

Web Scraping Assignment 1

Importing required libraries

In [1]:

```
1 import bs4
2 from bs4 import BeautifulSoup
3 import requests
4 import pandas as pd
```

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

In [2]:

```
1 page=requests.get("https://en.wikipedia.org/wiki/Main_Page")
2 page
```

Out[2]: <Response [200]>

In [3]:

```
1 soup=BeautifulSoup(page.content)
```

```
In [4]: 1 header = soup.find_all(['h1', 'h2', 'h3', 'h4', 'h5', 'h6'])
2 header
```

```
Out[4]: [<h1 class="firstHeading mw-first-heading" id="firstHeading" style="display: none"><span class="mw-page-title-main">
Main Page</span></h1>,
<h1><span class="mw-headline" id="Welcome_to_Wikipedia">Welcome to <a href="/wiki/Wikipedia" title="Wikipedia">Wiki
pedia</a></span></h1>,
<h2 class="mp-h2" id="mp-tfa-h2"><span id="From_today.27s_featured_article"></span><span class="mw-headline" id="Fr
om_today's_featured_article">From today's featured article</span></h2>,
<h2 class="mp-h2" id="mp-dyk-h2"><span class="mw-headline" id="Did_you_know_...">Did you know ...</span></h2>,
<h2 class="mp-h2" id="mp-itn-h2"><span class="mw-headline" id="In_the_news">In the news</span></h2>,
<h2 class="mp-h2" id="mp-otd-h2"><span class="mw-headline" id="On_this_day">On this day</span></h2>,
<h2 class="mp-h2" id="mp-tfp-h2"><span id="Today.27s_featured_picture"></span><span class="mw-headline" id="Today's
_featured_picture">Today's featured picture</span></h2>,
<h2 class="mp-h2" id="mp-other"><span class="mw-headline" id="Other_areas_of_Wikipedia">Other areas of Wikipedia</s
pan></h2>,
<h2 class="mp-h2" id="mp-sister"><span id="Wikipedia.27s_sister_projects"></span><span class="mw-headline" id="Wiki
pedia's_sister_projects">Wikipedia's sister projects</span></h2>,
<h2 class="mp-h2" id="mp-lang"><span class="mw-headline" id="Wikipedia_languages">Wikipedia languages</span></h2>]
```

```
In [5]: 1 df=pd.DataFrame(header, columns=['Header'])
```

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\internals\construction.py:576: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.

```
values = np.array([convert(v) for v in values])
```

In [6]: 1 df

Out[6]:

| 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 of office) from <https://presidentofindia.nic.in/former-presidents.htm> (<https://presidentofindia.nic.in/former-presidents.htm>) and make data frame.

In [7]: 1 page=requests.get("https://presidentofindia.nic.in/former-presidents.htm")
2 page

Out[7]: <Response [200]>

In [8]: 1 soup1=BeautifulSoup(page.content)

```
In [9]: 1 soup1
```

```
Out[9]: <!DOCTYPE html>
<html lang="en" xml:lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1"><title>
    Former Presidents - The President of India
</title><meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
<!--<meta http-equiv="Content-Style-Type" content="text/css" /><meta http-equiv="Content-Script-Type" content="typ
e" />-->
<meta content="telephone=no" name="format-detection"/><meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<!-- Start Favicon -->
<link href="favicon.ico" rel="shortcut icon" type="image/x-icon"/><link href="js/panorama_viewer.css" rel="stylesheet" type="text/css"/>
<!-- Start Javascript Library -->
<script language="JavaScript" type="text/javascript">
    if (top != self) top.location.replace(location);
    version = "3.6.0";
    prettyPhoto = "3.1.6";

</script>
<script src="js/jquery.min.js" type="text/javascript"></script>
```

```
In [10]: 1 n=soup1.find('div',class_="presidentListing")
2 name=n.text.split("\n")[1].split("(")[0]
3 term=n.text.split("\n")[2]
4 print(name)
5 print(term)
```

Shri Ram Nath Kovind
Term of Office: 25 July, 2017 to 25 July, 2022

```
In [11]: 1 names=[]
2
3 for name in soup1.find_all('div',class_="presidentListing"):
4     names.append(name.text.split("\n")[1].split("(")[0])
5
6 print(names)
7
```

```
['Shri Ram Nath Kovind ', 'Shri Pranab Mukherjee ', 'Smt Pratibha Devi Singh Patil ', 'DR. A.P.J. Abdul Kalam ', 'Shri K. R. Narayanan ', 'Dr Shankar Dayal Sharma ', 'Shri R Venkataraman ', 'Giani Zail Singh ', 'Shri Neelam Sanjiva Reddy ', 'Dr. Fakhruddin Ali Ahmed ', 'Shri Varahagiri Venkata Giri ', 'Dr. Zakir Husain ', 'Dr. Sarvepalli Radhakrishnan ', 'Dr. Rajendra Prasad ']
```

```
In [12]: 1 n=soup1.find('div',class_="presidentListing")
2
3 term=n.text.split('\n')[2].split(":")[1]
4
5 print(term)
```

```
25 July, 2017 to 25 July, 2022
```

```
In [13]: 1 terms=[]
2 for term in soup1.find_all('div',class_="presidentListing"):
3     terms.append(term.text.split("\n")[2].split(":")[1])
4
5 print(terms)
```

```
[' 25 July, 2017 to 25 July, 2022 ', ' 25 July, 2012 to 25 July, 2017 ', ' 25 July, 2007 to 25 July, 2012 ', ' 25 July, 2002 to 25 July, 2007 ', ' 25 July, 1997 to 25 July, 2002 ', ' 25 July, 1992 to 25 July, 1997 ', ' 25 July, 1987 to 25 July, 1992 ', ' 25 July, 1982 to 25 July, 1987 ', ' 25 July, 1977 to 25 July, 1982 ', ' 24 August, 1974 to 11 February, 1977', ' 3 May, 1969 to 20 July, 1969 and 24 August, 1969 to 24 August, 1974', ' 13 May, 1967 to 3 May, 1969', ' 13 May, 1962 to 13 May, 1967', ' 26 January, 1950 to 13 May, 1962']
```

```
In [14]: 1 print(len(names),len(terms))
```

```
14 14
```

In [15]:

```

1 preident_df=pd.DataFrame({"Name Of Preident":names,"Term of Office":terms})
2 preident_df

```

Out[15]:

| | Name Of Preident | Term of Office |
|----|------------------------------|--|
| 0 | Shri Ram Nath Kovind | 25 July, 2017 to 25 July, 2022 |
| 1 | Shri Pranab Mukherjee | 25 July, 2012 to 25 July, 2017 |
| 2 | Smt Pratibha Devisingh Patil | 25 July, 2007 to 25 July, 2012 |
| 3 | DR. A.P.J. Abdul Kalam | 25 July, 2002 to 25 July, 2007 |
| 4 | Shri K. R. Narayanan | 25 July, 1997 to 25 July, 2002 |
| 5 | Dr Shankar Dayal Sharma | 25 July, 1992 to 25 July, 1997 |
| 6 | Shri R Venkataraman | 25 July, 1987 to 25 July, 1992 |
| 7 | Giani Zail Singh | 25 July, 1982 to 25 July, 1987 |
| 8 | Shri Neelam Sanjiva Reddy | 25 July, 1977 to 25 July, 1982 |
| 9 | Dr. Fakhruddin Ali Ahmed | 24 August, 1974 to 11 February, 1977 |
| 10 | Shri Varahagiri Venkata Giri | 3 May, 1969 to 20 July, 1969 and 24 August, 1... |
| 11 | Dr. Zakir Husain | 13 May, 1967 to 3 May, 1969 |
| 12 | Dr. Sarvepalli Radhakrishnan | 13 May, 1962 to 13 May, 1967 |
| 13 | Dr. Rajendra Prasad | 26 January, 1950 to 13 May, 1962 |

3) Write a python program to scrape cricket rankings from icc-cricket.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.

