In [1]:
```
#1) Write a python program to display all the header tags from wikipedia.org

from bs4 import BeautifulSoup
import requests
import pandas as pd

url = 'https://en.wikipedia.org/wiki/Main_Page'

page = requests.get(url)

soup = BeautifulSoup (page.text, 'html.parser')
```

In [4]:
```
print (soup)
```

```
<script async="" src="/w/load.php?lang=en&amp;modules=startup&amp;only=s
cripts&amp;raw=1&amp;skin=vector-2022"></script>
<meta content="" name="ResourceLoaderDynamicStyles"/>
<link href="/w/load.php?lang=en&amp;modules=site.styles&amp;only=styles&
amp;skin=vector-2022" rel="stylesheet"/>
<meta content="MediaWiki 1.42.0-wmf.15" name="generator"/>
<meta content="origin" name="referrer"/>
<meta content="origin-when-cross-origin" name="referrer"/>
<meta content="max-image-preview:standard" name="robots"/>
<meta content="telephone=no" name="format-detection"/>
<meta content="https://upload.wikimedia.org/wikipedia/commons/5/58/Mars_
helicopter_on_sol_46.png" property="og:image"/>
<meta content="1200" property="og:image:width"/>
<meta content="1200" property="og:image:height"/>
<meta content="https://upload.wikimedia.org/wikipedia/commons/5/58/Mars_
helicopter_on_sol_46.png" property="og:image"/>
<meta content="800" property="og:image:width"/>
<meta content="800" property="og:image:height"/>
<meta content="640" property="og:image:width"/>
<meta content="640" property="og:image:height"/>
```

In [5]:
```
heading_tags = ["h1", "h2", "h3"]
for tags in soup.find_all(heading_tags):
    print(tags.name + ' -> ' + tags.text.strip())
```

```
h1 -> Main Page
h1 -> Welcome to Wikipedia
h2 -> From today's featured article
h2 -> Did you know ...
h2 -> In the news
h2 -> On this day
h2 -> Today's featured picture
h2 -> Other areas of Wikipedia
h2 -> Wikipedia's sister projects
h2 -> Wikipedia languages
```

In [6]:
```
df=pd.read_html(str())
df=pd.DataFrame(df[0])
df.head()
```

```
---------------------------------------------------------------------------
OSError                                   Traceback (most recent call la
st)
File ~\anaconda3\lib\site-packages\pandas\io\html.py:806, in _LxmlFrameP
arser._build_doc(self)
    804 else:
    805     # try to parse the input in the simplest way
--> 806     r = parse(self.io, parser=parser)
    807 try:

File ~\anaconda3\lib\site-packages\lxml\html\__init__.py:937, in parse(f
ilename_or_url, parser, base_url, **kw)
    936     parser = html_parser
--> 937 return etree.parse(filename_or_url, parser, base_url=base_url, *
*kw)

File src/lxml/etree.pyx:3538, in lxml.etree.parse()
```

In [22]:
```
#2) Write a python program to display list of respected former presidents of


from bs4 import BeautifulSoup
import requests
import pandas as pd

presidents = requests.get('https://presidentofindia.nic.in/former-presidents

presidents
```

Out[22]: <Response [200]>

In [23]:
```
in_presidents = BeautifulSoup(presidents.content)
in_presidents
```

Out[23]:
```
<!DOCTYPE html>
<html dir="ltr" lang="en">
<head>
<meta charset="utf-8"/>
<noscript><meta content="0; URL=/big_pipe/no-js?destination=/former-pres
idents" http-equiv="Refresh"/>
</noscript><meta content="Former Presidents of India - | President of In
dia" name="description"/>
<meta content="President of India | Former Presidents of India" name="ke
ywords"/>
<link href="http://presidentofindia.nic.in/former-presidents" rel="canon
ical"/>
<link href="/manifest.json" rel="manifest"/>
<meta content="" name="theme-color"/>
<meta content="Drupal 9 (https://www.drupal.org)" name="Generator"/>
<meta content="width" name="MobileOptimized"/>
<meta content="true" name="HandheldFriendly"/>
<meta content="width=device-width, initial-scale=1.0" name="viewport"/>
<link href="/sites/default/files/tiranga_1.png" rel="icon" type="image/p
```

In [24]:
```
name = []
for i in in_presidents.find_all('div', class_="desc-sec"):
    name.append(i.text.replace('\n',''))

name
```

