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

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')

1. Write a 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

import requests

from bs4 import BeautifulSoup

import pandas as pd

url = "https://presidentofindia.nic.in/former-presidents.htm"

response = requests.get(url)

soup = BeautifulSoup(response.content, "html.parser")

table = soup.find("table")

names = []

terms = []

for row in table.find\_all("tr")[1:]:

columns = row.find\_all("td")

name = columns[0].text.strip()

term = columns[1].text.strip()

names.append(name)

terms.append(term)

data = {"Name": names, "Term of Office": terms}

df = pd.DataFrame(data)

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.

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.

import requests

from bs4 import BeautifulSoup

import pandas as pd

url\_teams = "https://www.icc-cricket.com/rankings/womens/team-rankings/odi"

response\_teams = requests.get(url\_teams)

soup\_teams = BeautifulSoup(response\_teams.content, "html.parser")

teams\_data = []

table\_teams = soup\_teams.find("table", class\_="table")

rows\_teams = table\_teams.find\_all("tr")

for row in rows\_teams[1:11]:

team\_name = row.find("span", class\_="u-hide-phablet").text.strip()

matches = row.find\_all("td")[2].text.strip()

points = row.find\_all("td")[3].text.strip()

rating = row.find\_all("td")[4].text.strip()

teams\_data.append([team\_name, matches, points, rating])

url\_batting = "https://www.icc-cricket.com/rankings/womens/player-rankings/odi/batting"

response\_batting = requests.get(url\_batting)

soup\_batting = BeautifulSoup(response\_batting.content, "html.parser")

batting\_data = []

table\_batting = soup\_batting.find("table", class\_="table")

rows\_batting = table\_batting.find\_all("tr")

for row in rows\_batting[1:11]:

player\_name = row.find("td", class\_="table-body\_\_cell rankings-table\_\_name name").text.strip()

team = row.find("span", class\_="table-body\_\_logo-text").text.strip()

rating = row.find("td", class\_="table-body\_\_cell rating").text.strip()

batting\_data.append([player\_name, team, rating])

url\_allrounders = "https://www.icc-cricket.com/rankings/womens/player-rankings/odi/all-rounder"

response\_allrounders = requests.get(url\_allrounders)

soup\_allrounders = BeautifulSoup(response\_allrounders.content, "html.parser")

allrounders\_data = []

table\_allrounders = soup\_allrounders.find("table", class\_="table")

rows\_allrounders = table\_allrounders.find\_all("tr")

for row in rows\_allrounders[1:11]:

player\_name = row.find("td", class\_="table-body\_\_cell rankings-table\_\_name name").text.strip()

team = row.find("span", class\_="table-body\_\_logo-text").text.strip()

rating = row.find("td", class\_="table-body\_\_cell rating").text.strip()

allrounders\_data.append([player\_name, team, rating])

df\_teams = pd.DataFrame(teams\_data, columns=["Team", "Matches", "Points", "Rating"])

df\_batting = pd.DataFrame(batting\_data, columns=["Player", "Team", "Rating"])

df\_allrounders = pd.DataFrame(allrounders\_data, columns=["Player", "Team", "Rating"])

print("Top 10 ODI teams in women's cricket:")

print(df\_teams)

print("\nTop 10 women's ODI Batting players:")

print(df\_batting)

print("\nTop 10 women's ODI all-rounders:")

print(df\_allrounders)

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

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:

# Extract the headline

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)

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

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\_container = soup.find("div", class\_="pod-listing")

titles = []

authors = []

dates = []

urls = []

for article in articles\_container.find\_all("li"):

# Scrape the title

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").text.strip()

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)

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

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)