

# Web Scraping Homework - Mission to Mars

---



In this assignment, you will build a web application that scrapes various websites for data related to the Mission to Mars and displays the information in a single HTML page. The following outlines what you need to do.

## Before You Begin

1. Create a new repository for this project called [web-scraping-challenge](#). **Do not add this homework to an existing repository.**
2. Clone the new repository to your computer.
3. Inside your local git repository, create a directory for the web scraping challenge. Use a folder name to correspond to the challenge: **Missions\_to\_Mars**.
4. Add your notebook files to this folder as well as your flask app.
5. Push the above changes to GitHub or GitLab.

## Step 1 - Scraping

Complete your initial scraping using Jupyter Notebook, BeautifulSoup, Pandas, and Requests/Splinter.

- Create a Jupyter Notebook file called [mission\\_to\\_mars.ipynb](#) and use this to complete all of your scraping and analysis tasks. The following outlines what you need to scrape.

### NASA Mars News

- Scrape the [NASA Mars News Site](#) and collect the latest News Title and Paragraph Text. Assign the text to variables that you can reference later.

```
# Example:  
news_title = "NASA's Next Mars Mission to Investigate Interior of Red Planet"  
  
news_p = "Preparation of NASA's next spacecraft to Mars, InSight, has ramped up  
this summer, on course for launch next May from Vandenberg Air Force Base in  
central California -- the first interplanetary launch in history from America's  
West Coast."
```

## JPL Mars Space Images - Featured Image

- Visit the url for JPL Featured Space Image [here](#).
- Use splinter to navigate the site and find the image url for the current Featured Mars Image and assign the url string to a variable called `featured_image_url`.
- Make sure to find the image url to the full size `.jpg` image.
- Make sure to save a complete url string for this image.

```
# Example:  
featured_image_url =  
'https://www.jpl.nasa.gov/spaceimages/images/largesize/PIA16225_hires.jpg'
```

## Mars Facts

- Visit the Mars Facts webpage [here](#) and use Pandas to scrape the table containing facts about the planet including Diameter, Mass, etc.
- Use Pandas to convert the data to a HTML table string.

## Mars Hemispheres

- Visit the USGS Astrogeology site [here](#) to obtain high resolution images for each of Mar's hemispheres.
- You will need to click each of the links to the hemispheres in order to find the image url to the full resolution image.
- Save both the image url string for the full resolution hemisphere image, and the Hemisphere title containing the hemisphere name. Use a Python dictionary to store the data using the keys `img_url` and `title`.
- Append the dictionary with the image url string and the hemisphere title to a list. This list will contain one dictionary for each hemisphere.

```
# Example:  
hemisphere_image_urls = [  
    {"title": "Valles Marineris Hemisphere", "img_url": "..."},  
    {"title": "Cerberus Hemisphere", "img_url": "..."},  
    {"title": "Schiaparelli Hemisphere", "img_url": "..."},  
    {"title": "Syrtis Major Hemisphere", "img_url": "..."},  
]
```

---

## Step 2 - MongoDB and Flask Application

Use MongoDB with Flask templating to create a new HTML page that displays all of the information that was scraped from the URLs above.

- Start by converting your Jupyter notebook into a Python script called `scrape_mars.py` with a function called `scrape` that will execute all of your scraping code from above and return one Python dictionary containing all of the scraped data.
- Next, create a route called `/scrape` that will import your `scrape_mars.py` script and call your `scrape` function.
  - Store the return value in Mongo as a Python dictionary.
- Create a root route `/` that will query your Mongo database and pass the mars data into an HTML template to display the data.
- Create a template HTML file called `index.html` that will take the mars data dictionary and display all of the data in the appropriate HTML elements. Use the following as a guide for what the final product should look like, but feel free to create your own design.

# Mission to Mars

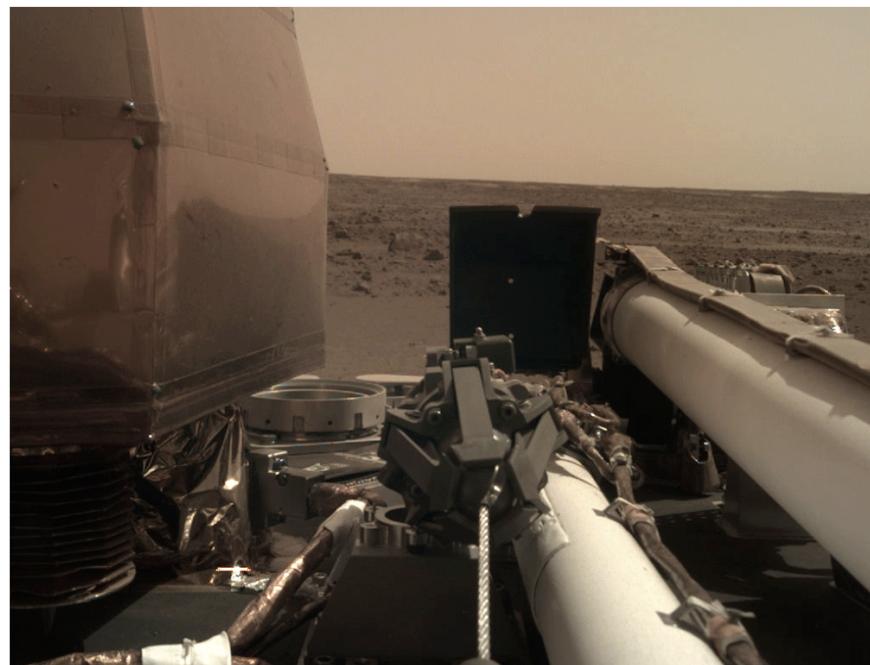
[Scrape New Data](#)

## Latest Mars News

### How NASA's Mars Helicopter Will Reach the Red Planet's Surface

The small craft will seek to prove that powered, controlled flight is possible on another planet. But just getting it onto the surface of Mars will take a whole lot of ingenuity.