3(a) Top 10 ODI teams in men's cricket

```
In [16]: 1 page_m_odi=requests.get("https://www.icc-cricket.com/rankings/mens/team-rankings/odi")
2 page_m_odi
```

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

```
In [17]: 1 soup_odi=BeautifulSoup(page_m_odi.content)
2 soup_odi
```

```
Out[17]: <!DOCTYPE html>
<html lang="en">
<head>
<meta content="ICC Men's ODI Team Rankings | ICC" name="twitter:title"/>
<meta content="website" property="og:type"/>
<meta content="summary_large_image" property="twitter:card"/>
<meta content="Official International Cricket Council ranking for One Day International (ODI) cricket teams. Discover latest ICC rankings table, predict upcoming matches, see points and ratings for all teams." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official International Cricket Council ranking for One Day International (ODI) cricket teams. Discover latest ICC rankings table, predict upcoming matches, see points and ratings for all teams." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-thumbnail.jpg" name="twitter:image"/>
<meta content="ICC Men's ODI Team Rankings | ICC" property="og:title"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-thumbnail.jpg" property="og:image"/>
<title>ICC Men's ODI Team Rankings | ICC</title>
<meta content="Official International Cricket Council ranking for One Day International (ODI) cricket teams. Discover latest ICC rankings table, predict upcoming matches, see points and ratings for all teams." property="og:description"/>
```

```
In [18]: 1 team=soup_odi.find('span',class_="u-hide-phablet")
2 team.text
```

```
Out[18]: 'Australia'
```

In [19]:

```
1 team=[]
2 for i in soup_odi.find_all('span',class_="u-hide-phablet"):
3     team.append(i.text)
4 team=team[0:10]
5
```

In [20]:

```
1 print(team)
```

```
['Australia', 'Pakistan', 'India', 'New Zealand', 'England', 'South Africa', 'Bangladesh', 'Afghanistan', 'Sri Lanka', 'West Indies']
```

In [21]:

```
1 matches=[]
2 match=soup_odi.find('td',class_="rankings-block__banner--matches")
3 matches.append(match.text)
```

In [22]:

```
1 matches
```

```
Out[22]: ['23']
```

In [23]:

```
1 points=[]
2 point=soup_odi.find('td',class_="rankings-block__banner--points")
3 points.append(point.text)
4 points
```

```
Out[23]: ['2,714']
```

In [24]:

```
1 rating=[]
2 rate=soup_odi.find('td',class_="rankings-block__banner--rating u-text-right")
3 rating.append(rate.text.split('\n')[1].strip(" "))
4 rating
```

```
Out[24]: ['118']
```

```
In [25]: 1 data=soup_odi.find("tr",class_="table-body")
2 data.text.split("\n") # index number 7th,8th and 9th data are required.
```

```
Out[25]: ['', '2', '', '', 'Pakistan', 'PAK', '', '20', '2,316', '116', '']
```

```
In [26]: 1 for i in soup_odi.find_all("tr",class_="table-body"):
2     matches.append(i.text.split("\n")[7])
3     points.append(i.text.split("\n")[8])
4     rating.append(i.text.split("\n")[9])
5 print(matches, '\n\n',points, '\n\n',rating)
```

```
['23', '20', '33', '27', '24', '19', '25', '10', '21', '25', '23', '18', '26', '34', '25', '28', '20', '17', '32',
'28']
```

```
['2,714', '2,316', '3,807', '2,806', '2,426', '1,910', '2,451', '878', '1,682', '1,797', '1,163', '826', '1,184',
'1,254', '865', '813', '525', '344', '565', '106']
```

```
['118', '116', '115', '104', '101', '101', '98', '88', '80', '72', '51', '46', '46', '37', '35', '29', '26', '20',
'18', '4']
```

```
In [27]: 1 matches=matches[0:10]
2 matches
```

```
Out[27]: ['23', '20', '33', '27', '24', '19', '25', '10', '21', '25']
```

```
In [28]: 1 points=points[0:10]
2 print(points)
```

```
['2,714', '2,316', '3,807', '2,806', '2,426', '1,910', '2,451', '878', '1,682', '1,797']
```

```
In [29]: 1 rating=rating[0:10]
```

```
In [30]: 1 print(rating)
```

```
['118', '116', '115', '104', '101', '101', '98', '88', '80', '72']
```

In [31]:

```
1 #making into dataframe
2 top_10_odi=pd.DataFrame({"Rank":list(range(1,11)),"Team":team,"Match":matches,"Point":points,"Rating":rating})
3 top_10_odi
```

Out[31]:

| Rank | Team | Match | Point | Rating |
|------|----------------|-------|-------|--------|
| 0 | 1 Australia | 23 | 2,714 | 118 |
| 1 | 2 Pakistan | 20 | 2,316 | 116 |
| 2 | 3 India | 33 | 3,807 | 115 |
| 3 | 4 New Zealand | 27 | 2,806 | 104 |
| 4 | 5 England | 24 | 2,426 | 101 |
| 5 | 6 South Africa | 19 | 1,910 | 101 |
| 6 | 7 Bangladesh | 25 | 2,451 | 98 |
| 7 | 8 Afghanistan | 10 | 878 | 88 |
| 8 | 9 Sri Lanka | 21 | 1,682 | 80 |
| 9 | 10 West Indies | 25 | 1,797 | 72 |

3(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 [32]:

```
1 page_m_bat=requests.get("https://www.icc-cricket.com/rankings/mens/player-rankings/odi")
2 page_m_bat
```

Out[32]: <Response [200]>

```
In [33]: 1 soup_m_bat=BeautifulSoup(page_m_bat.content)
2 soup_m_bat
```

```
Out[33]: <!DOCTYPE html>
<html lang="en">
<head>
<meta content="ICC Men's ODI Player Rankings | ICC" name="twitter:title"/>
<meta content="website" property="og:type"/>
<meta content="summary_large_image" property="twitter:card"/>
<meta content="Official International Cricket Council rankings for ODI match cricket players. Discover latest ICC
rankings table, predict upcoming matches, see points and ratings for all teams." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official International Cricket Council rankings for ODI match cricket players. Discover latest ICC
rankings table, predict upcoming matches, see points and ratings for all teams." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-thumbnail.jpg" name="twitter:image"/>
<meta content="ICC Men's ODI Player Rankings | ICC" property="og:title"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-thumbnail.jpg" property="og:image"/>
<title>ICC Men's ODI Player Rankings | ICC</title>
<meta content="Official International Cricket Council rankings for ODI match cricket players. Discover latest ICC
rankings table, predict upcoming matches, see points and ratings for all teams." property="og:description"/>
<!-- Designed and built by Pulselive - www.pulselive.com -->
<script>
```

```
In [34]: 1 pos=[]
2 for i in soup_m_bat.find_all("span",class_="rankings-block__pos-number"):
3     pos.append(i.text.split("\n")[1].strip(" "))
4 print(pos)
```

```
['1', '1', '1', '1', '1', '1']
```

```
In [35]: 1 pos_bat=pos[0:1]
2 pos_ball=pos[1:2]
3 print(pos_bat)
4 print(pos_ball)
```

```
['1']
['1']
```

In []:

1

In [36]:

```
1 player=[]
2 for plyr in soup_m_bat.find_all("div",class_="rankings-block__banner--name"):
3     player.append(plyr.text)
4 player
```

Out[36]:

```
['Babar Azam',
 'Josh Hazlewood',
 'Shakib Al Hasan',
 'Babar Azam',
 'Josh Hazlewood',
 'Shakib Al Hasan']
```

In [37]:

```
1 player_bat=player[0:1]
2 player_ball=player[1:2]
3 print(player_bat)
4 print(player_ball)
```

```
['Babar Azam']
['Josh Hazlewood']
```

In [38]:

```
1 team=[]
2 rating=[]
3 for tm in soup_m_bat.find_all("div",class_="rankings-block__banner--nationality"):
4     team.append(tm.text.split("\n")[2])
5     rating.append(tm.text.split("\n")[3].strip(" "))
6 print(team)
7 print(rating)
8
```

```
['PAK', 'AUS', 'BAN', 'PAK', 'AUS', 'BAN']
['886', '705', '367', '886', '705', '367']
```

In [39]:

```
1 team_bat=team[0:1]
2 team_ball=team[1:2]
3 print(team_bat,"\\n\\n",team_ball)
```

['PAK']

['AUS']

In [40]:

```
1 rating_bat=rating[0:1]
2 rating_ball=rating[1:2]
3 print(rating_bat)
4 print(rating_ball)
```

['886']

['705']

In [41]:

```
1 a=soup_m_bat.find("td",class_="table-body__cell table-body__cell--position u-text-right")
2 a.text.split('\\n')[3].strip(" ")
```

Out[41]: '2'

In [42]:

```
1 x=[]
2 for i in soup_m_bat.find_all("td",class_="table-body__cell table-body__cell--position u-text-right"):
3     x.append(i.text.split('\\n')[3].strip(" "))
4 print(x)
```

```
['2', '3', '4', '5', '6', '7', '8', '9', '10', '2', '3', '4', '5', '6', '7', '8', '9', '10', '2', '3', '4', '5',
'6', '7', '8', '9', '10', '2', '3', '4', '5', '6', '7', '8', '9', '10', '2', '3', '4', '5', '6', '7', '8', '9', '1
0', '2', '3', '4', '5', '6', '7', '8', '9', '10']
```

In [43]:

```
1 x[0:9]
```

Out[43]: ['2', '3', '4', '5', '6', '7', '8', '9', '10']

```
In [44]: 1 for i in x[0:9]:  
2     pos_bat.append(i)  
3 print(pos_bat)
```

```
['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']
```

```
In [45]: 1 x[9:18]
```

```
Out[45]: ['2', '3', '4', '5', '6', '7', '8', '9', '10']
```

```
In [46]: 1 for i in x[9:18]:  
2     pos_ball.append(i)  
3 print(pos_ball)
```

```
['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']
```

```
In [47]: 1 b=soup_m_bat.find("td",class_="table-body__cell name")  
2 b.text.split('\n')[1]
```

```
Out[47]: 'Rassie van der Dussen'
```

```
In [48]: 1 player_list=[]  
2 for b in soup_m_bat.find_all("td",class_="table-body__cell name"):  
3     player_list.append(b.text.split('\n')[1])  
4 print(player_list)
```

```
['Rassie van der Dussen', 'Fakhar Zaman', 'Imam-ul-Haq', 'Shubman Gill', 'David Warner', 'Harry Tector', 'Virat Kohli', 'Quinton de Kock', 'Rohit Sharma', 'Mohammed Siraj', 'Mitchell Starc', 'Matt Henry', 'Trent Boult', 'Rashid Khan', 'Adam Zampa', 'Mujeeb Ur Rahman', 'Mohammad Nabi', 'Shaheen Afridi', 'Mohammad Nabi', 'Rashid Khan', 'Sikandar Raza', 'Zeeshan Maqsood', 'Assad Vala', 'Mehedi Hasan', 'Mitchell Santner', 'Wanindu Hasaranga', 'Chris Woakes', 'Rassie van der Dussen', 'Fakhar Zaman', 'Imam-ul-Haq', 'Shubman Gill', 'David Warner', 'Harry Tector', 'Virat Kohli', 'Quinton de Kock', 'Rohit Sharma', 'Mohammed Siraj', 'Mitchell Starc', 'Matt Henry', 'Trent Boult', 'Rashid Khan', 'Adam Zampa', 'Mujeeb Ur Rahman', 'Mohammad Nabi', 'Shaheen Afridi', 'Mohammad Nabi', 'Rashid Khan', 'Sikandar Raza', 'Zeeshan Maqsood', 'Assad Vala', 'Mehedi Hasan', 'Mitchell Santner', 'Wanindu Hasaranga', 'Chris Woakes']
```

In [49]:

```
1 for i in player_list[0:9]:  
2     player_bat.append(i)  
3 for i in player_list[9:18]:  
4     player_ball.append(i)  
5 print(player_bat)  
6 print(player_ball)
```

```
['Babar Azam', 'Rassie van der Dussen', 'Fakhar Zaman', 'Imam-ul-Haq', 'Shubman Gill', 'David Warner', 'Harry Tector', 'Virat Kohli', 'Quinton de Kock', 'Rohit Sharma']  
['Josh Hazlewood', 'Mohammed Siraj', 'Mitchell Starc', 'Matt Henry', 'Trent Boult', 'Rashid Khan', 'Adam Zampa', 'Mujeeb Ur Rahman', 'Mohammad Nabi', 'Shaheen Afridi']
```

In []:

```
1
```

In [50]:

```
1 c=soup_m_bat.find('span',class_="table-body__logo-text")  
2 c.text
```

Out[50]:

```
'SA'
```

In [51]:

```
1 team_list=[]  
2 for c in soup_m_bat.find_all('span',class_="table-body__logo-text"):  
3     team_list.append(c.text)  
4 print(team_list)
```

```
['SA', 'PAK', 'PAK', 'IND', 'AUS', 'IRE', 'IND', 'SA', 'IND', 'IND', 'AUS', 'NZ', 'NZ', 'AFG', 'AUS', 'AFG', 'AFG',  
'PAK', 'AFG', 'AFG', 'ZIM', 'OMA', 'PNG', 'BAN', 'NZ', 'SL', 'ENG', 'SA', 'PAK', 'PAK', 'IND', 'AUS', 'IRE', 'IND',  
'SA', 'IND', 'IND', 'AUS', 'NZ', 'NZ', 'AFG', 'AUS', 'AFG', 'AFG', 'PAK', 'AFG', 'AFG', 'ZIM', 'OMA', 'PNG', 'BAN',  
'NZ', 'SL', 'ENG']
```

In [52]:

```
1 for i in team_list[0:9]:  
2     team_bat.append(i)  
3 for i in team_list[9:18]:  
4     team_ball.append(i)  
5 print(team_bat)  
6 print(team_ball)
```

```
['PAK', 'SA', 'PAK', 'PAK', 'IND', 'AUS', 'IRE', 'IND', 'SA', 'IND']  
['AUS', 'IND', 'AUS', 'NZ', 'NZ', 'AFG', 'AUS', 'AFG', 'AFG', 'PAK']
```

```
In [53]: 1 d=soup_m_bat.find('td',class_="table-body__cell u-text-right rating")
2 d.text
```

Out[53]: '777'

```
In [54]: 1 rating_list=[]
2 for d in soup_m_bat.find_all('td',class_="table-body__cell u-text-right rating"):
3     rating_list.append(d.text)
4 print(rating_list)
```

```
['777', '755', '745', '738', '726', '722', '719', '718', '707', '691', '686', '667', '660', '659', '652', '637', '631', '630', '310', '280', '253', '252', '248', '244', '234', '216', '215', '777', '755', '745', '738', '726', '722', '719', '718', '707', '691', '686', '667', '660', '659', '652', '637', '631', '630', '310', '280', '253', '252', '248', '244', '234', '216', '215']
```

```
In [55]: 1 for i in rating_list[0:9]:
2     rating_bat.append(i)
3 for i in rating_list[9:18]:
4     rating_ball.append(i)
5 print(rating_bat)
6 print(rating_ball)
```

```
['886', '777', '755', '745', '738', '726', '722', '719', '718', '707']
['705', '691', '686', '667', '660', '659', '652', '637', '631', '630']
```

In [56]:

```
1 #now making dataframe
2 top_10_ODI_Batting_mens=pd.DataFrame({"POS":pos_bat,"Player":player_bat,"Team":team_bat,"Rating":rating_bat})
3 top_10_ODI_Batting_mens
```

Out[56]:

| | POS | Player | Team | Rating |
|---|-----|-----------------------|------|--------|
| 0 | 1 | Babar Azam | PAK | 886 |
| 1 | 2 | Rassie van der Dussen | SA | 777 |
| 2 | 3 | Fakhar Zaman | PAK | 755 |
| 3 | 4 | Imam-ul-Haq | PAK | 745 |
| 4 | 5 | Shubman Gill | IND | 738 |
| 5 | 6 | David Warner | AUS | 726 |
| 6 | 7 | Harry Tector | IRE | 722 |
| 7 | 8 | Virat Kohli | IND | 719 |
| 8 | 9 | Quinton de Kock | SA | 718 |
| 9 | 10 | Rohit Sharma | IND | 707 |

3(c) Top 10 ODI bowlers(Mens) along with the records

```
In [57]: 1 top_10_ODI_Bowler_mens=pd.DataFrame({"POS":pos_ball,"Player":player_ball,"Team":team_ball,"Rating":rating_ball})
          2 top_10_ODI_Bowler_mens
```

Out[57]:

| | POS | Player | Team | Rating |
|---|-----|------------------|------|--------|
| 0 | 1 | Josh Hazlewood | AUS | 705 |
| 1 | 2 | Mohammed Siraj | IND | 691 |
| 2 | 3 | Mitchell Starc | AUS | 686 |
| 3 | 4 | Matt Henry | NZ | 667 |
| 4 | 5 | Trent Boult | NZ | 660 |
| 5 | 6 | Rashid Khan | AFG | 659 |
| 6 | 7 | Adam Zampa | AUS | 652 |
| 7 | 8 | Mujeeb Ur Rahman | AFG | 637 |
| 8 | 9 | Mohammad Nabi | AFG | 631 |
| 9 | 10 | Shaheen Afridi | PAK | 630 |

4) Write a python program to scrape cricket rankings from icc-cricket.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.

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

```
In [58]: 1 page_w_odi=requests.get("https://www.icc-cricket.com/rankings/womens/team-rankings/odi")
          2 page_w_odi
```

Out[58]: <Response [200]>

```
In [59]: 1 soup_w_odi=BeautifulSoup(page_w_odi.content)
2 soup_w_odi
```

```
Out[59]: <!DOCTYPE html>
<html lang="en">
<head>
<meta content="ICC Women's ODI Team Rankings | ICC" name="twitter:title"/>
<meta content="website" property="og:type"/>
<meta content="summary_large_image" property="twitter:card"/>
<meta content="Official International Cricket Council rankings for test match cricket teams. Discover latest ICC r
ankings table, predict upcoming matches, see points and ratings for all teams." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official International Cricket Council rankings for test match cricket teams. Discover latest ICC r
ankings table, predict upcoming matches, see points and ratings for all teams." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-thumbnail.jpg" name="twitter:image"/>
<meta content="ICC Women's ODI Team Rankings | ICC" property="og:title"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-thumbnail.jpg" property="og:image"/>
<title>ICC Women's ODI Team Rankings | ICC</title>
<meta content="Official International Cricket Council rankings for test match cricket teams. Discover latest ICC r
ankings table, predict upcoming matches, see points and ratings for all teams." property="og:description"/>
<!-- Designed and built by Pulselive - www.pulselive.com -->
<script>
```

```
In [60]: 1 pos=[]
2 position=soup_w_odi.find("td",class_="rankings-block__banner--pos")
3 pos.append(position.text)
4 pos
```

```
Out[60]: ['1']
```

```
In [61]: 1 team=[]
2 t=soup_w_odi.find("td",class_="rankings-block__banner--team-name")
3 team.append(t.text.split('\n')[2])
4 team
```

```
Out[61]: ['Australia']
```

```
In [62]: 1 matches=[]
2 m=soup_w_odi.find("td",class_="rankings-block__banner--matches")
3 matches.append(m.text)
4 matches
```

```
Out[62]: ['21']
```

```
In [63]: 1 points=[]
2 p=soup_w_odi.find("td",class_="rankings-block__banner--points")
3 points.append(p.text)
4 points
```

```
Out[63]: ['3,603']
```

```
In [64]: 1 rating=[]
2 r=soup_w_odi.find('td',class_="rankings-block__banner--rating u-text-right")
3 rating.append(r.text.split('\n')[1].strip())
4 rating
```

```
Out[64]: ['172']
```

```
In [65]: 1 for i in soup_w_odi.find_all("td",class_="table-body__cell table-body__cell--position u-text-right"):
2     pos.append(i.text)
3 pos
```

```
Out[65]: ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13']
```

```
In [66]: 1 pos=pos[0:10]
2 pos
```

```
Out[66]: ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']
```

```
In [67]: 1 team=[]
2 for i in soup_w_odis.find_all("span",class_="u-hide-phablet"):
3     team.append(i.text)
4 team
```

```
Out[67]: ['Australia',
'England',
'South Africa',
'India',
'New Zealand',
'West Indies',
'Thailand',
'Bangladesh',
'Pakistan',
'Sri Lanka',
'Ireland',
'Netherlands',
'Zimbabwe']
```

```
In [68]: 1 team=team[0:10]
2 print(team)
```

