[Flip Robo Technologies]

[WEB SCRAPING]

[Assingments 1]

NAME = MANOJ

SINGODIYA

batch : DS2308

In [2]:

```
!pip install bs4
!pip install requests
```

```
Requirement already satisfied: bs4 in c:\users\raj\anaconda3\lib\site-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in c:\users\raj\anaconda3\lib\site-packages (from bs4) (4.11.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\raj\anaconda3\lib\site-packages (from beautifulsoup4->bs4) (2.3.1)
Requirement already satisfied: requests in c:\users\raj\anaconda3\lib\site-packages (2.28.1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\raj\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\raj\anaconda3\lib\site-packages (from requests) (1.26.11)
Requirement already satisfied: idna<4,>=2.5 in c:\users\raj\anaconda3\lib\site-packages (from requests) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\raj\anaconda3\lib\site-packages (from requests) (2022.9.14)
```

In [3]:

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

In [4]:

```
pip install beautifulsoup4 pandas requests
```

```
Requirement already satisfied: beautifulsoup4 in c:\users\raj\anaconda3\lib\site-packages (4.11.1)
Requirement already satisfied: pandas in c:\users\raj\anaconda3\lib\site-packages (1.4.4)
Requirement already satisfied: requests in c:\users\raj\anaconda3\lib\site-packages (2.28.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\raj\anaconda3\lib\site-packages (from beautifulsoup4) (2.3.1)
Requirement already satisfied: pytz>=2020.1 in c:\users\raj\anaconda3\lib\site-packages (from pandas) (2022.1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\raj\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: numpy>=1.18.5 in c:\users\raj\anaconda3\lib\site-packages (from pandas) (1.24.4)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\raj\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\raj\anaconda3\lib\site-packages (from requests) (2022.9.14)
Requirement already satisfied: idna<4,>=2.5 in c:\users\raj\anaconda3\lib\site-packages (from requests) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\raj\anaconda3\lib\site-packages (from requests) (1.26.11)
Requirement already satisfied: six>=1.5 in c:\users\raj\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0) Note: you
may need to restart the kernel to use updated packages.
```

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

In [5]:

```
wiki = requests.get('https://en.wikipedia.org/wiki/Main_Page') soup =
BeautifulSoup(wiki.text, 'html.parser')

head_tag=soup.find_all(['h1','h2','h3'])
Header =[] for i in head_tag:
  Header.append(i.get_text().replace("\n",""))

### now creating the Dataframe

Wikipedia_Header=pd.DataFrame({})
Wikipedia_Header['Wikipedia Main Page Header Tag']=Header print('\033[1m'+'Wikipedia
Main Page Header Tag'+'\033[0m')
Wikipedia_Header
```

Wikipedia Main Page Header Tag

Out[5]:

	Wikipedia Main Page Header Tag
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 [6]:

```
page=requests.get('https://presidentofindia.nic.in/former-presidents')
soup=BeautifulSoup(page.content) tu_title=soup.find('div', class_="desc-
sec") headers=[] for i in soup.find_all('div', class_="desc-sec"):
headers.append(i.text.split('\n'))

## now creating the DataFrame

df=pd.DataFrame({'Headers':headers}) df
```

Out[6]:

	Headers
0	[, Shri Ram Nath Kovind, 14th President of Ind...
1	[, Shri Pranab Mukherjee, 13th President of In...
2	[, Smt Pratibha Devisingh Patil, 12th Presiden...
3	[, DR. A.P.J. Abdul Kalam, 11th President of I...
4	[, Shri K. R. Narayanan, 10th President of Ind...
5	[, Dr Shankar Dayal Sharma, 9th President of ...
6	[, Shri R Venkataraman, 8th President of India,]
7	[, Giani Zail Singh, 7th President of India,]
8	[, Shri Neelam Sanjiva Reddy, 6th President of...
9	[, Dr. Fakhruddin Ali Ahmed, 5th President of ...
10	[, Shri Varahagiri Venkata Giri, 4th President...
11	[, Dr. Zakir Husain, 3rd President of India,]
12	[, Dr. Sarvepalli Radhakrishnan, 2nd President...
13	[, Dr. Rajendra Prasad, 1st President of India,]

3) Write a python program to scrape cricket rankings from icccricket.com. You have to scrape and make data frame

a) 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 and rating.

c) Top 10 ODI bowlers along with the records of their team and rating.