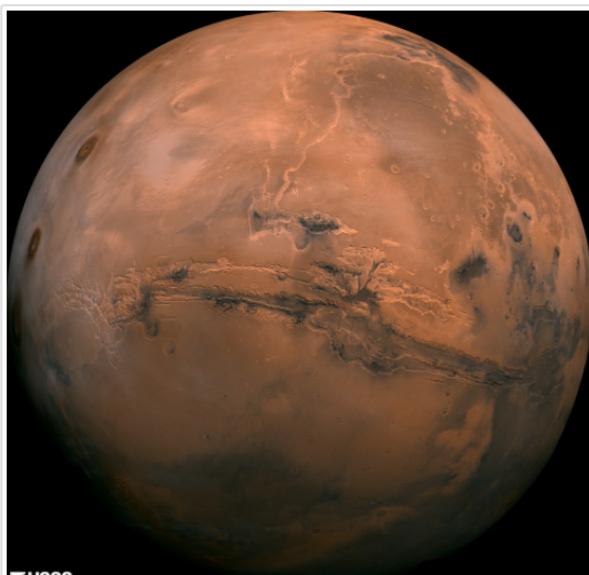
## Featured Mars Image



## Mars Facts

	Mars
Description	
Equatorial Diameter:	6,792 km
Polar Diameter:	6,752 km
Mass:	$6.39 \times 10^{23}$ kg (0.11 Earths)
Moons:	2 (Phobos & Deimos)
Orbit Distance:	227,943,824 km (1.38 AU)
Orbit Period:	687 days (1.9 years)
Surface Temperature:	-87 to -5 °C
First Record:	2nd millennium BC
Recorded By:	Egyptian astronomers

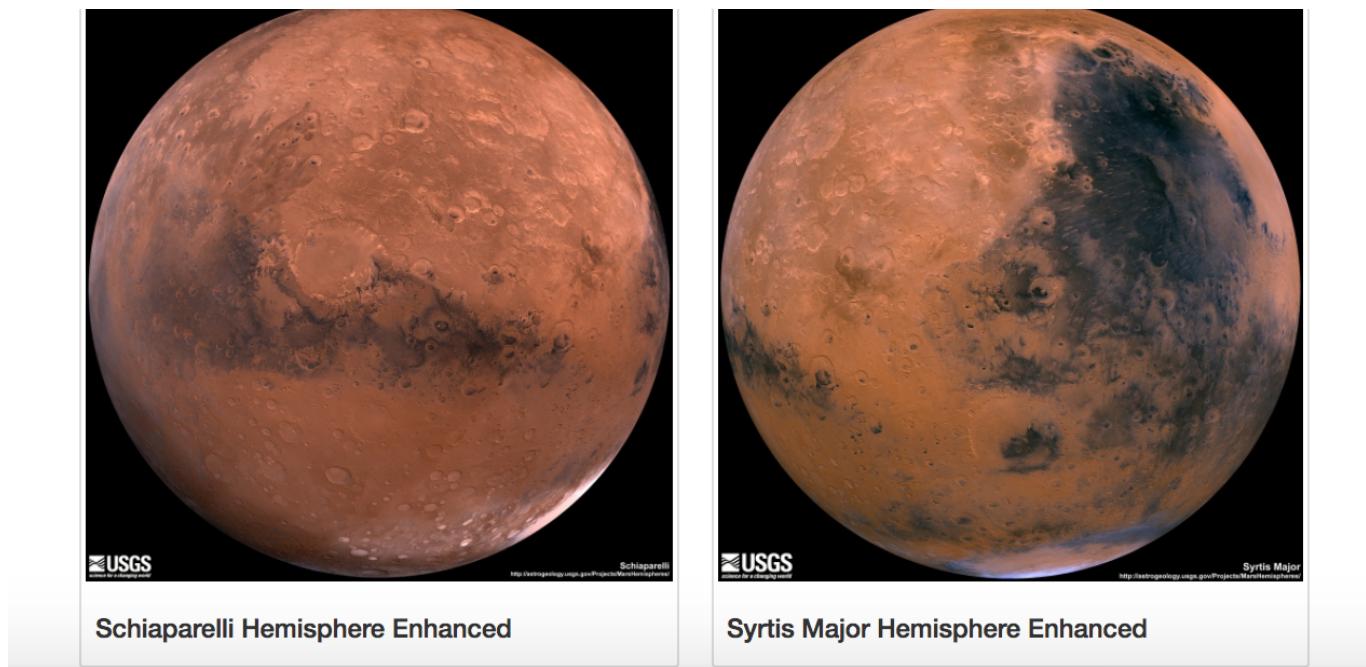
## Mars Hemispheres



Valles Marineris Hemisphere Enhanced



Cerberus Hemisphere Enhanced



## Step 3 - Submission

To submit your work to BootCampSpot, create a new GitHub repository and upload the following:

1. The Jupyter Notebook containing the scraping code used.
2. Screenshots of your final application.
3. Submit the link to your new repository to BootCampSpot.
4. Ensure your repository has regular commits (i.e. 20+ commits) and a thorough README.md file

## Hints

- Use Splinter to navigate the sites when needed and BeautifulSoup to help find and parse out the necessary data.
- Use Pymongo for CRUD applications for your database. For this homework, you can simply overwrite the existing document each time the `/scrape` url is visited and new data is obtained.
- Use Bootstrap to structure your HTML template.

## Copyright

Trilogy Education Services © 2020. All Rights Reserved.