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

```
In [69]: 1 t=[]
2 for i in soup_w_odis.find_all('td',class_="table-body__cell u-center-text"):
3     t.append(i.text)
4 print(t)
```

```
['28', '3,342', '26', '3,098', '27', '2,820', '25', '2,553', '27', '2,535', '11', '821', '14', '977', '27', '1,678',
'9', '479', '14', '548', '9', '0', '11', '0']
```

```
In [70]: 1 x=t[0:-1:2]
2 x
```

```
Out[70]: ['28', '26', '27', '25', '27', '11', '14', '27', '9', '14', '9', '11']
```

```
In [71]: 1 y=t[1:-1:2]
          2 print(y)
```

```
['3,342', '3,098', '2,820', '2,553', '2,535', '821', '977', '1,678', '479', '548', '0']
```

```
In [72]: 1 x=x[0:9]
          2 x
```

```
Out[72]: ['28', '26', '27', '25', '27', '11', '14', '27', '9']
```

```
In [73]: 1 y=y[0:9]
          2 y
```

```
Out[73]: ['3,342', '3,098', '2,820', '2,553', '2,535', '821', '977', '1,678', '479']
```

```
In [74]: 1 for i in x:
          2     matches.append(i)
          3 print(matches)
          4 for i in y:
          5     points.append(i)
          6 print(points)
```

```
['21', '28', '26', '27', '25', '27', '11', '14', '27', '9']
['3,603', '3,342', '3,098', '2,820', '2,553', '2,535', '821', '977', '1,678', '479']
```

```
In [75]: 1 for i in soup_w_odis.find_all("td", class_="table-body__cell u-text-right rating"):
          2     rating.append(i.text)
          3 print(rating)
```

```
['172', '119', '119', '104', '102', '94', '75', '70', '62', '53', '39', '0', '0']
```

```
In [76]: 1 rating=rating[0:10]
          2 print(rating)
```

```
['172', '119', '119', '104', '102', '94', '75', '70', '62', '53']
```

```
In [77]: 1 print(len(pos),len(team),len(matches),len(points),len(rating))
```

```
10 10 10 10 10
```

```
In [78]: 1 top_10_odi_womens_team=pd.DataFrame({"POS":pos,"Team":team,"Matches":matches,"Points":points,"Rating":rating})  
2 top_10_odi_womens_team
```

Out[78]:

| | POS | Team | Matches | Points | Rating |
|---|-----|--------------|---------|--------|--------|
| 0 | 1 | Australia | 21 | 3,603 | 172 |
| 1 | 2 | England | 28 | 3,342 | 119 |
| 2 | 3 | South Africa | 26 | 3,098 | 119 |
| 3 | 4 | India | 27 | 2,820 | 104 |
| 4 | 5 | New Zealand | 25 | 2,553 | 102 |
| 5 | 6 | West Indies | 27 | 2,535 | 94 |
| 6 | 7 | Thailand | 11 | 821 | 75 |
| 7 | 8 | Bangladesh | 14 | 977 | 70 |
| 8 | 9 | Pakistan | 27 | 1,678 | 62 |
| 9 | 10 | Sri Lanka | 9 | 479 | 53 |

4 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 [79]: 1 page_w=requests.get("https://www.icc-cricket.com/rankings/womens/player-rankings/odi")  
2 page_w
```

Out[79]: <Response [200]>

```
In [80]: 1 soup_w=BeautifulSoup(page_w.content)
          2 soup_w
```

```
Out[80]: <!DOCTYPE html>
<html lang="en">
<head>
<meta content="ICC Women's ODI Player Rankings | ICC" name="twitter:title"/>
<meta content="website" property="og:type"/>
<meta content="summary_large_image" property="twitter:card"/>
<meta content="Official International Cricket Council rankings for ODI match cricket players. Discover latest ICC
rankings table, predict upcoming matches, see points and ratings for all teams." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official International Cricket Council rankings for ODI match cricket players. Discover latest ICC
rankings table, predict upcoming matches, see points and ratings for all teams." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-thumbnail.jpg" name="twitter:image"/>
<meta content="ICC Women's ODI Player Rankings | ICC" property="og:title"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-thumbnail.jpg" property="og:image"/>
<title>ICC Women's ODI Player Rankings | ICC</title>
<meta content="Official International Cricket Council rankings for ODI match cricket players. Discover latest ICC
rankings table, predict upcoming matches, see points and ratings for all teams." property="og:description"/>
<!-- Designed and built by Pulselive - www.pulselive.com -->
<script>
```

```
In [81]: 1 pos=[]
          2 for i in soup_w.find_all("span",class_="rankings-block__pos-number"):
          3     pos.append(i.text.split('\n')[1].strip())
          4 pos
```

```
Out[81]: ['1', '1', '1']
```

```
In [82]: 1 pos_bat=pos[0:1]
          2 pos_all=pos[-1:-2:-1]
          3 print(pos_bat)
          4 print(pos_all)
```

```
['1']
['1']
```

```
In [83]: 1 player=[]
2 for i in soup_w.find_all("div",class_="rankings-block__banner--name"):
3     player.append(i.text)
4 player
```

```
Out[83]: ['Beth Mooney', 'Sophie Ecclestone', 'Hayley Matthews']
```

```
In [84]: 1 player_bat=player[0:1]
2 print(player_bat)
3 player_all=player[-1:-2:-1]
4 print(player_all)
```

```
['Beth Mooney']
['Hayley Matthews']
```

```
In [85]: 1 team=[]
2 rating=[]
3 for i in soup_w.find_all("div",class_="rankings-block__banner--nationality"):
4     team.append(i.text.split('\n')[2])
5     rating.append(i.text.split('\n')[3].strip())
6
7 print(team)
8 print(rating)
```

```
['AUS', 'ENG', 'WI']
['754', '751', '373']
```

In [86]:

```
1 team_bat=team[0:1]
2 print(team_bat)
3 team_all=team[-1:-2:-1]
4 print(team_all)
5 rating_bat=rating[0:1]
6 print(rating_bat)
7 rating_all=rating[-1:-2:-1]
8 print(rating_all)
```

```
['AUS']
['WI']
['754']
['373']
```

In [87]:

```
1 p=[]
2 for i in soup_w.find_all("span",class_="rankings-table__pos-number"):
3     p.append(i.text.split('\n')[1].strip())
4 print(p)
```

```
'2', '3', '4', '5', '6', '7', '8', '9', '10', '2', '3', '4', '5', '6', '7', '8', '9', '10', '2', '3', '4', '5',
'6', '7', '8', '9', '10']
```

In [88]:

```
1 for i in p[0:9]:
2     pos_bat.append(i)
3 for i in p[18:]:
4     pos_all.append(i)
5 print(pos_bat)
6 print(pos_all)
```

```
'1', '2', '3', '4', '5', '6', '7', '8', '9', '10'
['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']
```

In [89]:

```
1 plr=[]
2 for i in soup_w.find_all("td",class_="table-body__cell name"):
3     plr.append(i.text.split('\n')[1])
4 plr
```

Out[89]:

```
['Laura Wolvaardt',
 'Natalie Sciver',
 'Meg Lanning',
 'Harmanpreet Kaur',
 'Smriti Mandhana',
 'Chamari Athapaththu',
 'Ellyse Perry',
 'Tammy Beaumont',
 'Stafanie Taylor',
 'Jess Jonassen',
 'Shabnim Ismail',
 'Megan Schutt',
 'Hayley Matthews',
 'Kate Cross',
 'Ayabonga Khaka',
 'Rajeshwari Gayakwad',
 'Marizanne Kapp',
 'Deepti Sharma',
 'Natalie Sciver',
 'Ellyse Perry',
 'Marizanne Kapp',
 'Amelia Kerr',
 'Deepti Sharma',
 'Ashleigh Gardner',
 'Jess Jonassen',
 'Nida Dar',
 'Sophie Ecclestone']
```

In [90]:

```

1 for i in plr[0:9]:
2     player_bat.append(i)
3 print(player_bat)
4
5 for i in plr[18:]:
6     player_all.append(i)
7 print(player_all)

```

['Beth Mooney', 'Laura Wolvaardt', 'Natalie Sciver', 'Meg Lanning', 'Harmanpreet Kaur', 'Smriti Mandhana', 'Chamari Athapaththu', 'Ellyse Perry', 'Tammy Beaumont', 'Stafanie Taylor']
['Hayley Matthews', 'Natalie Sciver', 'Ellyse Perry', 'Marizanne Kapp', 'Amelia Kerr', 'Deepti Sharma', 'Ashleigh Gardner', 'Jess Jonassen', 'Nida Dar', 'Sophie Ecclestone']

In [91]:

```

1 tm=[]
2 for i in soup_w.find_all("span",class_="table-body__logo-text"):
3     tm.append(i.text)
4 print(tm)

```

['SA', 'ENG', 'AUS', 'IND', 'IND', 'SL', 'AUS', 'ENG', 'WI', 'AUS', 'SA', 'AUS', 'WI', 'ENG', 'SA', 'IND', 'SA', 'IND', 'ENG', 'AUS', 'SA', 'NZ', 'IND', 'AUS', 'AUS', 'PAK', 'ENG']

In [92]:

```

1 for i in tm[0:9]:
2     team_bat.append(i)
3 print(team_bat)
4
5 for i in tm[18:]:
6     team_all.append(i)
7 print(team_all)

```

['AUS', 'SA', 'ENG', 'AUS', 'IND', 'IND', 'SL', 'AUS', 'ENG', 'WI']
['WI', 'ENG', 'AUS', 'SA', 'NZ', 'IND', 'AUS', 'AUS', 'PAK', 'ENG']

In [93]:

```

1 rt=[]
2 for i in soup_w.find_all('td',class_="table-body__cell u-text-right rating"):
3     rt.append(i.text)
4 print(rt)

```

'732', '731', '717', '716', '714', '673', '626', '595', '588', '723', '722', '704', '660', '655', '634', '617', '598', '589', '371', '366', '349', '336', '322', '292', '250', '232', '205']

In [94]:

```
1 for i in rt[0:9]:  
2     rating_bat.append(i)  
3 print(rating_bat)  
4  
5 for i in rt[18:]:  
6     rating_all.append(i)  
7 print(rating_all)
```

```
['754', '732', '731', '717', '716', '714', '673', '626', '595', '588']  
['373', '371', '366', '349', '336', '322', '292', '250', '232', '205']
```

In [95]:

```
1 print(len(pos_bat),len(player_bat),len(team_bat),len(rating_bat))  
2 print(len(pos_all),len(player_all),len(team_all),len(rating_all))
```

```
10 10 10 10  
10 10 10 10
```

In [96]:

```
1 top_10_odi_women_bating_player=pd.DataFrame({'POS':pos_bat,"Player":player_bat,"Team":team_bat,"Rating":rating_ba  
2 top_10_odi_women_bating_player
```

Out[96]:

| | POS | Player | Team | Rating |
|---|-----|---------------------|------|--------|
| 0 | 1 | Beth Mooney | AUS | 754 |
| 1 | 2 | Laura Wolvaardt | SA | 732 |
| 2 | 3 | Natalie Sciver | ENG | 731 |
| 3 | 4 | Meg Lanning | AUS | 717 |
| 4 | 5 | Harmanpreet Kaur | IND | 716 |
| 5 | 6 | Smriti Mandhana | IND | 714 |
| 6 | 7 | Chamari Athapaththu | SL | 673 |
| 7 | 8 | Ellyse Perry | AUS | 626 |
| 8 | 9 | Tammy Beaumont | ENG | 595 |
| 9 | 10 | Stafanie Taylor | WI | 588 |

```
In [97]: 1 top_10_odi_women_all_rounder=pd.DataFrame({"POS":pos_all,"Player":player_all,"Team":team_all,"Rating":rating_all})
          2 top_10_odi_women_all_rounder
```

Out[97]:

| | POS | Player | Team | Rating |
|---|-----|-------------------|------|--------|
| 0 | 1 | Hayley Matthews | WI | 373 |
| 1 | 2 | Natalie Sciver | ENG | 371 |
| 2 | 3 | Ellyse Perry | AUS | 366 |
| 3 | 4 | Marizanne Kapp | SA | 349 |
| 4 | 5 | Amelia Kerr | NZ | 336 |
| 5 | 6 | Deepti Sharma | IND | 322 |
| 6 | 7 | Ashleigh Gardner | AUS | 292 |
| 7 | 8 | Jess Jonassen | AUS | 250 |
| 8 | 9 | Nida Dar | PAK | 232 |
| 9 | 10 | Sophie Ecclestone | ENG | 205 |

5) Write a python program to scrape mentioned news details from <https://www.cnbc.com/world/?region=world> (<https://www.cnbc.com/world/?region=world>) and make data frame-

- i) Headline
- ii) Time
- iii) News Link

```
In [98]: 1 page_news=requests.get("https://www.cnbc.com/world/?region=world")
          2 page_news
```

Out[98]: <Response [200]>

```
In [99]: 1 soup_news=BeautifulSoup(page_news.content)
2 soup_news
```

```
Out[99]: <!DOCTYPE html>
<html itemscope="" itemtype="https://schema.org/WebPage" lang="en" prefix="og=https://ogp.me/ns#><head><meta cont
ent="telephone=no" name="format-detection"/><style type="text/css">@charset "UTF-8";.RecaptchaAcknowledgement-ackn
owledgement{color:#747474;flex:1;font-size:11px;font-weight:600;line-height:15px;margin-bottom:7px;margin-top:24p
x;width:100%}.RecaptchaAcknowledgement-acknowledgement a{color:#747474;font-weight:500;text-decoration:none}.Recap
tchaAcknowledgement-acknowledgement a:hover{color:#747474;text-decoration:underline}.RecaptchaAcknowledgement-ackn
owledgement a:active{color:#747474}.RecaptchaAcknowledgement-reCaptchaPadding{margin-top:15px}.RecaptchaAcknowledg
ement-centerAligned{text-align:center}.RecaptchaAcknowledgement-leftAligned{text-align:left}.RecaptchaAcknowledgem
ent-rightAligned{text-align:right}.AuthForms-container{margin:0 auto;padding:0 10px;width:458px}@media (max-width:
759px){.AuthForms-container{max-width:458px;padding:20px 0 0;width:100%}}.AuthForms-container .AuthForms-signupCon
tainer{margin:0 auto;padding:0 41px;text-align:center;width:458px}@media (max-width:759px){.AuthForms-container .A
uthForms-signupContainer{padding:20px 0 0;width:100%}}.AuthForms-container .AuthForms-header{color:#171717;margin-
bottom:20px}.AuthForms-container .AuthForms-createAccountHeader{color:#171717;margin-bottom:20px;text-align:cente
r}.AuthForms-container .AuthForms-headerTitle{color:#071d39;font-size:20px;font-weight:700;text-align:center}.Auth
Forms-container .AuthForms-signinContent{font-size:14px;font-weight:450;padding-top:20px}.AuthForms-container .Aut
hForms-toggleCta{color:#424242;font-size:12px;font-weight:500;text-align:right}.AuthForms-container .AuthForms-ins
tructions{margin-bottom:20px;text-align:left}.AuthForms-container .AuthForms-instructions p{color:#171717;font-siz
e:14px;line-height:16px}.AuthForms-container .AuthForms-instructions p a{color:#005594;text-decoration:none}.AuthF
orms-container .AuthForms-instructions p a:hover{color:#002f6c}.AuthForms-container .AuthForms-formInputContainer
```

In [100]:

```
1 link=[]
2 for i in soup_news.find_all("a",class_="LatestNews-headline"):
3     link.append(i.get("href"))
4 link
```

```
Out[100]: ['https://www.cnbc.com/2023/05/26/marvell-technology-shares-surge-after-earnings-beat-heres-how-to-play-it.html',
'https://www.cnbc.com/2023/05/26/tech-stocks-are-back-driven-by-ai-craze-slowing-rate-hikes.html',
'https://www.cnbc.com/2023/05/26/next-week-hints-at-short-lived-debt-deal-relief-as-jobs-report-looms-.html',
'https://www.cnbc.com/2023/05/26/treasury-says-it-wont-run-out-money-until-at-least-june-5-buying-time-for-debt-ceiling-talks.html',
'https://www.cnbc.com/2023/05/26/disney-rips-desantis-bid-to-disqualify-judge.html',
'https://www.cnbc.com/2023/05/26/the-pause-on-student-loan-payments-has-been-a-win-for-public-servants.html',
'https://www.cnbc.com/2023/05/26/facebook-giphy-sale-shows-how-fear-of-regulators-is-slowing-ma-market.html',
'https://www.cnbc.com/2023/05/26/job-cuts-jpmorgan-chase-cut-about-500-tech-and-ops-jobs-.html',
'https://www.cnbc.com/2023/05/26/the-right-age-for-kids-to-be-on-social-media-experts-say.html',
'https://www.cnbc.com/2023/05/26/ai-excitement-leads-to-winning-week-for-nvidia-and-other-tech-stocks.html',
'https://www.cnbc.com/2023/05/26/nvidia-shares-jumped-25percent-this-week-and-got-cheaper-heres-why.html',
'https://www.cnbc.com/2023/05/26/needham-this-stock-plays-the-almost-perfect-marriage-between-ai-crypto.html',
'https://www.cnbc.com/2023/05/26/chatgpt-machine-learning-and-other-popular-ai-terms-to-know.html',
'https://www.cnbc.com/2023/05/26/taylor-swift-beyonc-metallica-summer-2023s-top-10-hottest-artist-.html',
'https://www.cnbc.com/2023/05/26/investors-shifted-into-these-gold-and-small-cap-etfs-this-week.html',
'https://www.cnbc.com/2023/05/26/ford-tesla-ev-charging-deal-puts-pressure-on-gm.html',
'https://www.cnbc.com/2023/05/26/paramount-shares-pop-after-bdt-capital-bets-on-the-media-giants-key-shareholder.html',
'https://www.cnbc.com/2023/05/26/stocks-making-the-biggest-moves-midday-ford-marvell-technology-paramount-gap-more.html',
'https://www.cnbc.com/2023/05/26/many-companies-offer-tuition-assistance-for-workers-to-go-to-college.html',
'https://www.cnbc.com/2023/05/26/stocked-up-on-short-term-t-bills-heres-why-it-might-be-time-to-buy-longer-term-bonds.html',
'https://www.cnbc.com/2023/05/26/club-name-ford-teams-up-with-tesla-sending-shares-soaring-.html',
'https://www.cnbc.com/2023/05/26/how-turo-cmo-andrew-mok-got-comfortable-embracing-his-asian-identity-at-work.html',
'https://www.cnbc.com/2023/05/26/biden-advances-renewable-energy-transmission-projects-in-nevada.html',
'https://www.cnbc.com/2023/05/26/morgan-stanley-upgrades-this-mining-stock-that-can-surge-more-than-20percent.html',
'https://www.cnbc.com/2023/05/26/apple-and-nvidia-are-in-my-top-5-holdings-portfolio-still-diversified-enough.html',
'https://www.cnbc.com/2023/05/26/marvell-shares-soar-25percent-after-semiconductor-firm-beats-on-earnings.html',
'https://www.cnbc.com/2023/05/26/jpmorgan-ceo-jamie-dimon-epstein-deposition.html',
'https://www.cnbc.com/2023/05/26/bank-of-america-hits-spacecraft-builder-terran-orbital-with-rare-double-downgrade.html',
'https://www.cnbc.com/2023/05/26/house-passes-bill-blocking-student-debt-forgiveness.html',
'https://www.cnbc.com/2023/05/26/employers-are-learning-gen-z-isnt-the-easiest-generation-to-work-with.html']
```

```
In [101]: 1 print(len(link))
```

30

```
In [102]: 1 headlines=[]
 2 for i in soup_news.find_all("a",class_="LatestNews-headline"):
 3     headlines.append(i.text)
 4 headlines
```

```
Out[102]: ["Marvell Technology shares surge after earnings beat. Here's how to play it",
 'The tech trade is back, driven by A.I. craze and prospect of slowing Fed hikes',
 'Next week hints at only short-lived debt deal relief as jobs report looms',
 'Treasury says it could run out of money June 5, buying time for debt ceiling talks',
 'Disney rips DeSantis bid to disqualify judge in free speech lawsuit',
 'Why the pause on student loan payments has been a win for public servants',
 'Facebook-Giphy sale shows how fear of regulators is slowing M&A market',
 'JPMorgan Chase cut about 500 jobs this week, including tech and operations roles',
 "Is there a 'right' age for kids to be on social media? Here's what an expert says",
 'A.I. excitement leads to a winning week for Nvidia and other tech stocks',
 "Nvidia shares jumped 25% this week – and got cheaper. Here's how that happens",
 "Needham says this stock plays the 'almost perfect marriage' between A.I., crypto",
 'AI is the latest buzzword in tech—but before investing, know these 4 terms',
 'Taylor Swift to Metallica: Top 10 most in-demand artists of summer',
 'Investors shifted into these gold and small-cap ETFs this week',
 "Ford's EV charging deal with Tesla puts pressure on GM, other rival automakers",
 "Paramount shares pop after BDT Capital bets on the media giant's key shareholder",
 'Stocks making the biggest moves midday: Ford, Marvell Technology, Gap and more',
 'Companies add, expand tuition assistance so workers can go back to college',
 'Here's why it might be time to buy longer-term bonds now',
 'Club name Ford teams up with Tesla, sending shares soaring',
 'How this CMO got comfortable embracing his Asian identity at work',
 'Biden Interior advances renewable energy transmission projects in Nevada',
 'Morgan Stanley upgrades this mining stock that can surge more than 20%',
 'Apple and Nvidia are in my top 5 holdings. Am I still diversified enough?',
 'Marvell shares soar 32% after the chip firm beats on top and bottom line results',
 'JPMorgan CEO Dimon testifies he had no involvement with Epstein account, bank says',
 'BofA hits spacecraft builder Terran Orbital with rare double downgrade',
 'House passes bill blocking student debt forgiveness—what borrowers need to know',
 'Companies are learning Gen Z isn't the easiest generation to work with']
```

```
In [103]: 1 print(len(headlines))
```

```
30
```

```
In [104]: 1 time=[]
2 for i in soup_news.find_all("time",class_="LatestNews-timestamp"):
3     time.append(i.text)
4 print(time)
```

```
['3 Hours Ago', '4 Hours Ago', '4 Hours Ago', '4 Hours Ago', '5 Hours Ago', '5 Hours Ago', '6 Hours Ago', '6 Hours Ago', '6 Hours Ago', '6 Hours Ago', '7 Hours Ago', '7 Hours Ago', '7 Hours Ago', '7 Hours Ago', '8 Hours Ago', '9 Hours Ago', '10 Hours Ago', '10 Hours Ago']
```

```
In [105]: 1 print(len(time))
```

```
30
```

```
In [106]:  
1 news_df=pd.DataFrame({"Headlines":headlines,"Link":link,"Time":time})  
2 news_df
```

Out[106]:

| | Headlines | Link | Time |
|----|---|---|-------------|
| 0 | Marvell Technology shares surge after earnings... | https://www.cnbc.com/2023/05/26/marvell-techno... | 3 Hours Ago |
| 1 | The tech trade is back, driven by A.I. craze a... | https://www.cnbc.com/2023/05/26/tech-stocks-ar... | 4 Hours Ago |
| 2 | Next week hints at only short-lived debt deal ... | https://www.cnbc.com/2023/05/26/next-week-hint... | 4 Hours Ago |
| 3 | Treasury says it could run out of money June 5... | https://www.cnbc.com/2023/05/26/treasury-says-... | 4 Hours Ago |
| 4 | Disney rips DeSantis bid to disqualify judge i... | https://www.cnbc.com/2023/05/26/disney-rips-de... | 5 Hours Ago |
| 5 | Why the pause on student loan payments has bee... | https://www.cnbc.com/2023/05/26/the-pause-on-s... | 5 Hours Ago |
| 6 | Facebook-Giphy sale shows how fear of regulato... | https://www.cnbc.com/2023/05/26/facebook-giph... | 6 Hours Ago |
| 7 | JPMorgan Chase cut about 500 jobs this week, i... | https://www.cnbc.com/2023/05/26/job-cuts-jpmor... | 6 Hours Ago |
| 8 | Is there a 'right' age for kids to be on socia... | https://www.cnbc.com/2023/05/26/the-right-age-... | 6 Hours Ago |
| 9 | A.I. excitement leads to a winning week for Nv... | https://www.cnbc.com/2023/05/26/ai-excitement-... | 6 Hours Ago |
| 10 | Nvidia shares jumped 25% this week — and got c... | https://www.cnbc.com/2023/05/26/nvidia-shares-... | 7 Hours Ago |
| 11 | Needham says this stock plays the 'almost perf... | https://www.cnbc.com/2023/05/26/needham-this-s... | 7 Hours Ago |
| 12 | AI is the latest buzzword in tech—but before i... | https://www.cnbc.com/2023/05/26/chatgpt-machin... | 7 Hours Ago |
| 13 | Taylor Swift to Metallica: Top 10 most in-dema... | https://www.cnbc.com/2023/05/26/taylor-swift-b... | 7 Hours Ago |
| 14 | Investors shifted into these gold and small-ca... | https://www.cnbc.com/2023/05/26/investors-shif... | 8 Hours Ago |
| 15 | Ford's EV charging deal with Tesla puts pressu... | https://www.cnbc.com/2023/05/26/ford-tesla-ev-... | 8 Hours Ago |
| 16 | Paramount shares pop after BDT Capital bets on... | https://www.cnbc.com/2023/05/26/paramount-shar... | 8 Hours Ago |
| 17 | Stocks making the biggest moves midday: Ford, ... | https://www.cnbc.com/2023/05/26/stocks-making-... | 8 Hours Ago |
| 18 | Companies add, expand tuition assistance so wo... | https://www.cnbc.com/2023/05/26/many-companies... | 8 Hours Ago |
| 19 | Here's why it might be time to buy longer-term... | https://www.cnbc.com/2023/05/26/stocked-up-on-... | 8 Hours Ago |
| 20 | Club name Ford teams up with Tesla, sending sh... | https://www.cnbc.com/2023/05/26/club-name-ford... | 9 Hours Ago |
| 21 | How this CMO got comfortable embracing his Asi... | https://www.cnbc.com/2023/05/26/how-turo-cmo-a... | 9 Hours Ago |
| 22 | Biden Interior advances renewable energy trans... | https://www.cnbc.com/2023/05/26/biden-advances... | 9 Hours Ago |
| 23 | Morgan Stanley upgrades this mining stock tha... | https://www.cnbc.com/2023/05/26/morgan-stanley... | 9 Hours Ago |
| 24 | Apple and Nvidia are in my top 5 holdings. Am ... | https://www.cnbc.com/2023/05/26/apple-and-nvid... | 9 Hours Ago |

| | Headlines | Link | Time |
|----|---|---|--------------|
| 25 | Marvell shares soar 32% after the chip firm be... | https://www.cnbc.com/2023/05/26/marvell-shares... | 9 Hours Ago |
| 26 | JPMorgan CEO Dimon testifies he had no involve... | https://www.cnbc.com/2023/05/26/jpmorgan-ceo-j... | 9 Hours Ago |
| 27 | BofA hits spacecraft builder Terran Orbital wi... | https://www.cnbc.com/2023/05/26/bank-of-americ... | 9 Hours Ago |
| 28 | House passes bill blocking student debt forgiv... | https://www.cnbc.com/2023/05/26/house-passes-b... | 10 Hours Ago |
| 29 | Companies are learning Gen Z isn't the easiest... | https://www.cnbc.com/2023/05/26/employers-are-... | 10 Hours Ago |

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>
[\(https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-articles\)](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 [107]: 1 page_j=requests.get("https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-articles")
           2 page_j
```

Out[107]: <Response [200]>

```
In [108]: 1 soup_j=BeautifulSoup(page_j.content)
2 soup_j
```

```
Out[108]: <!DOCTYPE html>
<html><head><meta charset="utf-8"/><meta content="width=device-width" name="viewport"/><meta content="en_US" name="og:locale"/><meta content="Most Downloaded Articles - Artificial Intelligence - Journal - Elsevier" property="og:title"/><meta content="The journal of Artificial Intelligence (AIJ) welcomes papers on broad aspects of AI that constitute advances in the overall field including, but not limited ..." property="og:description"/><meta content="http://ars.els-cdn.com/content/image/X00043702.jpg" name="og:image" property="og:image"/><meta content="http://ars.els-cdn.com/content/image/X00043702.jpg" name="og:image:url" property="og:image:url"/><meta content="https://ars.els-cdn.com/content/image/X00043702.jpg" name="og:image:secure_url" property="og:image:secure_url"/><meta content="journals.elsevier.com/artificial-intelligence/most-downloaded-articles" name="og:url"/><meta content="website" property="og:type"/><link href="/apple-touch-icon.png" rel="apple-touch-icon" sizes="180x180"/><link href="/favicon-32x32.png" rel="icon" sizes="32x32" type="image/png"/><link href="/favicon-16x16.png" rel="icon" sizes="16x16" type="image/png"/><link color="#ff6c00" href="/safari-pinned-tab.svg" rel="mask-icon"/><title>Most Downloaded Articles - Artificial Intelligence - Journal - Elsevier</title><meta content="The journal of Artificial Intelligence (AIJ) welcomes papers on broad aspects of AI that constitute advances in the overall field including, but not limited ..." name="description"/><meta content="Most Downloaded Articles - Artificial Intelligence - Journal - Elsevier" name="twitter:title"/><meta content="The journal of Artificial Intelligence (AIJ) welcomes papers on broad aspects of AI that constitute advances in the overall field including, but not limited ..." name="twitter:description"/><meta content="http://ars.els-cdn.com/content/image/X00043702.jpg" name="twitter:image"/><meta content="summary" name="twitter:card"/><script type="application/ld+json">{@context:"https://schema.org";@ty
```

In [109]:

```
1 title=[]
2 auther=[]
3 date=[]
4 url=[]
5
6 for i in soup_j.find_all("h2",class_="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg"):
7     title.append(i.text)
8
9 for i in soup_j.find_all("span",class_="sc-1w3fpd7-0 dnCnAO"):
10    auther.append(i.text)
11
12 for i in soup_j.find_all("span",class_="sc-1thf9ly-2 dvggWt"):
13    date.append(i.text)
14
15 for i in soup_j.find_all("a",class_="sc-5smygv-0 fIXTHm"):
16    url.append(i.get("href"))
17
18
19 print("title: \n",title)
20 print("auther: \n",auther)
21 print("date: \n",date)
22 print("URL : \n",url)
```

title:

['Reward is enough', 'Explanation in artificial intelligence: Insights from the social sciences', 'Creativity and a rtificial 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 w hen 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 comparis on 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 bet ween AI models and healthcare professionals: Explainability, utility and trust in AI-driven clinical decision-makin g', 'Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-r ates in XAI user studies', 'The Hanabi challenge: A new frontier for AI research', 'Wrappers for feature subset sele ction', '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 Lovela ce in the digital age', 'Robot ethics: Mapping the issues for a mechanized world', 'Reward (Mis)design for autonomou s driving', 'Planning and acting in partially observable stochastic domains', 'What do we want from Explainable Arti ficial Intelligence (XAI)? - A stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI r esearch']

author:

['David Silver, Satinder Singh, Doina Precup, Richard S. Sutton ', 'Tim Miller ', 'Margaret A. Boden ', 'Guni Sharo n, Roni Stern, Ariel Felner, Nathan R. Sturtevant ', 'Ilaria Tiddi, Stefan Schlobach ', 'Henry Prakken, Giovanni Sar tor ', 'Richard S. Sutton, Doina Precup, Satinder Singh ', 'Kjersti Aas, Martin Jullum, Anders Løland ', 'Wenhan Lu o, Junliang Xing and 4 more', 'Saurabh Arora, Prashant Doshi ', 'Jasper van der Waa, Elisabeth Nieuwburg, Anita Crem ers, Mark Neerincx ', 'Joe Collenette, Katie Atkinson, Trevor Bench-Capon ', 'Roel Dobbe, Thomas Krendl Gilbert, Yon atan 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, Ke ith Abney, George Bekey ', 'W. Bradley Knox, Alessandro Allievi and 3 more', 'Leslie Pack Kaelbling, Michael L. Litt man, Anthony R. Cassandra ', 'Markus Langer, Daniel Oster and 6 more']

date:

['October 2021', 'February 2019', 'August 1998', 'February 2015', 'January 2022', 'October 2015', 'August 1999', 'S eptember 2021', 'April 2021', 'August 2021', 'February 2021', 'April 2023', 'November 2021', 'March 2023', 'May 202 1', 'March 2020', 'December 1997', 'June 2017', 'June 2021', 'June 2016', 'April 2011', 'March 2023', 'May 1998', 'J uly 2021']

URL :

['<https://www.sciencedirect.com/science/article/pii/S0004370221000862>', '<https://www.sciencedirect.com/science/article/pii/S0004370218305988>', '[https://www.scie nedirect.com/science/article/pii/S0004370298000551](https://www.sciencedirect.com/science/article/pii/S0004370298000551)', '<https://www.sciedirect.com/science/article/pii/S0004370214001386>', '<https://www.sciencedirect.com/science/article/pii/S0004370221001788>', '[https://www.sciencedirect.com/scien ce/article/pii/S0004370215000910](https://www.sciencedirect.com/science/article/pii/S0004370215000910)', '[https://www.sciencedirect.com/scien ce/article/pii/S0004370299000521](https://www.sciencedirect.com/science/article/pii/S0004370299000521)', '<https://www.sciencedirect.com/science/article/pii/S0004370220301958>', '[https://w ww.sciencedirect.com/science/article/pii/S0004370221000539](https://www.sciencedirect.com/science/article/pii/S0004370221000539)', '<https://www.sciencedirect.com/science/article/pii/S00043702220301958>']

```
4370221000515', 'https://www.sciencedirect.com/science/article/pii/S0004370220301533', 'https://www.sciencedirect.co  
m/science/article/pii/S0004370223000073', 'https://www.sciencedirect.com/science/article/pii/S0004370221001065', 'ht  
tps://www.sciencedirect.com/science/article/pii/S0004370222001795', 'https://www.sciencedirect.com/science/article/p  
ii/S0004370221000102', 'https://www.sciencedirect.com/science/article/pii/S0004370219300116', 'https://www.sciencedi  
rect.com/science/article/pii/S000437029700043X', 'https://www.sciencedirect.com/science/article/pii/S000437021630079  
0', 'https://www.sciencedirect.com/science/article/pii/S0004370221000096', 'https://www.sciencedirect.com/science/ar  
ticle/pii/S0004370216300224', 'https://www.sciencedirect.com/science/article/pii/S0004370211000178', 'https://www.sc  
iencedirect.com/science/article/pii/S0004370222001692', 'https://www.sciencedirect.com/science/article/pii/S00043702  
9800023X', 'https://www.sciencedirect.com/science/article/pii/S0004370221000242']
```

In [110]: 1 print(len(title),len(auth),len(date),len(url))

```
24 24 24 24
```

```
In [111]: 1 articles_df=pd.DataFrame({"Pater Title":title,"Authors":auther,"Published Date":date,"Paper URL":url})  
          2 articles_df
```

Out[111]:

| | Paper Title | Authors | Published Date | Paper URL |
|----|---|---|----------------|---|
| 0 | Reward is enough | David Silver, Satinder Singh, Doina Precup, Ri... | October 2021 | https://www.sciencedirect.com/science/article/... |
| 1 | Explanation in artificial intelligence: Insights... | Tim Miller | February 2019 | https://www.sciencedirect.com/science/article/... |
| 2 | Creativity and artificial intelligence | Margaret A. Boden | August 1998 | https://www.sciencedirect.com/science/article/... |
| 3 | Conflict-based search for optimal multi-agent ... | Guni Sharon, Roni Stern, Ariel Felner, Nathan ... | February 2015 | https://www.sciencedirect.com/science/article/... |
| 4 | Knowledge graphs as tools for explainable mach... | Ilaria Tiddi, Stefan Schlobach | January 2022 | https://www.sciencedirect.com/science/article/... |
| 5 | Law and logic: A review from an argumentation ... | Henry Prakken, Giovanni Sartor | October 2015 | https://www.sciencedirect.com/science/article/... |
| 6 | Between MDPs and semi-MDPs: A framework for te... | Richard S. Sutton, Doina Precup, Satinder Singh | August 1999 | https://www.sciencedirect.com/science/article/... |
| 7 | Explaining individual predictions when feature... | Kjersti Aas, Martin Jullum, Anders Løland | September 2021 | https://www.sciencedirect.com/science/article/... |
| 8 | Multiple object tracking: A literature review | Wenhan Luo, Junliang Xing and 4 more | April 2021 | https://www.sciencedirect.com/science/article/... |
| 9 | A survey of inverse reinforcement learning: Ch... | Saurabh Arora, Prashant Doshi | August 2021 | https://www.sciencedirect.com/science/article/... |
| 10 | Evaluating XAI: A comparison of rule-based and... | Jasper van der Waa, Elisabeth Nieuwburg, Anita... | February 2021 | https://www.sciencedirect.com/science/article/... |
| 11 | Explainable AI tools for legal reasoning about... | Joe Collenette, Katie Atkinson, Trevor Bench-C... | April 2023 | https://www.sciencedirect.com/science/article/... |
| 12 | Hard choices in artificial intelligence | Roel Dobbe, Thomas Krendl Gilbert, Yonatan Mintz | November 2021 | https://www.sciencedirect.com/science/article/... |
| 13 | Assessing the communication gap between AI mod... | Oskar Wysocki, Jessica Katharine Davies and 5 ... | March 2023 | https://www.sciencedirect.com/science/article/... |
| 14 | Explaining black-box classifiers using post-ho... | Eoin M. Kenny, Courtney Ford, Molly Quinn, Mar... | May 2021 | https://www.sciencedirect.com/science/article/... |
| 15 | The Hanabi challenge: A new frontier for AI re... | Nolan Bard, Jakob N. Foerster and 13 more | March 2020 | https://www.sciencedirect.com/science/article/... |
| 16 | Wrappers for feature subset selection | Ron Kohavi, George H. John | December 1997 | https://www.sciencedirect.com/science/article/... |

| | Paper Title | Authors | Published Date | Paper URL |
|----|---|--|----------------|---|
| 17 | Artificial cognition for social human–robot interaction | Séverin Lemaignan, Mathieu Warnier and 3 more | June 2017 | https://www.sciencedirect.com/science/article/pii/S092658081730030X |
| 18 | A review of possible effects of cognitive bias on human–robot interaction | Tomáš Kliegr, Štěpán Bahník, Johannes Fürnkranz | June 2021 | https://www.sciencedirect.com/science/article/pii/S092658082100017X |
| 19 | The multifaceted impact of Ada Lovelace in the field of robotics | Luigia Carlucci Aiello | June 2016 | https://www.sciencedirect.com/science/article/pii/S092658081630017X |
| 20 | Robot ethics: Mapping the issues for a mechanistic approach | Patrick Lin, Keith Abney, George Bekey | April 2011 | https://www.sciencedirect.com/science/article/pii/S092658081100001X |
| 21 | Reward (Mis)design for autonomous driving | W. Bradley Knox, Alessandro Allievi and 3 more | March 2023 | https://www.sciencedirect.com/science/article/pii/S092658082300001X |
| 22 | Planning and acting in partially observable stochastic environments | Leslie Pack Kaelbling, Michael L. Littman, Andrew W. Moore | May 1998 | https://www.sciencedirect.com/science/article/pii/S092658089800001X |
| 23 | What do we want from Explainable Artificial Intelligence? | Markus Langer, Daniel Oster and 6 more | July 2021 | https://www.sciencedirect.com/science/article/pii/S092658082100001X |

7) Write a python program to scrape mentioned details from dineout.co.in and make data frame-

- i) Restaurant name
- ii) Cuisine
- iii) Location
- iv) Ratings
- v) Image URL

```
In [112]: 1 page_r=requests.get("https://www.dineout.co.in/delhi-restaurants/buffet-special")
2 page_r
```

Out[112]: <Response [200]>

```
In [113]: 1 soup_r=BeautifulSoup(page_r.content)
2 soup_r
```

```
Out[113]: <!DOCTYPE html>
<html lang="en"><head><meta charset="utf-8"/><meta content="IE=edge" http-equiv="X-UA-Compatible"/><meta content
="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" name="viewport"/><link href="/manife
st.json" rel="manifest"/><style type="text/css">
    @font-face {
        font-family: 'dineicon';
        src: url('/fonts/dineicon.eot');
        src: url('/fonts/dineicon.eot#iefix') format('embedded-opentype'),
        url('/fonts/dineicon.ttf') format('truetype'),
        url('/fonts/dineicon.woff') format('woff'),
        url('/fonts/dineicon.svg#dineicon') format('svg');
        font-weight: normal;
        font-style: normal;
        font-display: swap;
    }
    .hide {
        display: none !important;
    }
    .async-hide{
        .. . . . .
```

```
In [114]: 1 name=[]
2 for i in soup_r.find_all("a",class_="restnt-name ellipsis"):
3     name.append(i.text)
4 name
```

```
Out[114]: ['Castle Barbeque',
'Jungle Jamboree',
'Cafe Knosh',
"Castle's Barbeque",
'The Barbeque Company',
'India Grill',
'Delhi Barbeque',
'The Monarch - Bar Be Que Village',
'Indian Grill Room']
```

In [115]:

```
1 cuisine=[]
2 for i in soup_r.find_all("span",class_="double-line-ellipsis"):
3     cuisine.append(i.text.split('|')[1])
4 cuisine
```

Out[115]:

```
['Chinese, North Indian',
 'North Indian, Asian, Italian',
 'Italian, Continental',
 'Chinese, North Indian',
 'North Indian, Chinese',
 'North Indian, Italian',
 'North Indian',
 'North Indian',
 'North Indian, Mughlai']
```

In [116]:

```
1 location=[]
2 for i in soup_r("div",class_="restnt-loc ellipsis"):
3     location.append(i.text)
4 location
```

Out[116]:

```
['Connaught Place, Central Delhi',
 '3CS Mall,Lajpat Nagar - 3, South Delhi',
 'The Leela Ambience Convention Hotel,Shahdara, East Delhi',
 'Pacific Mall,Tagore Garden, West Delhi',
 'Gardens Galleria,Sector 38A, Noida',
 'Hilton Garden Inn,Saket, South Delhi',
 'Taurus Sarovar Portico,Mahipalpur, South Delhi',
 'Indirapuram Habitat Centre,Indirapuram, Ghaziabad',
 'Suncity Business Tower,Golf Course Road, Gurgaon']
```

In [117]:

```
1 rating=[]
2 for i in soup_r.find_all("div",class_="restnt-rating rating-4"):
3     rating.append(i.text)
4 rating
```

Out[117]:

```
['4', '3.9', '4.3', '3.9', '3.9', '3.9', '3.9', '3.7', '3.8', '4.3']
```

In [118]:

```
1 url=[]
2 for i in soup_r.find_all("img",class_="no-img"):
3     url.append(i.get("data-src"))
4 url
```

Out[118]:

```
['https://im1.dineout.co.in/images/uploads/restaurant/sharpen/8/k/b/p86792-16062953735fbe1f4d3fb7e.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/p/m/p59633-166088382462ff137009010.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/4/p/m/p406-15438184745c04cce491bc.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/j/o/p38113-15959192065f1fcbb666130c.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/7/p/k/p79307-16051787755fad1597f2bf9.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/2/v/t/p2687-1482477169585cce712b90f.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/d/i/p52501-1661855212630de5eceb6d2.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/n/o/p34822-15599107305cfaf594a13c24.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/y/f/p549-165000147262590640c0afc.jpg?tr=tr:n-medium']
```

In [119]:

```
1 print(len(name),len(cuisine),len(location),len(rating),len(url))
```

9 9 9 9 9

```
In [120]: 1 restaurants_df=pd.DataFrame({"Restaurant name":name,"Cuisine":cuisine,"Location":location,"Ratings":rating,"Image":image})
2 restaurants_df
```

Out[120]: