

**Name = Divyanshu Singh**

**Batch = DS2306**

**ID = 56**

.

```
In [1]: import requests  
from bs4 import BeautifulSoup
```

.

1

```
In [2]: ur= requests.get("https://en.wikipedia.org/wiki/Main_Page")  
ur
```

```
Out[2]: <Response [200]>
```

```
In [3]: soup = BeautifulSoup(ur.content)
```

```
In [4]: header= soup.find("span",class_="mw-headline")  
print(header.text)
```

Welcome to Wikipedia

```
In [5]: header =[]  
for i in soup.find_all("span",class_="mw-headline"):  
    header.append(i.text)  
header
```

```
Out[5]: ['Welcome to Wikipedia',  
        "From today's featured article",  
        'Did you know\xa0...',  
        'In the news',  
        'On this day',  
        "Today's featured picture",  
        'Other areas of Wikipedia',  
        "Wikipedia's sister projects",  
        'Wikipedia languages']
```

```
In [6]: import pandas as pd  
df = pd.DataFrame({'Header':header})  
df
```

Out[6]:

**Header**

- 0** Welcome to Wikipedia
- 1** From today's featured article
- 2** Did you know ...
- 3** In the news
- 4** On this day
- 5** Today's featured picture
- 6** Other areas of Wikipedia
- 7** Wikipedia's sister projects
- 8** Wikipedia languages

.

.

**2**

```
In [7]: lin= requests.get("https://presidentofindia.nic.in/former-presidents.htm")
lin
```

```
Out[7]: <Response [200]>
```

```
In [8]: soup = BeautifulSoup(lin.content)
```

```
In [9]: president_name= soup.find("div",class_="presidentListing")
print(president_name.text)
```

Shri Ram Nath Kovind (birth - 1945)  
Term of Office: 25 July, 2017 to 25 July, 2022  
<https://ramnathkovind.nic.in>

```
In [10]: term = soup.find("span",class_="terms")
term.text
```

```
Out[10]: 'Term of Office:'
```

```
In [11]: term = []
for i in soup.find_all("div",class_="presidentListing"):
    term.append(i.text)

term
```

```
Out[11]: ['\nShri Ram Nath Kovind (birth - 1945)\nTerm of Office: 25 July, 2017 to 25 July, 2022 \nhttps://ramnathkovind.nic.in\n',
  '\nShri Pranab Mukherjee (1935-2020)\nTerm of Office: 25 July, 2012 to 25 July, 2017\nhttp://pranabmukherjee.nic.in\n',
  '\nSmt Pratibha Devi Singh Patil (birth - 1934)\nTerm of Office: 25 July, 2007 to 25 July, 2012 \nhttp://pratibhapatil.nic.in\n',
  '\nDR. A.P.J. Abdul Kalam (1931-2015)\nTerm of Office: 25 July, 2002 to 25 July, 2007 \nhttp://abdulkalam.nic.in\n',
  '\nShri K. R. Narayanan (1920 - 2005)\nTerm of Office: 25 July, 1997 to 25 July, 2002 \n',
  '\nDr Shankar Dayal Sharma (1918-1999)\nTerm of Office: 25 July, 1992 to 25 July, 1997 \n',
  '\nShri R Venkataraman (1910-2009)\nTerm of Office: 25 July, 1987 to 25 July, 1992 \n',
  '\nGiani Zail Singh (1916-1994)\nTerm of Office: 25 July, 1982 to 25 July, 1987 \n',
  '\nShri Neelam Sanjiva Reddy (1913-1996)\nTerm of Office: 25 July, 1977 to 25 July, 1982 \n',
  '\nDr. Fakhruddin Ali Ahmed (1905-1977)\nTerm of Office: 24 August, 1974 to 11 February, 1977\n',
  '\nShri Varahagiri Venkata Giri (1894-1980)\nTerm of Office: 3 May, 1969 to 20 July, 1969 and 24 August, 1969 to 24 August, 1974\n',
  '\nDr. Zakir Husain (1897-1969)\nTerm of Office: 13 May, 1967 to 3 May, 1969\n',
  '\nDr. Sarvepalli Radhakrishnan (1888-1975)\nTerm of Office: 13 May, 1962 to 13 May, 1967\n',
  '\nDr. Rajendra Prasad (1884-1963) \nTerm of Office: 26 January, 1950 to 13 May, 1962\n']
```

```
In [12]: president_name = []
for i in soup.find_all("div", class_="presidentListing"):
    president_name.append(i.text)

president_name
```

```
Out[12]: ['\nShri Ram Nath Kovind (birth - 1945)\nTerm of Office: 25 July, 2017 to 25 July, 2022 \nhttps://ramnathkovind.nic.in\n',
  '\nShri Pranab Mukherjee (1935-2020)\nTerm of Office: 25 July, 2012 to 25 July, 2017\nhttp://pranabmukherjee.nic.in\n',
  '\nSmt Pratibha Devi Singh Patil (birth - 1934)\nTerm of Office: 25 July, 2007 to 25 July, 2012 \nhttp://pratibhapatil.nic.in\n',
  '\nDR. A.P.J. Abdul Kalam (1931-2015)\nTerm of Office: 25 July, 2002 to 25 July, 2007 \nhttp://abdulkalam.nic.in\n',
  '\nShri K. R. Narayanan (1920 - 2005)\nTerm of Office: 25 July, 1997 to 25 July, 2002 \n',
  '\nDr Shankar Dayal Sharma (1918-1999)\nTerm of Office: 25 July, 1992 to 25 July, 1997 \n',
  '\nShri R Venkataraman (1910-2009)\nTerm of Office: 25 July, 1987 to 25 July, 1992 \n',
  '\nGiani Zail Singh (1916-1994)\nTerm of Office: 25 July, 1982 to 25 July, 1987 \n',
  '\nShri Neelam Sanjiva Reddy (1913-1996)\nTerm of Office: 25 July, 1977 to 25 July, 1982 \n',
  '\nDr. Fakhruddin Ali Ahmed (1905-1977)\nTerm of Office: 24 August, 1974 to 11 February, 1977\n',
  '\nShri Varahagiri Venkata Giri (1894-1980)\nTerm of Office: 3 May, 1969 to 20 July, 1969 and 24 August, 1969 to 24 August, 1974\n',
  '\nDr. Zakir Husain (1897-1969)\nTerm of Office: 13 May, 1967 to 3 May, 1969\n',
  '\nDr. Sarvepalli Radhakrishnan (1888-1975)\nTerm of Office: 13 May, 1962 to 13 May, 1967\n',
  '\nDr. Rajendra Prasad (1884-1963) \nTerm of Office: 26 January, 1950 to 13 May, 1962\n']
```

```
In [13]: import pandas as pd
df = pd.DataFrame({'President and term':president_name})
df
```

Out[13]: **President and term**

0	\nShri Ram Nath Kovind (birth - 1945)\nTerm of...
1	\nShri Pranab Mukherjee (1935-2020)\nTerm of O...
2	\nSmt Pratibha Devi Singh Patil (birth - 1934)\n...
3	\nDR. A.P.J. Abdul Kalam (1931-2015)\nTerm of ...
4	\nShri K. R. Narayanan (1920 - 2005)\nTerm of ...
5	\nDr Shankar Dayal Sharma (1918-1999)\nTerm of...
6	\nShri R Venkataraman (1910-2009)\nTerm of Off...
7	\nGiani Zail Singh (1916-1994)\nTerm of Office...
8	\nShri Neelam Sanjiva Reddy (1913-1996)\nTerm ...
9	\nDr. Fakhruddin Ali Ahmed (1905-1977)\nTerm o...
10	\nShri Varahagiri Venkata Giri (1894-1980)\nTe...
11	\nDr. Zakir Husain (1897-1969)\nTerm of Office...
12	\nDr. Sarvepalli Radhakrishnan (1888-1975)\nTe...
13	\nDr. Rajendra Prasad (1884-1963) \nTerm of Of...

### 3. Cricket rankings

#### A.

```
In [14]: a= requests.get("https://www.icc-cricket.com/rankings/mens/team-rankings/odi")
a
```

Out[14]: <Response [200]>

```
In [15]: soup = BeautifulSoup(a.content)
```

```
In [16]: odi_teams= soup.find("span",class_="u-hide-phablet")
print(odi_teams.text)
```

Australia

```
In [17]: odi_teams=[]
for i in soup.find_all("span",class_="u-hide-phablet"):
    odi_teams.append(i.text)
```

```
matches= soup.find("td",class_="rankings-block--banner--matches""") print(matches.text)matches=[] for i in soup.find_all("td",class_="table-body__cell u-center-text"): matches.append(i.text.split(' ')) Matches= matches[:10]
```

```
In [18]: import pandas as pd
df = pd.DataFrame({'Country':odi_teams})
```

.

.

.

.

```
In [19]: b= requests.get("https://www.icc-cricket.com/rankings/mens/player-rankings/odi")
b
```

```
Out[19]: <Response [200]>
```

```
In [20]: soup = BeautifulSoup(b.content)
```

```
In [21]: players = soup.find("td",class_="table-body__cell name")
print(players.text)
```

Rassie van der Dussen

```
In [22]: players=[]
for i in soup.find_all("td",class_="table-body__cell name"):
    players.append(i.text)
```

```
In [23]: teams = soup.find("span",class_="table-body__logo-text")
teams.text
```

```
Out[23]: 'SA'
```

```
In [24]: teams = []
for i in soup.find_all("span",class_="table-body__logo-text"):
    teams.append(i.text)
```

```
In [25]: ratings = soup.find("td",class_="table-body__cell u-text-right rating")
ratings.text
```

```
Out[25]: '777'
```

```
In [26]: ratings = []
for i in soup.find_all("td",class_="table-body__cell u-text-right rating"):
    ratings.append(i.text)
```

```
In [27]: Ratings = ratings[0:10]
```

```
In [28]: Teams= teams[0:10]
```

```
In [29]: Players=players[0:10]
```

```
In [30]: import pandas as pd
df = pd.DataFrame({'Players':Players,'Teams':Teams,'Ratings':Ratings})
df
```

Out[30]:

	Players	Teams	Ratings
0	\nRassie van der Dussen\n	SA	777
1	\nFakhar Zaman\n	PAK	755
2	\nImam-ul-Haq\n	PAK	745
3	\nShubman Gill\n	IND	738
4	\nHarry Tector\n	IRE	726
5	\nDavid Warner\n	AUS	726
6	\nVirat Kohli\n	IND	719
7	\nQuinton de Kock\n	SA	718
8	\nRohit Sharma\n	IND	707
9	\nMohammed Siraj\n	IND	691

C.

```
In [31]: c = requests.get("https://www.icc-cricket.com/rankings/mens/player-rankings/odi/bowler")
```

```
In [32]: soup = BeautifulSoup(c.content)
```

```
In [33]: bowler = soup.find("td", class_="table-body__cell rankings-table__name name")
```

```
In [34]: bowler = []
for i in soup.find_all("td", class_="table-body__cell rankings-table__name name"):
    bowler.append(i.text)
```

```
In [35]: Bowler = bowler[:10]
Bowler
```

Out[35]:

```
['\nMohammed Siraj\n',
 '\nMitchell Starc\n',
 '\nMatt Henry\n',
 '\nTrent Boult\n',
 '\nAdam Zampa\n',
 '\nRashid Khan\n',
 '\nShaheen Afridi\n',
 '\nMujeeb Ur Rahman\n',
 '\nMohammad Nabi\n',
 '\nMark Watt\n']
```

```
In [36]: team = soup.find("span", class_="table-body__logo-text")
team.text
```

```
Out[36]: 'IND'
```

```
In [37]: team = []
for i in soup.find_all("span", class_="table-body__logo-text"):
    team.append(i.text)
```

```
In [38]: Teams = team[:10]
Teams
```

```
Out[38]: ['IND', 'AUS', 'NZ', 'NZ', 'AUS', 'AFG', 'PAK', 'AFG', 'AFG', 'SCO']
```

```
In [39]: ratings = []
for i in soup.find_all("td", class_="table-body__cell rating"):
    ratings.append(i.text)
```

```
In [40]: Ratings = ratings[:10]
Ratings
```

```
Out[40]: ['691', '686', '667', '660', '652', '640', '630', '630', '626', '621']
```

```
In [41]: import pandas as pd
Top_bowlers = pd.DataFrame({'Bowler':Bowler, 'Teams':Teams, 'Ratings':Ratings})
Top_bowlers
```

```
Out[41]:
```

	Bowler	Teams	Ratings
0	\nMohammed Siraj\n	IND	691
1	\nMitchell Starc\n	AUS	686
2	\nMatt Henry\n	NZ	667
3	\nTrent Boult\n	NZ	660
4	\nAdam Zampa\n	AUS	652
5	\nRashid Khan\n	AFG	640
6	\nShaheen Afridi\n	PAK	630
7	\nMujeeb Ur Rahman\n	AFG	630
8	\nMohammad Nabi\n	AFG	626
9	\nMark Watt\n	SCO	621
.			
.			

## 4. Women cricket rankings

# A.

.

```
In [42]: women_team_url = requests.get("https://www.icc-cricket.com/rankings/womens/team-rankir")
```

```
In [43]: soup = BeautifulSoup(women_team_url.content)
```

```
In [44]: Wteam = soup.find("span",class_="u-hide-phablet")
```

```
In [45]: Wteam = []
for i in soup.find_all("span",class_="u-hide-phablet"):
    Wteam.append(i.text)
```

```
In [46]: WTeams = Wteam[:10]
WTeams
```

```
Out[46]: ['Australia',
          'England',
          'South Africa',
          'India',
          'New Zealand',
          'West Indies',
          'Bangladesh',
          'Sri Lanka',
          'Thailand',
          'Pakistan']
```

```
In [47]: Wmatches= []
for i in soup.find_all("td",class_="table-body__cell u-center-text"):
    Wmatches.append(i.text)
Wmatches
```

```
Out[47]: ['28',
 '3,342',
 '26',
 '3,098',
 '27',
 '2,820',
 '28',
 '2,688',
 '29',
 '2,743',
 '14',
 '977',
 '12',
 '820',
 '12',
 '806',
 '27',
 '1,678',
 '16',
 '605',
 '10',
 '90',
 '11',
 '0']
```

## B.

```
In [48]: women_batsmen_url = requests.get("https://www.icc-cricket.com/rankings/womens/player-r
```

```
In [49]: soup = BeautifulSoup(women_batsmen_url.content)
```

```
In [50]: women_batsmen = []
for i in soup.find_all("td", class_="table-body__cell rankings-table__name name"):
    women_batsmen.append(i.text)
Women_Batsmen = women_batsmen[:10]
Women_Batsmen
```

```
Out[50]: ['\nJess Jonassen\n',
 '\nShabnim Ismail\n',
 '\nMegan Schutt\n',
 '\nHayley Matthews\n',
 '\nKate Cross\n',
 '\nAyabonga Khaka\n',
 '\nRajeshwari Gayakwad\n',
 '\nMarizanne Kapp\n',
 '\nDeepti Sharma\n',
 '\nEllyse Perry\n']
```

```
In [51]: women_team = []
for i in soup.find_all("span", class_="table-body__logo-text"):
    women_team.append(i.text)
Women_Team = women_team[:10]
Women_Team
```

```
Out[51]: ['AUS', 'SA', 'AUS', 'WI', 'ENG', 'SA', 'IND', 'SA', 'IND', 'AUS']
```

```
In [52]: women_ratings = []
for i in soup.find_all("td", class_="table-body__cell rating"):
    women_ratings.append(i.text)
Women_Ratings = women_ratings[:10]
Women_Ratings
```

```
Out[52]: ['723', '722', '704', '662', '655', '634', '617', '598', '589', '585']
```

```
In [53]: import pandas as pd
Women_odi = pd.DataFrame({'Women_Batsmen': Women_Batsmen, 'Women_Team': Women_Team, 'Women_Ratings': Women_Ratings})
```

	Women_Batsmen	Women_Team	Women_Ratings
0	\nJess Jonassen\n	AUS	723
1	\nShabnim Ismail\n	SA	722
2	\nMegan Schutt\n	AUS	704
3	\nHayley Matthews\n	WI	662
4	\nKate Cross\n	ENG	655
5	\nAyabonga Khaka\n	SA	634
6	\nRajeshwari Gayakwad\n	IND	617
7	\nMarizanne Kapp\n	SA	598
8	\nDeepti Sharma\n	IND	589
9	\nEllyse Perry\n	AUS	585

C.

```
In [54]: women_bowling_url = requests.get("https://www.icc-cricket.com/rankings/womens/player-rankings/bowling")
```

```
In [55]: soup = BeautifulSoup(women_bowling_url.content)
```

```
In [56]: women_bowling = []
for i in soup.find_all("td", class_="table-body__cell rankings-table__name name"):
    women_bowling.append(i.text)
Women_bowling = women_bowling[:10]
print(Women_bowling)
```

```
['\nJess Jonassen\n', '\nShabnim Ismail\n', '\nMegan Schutt\n', '\nHayley Matthews\n', '\nKate Cross\n', '\nAyabonga Khaka\n', '\nRajeshwari Gayakwad\n', '\nMarizanne Kapp\n', '\nDeepti Sharma\n', '\nEllyse Perry\n']
```

```
In [57]: women_bowling_team = []
for i in soup.find_all("span",class_="table-body__logo-text"):
    women_bowling_team.append(i.text)
Women_bowling_Team = women_bowling_team[:10]
Women_bowling_Team
```

Out[57]: ['AUS', 'SA', 'AUS', 'WI', 'ENG', 'SA', 'IND', 'SA', 'IND', 'AUS']

```
In [58]: women_bowling_ratings = []
for i in soup.find_all("td",class_="table-body__cell rating"):
    women_bowling_ratings.append(i.text)
Women_bowling_Ratings = women_bowling_ratings[:10]
Women_bowling_Ratings
```

Out[58]: ['723', '722', '704', '662', '655', '634', '617', '598', '589', '585']

```
In [59]: import pandas as pd
Women_odi = pd.DataFrame({'Women_bowling':Women_bowling,'Women_bowling_Team':Women_bowling_Team,
                           Women_bowling_Ratings:Women_bowling_Ratings})
```

	Women_bowling	Women_bowling_Team	Women_bowling_Ratings
0	\nJess Jonassen\n	AUS	723
1	\nShabnim Ismail\n	SA	722
2	\nMegan Schutt\n	AUS	704
3	\nHayley Matthews\n	WI	662
4	\nKate Cross\n	ENG	655
5	\nAyabonga Khaka\n	SA	634
6	\nRajeshwari Gayakwad\n	IND	617
7	\nMarizanne Kapp\n	SA	598
8	\nDeepti Sharma\n	IND	589
9	\nEllyse Perry\n	AUS	585
.			
.			

## 5. News details

```
In [60]: news = requests.get("https://www.cnbc.com/2023/07/06/janet-yellen-arrives-in-beijing-on-a-visit-to-china")
news
```

Out[60]: <Response [200]>

```
In [61]: soup= BeautifulSoup(news.content)
```

```
In [62]: cnbc= soup.find('h1',class_="ArticleHeader-headline")
print(cnbc.text)

Janet Yellen arrives in Beijing on mission to find common ground for U.S. and China

In [63]: time= soup.find("div",class_="ArticleHeader-time")

In [64]: cnbc=[]
for i in soup.find_all('h1',class_="ArticleHeader-headline"):
    cnbc.append(i.text)
cnbc

Out[64]: ['Janet Yellen arrives in Beijing on mission to find common ground for U.S. and Chin
a']

time=[] for i in soup.findall("div",class_="ArticleHeader-time"): time.append(i.text) time

In [65]: link= "https://www.cnbc.com/2023/07/06/trump-aide-walt-nauta-arrainment.html"

In [66]: news = pd.DataFrame({'CNBC':cnbc,'Link':link})
news

Out[66]:
```

CNBC	Link
0 Janet Yellen arrives in Beijing on mission to ...	<a href="https://www.cnbc.com/2023/07/06/trump-aide-wal...">https://www.cnbc.com/2023/07/06/trump-aide-wal...</a>
.	.
.	.

## 6. Most downloaded articles from AI in last 90 days

```
In [67]: ai_url = requests.get("https://www.journals.elsevier.com/artificial-intelligence/most-
ai_url

Out[67]: <Response [200]>

In [68]: soup = BeautifulSoup(ai_url.content)

In [69]: title = soup.find("p",class_="sc-1q3g1nv-0 sc-15ho9eu-1 iYcAGX cVARLc")

In [70]: authors = soup.find("span",class_="sc-1w3fpd7-0 dnCnAO")

In [71]: date = soup.find("span",class_="sc-1thf9ly-2 dvggwt")

In [72]: aiurl = "https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-art

In [73]: topics = soup.find("h2",class_="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu &

In [74]: title.text
```

```
Out[74]: 'The most downloaded articles from Artificial Intelligence in the last 90 days.'
```

.

```
In [75]: title=[]
for i in soup.find_all("p",class_="sc-1q3g1nv-0 sc-15ho9eu-1 iYcAGX cVARLc"):
    title.append(i.text)

title
```

```
Out[75]: ['The most downloaded articles from Artificial Intelligence in the last 90 days.']
```

```
In [76]: date=[]
for i in soup.find_all("span",class_="sc-1thf9ly-2 dvggWt"):
    date.append(i.text)

date
```

```
Out[76]: ['October 2021',
'February 2019',
'August 1998',
'February 2015',
'January 2022',
'October 2015',
'August 1999',
'September 2021',
'April 2021',
'August 2021',
'February 2021',
'April 2023',
'November 2021',
'March 2023',
'May 2021',
'March 2020',
'December 1997',
'June 2017',
'June 2021',
'June 2016',
'April 2011',
'March 2023',
'May 1998',
'July 2021']
```

```
In [77]: authors=[]
for i in soup.find_all("span",class_="sc-1w3fpd7-0 dnCnAO"):
    authors.append(i.text)

authors
```

```
Out[77]: ['David Silver, Satinder Singh, Doina Precup, Richard S. Sutton ',  
          'Tim Miller ',  
          'Margaret A. Boden ',  
          'Guni Sharon, Roni Stern, Ariel Felner, Nathan R. Sturtevant ',  
          'Ilaria Tiddi, Stefan Schlobach ',  
          'Henry Prakken, Giovanni Sartor ',  
          'Richard S. Sutton, Doina Precup, Satinder Singh ',  
          'Kjersti Aas, Martin Jullum, Anders Løland ',  
          'Wenhan Luo, Junliang Xing and 4 more',  
          'Saurabh Arora, Prashant Doshi ',  
          'Jasper van der Waa, Elisabeth Nieuwburg, Anita Cremers, Mark Neerincx ',  
          'Joe Collenette, Katie Atkinson, Trevor Bench-Capon ',  
          'Roel Dobbe, Thomas Krendl Gilbert, Yonatan Mintz ',  
          'Oskar Wysocki, Jessica Katharine Davies and 5 more',  
          'Eoin M. Kenny, Courtney Ford, Molly Quinn, Mark T. Keane ',  
          'Nolan Bard, Jakob N. Foerster and 13 more',  
          'Ron Kohavi, George H. John ',  
          'Séverin Lemaignan, Mathieu Warnier and 3 more',  
          'Tomáš Kliegr, Štěpán Bahník, Johannes Fürnkranz ',  
          'Luigia Carlucci Aiello ',  
          'Patrick Lin, Keith Abney, George Bekey ',  
          'W. Bradley Knox, Alessandro Allievi and 3 more',  
          'Leslie Pack Kaelbling, Michael L. Littman, Anthony R. Cassandra ',  
          'Markus Langer, Daniel Oster and 6 more']
```

```
In [78]: topics=[]  
for i in soup.find_all("h2",class_="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btct  
topics.append(i.text)  
  
topics
```

```
Out[78]: ['Reward is enough',
 'Explanation in artificial intelligence: Insights from the social sciences',
 'Creativity and artificial intelligence',
 'Conflict-based search for optimal multi-agent pathfinding',
 'Knowledge graphs as tools for explainable machine learning: A survey',
 'Law and logic: A review from an argumentation perspective',
 'Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning',
 'Explaining individual predictions when features are dependent: More accurate approximations to Shapley values',
 'Multiple object tracking: A literature review',
 'A survey of inverse reinforcement learning: Challenges, methods and progress',
 'Evaluating XAI: A comparison of rule-based and example-based explanations',
 'Explainable AI tools for legal reasoning about cases: A study on the European Court of Human Rights',
 'Hard choices in artificial intelligence',
 'Assessing the communication gap between AI models and healthcare professionals: Explainability, utility and trust in AI-driven clinical decision-making',
 'Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies',
 'The Hanabi challenge: A new frontier for AI research',
 'Wrappers for feature subset selection',
 'Artificial cognition for social human-robot interaction: An implementation',
 'A review of possible effects of cognitive biases on interpretation of rule-based machine learning models',
 'The multifaceted impact of Ada Lovelace in the digital age',
 'Robot ethics: Mapping the issues for a mechanized world',
 'Reward (Mis)design for autonomous driving',
 'Planning and acting in partially observable stochastic domains',
 'What do we want from Explainable Artificial Intelligence (XAI)? - A stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research']
```

```
In [79]: AI = pd.DataFrame({'Topics':topics,'Authors':authors,'Date':date,'AIurl':aiurl})
AI
```

Out[79]:

	Topics	Authors	Date	Alurl
0	Reward is enough	David Silver, Satinder Singh, Doina Precup, Ri...	October 2021	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
1	Explanation in artificial intelligence: Insight...	Tim Miller	February 2019	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
2	Creativity and artificial intelligence	Margaret A. Boden	August 1998	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
3	Conflict-based search for optimal multi-agent ...	Guni Sharon, Roni Stern, Ariel Felner, Nathan ...	February 2015	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
4	Knowledge graphs as tools for explainable mach...	Ilaria Tiddi, Stefan Schlobach	January 2022	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
5	Law and logic: A review from an argumentation ...	Henry Prakken, Giovanni Sartor	October 2015	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
6	Between MDPs and semi-MDPs: A framework for te...	Richard S. Sutton, Doina Precup, Satinder Singh	August 1999	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
7	Explaining individual predictions when feature...	Kjersti Aas, Martin Jullum, Anders Løland	September 2021	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
8	Multiple object tracking: A literature review	Wenhan Luo, Junliang Xing and 4 more	April 2021	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
9	A survey of inverse reinforcement learning: Ch...	Saurabh Arora, Prashant Doshi	August 2021	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
10	Evaluating XAI: A comparison of rule-based and...	Jasper van der Waa, Elisabeth Nieuwburg, Anita...	February 2021	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
11	Explainable AI tools for legal reasoning about...	Joe Collenette, Katie Atkinson, Trevor Bench-C...	April 2023	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
12	Hard choices in artificial intelligence	Roel Dobbe, Thomas Krendl Gilbert, Yonatan Mintz	November 2021	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
13	Assessing the communication gap between AI mod...	Oskar Wysocki, Jessica Katharine Davies and 5 ...	March 2023	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
14	Explaining black-box classifiers using post-ho...	Eoin M. Kenny, Courtney Ford, Molly Quinn, Mar...	May 2021	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>
15	The Hanabi challenge: A new frontier for AI re...	Nolan Bard, Jakob N. Foerster and 13 more	March 2020	<a href="https://www.journals.elsevier.com/artificial-i...">https://www.journals.elsevier.com/artificial-i...</a>

	Topics	Authors	Date	Alurl
16	Wrappers for feature subset selection	Ron Kohavi, George H. John	December 1997	<a href="https://www.journals.elsevier.com/artificial-intelligence">https://www.journals.elsevier.com/artificial-i... ...</a>
17	Artificial cognition for social human–robot in...	Séverin Lemaignan, Mathieu Warnier and 3 more	June 2017	<a href="https://www.journals.elsevier.com/artificial-intelligence">https://www.journals.elsevier.com/artificial-i... ...</a>
18	A review of possible effects of cognitive bias...	Tomáš Kliegr, Štěpán Bahník, Johannes Fürnkranz	June 2021	<a href="https://www.journals.elsevier.com/artificial-intelligence">https://www.journals.elsevier.com/artificial-i... ...</a>
19	The multifaceted impact of Ada Lovelace in the...	Luigia Carlucci Aiello	June 2016	<a href="https://www.journals.elsevier.com/artificial-intelligence">https://www.journals.elsevier.com/artificial-i... ...</a>
20	Robot ethics: Mapping the issues for a mechan...	Patrick Lin, Keith Abney, George Bekey	April 2011	<a href="https://www.journals.elsevier.com/artificial-intelligence">https://www.journals.elsevier.com/artificial-i... ...</a>
21	Reward (Mis)design for autonomous driving	W. Bradley Knox, Alessandro Allievi and 3 more	March 2023	<a href="https://www.journals.elsevier.com/artificial-intelligence">https://www.journals.elsevier.com/artificial-i... ...</a>
22	Planning and acting in partially observable st...	Leslie Pack Kaelbling, Michael L. Littman, Ant...	May 1998	<a href="https://www.journals.elsevier.com/artificial-intelligence">https://www.journals.elsevier.com/artificial-i... ...</a>
23	What do we want from Explainable Artificial In...	Markus Langer, Daniel Oster and 6 more	July 2021	<a href="https://www.journals.elsevier.com/artificial-intelligence">https://www.journals.elsevier.com/artificial-i... ...</a>

.

.

## 7. Dineout

```
In [80]: dine_url= requests.get("https://www.dineout.co.in/lucknow-restaurants/welcome-back")
dine_url
```

```
Out[80]: <Response [200]>
```

```
In [81]: soup = BeautifulSoup(dine_url.content)
```

```
In [82]: name= soup.find("a",class_="restnt-name ellipsis")
print(name.text)
```

Tanatan

```
In [83]: Address = soup.find("div",class_="restnt-loc ellipsis")
print(Address.text)
```

Hazratganj, Central Lucknow

```
In [84]: rating = soup.find("div",class_="restnt-rating rating-4")
print(rating.text)
```

## 4.1

```
In [85]: cuisine = soup.find("span",class_="double-line-ellipsis")
print(cuisine.text.split('|')[1])
```

North Indian, Awadhi, Malvani

```
In [86]: image = soup.find("img",class_="no-img")
print(image.get)
```

```
<bound method Tag.get of >
```

.

```
In [87]: Name = []
for i in soup.find_all("a",class_="restnt-name ellipsis"):
    Name.append(i.text)
Name
```

```
Out[87]: ['Tanatan',
'Branche Bar',
'Branche',
'Naimat Khana',
'Home Sweet Home',
'Al-baik',
"Karim's",
'Royal Cafe',
'The Chocolate Room',
'Pizza Hut',
'Pavillion',
'Woodpecker',
'EOS - Bar And Bistro',
'Kabila Restro - The Family Lounge',
'The Terrace Grill',
'Taste Of India',
'Atrium',
'Cafe 32',
'The Cherry Tree Cafe',
'Al-baik',
'On High']
```

```
In [88]: Cuisine = []
for i in soup.find_all("span",class_="double-line-ellipsis"):
    Cuisine.append(i.text.split('|')[1])
Cuisine
```

```
Out[88]: ['North Indian, Awadhi, Malvani',
  'North Indian, Finger Food',
  'Continental, North Indian, Chinese',
  'Awadhi, North Indian',
  'North Indian, Pizza',
  'Fast Food, Lebanese',
  'Mughlai, North Indian',
  'North Indian, Chinese, Continental, Fast Food',
  'Continental, Italian, Desserts, Beverages',
  'Pizza, Fast Food',
  'Chinese, North Indian, Continental',
  'North Indian, Chinese, Fast Food',
  'Finger Food',
  'Chinese, North Indian, Continental',
  'North Indian',
  'Chinese, North Indian, Mughlai',
  'North Indian, Continental, South Indian, Asian',
  'North Indian, Continental, Chinese',
  'Italian, North Indian, American, Asian',
  'Fast Food, Lebanese',
  'North Indian, Italian, Chinese, Continental']
```

```
In [89]: Location = []
for i in soup.find_all("div", class_="restnt-loc ellipsis"):
    Location.append(i.text)
Location
```

```
Out[89]: ['Hazratganj, Central Lucknow',
  'Golden Tulip Lucknow,Husainganj, Central Lucknow',
  'Golden Tulip Lucknow,Husainganj, Central Lucknow',
  'Hazratganj, Central Lucknow',
  'Hazratganj, Central Lucknow',
  'Husainganj, Central Lucknow',
  'Lalbagh, Central Lucknow',
  'Hotel Royal Inn,Hazratganj, Central Lucknow',
  'Hazratganj, Central Lucknow',
  'Hazratganj, Central Lucknow',
  'La Place Sarovar Portico,Hazratganj, Central Lucknow',
  'Hotel Charans Plaza,Hazratganj, Central Lucknow',
  'Hotel Levana Suites,Hazratganj, Central Lucknow',
  'Mahmoodbad Building,Hazratganj, Central Lucknow',
  'La Place Sarovar Portico,Hazratganj, Central Lucknow',
  'Hazratganj, Central Lucknow',
  'Hotel Levana Suites,Hazratganj, Central Lucknow',
  'Hotel India Awadh,Hazratganj, Central Lucknow',
  'Hazratganj, Central Lucknow',
  'Hazratganj, Central Lucknow',
  'Hotel Silvete,Hazratganj, Central Lucknow']
```

```
In [90]: Ratings =[]
for i in soup.find_all("div", class_="restnt-rating rating-4"):
    Ratings.append(i.text)
Ratings
```

```
Out[90]: ['4.1',
 '4.4',
 '4.4',
 '4.1',
 '4.1',
 '3.8',
 '4.2',
 '4.1',
 '4.1',
 '4',
 '3.6',
 '4.4',
 '3.8',
 '3.9',
 '4.4']
```

```
In [91]: Image_URL = []
for i in soup.find_all("img", class_="no-img"):
    Image_URL.append(i.get('data-src'))
Image_URL
```

```
Out[91]: ['https://im1.dineout.co.in/images/uploads/restaurant/sharpen/8/n/e/p89909-16160535306053051aed289.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/i/e/p56414-15586218045ce6ae6c14bdb.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/1/h/j/p17524-15569328235cce8d779e46.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/y/k/p56410-15585240555ce5309719abe.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/8/c/c/p82546-16121678606017bab4311e8.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/u/y/p5358-15571214265ccfc99296891.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/9/x/z/p97332-163309342661570732c5241.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/t/r/p3383-15498721025c612be6008fb.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/h/o/p58180-1647405248623168c0afb55.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/p/v/p3382-15494491505c5ab7bea9b4c.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/1/v/l/p17345-168863376964a681a910832.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/7/g/n/p73841-16454563546213abe2d7d46.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/1/o/g/p17350-15580806885cde6cb0f14e1.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/a/l/p53964-165164166662720d42c4fad.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/c/i/p56403-15568020785ccaea1e1ae6f.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/u/c/p56092-15804646455e33fa0504333.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/1/h/c/p17352-15557555365cbaf210a9391.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/1/q/h/p17379-15569779295ccd990924273.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/1/o/k/p17300-15581693565cdfc70ca5301.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/u/y/p5358-15571214265ccfc99296891.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/u/h/p53968-15673338055d6b9dad2b950.jpg?tr=tr:n-medium']
```

In [92]: `len(Ratings)`

Out[92]: 15

In [93]: `DINEOUT = pd.DataFrame({'Restaurant name':Name,'Cuisine':Cuisine,'Location':Cuisine,'']})`

In [94]: `DINEOUT`

Out[94]:

	<b>Restaurant name</b>	<b>Cuisine</b>	<b>Location</b>	<b>Image_URL</b>
0	Tanatan	North Indian, Awadhi, Malvani	North Indian, Awadhi, Malvani	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
1	Branche Bar	North Indian, Finger Food	North Indian, Finger Food	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
2	Branche	Continental, North Indian, Chinese	Continental, North Indian, Chinese	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
3	Naimat Khana	Awadhi, North Indian	Awadhi, North Indian	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
4	Home Sweet Home	North Indian, Pizza	North Indian, Pizza	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
5	Al-baik	Fast Food, Lebanese	Fast Food, Lebanese	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
6	Karim's	Mughlai, North Indian	Mughlai, North Indian	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
7	Royal Cafe	North Indian, Chinese, Continental, Fast Food	North Indian, Chinese, Continental, Fast Food	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
8	The Chocolate Room	Continental, Italian, Desserts, Beverages	Continental, Italian, Desserts, Beverages	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
9	Pizza Hut	Pizza, Fast Food	Pizza, Fast Food	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
10	Pavillion	Chinese, North Indian, Continental	Chinese, North Indian, Continental	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
11	Woodpecker	North Indian, Chinese, Fast Food	North Indian, Chinese, Fast Food	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
12	EOS - Bar And Bistro	Finger Food	Finger Food	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
13	Kabila Restro - The Family Lounge	Chinese, North Indian, Continental	Chinese, North Indian, Continental	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
14	The Terrace Grill	North Indian	North Indian	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
15	Taste Of India	Chinese, North Indian, Mughlai	Chinese, North Indian, Mughlai	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
16	Atrium	North Indian, Continental, South Indian, Asian	North Indian, Continental, South Indian, Asian	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
17	Cafe 32	North Indian, Continental,	North Indian, Continental,	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>

	<b>Restaurant name</b>	<b>Cuisine</b>	<b>Location</b>	<b>Image_URL</b>
		Chinese	Chinese	
18	The Cherry Tree Cafe	Italian, North Indian, American, Asian	Italian, North Indian, American, Asian	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
19	Al-baik	Fast Food, Lebanese	Fast Food, Lebanese	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>
20	On High	North Indian, Italian, Chinese, Continental	North Indian, Italian, Chinese, Continental	<a href="https://im1.dineout.co.in/images/uploads/resta...">https://im1.dineout.co.in/images/uploads/resta...</a>