visually communicating your insights.

What You're Creating

This new assignment consists of two technical products. You will submit the following deliverables:

- Deliverable 1: Scrape titles and preview text from Mars news articles. Optionally export the data into a JSON file or a MongoDB database.
- Deliverable 2: Scrape and analyse Mars weather data, which exists in a table.

Files

Download the following files to help you get started:

Module 12 Challenge files (https://static.bc-edx.com/data/dla-1-1/m12/lms/starter/Starter_Code.zip)

Instructions

Deliverable 1: Scrape Titles and Preview Text from Mars News

Open the Jupyter Notebook in the starter code folder named `part_1_mars_news.ipynb`. You will work in this code as you follow the steps below to scrape the Mars News website.

1. Use automated browsing to visit the **Mars NASA news site (https://redplanetscience.com)**. Inspect the page to identify which elements to scrape.

   SHOW HINT

2. Create a Beautiful Soup object and use it to extract text elements from the website.
3. Extract the titles and preview text of the news articles that you scraped. Store the scraping results in Python data structures as follows:
   - Store each title-and-preview pair in a Python dictionary. And, give each dictionary two keys: `title` and `preview`. An example is the following:

     ```
     {'title': "Mars Rover Begins Mission!",
          'preview': "NASA's Mars Rover begins a multiyear mission to collect data about the little-explored planet."}
     ```

   - Store all the dictionaries in a Python list.
   - Print the list in your notebook.
4. Optionally, store the scraped data in a file or database (to ease sharing the data with others). To do so, export the scraped data to either a JSON file or a MongoDB database.

Deliverable 2: Scrape and Analyse Mars Weather Data

Open the Jupyter Notebook in the starter code folder named `part_2_mars_weather.ipynb`. You will work in this code as you follow the steps below to scrape and analyse Mars weather data.

1. Use automated browsing to visit the **Mars Temperature Data Site** (https://data-class-mars-challenge.s3.amazonaws.com/Mars/index.html). Inspect the page to identify which elements to scrape. Note that the URL is `https://data-class-mars-challenge.s3.amazonaws.com/Mars/index.html`.

SHOW HINT

2. Create a Beautiful Soup object and use it to scrape the data in the HTML table. Note that this can also be achieved by using the Pandas `read_html` function. However, use Beautiful Soup here to continue sharpening your web scraping skills.
3. Assemble the scraped data into a Pandas DataFrame. The columns should have the same headings as the table on the website. Here's an explanation of the column headings:
   - `id`: the identification number of a single transmission from the Curiosity rover
   - `terrestrial_date`: the date on Earth
   - `sol`: the number of elapsed sols (Martian days) since Curiosity landed on Mars
   - `ls`: the solar longitude
   - `month`: the Martian month
   - `min_temp`: the minimum temperature, in Celsius, of a single Martian day (sol)
   - `pressure`: The atmospheric pressure at Curiosity's location
4. Examine the data types that are currently associated with each column. If necessary, cast (or convert) the data to the appropriate `datetime`, `int`, or `float` data types.

SHOW HINT

5. Analyse your dataset by using Pandas functions to answer the following questions:

    1. How many months exist on Mars?
    2. How many Martian (and not Earth) days worth of data exist in the scraped dataset?
    3. What are the coldest and the warmest months on Mars (at the location of Curiosity)? To answer this question:
        - Find the average the minimum daily temperature for all of the months.
        - Plot the results as a bar chart.
    4. Which months have the lowest and the highest atmospheric pressure on Mars? To answer this question:
        - Find the average the daily atmospheric pressure of all the months.
        - Plot the results as a bar chart.
    5. About how many terrestrial (Earth) days exist in a Martian year? To answer this question:
        - Consider how many days elapse on Earth in the time that Mars circles the Sun once.
        - Visually estimate the result by plotting the daily minimum temperature.

6. Export the DataFrame to a CSV file.

Requirements

Deliverable 1: Scrape Titles and Preview Text from Mars News (40 points)

- Automated browsing (with Splinter) was used to visit the Mars news site, and the HTML code was extracted (with Beautiful Soup). (10 points)
- The titles and preview text of the news articles were scraped and extracted. (20 points)
- The scraped information was stored in the specified Python data structure—specifically, a list of dictionaries. (10 points)

Deliverable 2: Scrape and Analyse Mars Weather Data (60 points)

- The HTML table was extracted into a Pandas DataFrame. Splinter and Beautiful Soup were used to scrape the data. The columns have the correct headings and data types. (15 points)
- The data was analysed to answer the following questions: (10 points)

  - How many months exist on Mars? (5 points)
  - How many Martian days' worth of data are there? (5 points)

- The data was analysed to answer the following questions, and a data visualisation was created to support each answer: (30 points)
  - Which month, on average, has the lowest temperature? The highest? (10 points)
  - Which month, on average, has the lowest atmospheric pressure? The highest? (10 points)
  - How many terrestrial days exist in a Martian year? A visual estimate within 25% was made. (10 points)

- The DataFrame was exported into a CSV file. (5 points)

Grading

This assignment will be evaluated against the requirements and assigned a grade according to the following table:

Grade	Points
A (+/-)	90+
B (+/-)	80–89
C (+/-)	70–79
D (+/-)	60–69
F (+/-)	< 60

Submission

As a reminder, the deliverables for this Challenge are as follows:

- Deliverable 1: A Jupyter notebook containing code that scrapes the Mars news titles and preview text.

- Deliverable 2: A Jupyter notebook containing code that scrapes the Mars weather data and that cleans, visualises, and analyses that data.

To submit your Challenge assignment, click Submit, and then provide the URL of your GitHub repository for grading.

NOTE

You are allowed to miss up to two Challenge assignments and still earn your certificate. If you complete all Challenge assignments, your lowest two grades will be dropped. If you wish to skip this assignment, click Next, and move on to the next module.

Comments are disabled for graded submissions in BootCamp Spot. If you have questions about your feedback, please notify your instructional staff or your Student Success Manager. If you would like to resubmit your work for an additional review, you can use the Resubmit Assignment button to upload new links.