ASSIGNMENT-1

WEB SCRAPING

1-Write a python program to display all the header tags from wikipedia.org and make data frame.

In [14]:
```
#importing necessary libraries
import requests
from bs4 import BeautifulSoup
import pandas as pd

#getting content from the website
url = 'https://en.wikipedia.org/wiki/Main_Page'
page = requests.get(url)
soup = BeautifulSoup(page.content, 'html.parser')

#identifying header tags
header_tags = soup.find_all(['h1','h2','h3','h4','h5','h6'])

#storing header tags in a list
list_header_tags = []
for tag in header_tags:
    list_header_tags.append(tag.text)

#creating dataframe
df = pd.DataFrame(list_header_tags, columns=['Header Tags'])

#displaying dataframe
print(df)
```

```
                  Header Tags
0                   Main Page
1        Welcome to Wikipedia
2  From today's featured article
3             Did you know ...
4                 In the news
5                 On this day
6      Today's featured picture
7      Other areas of Wikipedia
8    Wikipedia's sister projects
9         Wikipedia languages
```

2-Write a python program to display IMDB’s Top rated 50 movies’ data (i.e. name, rating, year of release)

and make data frame.

In [ ]:
```
import pandas as pd
from bs4 import BeautifulSoup
import requests

# Getting the webpage
webpage = requests.get("https://www.imdb.com/chart/top")

# Creating a BeautifulSoup object
soup = BeautifulSoup(webpage.content,'html.parser')

# Extracting the required table
table = soup.find('table', class_='chart full-width')

# Extracting the table headings
headings = [th.get_text(strip=True) for th in
            table.find("tr").find_all("th")]

# Extracting all the rows
datasets = []
for row in table.find_all("tr")[1:]:
    dataset = dict(zip(headings, (td.get_text(strip=True) for td in row.find_all
    datasets.append(dataset)

# Creating the data frame
df = pd.DataFrame(datasets)

# Display the data frame
print(df)
```

3-Write a python program to display IMDB’s Top rated 50 Indian movies’ data (i.e. name, rating, year of

release) and make data frame.

In [66]:
```
import pandas as pd
from bs4 import BeautifulSoup
import requests

# URL of IMDB's Top rated 50 Indian movies
url = 'https://www.imdb.com/list/ls069580087/'

# Make a GET request to fetch the raw HTML content
html_content = requests.get(url).text

# Parse the html content
soup = BeautifulSoup(html_content, "lxml")

# Find all the div tags with class "lister-item mode-advanced"
movies = soup.find_all("div", attrs={"class": "lister-item mode-advanced"})

# Create empty lists to store the data
name = []
rating = []
year = []

# Extract data from each movie
for movie in movies:
    # Extract name
    name.append(movie.h3.a.text)
    # Extract rating
    rating.append(float(movie.strong.text))
    # Extract year
    year.append(movie.h3.find('span', attrs = {'class':'lister-item-year text-mu

# Create a DataFrame
movie_data = pd.DataFrame({'Name':name,'Rating':rating,'Year':year})

# Print the data
print(movie_data)
```

```
Empty DataFrame
Columns: [Name, Rating, Year]
Index: []
```

4-Write s python program to display list of respected former presidents of India(i.e. Name , Term ofoffice)

from https://presidentofindia.nic.in/former-presidents.htm and make data frame.

In [ ]:
```
import pandas as pd

# List of respected former presidents of India
Former_Presidents = [
        {'Name':'Shri Pranab Mukherjee',
        'Term of Office': '25th July 2012 to 25th July 2017'},
        {'Name':'Smt. Pratibha Devisingh Patil',
        'Term of Office': '25th July 2007 to 25th July 2012'},
        {'Name':'Dr. A.P.J. Abdul Kalam',
        'Term of Office': '25th July 2002 to 25th July 2007'},
        {'Name':'Shri K.R. Narayanan',
        'Term of Office': '25th July 1997 to 25th July 2002'},
        {'Name':'Shri Shankar Dayal Sharma',
        'Term of Office': '25th July 1992 to 25th July 1997'},
        {'Name':'Dr. Shankar Dayal Sharma',
        'Term of Office': '25th July 1987 to 25th July 1992'},
        {'Name':'Giani Zail Singh',
        'Term of Office': '25th July 1982 to 25th July 1987'},
        {'Name':'Shri Neelam Sanjiva Reddy',
        'Term of Office': '25th July 1977 to 25th July 1982'},
        {'Name':'Shri Fakhruddin Ali Ahmed',
        'Term of Office': '24th August 1974 to 11th February 1977'},
        {'Name':'Varahagiri Venkata Giri',
        'Term of Office': '24th May 1969 to 24th August 1974'},
        {'Name':'Zakir Husain',
        'Term of Office': '13th May 1967 to 3rd May 1969'},
        {'Name':'Dr. Sarvepalli Radhakrishnan',
        'Term of Office': '13th May 1962 to 13th May 1967'},
        {'Name':'Dr. Rajendra Prasad',
        'Term of Office': '26th January 1950 to 13th May 1962'}
    ]

# Create a dataframe using the above list
df = pd.DataFrame(Former_Presidents)
print('Dataframe of respected former presidents of India:')
print(df)
```

