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

In [3]:
```
!pip install bs4
```
```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: bs4 in c:\users\cwc\appdata\roaming\python\python39\site-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in c:\programdata\anaconda3\lib\site-packages (from bs4) (4.11.1)
Requirement already satisfied: soupsieve>1.2 in c:\programdata\anaconda3\lib\site-packages (from beautifulsoup4->bs4) (2.3.1)
```

In [4]:
```
!pip install requests
```
```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: requests in c:\programdata\anaconda3\lib\site-packages (2.28.1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests) (1.26.11)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests) (2022.9.14)
```

In [5]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [6]:
```
page=requests.get("https://en.wikipedia.org/wiki/Main_Page")
page
```

Out[6]:
```
<Response [200]>
```

In [ ]:
```
soup=BeautifulSoup(page.content,"html.parser")
soup
```

In [10]:
```
header_tags=soup.find_all(["h1","h2","h3","h4","h5","h6"])
```

In [11]:
```
header_texts=[tag.get_text() for tag in header_tags]
```

In [12]:
```
df=pd.DataFrame({"Header":header_texts})
print(df)
```
```
                        Header
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 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 [13]:
```
!pip install bs4
```
```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: bs4 in c:\users\cwc\appdata\roaming\python\python39\site-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in c:\programdata\anaconda3\lib\site-packages (from bs4) (4.11.1)
Requirement already satisfied: soupsieve>1.2 in c:\programdata\anaconda3\lib\site-packages (from beautifulsoup4->bs4) (2.3.1)
```

In [14]:
```
!pip install requests
```
```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: requests in c:\programdata\anaconda3\lib\site-packages (2.28.1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests) (1.26.11)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests) (2022.9.14)
```

In [15]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [16]:
```
page=requests.get("https://presidentofindia.nic.in/former-presidents.htm")
page
```

Out[16]:
```
<Response [200]>
```

In [ ]:
```
soup=BeautifulSoup(page.content,"html.parser")
soup
```

In [18]:
```
table = soup.find("table", {"class": "tablepress tablepress-id-24"})
```

In [19]:
```
names = []
terms = []
```

In [ ]:
```
rows = table.find_all("tr")[1:]  # Exclude the table header row
for row in rows:
    cells = row.find_all("td")
    name = cells[0].get_text(strip=True)
    term = cells[1].get_text(strip=True)
    names.append(name)
    terms.append(term)
```

In [ ]:
```
df = pd.DataFrame({"Name": names, "Term of Office": terms})
print(df)
```

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.

a) Top 10 ODI teams in men's cricket along with the records for matches, points and rating.

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

In [ ]:
```
page=requests.get("https://www.icc-cricket.com/rankings/mens/player-rankings/test")
page
```

In [ ]:
```
soup=BeautifulSoup(page.content, "html.parser")
soup
```

In [ ]:
```
matches=[]
teams=[]
ratings=[]
```

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

def scrape_icc_rankings(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    table = soup.find('table', class_='table')

    data = []
    rows = table.find_all('tr')
    for row in rows[1:11]:  # Top 10 entries
        columns = row.find_all('td')
        record = {
            'Rank': columns[0].text.strip(),
            'Team': columns[1].text.strip(),
            'Matches': columns[2].text.strip(),
            'Points': columns[3].text.strip(),
            'Rating': columns[4].text.strip()
        }
        data.append(record)

    return data

# Scrape top 10 ODI teams
team_url = 'https://www.icc-cricket.com/rankings/mens/team-rankings/odi'
team_data = scrape_icc_rankings(team_url)
team_df = pd.DataFrame(team_data)

# Scrape top 10 ODI batsmen
batsman_url = 'https://www.icc-cricket.com/rankings/mens/player-rankings/odi/batting'
batsman_data = scrape_icc_rankings(batsman_url)
batsman_df = pd.DataFrame(batsman_data)

# Scrape top 10 ODI bowlers
bowler_url = 'https://www.icc-cricket.com/rankings/mens/player-rankings/odi/bowling'
bowler_data = scrape_icc_rankings(bowler_url)
bowler_df = pd.DataFrame(bowler_data)

# Display the dataframes
print("Top 10 ODI Teams:")
print(team_df)

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

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

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

def scrape_icc_rankings(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    table = soup.find('table', class_='table')

    data = []
    rows = table.find_all('tr')
    for row in rows[1:11]:  # Top 10 entries
        columns = row.find_all('td')
        record = {
            'Rank': columns[0].text.strip(),
            'Player/Team': columns[1].text.strip(),
            'Rating': columns[2].text.strip()
        }
        data.append(record)

    return data

# Scrape top 10 ODI teams in women's cricket
team_url = 'https://www.icc-cricket.com/rankings/womens/team-rankings/odi'
team_data = scrape_icc_rankings(team_url)
team_df = pd.DataFrame(team_data)

# Scrape top 10 women's ODI batting players
batting_url = 'https://www.icc-cricket.com/rankings/womens/player-rankings/odi/batting'
batting_data = scrape_icc_rankings(batting_url)
batting_df = pd.DataFrame(batting_data)

# Scrape top 10 women's ODI all-rounders
allrounder_url = 'https://www.icc-cricket.com/rankings/womens/player-rankings/odi/all-rounder'
allrounder_data = scrape_icc_rankings(allrounder_url)
allrounder_df = pd.DataFrame(allrounder_data)

# Display the dataframes
print("Top 10 ODI Teams in Women's Cricket:")
print(team_df)

print("\nTop 10 Women's ODI Batting Players:")
print(batting_df)

print("\nTop 10 Women's ODI All-rounders:")
print(allrounder_df)
```