In [7]:

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

```
url = 'https://www.icc-cricket.com/rankings/mens/team-rankings/odi'

response = requests.get(url)
print(response.status_code, '--->',url) print('\n')
soup= BeautifulSoup(response.content, 'lxml')


Team=[]
Matches=[]
Points=[]
Rating=[]
Country = soup.find_all('span',class_="u-hide-phablet") for i
in Country:
   Team.append(i.get_text().replace("\n","")) Team=Team[0:10]

match=soup.find_all('td',class_='rankings-block__banner--matches')
matchs=soup.find_all('td',class_='table-body__cell u-center-text') mtc =
match + matchs

for i in mtc:
   Matches.append(i.text)
   Matches=Matches[0:10]

pt=soup.find_all('td',class_="rankings-block__banner--points")

pts= soup.find_all('td',class_ ="table-body__cell u-center-text")
Point= pt + pts for i in Point:
   Points.append(i.get_text().replace("\n","")) Points=Points[0:10] rating
= soup.find_all('td',class_="table-body__cell u-text-right rating") for i in
rating:
   Rating.append(i.get_text().replace("\n",""))
   Rating=Rating[0:10]

### now we are creating the DataFrame


ODI=pd.DataFrame({})
ODI['Country']=Team
ODI['Matches']=Matches
ODI['Rating']=Rating ODI['Points']=Points print('\033[1m'+'ICC MENS ODI
RANKING'+'\033[0m') # Print Title in bold case
ODI
```

```
200 ---> https://www.icc-cricket.com/rankings/mens/team-rankings/odi
```

ICC MENS ODI RANKING

Out[7]:

	Country	Matches	Rating	Points
0	Pakistan	27	115	3,102
1	India	41	113	41
2	Australia	4,701	106	4,701
3	South Africa	28	105	28
4	England	3,166	100	3,166
5	New Zealand	24	94	24
6	Bangladesh	2,551	92	2,551
7	Sri Lanka	28	80	28
8	Afghanistan	2,942	68	2,942
9	West Indies	31	55	31

In [8]:

```
## 3.(b) Top 10 ODI Batsmen along with the records of their team and rating.

url = 'https://www.icc-cricket.com/rankings/mens/player-rankings/odi/batting' headers =
{
   'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36'
}

response = requests.get(url)
print(response.status_code, '--->',url)
print('\n\n') soup=
BeautifulSoup(response.content, 'lxml')
Position =[]
```

```
Player =[]
Country =[]
Rating =[]
# Extracting Data of Top Player from Banner block_1= soup.find('tr', attrs={'class':'rankings-
block__banner'}) # contains Top Player ranking detail

Position.append(block_1.find('td',class_='rankings-block__position').text)# Ranking Position
Player.append(block_1.find('div', class_="rankings-block__banner--name-large").text) # Extract Player Name
Country.append(block_1.find('span', class_='rankings-block__banner--nation').text)# Extract Country Name
Rating.append(block_1.find('div', class_="rankings-block__banner--rating").text) # Extract Rating

# Extracting other Player Ranking table_rows
=soup.find_all('tr', attrs={'class':'table-body'})

for row in table_rows[:10]:
    Position.append(row.find('td', class_='table-body__cell table-body__cell--position u-text-right').text.replace('\n',''))    Player.append(row.find('a').text)
    Country.append(row.find('span', class_='table-body__logo-text').text)
    Rating.append(row.find('td', class_='table-body__cell rating').text)

# Storing data in Dataframe
ODI_Batmans=pd.DataFrame({'Ranking':Position,'Player_Name':Player, 'Team':Country, 'Rating':Rating})

print('\033[1m'+'ICC ODI MENS BATTING RANKING'+'\033[0m') # Print Title in bold case
ODI_Batmans
```

```
200 ---> https://www.icc-cricket.com/rankings/mens/player-rankings/odi/batting
```

ICC ODI MENS BATTING RANKING

Out[8]:

	Ranking	Player_Name	Team	Rating
0	\n\n\n 1\n ...	Babar Azam		863
1	2 ...	Shubman Gill	IND	759
2	3 ...	Rassie van der Dussen	SA	745
3	4 ...	David Warner	AUS	739
4	5 ...	Imam-ul-Haq	PAK	735
5	6 ...	Harry Tector	IRE	726
6	7 ...	Quinton de Kock	SA	721
7	8 ...	Virat Kohli	IND	715
8	9 ...	Rohit Sharma	IND	707
9	10 ...	Fakhar Zaman	PAK	705
10	11 ...	Temba Bavuma	SA	691