Out[24]:
```
['Shri Ram Nath Kovind14th President of India',
 'Shri Pranab Mukherjee13th President of India',
 'Smt Pratibha Devisingh Patil12th President of India',
 'DR. A.P.J. Abdul Kalam11th President of India',
 'Shri K. R. Narayanan10th President of India',
 'Dr Shankar Dayal Sharma9th  President of India',
 'Shri R Venkataraman8th President of India',
 'Giani Zail Singh7th President of India',
 'Shri Neelam Sanjiva Reddy6th President of India',
 'Dr. Fakhruddin Ali Ahmed5th President of India',
 'Shri Varahagiri Venkata Giri4th President of India',
 'Dr. Zakir Husain3rd President of India',
 'Dr. Sarvepalli Radhakrishnan2nd President of India',
 'Dr. Rajendra Prasad1st President of India']
```

In [26]:
```
df_presidents = pd.DataFrame({'Name of President and Term of office': name }
df_presidents
```

Out[26]:

	Name of President and Term of office
0	Shri Ram Nath Kovind14th President of India
1	Shri Pranab Mukherjee13th President of India
2	Smt Pratibha Devisingh Patil12th President of ...
3	DR. A.P.J. Abdul Kalam11th President of India
4	Shri K. R. Narayanan10th President of India
5	Dr Shankar Dayal Sharma9th President of India
6	Shri R Venkataraman8th President of India
7	Giani Zail Singh7th President of India
8	Shri Neelam Sanjiva Reddy6th President of India
9	Dr. Fakhruddin Ali Ahmed5th President of India
10	Shri Varahagiri Venkata Giri4th President of I...
11	Dr. Zakir Husain3rd President of India
12	Dr. Sarvepalli Radhakrishnan2nd President of I...
13	Dr. Rajendra Prasad1st President of India

In [28]:
```
pat = '(\D+)'
all_names = df_presidents['Name of President and Term of office'].str.extrac
all_names
```

Out[28]:
```
0             Shri Ram Nath Kovind
1            Shri Pranab Mukherjee
2     Smt Pratibha Devisingh Patil
3           DR. A.P.J. Abdul Kalam
4             Shri K. R. Narayanan
5          Dr Shankar Dayal Sharma
6              Shri R Venkataraman
7                 Giani Zail Singh
8        Shri Neelam Sanjiva Reddy
9         Dr. Fakhruddin Ali Ahmed
10    Shri Varahagiri Venkata Giri
11                Dr. Zakir Husain
12    Dr. Sarvepalli Radhakrishnan
13             Dr. Rajendra Prasad
Name: Name of President and Term of office, dtype: object
```

In [29]:
```
dict = pd.DataFrame({'Name of President': all_names,
                     'Term of Office': ['25th July, 2017 - 25th July 2022','
dict
```

Out[29]:

	Name of President	Term of Office
0	Shri Ram Nath Kovind	25th July, 2017 - 25th July 2022
1	Shri Pranab Mukherjee	25th July 2012 - 25th July 2017
2	Smt Pratibha Devisingh Patil	25th July 2007 - 25th July 2012
3	DR. A.P.J. Abdul Kalam	July 2002 - 25th July 2007
4	Shri K. R. Narayanan	25 July, 1997 - 25 July 2002
5	Dr Shankar Dayal Sharma	25 July, 1992 - 25 July, 1997
6	Shri R Venkataraman	25 July, 1987 - 25 July, 1992
7	Giani Zail Singh	25 July, 1982 - 25 July, 1987
8	Shri Neelam Sanjiva Reddy	25 July, 1977 - 25 July, 1982
9	Dr. Fakhruddin Ali Ahmed	24 August, 1974 - 11 February, 1977
10	Shri Varahagiri Venkata Giri	24 August, 1969 - 24 August, 1974
11	Dr. Zakir Husain	13 May, 1967 - 03 May, 1969
12	Dr. Sarvepalli Radhakrishnan	13 May, 1962 - 13 May , 1967
13	Dr. Rajendra Prasad	26 January, 1950 - 13 May, 1962

In [3]:
```
#3) Write a python program to scrape cricket rankings from icc-cricket.com.
#a) Top 10 ODI teams in men's cricket along with the records for matches, po
#b) Top 10 ODI Batsmen along with the records of their team and rating.
#c) Top 10 ODI bowlers along with the records of their team and rating.

#(a)

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)

#b) To scrape the top 10 ODI batsmen along with the records of their team ar

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)

#c) To scrape the top 10 ODI bowlers along with the records of their team ar

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)

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)
```

```
---------------------------------------------------------------------------
-
AttributeError                            Traceback (most recent call las
t)
Cell In[3], line 18
     16 team_data = []
     17 table = soup.find("table", class_="table")
---> 18 rows = table.find_all("tr")
     20 for row in rows[1:11]:
     21   cells = row.find_all("td")

AttributeError: 'NoneType' object has no attribute 'find_all'
```

In [4]:
```
#4) Write a python program to scrape cricket rankings from icc-cricket.com.
#a) Top 10 ODI teams in women's cricket along with the records for matches,
#b) Top 10 women's ODI Batting players along with the records of their team
#c) Top 10 women's ODI all-rounder along with the records of their team and

#a) To scrape the top 10 ODI teams in men's cricket along with the records f

import requests
from bs4 import BeautifulSoup
import pandas as pd

url = "https://www.icc-cricket.com/rankings/womens/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)

df.head(10)

#b) To scrape the top 10 ODI batsmen along with the records of their team ar

url = "https://www.icc-cricket.com/rankings/womens/player-rankings/odi/batti
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)

#c) To scrape the top 10 ODI bowlers along with the records of their team ar

url = "https://www.icc-cricket.com/rankings/womens/player-rankings/odi/bowli
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)
```

```
---------------------------------------------------------------------------
-
AttributeError                            Traceback (most recent call las
t)
Cell In[4], line 18
     16 team_data = []
     17 table = soup.find("table", class_="table")
---> 18 rows = table.find_all("tr")
     20 for row in rows[1:11]:
     21   cells = row.find_all("td")

AttributeError: 'NoneType' object has no attribute 'find_all'
```

In [63]:
```
#Write a python program to scrape mentioned news details from https://www.cr
#i) Headline
#ii) Time
#iii) News Link

import requests
from bs4 import BeautifulSoup
import pandas as pd

url = "https://www.cnbc.com/world/?region=world"
response = requests.get(url)
soup = BeautifulSoup(response.content, "html.parser")

articles = soup.find_all("div", class_="Card-titleContainer")
headlines = []
times = []
links = []

for article in articles:
  headline = article.find("a").text.strip()
  headlines.append(headline)

  time = article.find("time").text.strip()
  times.append(time)

  link = article.find("a")["href"]
  links.append(link)

data = {
  "Headline": headlines,
  "Time": times,
  "News Link": links
}
df = pd.DataFrame(data)

print(df)
```

```
---------------------------------------------------------------------------
-
AttributeError                            Traceback (most recent call las
t)
Cell In[63], line 23
     20 headline = article.find("a").text.strip()
     21 headlines.append(headline)
---> 23 time = article.find("time").text.strip()
     24 times.append(time)
     26 link = article.find("a")["href"]

AttributeError: 'NoneType' object has no attribute 'text'
```

In [64]:
```
#6) Write a python program to scrape the details of most downloaded articles
#i) Paper Title ii) Authors iii) Published Date iv) Paper URL

import requests
from bs4 import BeautifulSoup
import pandas as pd

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

articles_container = soup.find("div", class_="pod-listing")
titles = []
authors = []
dates = []
urls = []

for article in articles_container.find_all("li"):
  title = article.find("h3").text.strip()
  titles.append(title)

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

  date = article.find("span", class_="text-xs").find_next_sibling("span").te
  dates.append(date)

  url = article.find("a")["href"]
  urls.append(url)

data = {"Paper Title": titles,
  "Authors": authors,
  "Published Date": dates,
  "Paper URL": urls
}
df = pd.DataFrame(data)

print(df)
```

```
---------------------------------------------------------------------------
-
AttributeError                            Traceback (most recent call las
t)
Cell In[64], line 18
     15 dates = []
     16 urls = []
---> 18 for article in articles_container.find("li"):
     19   title = article.find("h3").text.strip()
     20   titles.append(title)

AttributeError: 'NoneType' object has no attribute 'find'
```

In [60]:
```
#7) Write a python program to scrape mentioned details from dineout.co.in ar
#i) Restaurant name ii) Cuisine iii) Location iv) Ratings v) Image URL

import requests
from bs4 import BeautifulSoup
import pandas as pd

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

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')

restaurant_list = []
cuisine_list = []
location_list = []
rating_list = []
image_url_list = []

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'])

data = {'Restaurant Name': restaurant_list,
  'Cuisine': cuisine_list,
  'Location': location_list,
  'Ratings': rating_list,
  'Image URL': image_url_list
}

df = pd.DataFrame(data)
print(df)
```

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

In [ ]: