ASSIGNMENT-1

WEB SCRAPING

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

In [1]:
```
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')
```

```
List all the header tags :

<h1 class="firstHeading mw-first-heading" id="firstHeading" style="display: n
one"><span class="mw-page-title-main">Main Page</span></h1>

<h1><span class="mw-headline" id="Welcome_to_Wikipedia">Welcome to <a href="/
wiki/Wikipedia" title="Wikipedia">Wikipedia</a></span></h1>

<h2 class="mp-h2" id="mp-tfa-h2"><span id="From_today.27s_featured_article">
</span><span class="mw-headline" id="From_today's_featured_article">From toda
y's featured article</span></h2>

<h2 class="mp-h2" id="mp-dyk-h2"><span class="mw-headline" id="Did_you_know
_...">Did you know ...</span></h2>

<h2 class="mp-h2" id="mp-itn-h2"><span class="mw-headline" id="In_the_news">I
n the news</span></h2>

<h2 class="mp-h2" id="mp-otd-h2"><span class="mw-headline" id="On_this_day">O
n this day</span></h2>

<h2 class="mp-h2" id="mp-tfp-h2"><span id="Today.27s_featured_picture"></span
><span class="mw-headline" id="Today's_featured_picture">Today's featured pic
ture</span></h2>

<h2 class="mp-h2" id="mp-other"><span class="mw-headline" id="Other_areas_of_
Wikipedia">Other areas of Wikipedia</span></h2>

<h2 class="mp-h2" id="mp-sister"><span id="Wikipedia.27s_sister_projects"></s
pan><span class="mw-headline" id="Wikipedia's_sister_projects">Wikipedia's si
ster projects</span></h2>

<h2 class="mp-h2" id="mp-lang"><span class="mw-headline" id="Wikipedia_langua
ges">Wikipedia languages</span></h2>
```

3) Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape and make data framea) Top 10 ODI teams in men’s cricket along with the records for matches, points and rating.

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 [2]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd

# Function to scrape and create a data frame
def scrape_and_create_dataframe(url, columns):
    response = requests.get(url)
    if response.status_code == 200:
        soup = BeautifulSoup(response.content, 'html.parser')
        data = []

        # Scrape data from the table
        table = soup.find('table', class_='table')
        rows = table.find_all('tr')[1:]  # Skip header row

        for row in rows:
            cols = row.find_all('td')
            record = [col.text.strip() for col in cols]
            data.append(record)

        # Create a data frame
        df = pd.DataFrame(data, columns=columns)
        return df
    else:
        print("Error:", response.status_code)
        return None

# Scrape and create data frames for top 10 ODI teams, batsmen, and bowlers
teams_url = "https://www.icc-cricket.com/rankings/mens/team-rankings/odi"
batsmen_url = "https://www.icc-cricket.com/rankings/mens/player-rankings/odi/ba
bowlers_url = "https://www.icc-cricket.com/rankings/mens/player-rankings/odi/bo

team_columns = ["Position", "Team", "Matches", "Points", "Rating"]
batsmen_columns = ["Position", "Batsman", "Team", "Rating", "Career Best Rating
bowlers_columns = ["Position", "Bowler", "Team", "Rating", "Career Best Rating'

top_10_teams_df = scrape_and_create_dataframe(teams_url, team_columns)
top_10_batsmen_df = scrape_and_create_dataframe(batsmen_url, batsmen_columns)
top_10_bowlers_df = scrape_and_create_dataframe(bowlers_url, bowlers_columns)

# Display the data frames
print("Top 10 ODI Teams:")
print(top_10_teams_df)

print("\nTop 10 ODI Batsmen:")
print(top_10_batsmen_df)

print("\nTop 10 ODI Bowlers:")
print(top_10_bowlers_df)
```

```
Top 10 ODI Teams:
   Position               Team Matches Points Rating
0         1      Australia\nAUS      23  2,714    118
1         2       Pakistan\nPAK      20  2,316    116
2         3          India\nIND      36  4,081    113
3         4     New Zealand\nNZ      27  2,806    104
4         5        England\nENG      24  2,426    101
5         6    South Africa\nSA      19  1,910    101
6         7     Bangladesh\nBAN      28  2,661     95
7         8    Afghanistan\nAFG      16  1,404     88
8         9       Sri Lanka\nSL      32  2,794     87
9        10     West Indies\nWI      38  2,582     68
10       11       Zimbabwe\nZIM      30  1,641     55
11       12       Scotland\nSCO      33  1,662     50
12       13        Ireland\nIRE      24  1,052     44
13       14    Netherlands\nNED      28  1,044     37
14       15          Nepal\nNEP      40  1,396     35
15       16        Namibia\nNAM      28    813     29
16       17  United States\nUSA      31    808     26
17       18           Oman\nOMA      24    525     22
18       19            UAE\nUAE      41    617     15

Top 10 ODI Batsmen:
                                                Position               Batsman  \
0                1\n                         \n\n\n(0)            Babar Azam
1        2\n                                 \n\n\n(0)  Rassie van der Dussen
2        3\n                                 \n\n\n(0)           Fakhar Zaman
3        4\n                                 \n\n\n(0)            Imam-ul-Haq
4   5\n                                 \n\n\n\n\n(...           Shubman Gill
..                                                   ...                   ...
95      96\n                                 \n\n\n(0)         Dasun Shanaka
96      97\n                                 \n\n\n(0)      Avishka Fernando
97      98\n                                 \n\n\n(0)             Ryan Burl
98      99\n                                 \n\n\n(0)            Finn Allen
99     100\n                                 \n\n\n(0)     Wanindu Hasaranga

   Team Rating            Career Best Rating
0   PAK    886   898 v West Indies, 10/06/2022
1    SA    777       796 v England, 19/07/2022
2   PAK    755   784 v New Zealand, 29/04/2023
3   PAK    745   815 v West Indies, 12/06/2022
4   IND    743   743 v West Indies, 01/08/2023
..  ...    ...                             ...
95   SL    440         506 v India, 10/01/2023
96   SL    439  591 v South Africa, 02/09/2021
97  ZIM    437      437 v Scotland, 04/07/2023
98   NZ    434      500 v Pakistan, 09/01/2023
99   SL    433         433 v India, 12/01/2023

[100 rows x 5 columns]

Top 10 ODI Bowlers:
                                                Position           Bowler Team  \
0                1\n                         \n\n\n(0)   Josh Hazlewood  AUS
1        2\n                                 \n\n\n(0)   Mitchell Starc  AUS
```

```
2       3\n                               \n\n\n(0)      Rashid Khan  AFG
3       4\n                               \n\n\n(0)   Mohammed Siraj  IND
4       5\n                               \n\n\n(0)       Matt Henry   NZ
..                                              ...              ...  ...
95  96\n                               \n\n\n\n\n...    Henry Shipley   NZ
96  97\n                               \n\n\n\n\n...    Kasun Rajitha   SL
97      98\n                               \n\n\n(0)  Lalit Rajbanshi  NEP
98  99\n                               \n\n\n\n\n...   Ebadot Hossain  BAN
99  =\n                               \n\n\n\n\n\...       Hamza Tahir  SCO

   Rating             Career Best Rating
0     705      733 v England, 26/01/2018
1     686  783 v New Zealand, 29/03/2015
2     682     806 v Pakistan, 21/09/2018
3     670  736 v New Zealand, 21/01/2023
4     667   691 v Bangladesh, 26/03/2021
..    ...                            ...
95    400     400 v Pakistan, 07/05/2023
96    395     411 v Scotland, 27/06/2023
97    388  397 v West Indies, 22/06/2023
98    385  389 v Afghanistan, 08/07/2023
99    385        484 v Nepal, 17/07/2022