In [9]:

```
## 3.(c) Top 10 ODI bowlers along with the records of their team and rating.

url = 'https://www.icc-cricket.com/rankings/mens/player-rankings/odi/bowling' headers =
{
    'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36'
}

response = requests.get(url)
print(response.status_code, '--->',url)
print('\n\n') soup=
BeautifulSoup(response.content, 'lxml')
Position =[]
Player =[]
Country =[]
Rating =[]

# Extracting Data of Top Player from Banner block_1= soup.find('tr', attrs={'class':'rankings-
block__banner'}) # contains Top Player ranking detail

Position.append(block_1.find('td',class_='rankings-block__position').text)# Ranking Position
Player.append(block_1.find('div', class_="rankings-block__banner--name-large").text) # Extract Player Name
Country.append(block_1.find('span', class_='rankings-block__banner--nation').text)# Extract Country Name
Rating.append(block_1.find('div', class_="rankings-block__banner--rating").text) # Extract Rating

# Extracting other Player Ranking table_rows
=soup.find_all('tr', attrs={'class':'table-body'})

for row in table_rows[:10]:
```

```
    Position.append(row.find('td', class_='table-body__cell table-body__cell--position u-text-right').text.replace('\n',''))    Player.append(row.find('a').text)
    Country.append(row.find('span', class_='table-body__logo-text').text)
    Rating.append(row.find('td', class_='table-body__cell rating').text)

# Storing data in Dataframe
ODI_Bowling=pd.DataFrame({'Ranking':Position,'Player_Name':Player, 'Team':Country, 'Rating':Rating})

print('\033[1m'+'ICC ODI MENS BOWLING RANKING'+'\033[0m') # Print Title in bold case
ODI_Bowling
```

200 ---> https://www.icc-cricket.com/rankings/mens/player-rankings/odi/bowling

ICC ODI MENS BOWLING RANKING

Out[9]:

	Ranking	Player_Name	Team	Rating
0	\n\n\n 1\n ...	Josh Hazlewood		692
1	2 ...	Mitchell Starc	AUS	666
2	= ...	Trent Boult	NZ	666
3	4 ...	Adam Zampa	AUS	663
4	5 ...	Matt Henry	NZ	658
5	6 ...	Mujeeb Ur Rahman	AFG	657
6	7 ...	Kuldeep Yadav	IND	656
7	8 ...	Rashid Khan	AFG	655
8	9 ...	Mohammed Siraj	IND	643
9	10 ...	Shaheen Afridi	PAK	635
10	11 ...	Mohammad Nabi	AFG	621

4) Write a python program to scrape cricket rankings from icccricket.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 [10]:

```
# 4.(a) Top 10 ODI teams in women's cricket along with the records for matches, points and rating.


url = 'https://www.icc-cricket.com/rankings/womens/team-rankings/odi'

response = requests.get(url)
print(response.status_code, '--->',url)
print('\n')
SOUP= BeautifulSoup(response.content, 'lxml')

# Creating empty list
Team=[]
Matches=[]
Points=[]
Rating=[]

# Extracting Team Name
Country = SOUP.find_all('span',class_="u-hide-phablet") for i in
Country:
  Team.append(i.get_text().replace("\n",""))
  Team=Team[0:10]

# Extracting No of Matches  match=SOUP.find_all('td',class_='rankings-
block__banner--matches') matchs=SOUP.find_all('td',class_='table-
body__cell u-center-text') mtc = match + matchs
for i in  mtc:
  Matches.append(i.text)
  Matches=Matches[0:10]

# Extracting Points gain  pt=SOUP.find_all('td',class_="rankings-
block__banner--points") pts= SOUP.find_all('td',class_ ="table-
body__cell u-center-text") Point= pt + pts for i in Point:
  Points.append(i.get_text().replace("\n",""))
```

```
    Points=Points[0:10]

# Extracting Rating rat=SOUP.find_all('td',class_="rankings-block__banner--rating")
rating = SOUP.find_all('td',class_="table-body__cell u-text-right rating") RATING=rat +
rating for i in RATING:
    Rating.append(i.get_text().replace("\n",""))
    Rating=Rating[0:10]
Rating

# Creating dataframe to store data
ODI=pd.DataFrame({})
ODI['Country']=Team
ODI['Matches']=Matches
ODI['Rating']=Rating ODI['Points']=Points

print('\033[1m'+'ICC ODI WOMENS RANKING'+'\033[0m') # Print Title in bold case
ODI
```

200 ---> https://www.icc-cricket.com/rankings/womens/team-rankings/odi

ICC ODI WOMENS RANKING

Out[10]:

	Country	Matches	Rating	Points
0	Australia	26	165 ...	4,290
1	England	31	125	31
2	South Africa	3,875	119	3,875
3	India	26	101	26
4	New Zealand	3,098	96	3,098
5	West Indies	30	95	30
6	Bangladesh	3,039	76	3,039
7	Sri Lanka	28	68	28
8	Thailand	2,688	68	2,688
9	Pakistan	29	62	29

In [11]:

```
# 4.(b) Top 10 women's ODI Batting players along with the records of their team and rating.

url = 'https://www.icc-cricket.com/rankings/womens/player-rankings/odi/batting' headers =
{
   'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36'
}

response = requests.get(url)
print(response.status_code, '--->',url)
print('\n\n') soup=
BeautifulSoup(response.text, 'lxml')
Position =[] Player
=[]
Country =[]
Rating =[]

# Extracting Data of Top Player from Banner block_1= soup.find('tr', attrs={'class':'rankings-
block__banner'}) # contains Top Player ranking detail

Position.append(block_1.find('td',class_='rankings-block__position').text)# Ranking Position
Player.append(block_1.find('div', class_="rankings-block__banner--name-large").text) # Extract Player Name
Country.append(block_1.find('span', class_='rankings-block__banner--nation').text)# Extract Country Name
Rating.append(block_1.find('div', class_="rankings-block__banner--rating").text) # Extract Rating

# Extracting other Player Ranking table_rows
=soup.find_all('tr', attrs={'class':'table-body'})

for row in table_rows[:10]:
    Position.append(row.find('td', class_='table-body__cell table-body__cell--position u-text-right').text.replace('\n',''))    Player.append(row.find('a').text)
    Country.append(row.find('span', class_='table-body__logo-text').text)
    Rating.append(row.find('td', class_='table-body__cell rating').text)

# Storing data in Dataframe
ODI_Batmans=pd.DataFrame({'Ranking':Position,'Player_Name':Player, 'Team':Country, 'Rating':Rating})

print('\033[1m'+'ICC ODI WOMENS BATTING RANKING'+'\033[0m') # Print Title in bold case
ODI_Batmans
```

200 ---> https://www.icc-cricket.com/rankings/womens/player-rankings/odi/batting

ICC ODI WOMENS BATTING RANKING

Out[11]:

	Ranking	Player_Name	Team	Rating
0	\n\n\n 1\n ...	Natalie Sciver-Brunt		801
1	2 ...	Beth Mooney	AUS	751
2	3 ...	Chamari Athapaththu	SL	743
3	4 ...	Laura Wolvaardt	SA	708
4	= ...	Smriti Mandhana	IND	708
5	6 ...	Alyssa Healy	AUS	702
6	7 ...	Harmanpreet Kaur	IND	694
7	8 ...	Ellyse Perry	AUS	686
8	9 ...	Meg Lanning	AUS	682
9	10 ...	Stafanie Taylor	WI	618
10	11 ...	Marizanne Kapp	SA	617

In [12]:

```
# 4.(c) Top 10 women's ODI all-rounder along with the records of their team and rating.

import re url = 'https://www.icc-cricket.com/rankings/womens/player-
rankings/odi/bowling' headers = {
  'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36'
}

response = requests.get(url)
print(response.status_code, '--->',url) print('\n\n')
soup= BeautifulSoup(response.text, 'html.parser')
Position =[]
Player =[]
Country =[]
Rating =[]

# Extracting Data of Top Player from Banner block_1= soup.find('tr', attrs={'class':'rankings-
block__banner'}) # contains Top Player ranking detail

Position.append(block_1.find('td',class_='rankings-block__position').text)# Ranking Position
Player.append(block_1.find('div', class_="rankings-block__banner--name-large").text) # Extract Player Name
Country.append(block_1.find('span', class_='rankings-block__banner--nation').text)# Extract Country Name
Rating.append(block_1.find('div', class_="rankings-block__banner--rating").text) # Extract Rating

# Extracting other Player Ranking table_rows
=soup.find_all('tr', attrs={'class':'table-body'})

for row in table_rows[:10]:
  Position.append(row.find('td', class_='table-body__cell table-body__cell--position u-text-right').text.replace('\n','')) Player.append(row.find('a').text)
  Country.append(row.find('span', class_='table-body__logo-text').text)
  Rating.append(row.find('td', class_='table-body__cell rating').text)

# Storing data in Dataframe
ODI_Bowling=pd.DataFrame({'Ranking':Position,'Player_Name':Player, 'Team':Country, 'Rating':Rating})

print('\033[1m'+'ICC ODI WOMENS BOWLING RANKING'+'\033[0m') # Print Title in bold case
ODI_Bowling
```

