In [ ]:

```
ASSIGNMENT-1
WEB SCRAPING
```

In [ ]:

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

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: none"><sp
an class="mw-page-title-main">Main Page</span></h1>

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

<h2 class="mp-h2" id="mp-tfa-h2"><span id="From_today.27s_featured_article"></span><s
pan class="mw-headline" id="From_today's_featured_article">From today's featured arti
cle</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">In the ne
ws</span></h2>

<h2 class="mp-h2" id="mp-otd-h2"><span class="mw-headline" id="On_this_day">On this d
ay</span></h2>

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

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

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

<h2 class="mp-h2" id="mp-lang"><span class="mw-headline" id="Wikipedia_languages">Wik
ipedia languages</span></h2>
```

In [2]:

```
pip install requests beautifulsoup4 pandas
```

```
Requirement already satisfied: requests in c:\users\gaura\anaconda3\lib\site-packages
(2.31.0)
Requirement already satisfied: beautifulsoup4 in c:\users\gaura\anaconda3\lib\site-pa
ckages (4.12.2)
Requirement already satisfied: pandas in c:\users\gaura\anaconda3\lib\site-packages
(2.0.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\gaura\anaconda3\l
ib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\gaura\anaconda3\lib\site-pack
ages (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\gaura\anaconda3\lib\sit
e-packages (from requests) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\gaura\anaconda3\lib\sit
e-packages (from requests) (2023.7.22)
Requirement already satisfied: soupsieve>1.2 in c:\users\gaura\anaconda3\lib\site-pac
kages (from beautifulsoup4) (2.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\gaura\anaconda3\lib
\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\gaura\anaconda3\lib\site-pack
ages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\gaura\anaconda3\lib\site-pa
ckages (from pandas) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in c:\users\gaura\anaconda3\lib\site-pac
kages (from pandas) (1.24.3)
Requirement already satisfied: six>=1.5 in c:\users\gaura\anaconda3\lib\site-packages
(from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [ ]:

```
Q-2 Write s python program to display list of respected former presidents of India(i.e
from https://presidentofindia.nic.in/former-presidents.htm and make data frame.
```

In [ ]:

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

# Send a GET request to the website
url = "https://presidentofindia.nic.in/former-presidents.htm"
response = requests.get(url)

# Create a BeautifulSoup object to parse the HTML content
soup = BeautifulSoup(response.content, "html.parser")

# Find the table containing the information
table = soup.find("table")

# Create empty lists to store the data
names = []
terms = []
# Iterate over each row in the table

for row in table.find_all("tr")[1:]:
  # Extract the name and term of office from the columns
  columns = row.find_all("td")
  name = columns[0].text.strip()
  term = columns[1].text.strip()

  # Append the data to the respective lists
  names.append(name)
  terms.append(term)
```

```
# Create a data frame using the lists
data = {"Name": names, "Term of Office": terms}
df = pd.DataFrame(data)

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

In [ ]:

```
Q-3 Write a python program to scrape cricket rankings from icc-cricket.com. You have t
a) To scrape the top 10 ODI teams in men's cricket along with the records for matches,
```

In [ ]:

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

url = "https://www.icc-cricket.com/rankings/mens/team-rankings/odi"
response = requests.get(url)
soup = BeautifulSoup(response.content, "html.parser")

team_data = []
table = soup.find("table", class_="table")
rows = table.find_all("tr")

for row in rows[1:11]:
  cells = row.find_all("td")
  team = cells[1].text.strip()
  matches = cells[2].text.strip()
  points = cells[3].text.strip()
  rating = cells[4].text.strip()
  team_data.append([team, matches, points, rating])

df = pd.DataFrame(team_data, columns=["Team", "Matches", "Points", "Rating"])
print(df)
```

In [ ]:

```
Q-3 Write a python program to scrape cricket rankings from icc-cricket.com. You have t
b) Top 10 ODI Batsmen along with the records of their team andrating.
```

In [ ]:

```
url = "https://www.icc-cricket.com/rankings/mens/player-rankings/odi/batting"
response = requests.get(url)
soup = BeautifulSoup(response.content, "html.parser")

batsman_data = []
table = soup.find("table", class_="table")
rows = table.find_all("tr")

for row in rows[1:11]:
  cells = row.find_all("td")
  batsman = cells[1].text.strip()
  team = cells[2].text.strip()
  rating = cells[3].text.strip()
  batsman_data.append([batsman, team, rating])

df = pd.DataFrame(batsman_data, columns=["Batsman", "Team", "Rating"])
print(df)
```

In [ ]:

```
Q-3 Write a python program to scrape cricket rankings from icc-cricket.com. You have t
c) Top 10 ODI bowlers along with the records of their team andrating.
```

In [ ]:

```
url = "https://www.icc-cricket.com/rankings/mens/player-rankings/odi/bowling"
response = requests.get(url)
soup = BeautifulSoup(response.content, "html.parser")

bowler_data = []
table = soup.find("table", class_="table")
rows = table.find_all("tr")

for row in rows[1:11]:
  cells = row.find_all("td")
  bowler = cells[1].text.strip()
  team = cells[2].text.strip()
  rating = cells[3].text.strip()
  bowler_data.append([bowler, team, rating])

df = pd.DataFrame(bowler_data, columns=["Bowler", "Team", "Rating"])
print(df)
```

In [ ]:

```
Q-4 Write a python program to scrape cricket rankings from icc-cricket.com. You have t
a) Top 10 ODI teams in women’s cricket along with the records for matches, points and
```

In [ ]:

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

url = "https://www.icc-cricket.com/rankings/mens/team-rankings/odi"
response = requests.get(url)
soup = BeautifulSoup(response.content, "html.parser")

team_data = []
table = soup.find("table", class_="table")
rows = table.find_all("tr")

for row in rows[1:11]:
  cells = row.find_all("td")
  team = cells[1].text.strip()
  matches = cells[2].text.strip()
  points = cells[3].text.strip()
  rating = cells[4].text.strip()
  team_data.append([team, matches, points, rating])

df = pd.DataFrame(team_data, columns=["Team", "Matches", "Points", "Rating"])
print(df)
```

In [ ]:

```
Q-4 Write a python program to scrape cricket rankings from icc-cricket.com. You have t
b) Top 10 women’s ODI Batting players along with the records of their team and rating
```

In [ ]:

```
url = "https://www.icc-cricket.com/rankings/mens/player-rankings/odi/batting"
response = requests.get(url)
soup = BeautifulSoup(response.content, "html.parser")

batsman_data = []
table = soup.find("table", class_="table")
rows = table.find_all("tr")

for row in rows[1:11]:
  cells = row.find_all("td")
  batsman = cells[1].text.strip()
```

```
    team = cells[2].text.strip()
    rating = cells[3].text.strip()
    batsman_data.append([batsman, team, rating])

df = pd.DataFrame(batsman_data, columns=["Batsman", "Team", "Rating"])
print(df)
```

In [ ]:

```
Q-4 Write a python program to scrape cricket rankings from icc-cricket.com. You have t
c) Top 10 women’s ODI all-rounder along with the records of their team and rating.
```

In [ ]:

```
url = "https://www.icc-cricket.com/rankings/mens/player-rankings/odi/bowling"
response = requests.get(url)
soup = BeautifulSoup(response.content, "html.parser")

bowler_data = []
table = soup.find("table", class_="table")
rows = table.find_all("tr")

for row in rows[1:11]:
    cells = row.find_all("td")
    bowler = cells[1].text.strip()
    team = cells[2].text.strip()
    rating = cells[3].text.strip()
    bowler_data.append([bowler, team, rating])

df = pd.DataFrame(bowler_data, columns=["Bowler", "Team", "Rating"])
print(df)
```

In [ ]:

```
Q-5 Write a python program to scrape mentioned news details from https://www.cnbc.com/
make data frame
i) Headline
ii) Time
iii) News Link
```

In [ ]:

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

# Send a GET request to the website
url = "https://www.cnbc.com/world/?region=world"
response = requests.get(url)

# Create a BeautifulSoup object to parse the HTML content
soup = BeautifulSoup(response.content, "html.parser")

# Find all the news articles on the page
articles = soup.find_all("div", class_="Card-titleContainer")

# Initialize empty lists to store the scraped data
headlines = []
times = []
links = []

# Loop through each article and extract the required information
for article in articles:
    # Extract the headline
    headline = article.find("a").text.strip()
    headlines.append(headline)
```

```
# Extract the time
  time = article.find("time").text.strip()
  times.append(time)

  # Extract the news link
  link = article.find("a")["href"]
  links.append(link)

# Create a dataframe using the scraped data
data = {
  "Headline": headlines,
  "Time": times,
  "News Link": links
}
df = pd.DataFrame(data)

# Print the dataframe
print(df)
```

In [4]:

```
pip install requests beautifulsoup4 pandas
```

```
Requirement already satisfied: requests in c:\users\gaura\anaconda3\lib\site-packages
(2.31.0)
Requirement already satisfied: beautifulsoup4 in c:\users\gaura\anaconda3\lib\site-pa
ckages (4.12.2)
Requirement already satisfied: pandas in c:\users\gaura\anaconda3\lib\site-packages
(2.0.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\gaura\anaconda3\l
ib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\gaura\anaconda3\lib\site-pack
ages (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\gaura\anaconda3\lib\sit
e-packages (from requests) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\gaura\anaconda3\lib\sit
e-packages (from requests) (2023.7.22)
Requirement already satisfied: soupsieve>1.2 in c:\users\gaura\anaconda3\lib\site-pac
kages (from beautifulsoup4) (2.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\gaura\anaconda3\lib
\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\gaura\anaconda3\lib\site-pack
ages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\gaura\anaconda3\lib\site-pa
ckages (from pandas) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in c:\users\gaura\anaconda3\lib\site-pac
kages (from pandas) (1.24.3)
Requirement already satisfied: six>=1.5 in c:\users\gaura\anaconda3\lib\site-packages
(from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [ ]:

```
Q-6 Write a python program to scrape the details of most downloaded articles from AI i
days.https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-article
Scrape below mentioned details and make data frame
i) Paper Title
ii) Authors
iii) Published Date
iv) Paper URL
```

In [ ]:

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

# Send a GET request to the URL
url = "https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-artic
response = requests.get(url)

# Create a BeautifulSoup object to parse the HTML content
soup = BeautifulSoup(response.content, "html.parser")

# Find the container that holds the article details
articles_container = soup.find("div", class_="pod-listing")

# Initialize empty lists to store the scraped data
titles = []
authors = []
dates = []
urls = []

# Iterate over each article in the container
for article in articles_container.find_all("li"):
  # Scrape the title
  title = article.find("h3").text.strip()
  titles.append(title)

  # Scrape the authors
  author = article.find("span", class_="text-xs").text.strip()
  authors.append(author)

  # Scrape the published date
  date = article.find("span", class_="text-xs").find_next_sibling("span").text.strip()
  dates.append(date)

  # Scrape the paper URL
  url = article.find("a")["href"]
  urls.append(url)

# Create a dataframe with the scraped data
data = {
  "Paper Title": titles,
  "Authors": authors,
  "Published Date": dates,
  "Paper URL": urls
}
df = pd.DataFrame(data)

# Print the dataframe
print(df)
```

In [ ]:

```
Q-7 Write a python program to scrape mentioned details from dineout.co.inand make data
i) Restaurant name
ii) Cuisine
iii) Location
iv) Ratings
v) Image URL
```

In [ ]:

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

# Send a GET request to the website
url = "https://www.dineout.co.in"
response = requests.get(url)

# Create a BeautifulSoup object to parse the HTML content
soup = BeautifulSoup(response.content, 'html.parser')

# Find the elements containing the details you want to scrape
restaurant_names = soup.find_all('h2', class_='restnt-name ellipsis')
cuisines = soup.find_all('span', class_='double-line-ellipsis')
locations = soup.find_all('span', class_='double-line-ellipsis')
ratings = soup.find_all('span', class_='rating-value')
image_urls = soup.find_all('img', class_='img-responsive')

# Create empty lists to store the scraped data
restaurant_list = []
cuisine_list = []
location_list = []
rating_list = []
image_url_list = []

# Extract the data from the elements and append them to the respective lists
for name in restaurant_names:
  restaurant_list.append(name.text.strip())

for cuisine in cuisines:
  cuisine_list.append(cuisine.text.strip())

for location in locations:
  location_list.append(location.text.strip())

for rating in ratings:
  rating_list.append(rating.text.strip())

for image in image_urls:
  image_url_list.append(image['src'])

# Create a dictionary from the lists
data = {
  'Restaurant Name': restaurant_list,
  'Cuisine': cuisine_list,
  'Location': location_list,
  'Ratings': rating_list,
  'Image URL': image_url_list
}

# Create a dataframe from the dictionary
df = pd.DataFrame(data)

# Print the dataframe
print(df)
```