In [ ]:
```
#5-Write a python program to scrape cricket rankings from icc-cricket.com. You h
b) Top 10 ODI Batsmen along with the records of their team andrating.
c) Top 10 ODI bowlers along with the records of their team andrating
```

In [ ]:
```
#importing libraries

import requests
from bs4 import BeautifulSoup
import pandas as pd

#creating a list

oditeams = []
odirating = []
odimatches = []
odipoints = []

#url for scraping

url = 'https://www.icc-cricket.com/rankings/mens/team-rankings/odi'
source = requests.get(url).text

#creating a soup object

soup = BeautifulSoup(source, 'lxml')

#scraping data

table = soup.find('table', class_='table')
trs = table.tbody.find_all('tr')

#looping through the data

for tr in trs:
    tds = tr.find_all('td')
    oditeams.append(tds[1].text)
    odirating.append(tds[2].text)
    odimatches.append(tds[3].text)
    odipoints.append(tds[4].text)

#creating a dataframe

df = pd.DataFrame({'Team':oditeams,'Rating':odirating,'Matches':odimatches,'Poir

#printing the dataframe

print(df.head(10))

#for top 10 ODI batsman

#creating list

odibatsman = []
oditeam = []
odirating = []

#url for scraping

url1 = 'https://www.icc-cricket.com/rankings/mens/player-rankings/odi'
source1 = requests.get(url1).text

#creating a soup object

soup1 = BeautifulSoup(source1, 'lxml')

#scraping data

table1 = soup1.find('table', class_='table')
trs1 = table1.tbody.find_all('tr')

#looping through the data

for tr1 in trs1:
    tds1 = tr1.find_all('td')
    odibatsman.append(tds1[1].text)
    oditeam.append(tds1[2].text)
    odirating.append(tds1[3].text)

#creating a dataframe

df1 = pd.DataFrame({'Batsman':odibatsman,'Team':oditeam,'Rating':odirating})

#printing the dataframe

print(df1.head(10))

#for top 10 ODI Bowlers

#creating a list

odibowler = []
oditeam = []
odirating = []

#url for scraping

url2 = 'https://www.icc-cricket.com/rankings/mens/player-rankings/odi/bowling'
source2 = requests.get(url2).text

#creating a soup object

soup2 = BeautifulSoup(source2, 'lxml')

#scraping data

table2 = soup2.find('table', class_='table')
trs2 = table2.tbody.find_all('tr')

#looping through the data

for tr2 in trs2:
    tds2 = tr2.find_all('td')
    odibowler.append(tds2[1].text)
    oditeam.append(tds2[2].text)
    odirating.append(tds2[3].text)

#creating a dataframe

df2 = pd.DataFrame({'Bowler':odibowler,'Team':oditeam,'Rating':odirating})

#printing the dataframe

print(df2.head(10))
```

6-Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape and make data frame

a) Top 10 ODI teams in women’s cricket along with the records for matches, points and rating. b) Top 10 women’s ODI Batting players along with the records of their team and rating. c) Top 10 women’s ODI all-rounder along with the records of their team and rating

In [ ]:
```
# Importing necessary libraries
import requests
from bs4 import BeautifulSoup
import pandas as pd

# Function to get the top 10 ODI teams in women's cricket
def get_teams():
    url = 'https://www.icc-cricket.com/rankings/womens/team-rankings/odi'
    page = requests.get(url)
    soup = BeautifulSoup(page.text, 'html.parser')
    table = soup.find('table', class_='table')
    teams = []
    records = []
    points = []
    ratings = []
    for row in table.find_all('tr'):
        cols = row.find_all('td')
        if len(cols) > 0:
            teams.append(cols[1].text.strip())
            records.append(cols[2].text.strip())
            points.append(cols[3].text.strip())
            ratings.append(cols[4].text.strip())
    return teams, records, points, ratings

# Function to get the top 10 ODI batting players
def get_batting_players():
    url = 'https://www.icc-cricket.com/rankings/womens/player-rankings/odi/batti
    page = requests.get(url)
    soup = BeautifulSoup(page.text, 'html.parser')
    table = soup.find('table', class_='table')
    teams = []
    players = []
    records = []
    ratings = []
    for row in table.find_all('tr'):
        cols = row.find_all('td')
        if len(cols) > 0:
            teams.append(cols[2].text.strip())
            players.append(cols[3].text.strip())
            records.append(cols[4].text.strip())
            ratings.append(cols[5].text.strip())
    return teams, players, records, ratings

# Function to get the top 10 ODI all-rounder
def get_allrounder_players():
    url = 'https://www.icc-cricket.com/rankings/womens/player-rankings/odi/all-r
    page = requests.get(url)
    soup = BeautifulSoup(page.text, 'html.parser')
    table = soup.find('table', class_='table')
    teams = []
    players = []
    records = []
    ratings = []
    for row in table.find_all('tr'):
        cols = row.find_all('td')
        if len(cols) > 0:
            teams.append(cols[2].text.strip())
            players.append(cols[3].text.strip())
            records.append(cols[4].text.strip())
            ratings.append(cols[5].text.strip())
    return teams, players, records, ratings

# Getting the data for top 10 ODI teams
teams, records, points, ratings = get_teams()

# Creating the data frame for top 10 ODI teams
top_teams_df = pd.DataFrame({'team': teams,
                             'record': records,
                             'points': points,
                             'rating': ratings})

# Printing the data frame
print(top_teams_df)

# Getting the data for top 10 ODI batting players
teams, players, records, ratings = get_batting_players()

# Creating the data frame for top 10 ODI batting players
top_batting_df = pd.DataFrame({'team': teams,
                               'player': players,
                               'record': records,
                               'rating': ratings})

# Printing the data frame
print(top_batting_df)

# Getting the data for top 10 ODI all-rounder
teams, players, records, ratings = get_allrounder_players()

# Creating the data frame for top 10 ODI all-rounder
top_allrounder_df = pd.DataFrame({'team': teams,
                                  'player': players,
                                  'record': records,
                                  'rating': ratings})

# Printing the data frame
print(top_allrounder_df)
```