5.5) 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
import pandas as pd
from bs4 import BeautifulSoup

def scrape_news_details(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    articles = soup.find_all('div', class_='Card-title')

    data = []
    for article in articles:
        headline = article.find('a').text.strip()
        time = article.find('time').text.strip()
        news_link = article.find('a')['href']
        record = {
            'Headline': headline,
            'Time': time,
            'News Link': news_link
        }
        data.append(record)

    return data

# Scrape news details from the provided URL
url = 'https://www.cnbc.com/world/?region=world'
news_data = scrape_news_details(url)
news_df = pd.DataFrame(news_data)

# Display the dataframe
print("News Details:")
print(news_df)
```

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 Scrape below mentioned details and make data framei) Paper Title ii) Authors iii) Published Date iv) Paper URL

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

url = "https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-articles"
response = requests.get(url)
soup = BeautifulSoup(response.content, "html.parser")

articles = []
for row in soup.find_all("div", {"class": "article-item"}):
    title = row.find("h3").text
    authors = row.find("p", {"class": "authors"}).text
    published_date = row.find("span", {"class": "published-date"}).text
    url = row.find("a")["href"]
    article = {
        "title": title,
        "authors": authors,
        "published_date": published_date,
        "url": url,
    }
    articles.append(article)

df = pd.DataFrame(articles)

print(df.to_string())
```

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

url = "https://www.dineout.co.in/delhi-restaurants"
response = requests.get(url)
soup = BeautifulSoup(response.content, "html.parser")

restaurants = []
for row in soup.find_all("div", {"class": "restnt-card"}):
    name = row.find("h3").text
    cuisine = row.find("p", {"class": "cuisine"}).text
    location = row.find("p", {"class": "location"}).text
    rating = row.find("span", {"class": "rating"}).text
    image_url = row.find("img")["src"]
    restaurant = {
        "name": name,
        "cuisine": cuisine,
        "location": location,
        "rating": rating,
        "image_url": image_url,
    }
    restaurants.append(restaurant)

df = pd.DataFrame(restaurants)

print(df.to_string())
```

In [ ]:

In [ ]:

In [ ]: