WEB SCRAPING

In all the following questions, you have to use BeautifulSoup to scrape different websites and collect data as per the requirement of the question.

Every answer to the question should be in form of a python function which should take URL as the parameter. Use Jupyter Notebooks to program, upload it on your GitHub and send the link of the Jupyter notebook to your SME.

1) **Write a python program to display all the header tags from wikipedia.org.**

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
html = urlopen('https://en.wikipedia.org/wiki/Main_Page')
bs = BeautifulSoup(html, "html.parser")
titles = bs.find_all(['h1', 'h2','h3','h4','h5','h6'])
print('List all the header tags :', *titles, sep='\n\n')
```

2) **Write a python program to display IMDB's Top rated 100 movies' data (i.e. name, rating, year of release) and make data frame.**

```
import pandas as pd
from imdb import IMDb

def fetch_top_rated_movies():
    ia = IMDb()
    top_100_movies = ia.get_top250_movies()[:100]

    movies_data = []

    for movie in top_100_movies:
        title = movie['title']
        year = movie['year']
        rating = movie.data['rating']
        data = {"Title": title, "Year": year, "Rating": rating}
        movies_data.append(data)

    return movies_data

def create_dataframe(movies_data):
    df = pd.DataFrame(movies_data)
    return df

def main():
movies_data = fetch_top_rated_movies()

    if movies_data:
        df = create_dataframe(movies_data)
        print("Top 100 IMDb Movies:")
        print(df)
```

3) **Write a python program to scrape mentioned details from dineout.co.in : i) Restaurant name ii) Cuisine iii) Location iv) Ratings v) Image URL.**

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

def scrape_dineout_data(url="https://www.dineout.co.in/delhi-restaurants"):
    """Scrapes restaurant data from the given Dineout URL and returns a list of dictionaries.

    Args:
        url (str, optional): The URL of the Dineout restaurant listing page. Defaults to
"https://www.dineout.co.in/delhi-restaurants".

    Returns:
        list: A list of dictionaries, where each dictionary represents a restaurant with the following
keys:
            "Name": The name of the restaurant.
            "Cuisine": The type of cuisine served by the restaurant.
            "Location": The location of the restaurant.
            "Rating": The overall rating of the restaurant.
            "Image_URL": The URL of the restaurant's image.
    """

    try:
        response = requests.get(url)
        response.raise_for_status()  # Raise an exception for non-200 status codes

        soup = BeautifulSoup(response.content, 'html.parser')
        restaurant_data = []

        for restaurant in soup.find_all('div', class_='restnt-card'):
            try:
                name = restaurant.find('div', class_='restnt-info').h2.a.text.strip()
                cuisine = restaurant.find('span', class_='double-line-ellipsis').text.strip()
                location = restaurant.find('div', class_='restnt-loc').span.text.strip()
                rating = restaurant.find('div', class_='restnt-rating').text.strip()
                image_url = restaurant.find('div', class_='restnt-thumbnail').img['data-src']
                data = {"Name": name, "Cuisine": cuisine, "Location": location,
                        "Rating": rating, "Image_URL": image_url}
                restaurant_data.append(data)
            except AttributeError:
                # Handle missing elements gracefully
                # (e.g., log it or skip the restaurant)
                pass

        return restaurant_data

    except requests.exceptions.RequestException as e:
        print(f"Error fetching data: {e}")
        return None

def create_dataframe(restaurant_data):
    """Creates a Pandas DataFrame from a list of restaurant data dictionaries.
```

```
    Args:
        restaurant_data (list): A list of dictionaries, where each dictionary represents a restaurant.

    Returns:
        pd.DataFrame: A Pandas DataFrame containing the restaurant data.
    """

    df = pd.DataFrame(restaurant_data)
    return df

def main():
    url = input("Enter a Dineout URL (or press Enter for the default): ")
    if not url:
        url = "https://www.dineout.co.in/delhi-restaurants"

    restaurant_data = scrape_dineout_data(url)

    if restaurant_data:
        df = create_dataframe(restaurant_data)
        print("Dineout Restaurants Data:")
        print(df)

if __name__ == "__main__":
    main()
```

4) Write s python program to display list of respected former finance minister of India(i.e. Name , Term of office) from https://presidentofindia.nic.in/former-presidents.htm and make data frame.

```
import requests

from bs4 import BeautifulSoup

import pandas as pd


def scrape_finance_ministers():

    url = "https://presidentofindia.nic.in/former-presidents.htm"

    response = requests.get(url)


    if response.status_code == 200:

        soup = BeautifulSoup(response.text, 'html.parser')

        ministers_data = []


        finance_ministers_section = soup.find('div', class_='your-class-name')  # Replace 'your-class-name' with the actual class name or tag.
```

```
        for minister in finance_ministers_section.find_all('div', class_='individual-minister-class'):  # Replace 'individual-minister-class' with the actual class name or tag.
            name = minister.find('span', class_='minister-name').text.strip()
            term_of_office = minister.find('span', class_='minister-term').text.strip()

            data = {"Name": name, "Term of Office": term_of_office}
            ministers_data.append(data)

        return ministers_data
    else:
        print(f"Failed to retrieve data. Status code: {response.status_code}")
        return None

def create_dataframe(ministers_data):
    df = pd.DataFrame(ministers_data)
    return df

def main():
    finance_ministers_data = scrape_finance_ministers()

    if finance_ministers_data:
        df = create_dataframe(finance_ministers_data)
        print("Former Finance Ministers of India:")
        print(df)

if __name__ == "__main__":
    main()
```