In [1]:

```
# Import necessary libraries

from flask import Flask, render_template, redirect
from flask_pymongo import PyMongo

from splinter import Browser
from bs4 import BeautifulSoup as bs
import requests

import pandas as pd
```

In [2]:

```
executable_path ={'executable_path': 'chromedriver.exe'}
browser = Browser('chrome', **executable_path, headless=False)
```

In [3]:

```
# Sec A: Scraping
# Complete your initial scraping using Jupyter Notebook, BeautifulSoup, Panda
s, and Requests/Splinter.
```

In [4]:

```
# NASA Mars News
# Scrape the NASA Mars News Site and collect the latest News Title and Paragra
ph Text.
# Assign the text to variables that you can reference later.
# Example:
# news_title = "NASA's Next Mars Mission to Investigate Interior of Red Plane
t"
# 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."
```

In [5]:

```
url = 'https://mars.nasa.gov/news/?page=0&per_page=40&order=publish_date+desc%
2Ccreated_at+desc&search=&category=19%2C165%2C184%2C204&blank_scope=Latest'
browser.visit(url)
html = browser.html
soup = bs(html, 'html.parser')
```

In [6]:

```
news_title = soup.find('div', class_="content_title").text
news_text = soup.find('div', class_="article_teaser_body").text
print(news_title, "********", news_text)
```

```
NASA's Briefcase-Size MarCO Satellite Picks Up Honors ******** The twin space
craft, the first of their kind to fly into deep space, earn a Laureate from A
viation Week & Space Technology.
```

In [7]:

```
# 2. JPL Mars Space Images - Featured Image

# https://www.jpl.nasa.gov/spaceimages/?search=&category=Mars

# 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'
```

In [8]:

```
url = 'https://www.jpl.nasa.gov/spaceimages/?search=&category=Mars'
browser.visit(url)
html = browser.html
soup = bs(html, 'html.parser')
```

In [9]:

```
imgbody = soup.find("a", class_="fancybox").get('data-fancybox-href').strip()
imgurl = 'https://www.jpl.nasa.gov'+imgbody
print(imgurl)
```

```
https://www.jpl.nasa.gov/spaceimages/images/mediumsize/PIA17794_ip.jpg
```

In [10]:

```
# 3. Mars Weather

# https://twitter.com/marswxreport?lang=en

# Visit the Mars Weather twitter account
# and scrape the latest Mars weather tweet from the page. Save the tweet text
# for the weather report as a variable called mars_weather.
# Example:
# mars_weather = 'Sol 1801 (Aug 30, 2017), Sunny, high -21C/-5F, low -80C/-112
F, pressure at 8.82 hPa, daylight 06:09-17:55'
```

In [11]:

```
url = 'https://twitter.com/marswxreport?lang=en'
browser.visit(url)
html = browser.html
soup = bs(html, 'html.parser')
```

In [12]:

```
tweet = soup.find(class_="js-tweet-text").text
weather = tweet.split('pic')[0]
print(weather)
```

```
InSight sol 366 (2019-12-07) low -98.9ºC (-146.1ºF) high -20.4ºC (-4.8ºF)
winds from the SSE at 5.7 m/s (12.6 mph) gusting to 20.4 m/s (45.5 mph)
pressure at 6.60 hPa
```

In [13]:
```
# 4. Mars Facts

# https://space-facts.com/mars/

# Visit the Mars Facts webpage here and use Pandas to scrape the table contain
ing
# facts about the planet including Diameter, Mass, etc.

# Use Pandas to convert the data to a HTML table string.
```

In [14]:
```
url = 'https://space-facts.com/mars/'
```

In [15]:
```
tables = pd.read_html(url)
tables
```

Out[15]:
```
[                      0                              1
 0  Equatorial Diameter:                       6,792 km
 1       Polar Diameter:                       6,752 km
 2                 Mass:  6.39 × 10^23 kg (0.11 Earths)
 3                Moons:            2 (Phobos & Deimos)
 4       Orbit Distance:       227,943,824 km (1.38 AU)
 5         Orbit Period:           687 days (1.9 years)
 6  Surface Temperature:                   -87 to -5 °C
 7         First Record:              2nd millennium BC
 8          Recorded By:           Egyptian astronomers,
   Mars - Earth Comparison             Mars            Earth
 0               Diameter:         6,779 km        12,742 km
 1                   Mass:  6.39 × 10^23 kg  5.97 × 10^24 kg
 2                  Moons:                2                1
 3       Distance from Sun:   227,943,824 km   149,598,262 km
 4         Length of Year:   687 Earth days      365.24 days
 5            Temperature:    -153 to 20 °C      -88 to 58°C,
                       0                              1
 0  Equatorial Diameter:                       6,792 km
 1       Polar Diameter:                       6,752 km
 2                 Mass:  6.39 × 10^23 kg (0.11 Earths)
 3                Moons:            2 (Phobos & Deimos)
 4       Orbit Distance:       227,943,824 km (1.38 AU)
 5         Orbit Period:           687 days (1.9 years)
 6  Surface Temperature:                   -87 to -5 °C
 7         First Record:              2nd millennium BC
 8          Recorded By:           Egyptian astronomers]
```

In [16]:
```
df = tables[0]
df.columns = ['description','Value']
df.set_index('description', inplace=True)
df
```

Out[16]:

	Value
description	
Equatorial Diameter:	6,792 km
Polar Diameter:	6,752 km
Mass:	6.39 × 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

In [17]:
```
# 5. Mars Hemispheres
# https://astrogeology.usgs.gov/search/results?q=hemisphere+enhanced&k1=target
&v1=Mars

# 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 dictio
nary
# 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": "..."},
# ]
```

In [18]:
```
url = 'https://astrogeology.usgs.gov/search/results?q=hemisphere+enhanced&k1=t
arget&v1=Mars'
browser.visit(url)
html = browser.html
soup = bs(html, 'html.parser')
```

In [19]:

```
url_list = []
itemlinks = soup.find_all(class_="description")

for itemlink in itemlinks:
    item = itemlink.find('a', class_="itemLink product-item").get('href')
    url_list.append(item)
item_url_list = ['https://astrogeology.usgs.gov/' + item_url for item_url in u
rl_list]

print(item_url_list[0],item_url_list[1], item_url_list[2], item_url_list[3] )
```

```
https://astrogeology.usgs.gov//search/map/Mars/Viking/cerberus_enhanced http
s://astrogeology.usgs.gov//search/map/Mars/Viking/schiaparelli_enhanced http
s://astrogeology.usgs.gov//search/map/Mars/Viking/syrtis_major_enhanced http
s://astrogeology.usgs.gov//search/map/Mars/Viking/valles_marineris_enhanced
```

In [20]:

```
title_list = []
image_urls_list = []
for x in range (0,4):

    item_url = item_url_list[x]
    browser.visit(item_url)
    html = browser.html
    soup = bs(html, 'html.parser')
#     print(item_url)
    image_urls = soup.find("li").find('a')['href']
    title = soup.find('h2', class_="title").text
    title_list.append(title)
    image_urls_list.append(image_urls)
print(title_list)
print(image_urls_list)
# hemisphere_image_urls = zip(title_list,imgurl_list )
hemisphere_image_urls = [{"title": title_list, "img_url": image_urls_list}]
print(hemisphere_image_urls)
```

```
['Cerberus Hemisphere Enhanced', 'Schiaparelli Hemisphere Enhanced', 'Syrtis
Major Hemisphere Enhanced', 'Valles Marineris Hemisphere Enhanced']
['http://astropedia.astrogeology.usgs.gov/download/Mars/Viking/cerberus_enhan
ced.tif/full.jpg', 'http://astropedia.astrogeology.usgs.gov/download/Mars/Vik
ing/schiaparelli_enhanced.tif/full.jpg', 'http://astropedia.astrogeology.usg
s.gov/download/Mars/Viking/syrtis_major_enhanced.tif/full.jpg', 'http://astro
pedia.astrogeology.usgs.gov/download/Mars/Viking/valles_marineris_enhanced.ti
f/full.jpg']
[{'title': ['Cerberus Hemisphere Enhanced', 'Schiaparelli Hemisphere Enhance
d', 'Syrtis Major Hemisphere Enhanced', 'Valles Marineris Hemisphere Enhance
d'], 'img_url': ['http://astropedia.astrogeology.usgs.gov/download/Mars/Vikin
g/cerberus_enhanced.tif/full.jpg', 'http://astropedia.astrogeology.usgs.gov/d
ownload/Mars/Viking/schiaparelli_enhanced.tif/full.jpg', 'http://astropedia.a
strogeology.usgs.gov/download/Mars/Viking/syrtis_major_enhanced.tif/full.jp
g', 'http://astropedia.astrogeology.usgs.gov/download/Mars/Viking/valles_mari
neris_enhanced.tif/full.jpg']}]
```

In [ ]:

In [21]:

```
# Sec B - 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.
```

In [ ]:

```
```