200 ---> https://www.icc-cricket.com/rankings/womens/player-rankings/odi/bowling

ICC ODI WOMENS BOWLING RANKING

Out[12]:

	Ranking	Player_Name	Team	Rating
0	\n\n\n 1\n ...	Sophie Ecclestone		761
1	2 ...	Shabnim Ismail	SA	708
2	3 ...	Jess Jonassen	AUS	682
3	4 ...	Ashleigh Gardner	AUS	673
4	5 ...	Megan Schutt	AUS	666
5	6 ...	Hayley Matthews	WI	662

6	7 ...	Kate Cross	ENG	660
7	8 ...	Ayabonga Khaka	SA	646
8	9 ...	Deepti Sharma	IND	607
9	10 ...	Rajeshwari Gayakwad	IND	599
10	11 ...	Marizanne Kapp	SA	587

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

make data frame

i) Headline

ii) Time

iii) News Link

In [13]:

```
# URL of the CNBC World news page 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.text, "html.parser")

  # Initialize empty lists to store data
headlines = []    times = []    news_links
= []

  # Find news articles on the page        article_elements =
soup.find_all("div",  class_="Card-title")          for  article  in
article_elements:
    # Headline            headline =
article.text.strip()
headlines.append(headline)

  time_elements = soup.find_all("div", class_="Card-timestamp")    for
time_element in time_elements:
    # Time       time =
time_element.text.strip()
times.append(time)

  link_elements = soup.find_all("div", class_="Card-title")    for
link_element in link_elements:
    # News Link       news_link =
link_element["href"]
news_links.append(news_link)

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

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

```
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 Scrape below mentioned details and make data frame i) Paper

Title

ii) Authors

iii) Published Date

iv) Paper URL

In [14]:

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

# 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.text, "html.parser")

  # Initialize empty lists to store data
paper_titles = []        authors_list = []
published_dates = []    paper_urls = []

  # Find articles on the page    article_elements = soup.find_all("li",
class_="js-article-list-item")

  for article in article_elements:
    # Paper Title       title = article.find("a", class_="article-
title").text.strip()       paper_titles.append(title)

    # Authors       authors = article.find("div", class_="js-
authors").text.strip()       authors_list.append(authors)

    # Published Date       date = article.find("div", class_="js-pub-
date").text.strip()       published_dates.append(date)

    # Paper URL       paper_url = article.find("a",
class_="article-title")["href"]       paper_urls.append(paper_url)

  # Create a DataFrame    df
= pd.DataFrame({
    "Paper Title": paper_titles,
    "Authors": authors_list,
    "Published Date": published_dates,
    "Paper URL": paper_urls
  })

  # Print the DataFrame
  print(df)

else:    print("Failed to retrieve the web
page.")
```

```
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 frame

i) Restaurant name

ii) Cuisine

iii) Location

iv) Ratings

v) Image URL

In [ ]:

```
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)

# Check if the request was successful if
response.status_code == 200:    soup =
BeautifulSoup(response.text, "html.parser")
```

```
    # Initialize empty lists to store data
restaurant_names = []  cuisines = []
locations = []  ratings = []
image_urls = []

    # Find restaurant details on the page  restaurant_elements =
soup.find_all("div", class_="restnt-info")  for restaurant in
restaurant_elements:
    # Restaurant Name  name = restaurant.find("div", class_="restnt-info-
dtls").h3.text.strip()  restaurant_names.append(name)

    # Cuisine  cuisine = restaurant.find("div", class_="restnt-info-
dtls").p.text.strip()  cuisines.append(cuisine)

    # Location  location = restaurant.find("div", class_="restnt-info-
loc").p.text.strip()  locations.append(location)

    # Ratings  rating = restaurant.find("div", class_="rating-
sec").span.text.strip()  ratings.append(rating)

    # Image URL  image_url =
restaurant.find("img")["src"]
image_urls.append(image_url)

    # Create a DataFrame
df = pd.DataFrame({
    "Restaurant Name": restaurant_names,
    "Cuisine": cuisines,
    "Location": locations,
    "Ratings": ratings,
    "Image URL": image_urls
  })

    # Print the DataFrame
    print(df)

else:  print("Failed to retrieve the web
page.")
----------THANK YOU----------

In [ ]:
```