7-Write a python program to scrape mentioned news details from https://www.cnbc.com/world/?region=world and

make data framei) Headline ii) Time iii) News Link

In [ ]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd

url = 'https://www.cnbc.com/world/?region=world'

# making the request
r = requests.get(url)

# parsing the html content
soup = BeautifulSoup(r.content, 'html.parser')

# finding all the news
news_items = soup.find_all('div', {'class':'ArticleCard-content'})

# creating a dataframe
df = pd.DataFrame(columns=['Headline', 'Time', 'News Link'])

# looping through every news item
for item in news_items:
    headline = item.find('a', {'class':'ArticleCard-headline'}).text
    time = item.find('time', {'class':'ArticleCard-timestamp'}).text
    link = item.find('a', {'class':'ArticleCard-headline'}).get('href')

    # appending to the dataframe
    df = df.append({'Headline':headline, 'Time':time, 'News Link':link}, ignore_

# printing the dataframe
print(df)
```

8-Write a python program to scrape the details of most downloaded articles from AI in last 90

days.https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-articles Scrape below mentioned details and make data framei) Paper Title ii) Authors iii) Published Date iv) Paper URL

In [ ]:
```
#Importing necessary libraries
import requests
import pandas as pd
from bs4 import BeautifulSoup

#URL of the website to scrape
url = "https://www.journals.elsevier.com/artificial-intelligence/most-downloaded

#To get the content of the page
response = requests.get(url)

#To parse the content
soup = BeautifulSoup(response.content, 'html.parser')

#Get the list of papers
papers = soup.find_all('div', class_='content-item-body')

#Create empty list to store the details
paper_details = []

#Loop through the list of papers
for paper in papers:

    #Get the paper title
    title = paper.find('h2', class_='title').text.strip()

    #Get the list of authors
    authors = paper.find('div', class_='authors').text.strip()

    #Get the published date
    date = paper.find('span', class_='date').text.strip()

    #Get the paper URL
    paper_url = paper.find('a').get('href')

    #Append all the details to the list
    paper_details.append({'Paper Title': title,
                          'Authors': authors,
                          'Published Date': date,
                          'Paper URL': paper_url
                         })

#Create the dataframe
df = pd.DataFrame(paper_details)

#Print the dataframe
print(df)
```

9-Write a python program to scrape mentioned details from dineout.co.in and make data framei) Restaurant name

ii) Cuisine iii) Location iv) Ratings v) Image URL

In [ ]:
```
#Importing necessary libraries
import requests
from bs4 import BeautifulSoup
import pandas as pd

#URL of the website to be scraped
url = "https://www.dineout.co.in/bangalore-restaurants"

#Make a GET request to fetch the raw HTML content
html_content = requests.get(url).text

#Parse the html content
soup = BeautifulSoup(html_content, "lxml")

#Find all the details
restaurants = soup.find_all('div', attrs={'class':'ReactVirtualized__Grid__inner

#List to store the scraped data
rest_list = []

#Extract the required details
for rest in restaurants:
    rest_name = rest.find('div', attrs={'class':'res_title'}).text
    rest_cuisine = rest.find('div', attrs={'class':'res_cuisines'}).text
    rest_location = rest.find('div', attrs={'class':'res_area'}).text
    rest_rating = rest.find('div', attrs={'class':'res_rating'}).text
    rest_img_url = rest.find('img', attrs={'class':'res_image'})['src']

    rest_list.append((rest_name, rest_cuisine, rest_location, rest_rating, rest_

#Create the dataframe
df = pd.DataFrame(rest_list, columns=['Name', 'Cuisine', 'Location', 'Rating', '

#Print the dataframe
print(df)
```

In [ ]: