

Module 11 Challenge

[Start Assignment](#)

Due Nov 9 by 11:59pm **Points** 100

Submitting a text entry box or a website url

Background

Robin is now ready to take on the full web-scraping and data analysis project for the mission to Mars. She's 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 BeautifulSoup. She's also learned to scrape various types of information. These include HTML tables and recurring elements, like multiple news articles on a webpage.

You'll help Robin scrape, organize, analyze, and visualize the data. 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, analyzing 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 analyze Mars weather data, which exists in a table.

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

Deliverable 1 Instructions

1. Create a new Jupyter notebook named `mars_data_challenge_part_1.ipynb`.
2. Scrape the [Mars News](https://redplanetscience.com) [↗\(https://redplanetscience.com\)](https://redplanetscience.com) website by using Splinter and BeautifulSoup. Specifically, scrape the title and preview text, or summary text, of each article on the landing page.
 - If you'd like a hint on identifying which elements to scrape, that's totally okay. If not, that's great too. You can always revisit this later if you change your mind.

[SHOW HINT](#)

3. 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 col
```

- 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.



REWIND

For this deliverable, you have done the following in this module:

- [Lesson 11.4.1](#): Gaining a Deeper Understanding of a Website Structure
- [Lesson 11.5.1](#): Performing an Automated Web Scrape
- [Lesson 11.6.2](#): Scraping Mars Facts
- [Lesson 11.6.3](#): Scraping Mars News


Deliverable 1 Requirements

- Automated browsing (with Splinter) was used to visit the Mars news site, and the HTML code was extracted (with BeautifulSoup). (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 Analyze Mars Weather Data (60 points)

Deliverable 2 Instructions

1. Create a Jupyter notebook named `mars_data_challenge_part_2.ipynb`. Import the relevant dependencies for web scraping, Pandas, and Matplotlib.
2. With your automated browser, visit the [Mars Temperature Data](https://data-class-mars-challenge.s3.amazonaws.com/Mars/index.html)  (<https://data-class-mars-challenge.s3.amazonaws.com/Mars/index.html>) site. Note that the URL is `https://data-class-mars-challenge.s3.amazonaws.com/Mars/index.html`.
3. Scrape the data in the HTML table. To do so, choose one of two ways. The first, simpler way is to use Pandas's `read_html` method. The second, slightly more challenging way is to manually scrape the data by using Splinter and BeautifulSoup. We highly encourage you to choose the latter to reinforce your scraping skills.
 - If you'd like a hint on manually scraping the data, that's totally okay. If not, that's great too. You can always revisit this later if you change your mind.

[SHOW HINT](#)

4. 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:
 - The `id` heading: The identification number of a single transmission from the Curiosity rover.
 - The `terrestrial_date` heading: The date on Earth.
 - The `sol` heading: The number of elapsed sols (Martian days) since Curiosity landed on Mars.
 - The `ls` heading: The solar longitude.
 - The `month` heading: The Martian month.
 - The `min_temp` heading: The minimum temperature, in Celsius, of a single Martian day (sol).
 - The `pressure` heading: The atmospheric pressure at Curiosity's location.
5. Examine the data types of all the DataFrame columns. If necessary, cast (or convert) the data to the appropriate `datetime`, `int`, or `float` data types.

- If you'd like a hint on how to convert the data, that's totally okay. If not, that's great too. You can always revisit this later if you change your mind.

[**SHOW HINT**](#)

6. Answer the following question: How many months exist on Mars?
7. Answer the following question: How many Martian (and not Earth) days worth of data exist in the scraped dataset?
8. Answer the following question: What are the coldest and the warmest months on Mars (at the location of Curiosity)? Get the answer by averaging the minimum daily temperature of all the months. Plot the results as a bar chart.
9. Answer the following question: Which months have the lowest and the highest atmospheric pressure on Mars? Get the answer by averaging the daily atmospheric pressure of all the months. Plot the results as a bar chart.
10. Answer the following question: About how many terrestrial (Earth) days exist in a Martian year? That is, in the time that Mars circles the Sun once, how many days elapse on Earth? Visually estimate the result by plotting the daily minimum temperature.
11. Export the DataFrame to a CSV file.

**REWIND**

For this deliverable, you have done the following in this module:

- [Lesson 11.6.5](#): Scraping a Table with Pandas

Deliverable 2 Requirements

- The HTML table was extracted into a Pandas DataFrame. Either Pandas or Splinter and Beautiful Soup were used to scrape the data. The columns have the correct headings and data types. (15 points)
- The data was analyzed to answer the following questions, and a data visualization was created to support each answer: (40 points)
 - How many months exist on Mars?
 - Which month, on average, has the lowest temperature? The highest?
 - Which month, on average, has the lowest atmospheric pressure? The highest?
 - How many terrestrial days exist in a Martian year? A visual estimate within 25% was made.

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

Submission

Once you're ready to submit, make sure to check your work against the rubric to ensure you meet the requirements for this Challenge one final time. It's easy to overlook items when you're in the zone! Then, commit the deliverables to your Mission-to-Mars GitHub repository.

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, visualizes, and analyzes that data.

To submit your Challenge assignment in Canvas, click Submit, then provide the URL of your Mission-to-Mars GitHub repository for grading. Comments are disabled for graded submissions in Bootcamp Spot. If you have questions about your feedback, please notify your instructional staff or Student Success Manager. If you would like to resubmit your work for an improved grade, you can use the Re-Submit Assignment button to upload new links. You can resubmit up to three times, for a total of four submissions.

IMPORTANT

When you receive feedback on your Challenge, make any suggested updates or adjustments to your work. Then add this week's Challenge to your professional portfolio.

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 Submit, and then indicate you are skipping by typing "I choose to skip this assignment" in the text box.

© 2022 edX Boot Camps LLC