

Module 11 Challenge

[Start Assignment](#)

Due Thursday by 11:59pm **Points** 100 **Submitting** a text entry box or a website url

Background

You're now ready to take on a full web-scraping and data analysis project. 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 BeautifulSoup. 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, 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.
- Deliverable 2: Scrape and analyze Mars weather data, which exists in a table.

Files


Download the following files to help you get started:

[Module 11 Challenge files](https://static.bc-edx.com/data/dl-1-2/m11/lms/starter/Starter_Code_v1.2.2.zip)  (https://static.bc-edx.com/data/dl-1-2/m11/lms/starter/Starter_Code_v1.2.2.zip)

Instructions

Part 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 news site](https://static.bc-edx.com/data/web/mars_news/index.html)  (https://static.bc-edx.com/data/web/mars_news/index.html). Inspect the page to identify which elements to scrape.

[HIDE HINT](#)

To identify which elements to scrape, you might want to inspect the page by using Chrome DevTools.

2. Create a BeautifulSoup 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': "NASA's MAVEN Observes Martian Light Show Caused by Major Solar Storm",  
 'preview': "For the first time in its eight years orbiting Mars, NASA's MAVEN mission witnessed two d
```

- Store all the dictionaries in a Python list.
 - Print the list in your notebook.
4. Optionally, store the scraped data in a file (to ease sharing the data with others). To do so, export the scraped data to a JSON file. (Note: there will be no extra points for completing this.)

Part 2: Scrape and Analyze 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 analyze Mars weather data.

1. Use automated browsing to visit the [Mars Temperature Data Site](https://static.bc-edx.com/data/web/mars_facts/temperature.html) (https://static.bc-edx.com/data/web/mars_facts/temperature.html). Inspect the page to identify which elements to scrape. Note that the URL is `https://static.bc-edx.com/data/web/mars_facts/temperature.html`.

HIDE HINT

To identify which elements to scrape, you might want to inspect the page by using Chrome DevTools to discover whether the table contains usable classes.

2. Create a BeautifulSoup 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 BeautifulSoup 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.

[HIDE HINT](#)

You can use the Pandas `astype` and `to_datetime` methods to accomplish this task.

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

- How many months exist on Mars?
- How many Martian (and not Earth) days worth of data exist in the scraped dataset?
- What are the coldest and the warmest months on Mars (at the location of Curiosity)? To answer this question:
 - Find the average minimum daily temperature for all of the months.
 - Plot the results as a bar chart.
- Which months have the lowest and the highest atmospheric pressure on Mars? To answer this question:
 - Find the average daily atmospheric pressure of all the months.
 - Plot the results as a bar chart.
- 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

Part 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 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)

Part 2: Scrape and Analyze Mars Weather Data (60 points)

- The HTML table was extracted into a Pandas DataFrame. Either Pandas or Splinter and BeautifulSoup 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: (10 points)
 - How many months exist on Mars? (5 points)
 - How many Martian days' worth of data are there? (5 points)
- The data was analyzed to answer the following questions, and a data visualization 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, visualizes, and analyzes 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. You may resubmit up to three times for a total of four submissions.

IMPORTANT

It is your responsibility to include a note in the README section of your repo specifying code source and its location within your repo. This applies if you have worked with a peer on an assignment, used code in which you did not author or create sourced from a forum such as Stack Overflow, or you received code outside curriculum content from support staff such as an Instructor, TA, Tutor, or Learning Assistant. This will provide visibility to grading staff of your circumstance in order to avoid flagging your work as plagiarized.

If you are struggling with a Challenge or any aspect of the curriculum, please remember that there are student support services available for you:

1. Office hours facilitated by your TA(s)
2. Tutor sessions ([sign up](https://tinyurl.com/BootCampTutorTeam) [↗\(https://tinyurl.com/BootCampTutorTeam\)](https://tinyurl.com/BootCampTutorTeam))
3. Ask the class Slack channel/get peer support
4. AskBCS Learning Assistants

References

[The Mars News website](https://static.bc-edx.com/data/web/mars_news/index.html) [↗\(https://static.bc-edx.com/data/web/mars_news/index.html\)](https://static.bc-edx.com/data/web/mars_news/index.html) is operated by edX Boot Camps LLC for educational purposes only. The news article titles, summaries, dates, and images were scraped from [NASA's Mars News](https://mars.nasa.gov/) [↗\(https://mars.nasa.gov/\)](https://mars.nasa.gov/) website in November 2022. Images are used according to the [JPL Image Use](#)

© 2023 edX Boot Camps LLC