[100 rows x 5 columns]
```

4) Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape and make data framea) 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 [3]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd

# Function to scrape data for a given URL
def scrape_data(url):
    response = requests.get(url)
    if response.status_code == 200:
        return response.content
    else:
        raise Exception(f"Failed to fetch data from {url}")

# Function to create a dataframe from the scraped data
def create_dataframe(data, columns):
    return pd.DataFrame(data, columns=columns)

# Scrape and create dataframe for Top 10 ODI teams
def scrape_top_10_teams():
    url = "https://www.icc-cricket.com/rankings/womens/team-rankings/odi"
    content = scrape_data(url)
    soup = BeautifulSoup(content, "html.parser")

    teams = []
    matches = []
    points = []
    rating = []

    table = soup.find("table", class_="table")
    rows = table.find_all("tr")[1:11]  # Exclude header row and get top 10 tean

    for row in rows:
        cols = row.find_all("td")
        teams.append(cols[1].text.strip())
        matches.append(cols[2].text.strip())
        points.append(cols[3].text.strip())
        rating.append(cols[4].text.strip())

    data = {
        "Team": teams,
        "Matches": matches,
        "Points": points,
        "Rating": rating
    }

    df = create_dataframe(data, columns=["Team", "Matches", "Points", "Rating"]
    return df

# Scrape and create dataframe for Top 10 women's ODI batting players
def scrape_top_10_batting_players():
    url = "https://www.icc-cricket.com/rankings/womens/player-rankings/odi/batt
    content = scrape_data(url)
    soup = BeautifulSoup(content, "html.parser")

    players = []
    teams = []
    ratings = []
    table = soup.find("table", class_="table")
    rows = table.find_all("tr")[1:11]  # Exclude header row and get top 10 play

    for row in rows:
        cols = row.find_all("td")
        players.append(cols[1].text.strip())
        teams.append(cols[2].text.strip())
        ratings.append(cols[4].text.strip())

    data = {
        "Player": players,
        "Team": teams,
        "Rating": ratings
    }

    df = create_dataframe(data, columns=["Player", "Team", "Rating"])
    return df

# Scrape and create dataframe for Top 10 women's ODI all-rounders
def scrape_top_10_allrounders():
    url = "https://www.icc-cricket.com/rankings/womens/player-rankings/odi/all-
    content = scrape_data(url)
    soup = BeautifulSoup(content, "html.parser")

    players = []
    teams = []
    ratings = []

    table = soup.find("table", class_="table")
    rows = table.find_all("tr")[1:11]  # Exclude header row and get top 10 play

    for row in rows:
        cols = row.find_all("td")
        players.append(cols[1].text.strip())
        teams.append(cols[2].text.strip())
        ratings.append(cols[4].text.strip())

    data = {
        "Player": players,
        "Team": teams,
        "Rating": ratings
    }

    df = create_dataframe(data, columns=["Player", "Team", "Rating"])
    return df

if __name__ == "__main__":
    top_10_teams_df = scrape_top_10_teams()
    top_10_batting_players_df = scrape_top_10_batting_players()
    top_10_allrounders_df = scrape_top_10_allrounders()

    print("Top 10 ODI Teams:")
    print(top_10_teams_df)

    print("\nTop 10 Women's ODI Batting Players:")
    print(top_10_batting_players_df)
    print("\nTop 10 Women's ODI All-Rounders:")
    print(top_10_allrounders_df)
```

```
Top 10 ODI Teams:
               Team Matches Points Rating
0    Australia\nAUS      26  4,290    165
1      England\nENG      31  3,875    125
2  South Africa\nSA      26  3,098    119
3        India\nIND      30  3,039    101
4   New Zealand\nNZ      28  2,688     96
5   West Indies\nWI      29  2,743     95
6   Bangladesh\nBAN      17  1,284     76
7     Sri Lanka\nSL      12    820     68
8     Thailand\nTHA      13    883     68
9     Pakistan\nPAK      27  1,678     62

Top 10 Women's ODI Batting Players:
                 Player Team                          Rating
0  Natalie Sciver-Brunt  ENG    803 v Australia, 18/07/2023
1   Chamari Athapaththu   SL  758 v New Zealand, 03/07/2023
2           Beth Mooney  AUS      776 v England, 12/07/2023
3       Laura Wolvaardt   SA    741 v Australia, 22/03/2022
4       Smriti Mandhana  IND      797 v England, 28/02/2019
5          Alyssa Healy  AUS      785 v England, 03/04/2022
6      Harmanpreet Kaur  IND      731 v England, 21/09/2022
7          Ellyse Perry  AUS  766 v West Indies, 11/09/2019
8           Meg Lanning  AUS  834 v New Zealand, 24/02/2016
9       Stafanie Taylor   WI     766 v Pakistan, 07/07/2021

Top 10 Women's ODI All-Rounders:
                 Player Team                           Rating
0  Natalie Sciver-Brunt  ENG     421 v Australia, 18/07/2023
1      Ashleigh Gardner  AUS       389 v Ireland, 28/07/2023
2       Hayley Matthews   WI       392 v Ireland, 26/06/2023
3        Marizanne Kapp   SA   419 v West Indies, 10/09/2021
4          Ellyse Perry  AUS   548 v West Indies, 11/09/2019
5           Amelia Kerr   NZ   356 v West Indies, 25/09/2022
6         Deepti Sharma  IND  397 v South Africa, 09/10/2019
7        Jess Jonassen  AUS   308 v West Indies, 11/09/2019
8         Sophie Devine   NZ     305 v Australia, 05/10/2020
9              Nida Dar  PAK     232 v Australia, 21/01/2023
```

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

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

In [4]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd

# Define the URL to scrape
url = "https://www.cnbc.com/world/?region=world"

# Send a GET request to the URL
response = requests.get(url)

# Check if the request was successful
if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    news_list = soup.find_all('div', class_='Card-titleContainer')

    headlines = []
    times = []
    news_links = []

    for news in news_list:
        headline_elem = news.find('a', class_='Card-titleLink')
        time_elem = news.find('time')
        link_elem = news.find('a', class_='Card-titleLink')

        if headline_elem and time_elem and link_elem:
            headline = headline_elem.text
            time = time_elem.text
            link = link_elem['href']

            headlines.append(headline)
            times.append(time)
            news_links.append(link)

    # Create a DataFrame
    news_data = {
        'Headline': headlines,
        'Time': times,
        'News Link': news_links
    }
    df = pd.DataFrame(news_data)

    # Print the DataFrame
    print(df)
else:
    print("Failed to retrieve the webpage.")
```

```
Empty DataFrame
Columns: [Headline, Time, News Link]
Index: []
```

6) 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 (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 [5]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd

# URL of the most downloaded articles page
url = "https://www.journals.elsevier.com/artificial-intelligence/most-downloade

# Send an HTTP GET request to the URL
response = requests.get(url)
content = response.content

# Parse the HTML content using BeautifulSoup
soup = BeautifulSoup(content, "html.parser")

# Find all the article details
articles = soup.find_all("div", class_="pod-listing-header")

# Initialize empty lists to store the scraped data
titles = []
authors = []
published_dates = []
paper_urls = []

# Loop through each article and extract the required details
for article in articles:
    # Extract paper title
    title = article.find("h2").text.strip()
    titles.append(title)

    # Extract authors
    author = article.find("div", class_="text-s").text.strip()
    authors.append(author)

    # Extract published date
    published_date = article.find("span", class_="text-xs").text.strip()
    published_dates.append(published_date)

    # Extract paper URL
    paper_url = "https://www.journals.elsevier.com" + article.find("a")["href"]
    paper_urls.append(paper_url)

# Create a DataFrame from the scraped data
data = {
    "Paper Title": titles,
    "Authors": authors,
    "Published Date": published_dates,
    "Paper URL": paper_urls
}

df = pd.DataFrame(data)

# Display the DataFrame
print(df)
```

```
Empty DataFrame
Columns: [Paper Title, Authors, Published Date, Paper URL]
Index: []
```

7) Write a python program to scrape mentioned details from dineout.co.inand make data framei) Restaurant name

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

In [6]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd

# URL of the dineout.co.in page you want to scrape
url = 'https://www.dineout.co.in/delhi-restaurants'

# Send a GET request to the URL
response = requests.get(url)

# Parse the HTML content using Beautiful Soup
soup = BeautifulSoup(response.content, 'html.parser')

# Lists to store scraped data
restaurant_names = []
cuisines = []
locations = []
ratings = []
image_urls = []

# Find all restaurant containers
restaurant_containers = soup.find_all('div', class_='restnt-info-section')

# Loop through each restaurant container
for container in restaurant_containers:
    # Restaurant Name
    name = container.find('div', class_='restnt-name').text.strip()
    restaurant_names.append(name)

    # Cuisine
    cuisine = container.find('div', class_='restnt-cuisine').text.strip()
    cuisines.append(cuisine)

    # Location
    location = container.find('div', class_='restnt-loc').text.strip()
    locations.append(location)

    # Ratings
    rating = container.find('div', class_='restnt-rating').text.strip()
    ratings.append(rating)

    # Image URL
    image = container.find('img')['src']
    image_urls.append(image)

# Create a DataFrame
data = {
    'Restaurant Name': restaurant_names,
    'Cuisine': cuisines,
    'Location': locations,
    'Ratings': ratings,
    'Image URL': image_urls
}

df = pd.DataFrame(data)

# Display the DataFrame
```

```
print(df)
```

```
Empty DataFrame
Columns: [Restaurant Name, Cuisine, Location, Ratings, Image URL]
Index: []
```