BeautifulSoup Assignment

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

```
In [1]:  !pip install bs4
         !pip install requests
```

Requirement already satisfied: bs4 in c:\users\dell\anaconda3\lib\site-pac
kages (0.0.1)
Requirement already satisfied: beautifulsoup4 in c:\users\dell\anaconda3\l
ib\site-packages (from bs4) (4.11.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\dell\anaconda3\li
b\site-packages (from beautifulsoup4->bs4) (2.3.2.post1)
Requirement already satisfied: requests in c:\users\dell\anaconda3\lib\sit
e-packages (2.28.1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\dell\a
naconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\dell\anac
onda3\lib\site-packages (from requests) (1.26.14)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dell\anaconda3\lib
\site-packages (from requests) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\anacond
a3\lib\site-packages (from requests) (2022.12.7)

```
In [2]:  from bs4 import BeautifulSoup
         import requests
```

```
In [3]:  page = requests.get('https://en.wikipedia.org/wiki/Main_Page')
```

```
In [4]:  page
```

Out[4]: <Response [200]>

In [5]:
```python
soup = BeautifulSoup(page.content)
soup
```

["",""],
"wgDefaultDateFormat":"dmy","wgMonthNames":["","January","February","Ma
rch","April","May","June","July","August","September","October","Novemb
er","December"],"wgRequestId":"92739914-b967-4a57-8c3b-b563bf5047a5","w
gCanonicalNamespace":"","wgCanonicalSpecialPageName":false,"wgNamespace
Number":0,"wgPageName":"Main_Page","wgTitle":"Main Page","wgCurRevision
Id":1114291180,"wgRevisionId":1114291180,"wgArticleId":15580374,"wgIsAr
ticle":true,"wgIsRedirect":false,"wgAction":"view","wgUserName":null,"w
gUserGroups":["*"],"wgCategories":[],"wgPageViewLanguage":"en","wgPageC
ontentLanguage":"en","wgPageContentModel":"wikitext","wgRelevantPageNam
e":"Main_Page","wgRelevantArticleId":15580374,"wgIsProbablyEditable":fa
lse,"wgRelevantPageIsProbablyEditable":false,"wgRestrictionEdit":["syso
p"],"wgRestrictionMove":["sysop"],"wgIsMainPage":true,"wgNoticeProjec
t":"wikipedia","wgFlaggedRevsParams":{"tags":{"status":{"levels":
1}}},"wgMediaViewerOnClick":true,"wgMediaViewerEnabledByDefault":tru
e,"wgPopupsFlags"
:10,"wgVisualEditor":{"pageLanguageCode":"en","pageLanguageDir":"lt
r","pageVariantFallbacks":"en"},"wgMFDisplayWikibaseDescriptions":{"sea
rch":true,"watchlist":true,"tagline":false,"nearby":true},"wgWMESchemaE
ditAttemptStepOversample":false,"wgWMEPageLength":3000,"wgULSCurrentAut

In [6]:
```python
Headers = []

for i in soup.find_all('span',class_="mw-headline"):

    Headers.append(i.text)

Headers
```

Out[6]:
```
['Welcome to Wikipedia',
 "From today's featured article",
 'Did you know\xa0...',
 'In the news',
 'On this day',
 "From today's featured list",
 "Today's featured picture",
 'Other areas of Wikipedia',
 "Wikipedia's sister projects",
 'Wikipedia languages']
```

In [7]:
```python
import pandas as pd
```

In [8]:
```python
df = pd.DataFrame({"Headings ":Headers})
```

In [9]:
```python
df
```

Out[9]:

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

Question 2: Write s python program to display list of respected former presidents of India(i.e. Name , Term ofoffice) from [https://presidentofindia.nic.in/former-presidents.htm (https://presidentofindia.nic.in/former-presidents.htm)](https://presidentofindia.nic.in/former-presidents.htm) and make data frame.

In [10]:
```python
url = requests.get('https://presidentofindia.nic.in/former-presidents.htm')
url
```

Out[10]: <Response [404]>

Question 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.

In [11]:
```python
url = requests.get('https://www.icc-cricket.com/rankings/mens/team-rankings
url
```

Out[11]: <Response [200]>

In [12]:
```python
Content = BeautifulSoup(url.content)
Content
```

```
<title>ICC Men's ODI Team Rankings | ICC</title>
<meta content="Official International Cricket Council ranking for One D
ay International (ODI) cricket teams. Discover latest ICC rankings tabl
e, predict upcoming matches, see points and ratings for all teams." pro
perty="og:description"/>
<!-- Designed and built by Pulselive - www.pulselive.com -->
<script>

    var dataLayer = [{
        'user_language':'English',
        'web_domain':'www.icc-cricket.com',
        'page_type':'/rankings/mens/team-rankings/odi'
    }];

</script>
<!-- Google Tag Manager -->
<script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)
[0],
j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=
```

In [13]:
```python
Teams =[]

for i in Content.find_all('span',class_="u-hide-phablet"):
    Teams.append(i.text)
Teams
```

Out[13]:
```
['India',
 'Australia',
 'Pakistan',
 'South Africa',
 'New Zealand',
 'England',
 'Sri Lanka',
 'Bangladesh',
 'Afghanistan',
 'West Indies',
 'Zimbabwe',
 'Scotland',
 'Ireland',
 'Netherlands',
 'Nepal',
 'Namibia',
 'United States',
 'Oman',
 'UAE']
```

In [14]:
```python
matches =[]

for i in Content.find_all('td',class_="rankings-block__banner--matches"):
    matches.append(i.text)
matches
```

Out[14]: ['49']

```
In [15]: points =[]

         for i in Content.find_all('td',class_="rankings-block__banner--points"):
             points.append(i.text)
         points
```

Out[15]: ['5,839']

```
In [16]: rating = []

         for i in Content.find_all('td',class_="rankings-block__banner--rating u-tex
             rating.append(i.text.replace('\n',''))
         rating
```

Out[16]: ['   119   ']

```
In [17]: matches =[]

         for i in Content.find_all('a',href_="/rankings/mens/player-rankings/3761"):
             matches.append(i.text)
         matches
```

Out[17]: []

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

```
In [18]: page = requests.get('https://www.icc-cricket.com/rankings/mens/player-ranki
         page
```

Out[18]: <Response [200]>

```
In [19]: tag = BeautifulSoup(page.content)
         tag
```

```
ault-thumbnail.jpg" name="twitter:image"/>
<meta content="ICC Men's ODI Batting | Player Rankings | ICC" property
="og:title"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/def
ault-thumbnail.jpg" property="og:image"/>
<title>ICC Men's ODI Batting | Player Rankings | ICC</title>
<meta content="Official ICC Cricket website - live matches, scores, new
s, highlights, commentary, rankings, videos and fixtures from the Inter
national Cricket Council." property="og:description"/>
<!-- Designed and built by Pulselive - www.pulselive.com -->
<script>

    var dataLayer = [{
        'user_language':'English',
        'web_domain':'www.icc-cricket.com',
        'page_type':'/rankings/mens/player-rankings/odi/batting'
    }];

</script>
<!-- Google Tag Manager -->
```

In [20]:
```python
plname = []

for i in tag.find_all('td',class_="table-body__cell rankings-table__name na
    plname.append(i.text.replace('\n',''))
plname
```

```
Out[20]: ['Shubman Gill',
          'Quinton de Kock',
          'Heinrich Klaasen',
          'David Warner',
          'Virat Kohli',
          'Harry Tector',
          'Rohit Sharma',
          'Rassie van der Dussen',
          'Imam-ul-Haq',
          'Dawid Malan',
          'Shai Hope',
          'Daryl Mitchell',
          'Fakhar Zaman',
          'David Miller',
          'Rahmanullah Gurbaz',
          'Steve Smith',
          'Ibrahim Zadran',
          'Jos Buttler',
          'Lokesh Rahul',
          'Temba Bavuma',
          'Kariyawasa Asalanka',
          'Aiden Markram',
          'Mushfiqur Rahim',
          'Paul Stirling',
          'Jason Roy',
          'Pathum Nissanka',
          'Scott Edwards',
          'Sikandar Raza',
          'Marnus Labuschagne',
          'Gerhard Erasmus',
          'Shreyas Iyer',
          'Jonny Bairstow',
          'Nicholas Pooran',
          'Joe Root',
          'Kane Williamson',
          'Tom Latham',
          'Mohammad Rizwan',
          'Travis Head',
          'Devon Conway',
          'Ishan Kishan',
          'Rahmat Shah',
          'Kusal Mendis',
          'Shakib Al Hasan',
          'Sean Williams',
          'Alex Carey',
          'Mitchell Marsh',
          'Ben Stokes',
          'Litton Das',
          'Glenn Maxwell',
          'Tamim Iqbal',
          'Mahmudullah',
          'Assad Vala',
          'George Munsey',
          'Hashmatullah Shaidi',
          'Andrew Balbirnie',
          'Vriitya Aravind',
          'Will Young',
          'Gajanand Singh',
          'Aaron Jones',
          'Kyle Coetzer',
          "Max O'Dowd",
```

```
 'Aqib Ilyas',
 'Muhammad Waseem',
 'Sadeera Samarawickrama',
 'Richard Berrington',
 'Monank Patel',
 'Martin Guptill',
 'Shikhar Dhawan',
 'Zeeshan Maqsood',
 'Michael Leask',
 'Nazmul Shanto',
 'Aasif Sheikh',
 'Hardik Pandya',
 'Najibullah Zadran',
 'Asif Khan',
 'Ayaan Khan',
 'Henry Nicholls',
 'Mohammad Nabi',
 'Rachin Ravindra',
 'Kashyap Prajapati',
 'Craig Ervine',
 'Steven Taylor',
 'R.K. Paudel',
 'George Dockrell',
 'Brandon King',
 'Michael van Lingen',
 'Jatinder Singh',
 'Haris Sohail',
 'Vikramjit Singh',
 'Brandon McMullen',
 'Bas de Leede',
 'Iftikhar Ahmed',
 'Janneman Malan',
 'Curtis Campher',
 'Ryan Burl',
 'Dhananjaya de Silva',
 'Dimuth Karunaratne',
 'Kushal Bhurtel',
 'Tony Ura']
```

In [21]:
```python
team = []

for i in tag.find_all('span',class_="table-body__logo-text"):
    team.append(i.text)
team
```

```
Out[21]: ['IND',
          'SA',
          'SA',
          'AUS',
          'IND',
          'IRE',
          'IND',
          'SA',
          'PAK',
          'ENG',
          'WI',
          'NZ',
          'PAK',
          'SA',
          'AFG',
          'AUS',
          'AFG',
          'ENG',
          'IND',
          'SA',
          'SL',
          'SA',
          'BAN',
          'IRE',
          'ENG',
          'SL',
          'NED',
          'ZIM',
          'AUS',
          'NAM',
          'IND',
          'ENG',
          'WI',
          'ENG',
          'NZ',
          'NZ',
          'PAK',
          'AUS',
          'NZ',
          'IND',
          'AFG',
          'SL',
          'BAN',
          'ZIM',
          'AUS',
          'AUS',
          'ENG',
          'BAN',
          'AUS',
          'BAN',
          'BAN',
          'PNG',
          'SCO',
          'AFG',
          'IRE',
          'UAE',
          'NZ',
          'USA',
          'USA',
          'SCO',
          'NED',
```

```
'OMA',
'UAE',
'SL',
'SCO',
'USA',
'NZ',
'IND',
'OMA',
'SCO',
'BAN',
'NEP',
'IND',
'AFG',
'UAE',
'OMA',
'NZ',
'AFG',
'NZ',
'OMA',
'ZIM',
'USA',
'NEP',
'IRE',
'WI',
'NAM',
'OMA',
'PAK',
'NED',
'SCO',
'NED',
'PAK',
'SA',
'IRE',
'ZIM',
'SL',
'SL',
'NEP',
'PNG']
```

In [22]:
```python
rate = []

for i in tag.find_all('td',class_="table-body__cell rating"):
    rate.append(i.text)
rate
```

```
Out[22]:  ['823',
           '769',
           '756',
           '747',
           '747',
           '729',
           '725',
           '716',
           '704',
           '699',
           '677',
           '668',
           '654',
           '651',
           '649',
           '647',
           '646',
           '637',
           '637',
           '635',
           '632',
           '632',
           '627',
           '623',
           '617',
           '615',
           '615',
           '615',
           '615',
           '614',
           '607',
           '605',
           '601',
           '600',
           '596',
           '592',
           '588',
           '587',
           '583',
           '582',
           '573',
           '573',
           '567',
           '566',
           '560',
           '554',
           '551',
           '548',
           '548',
           '544',
           '543',
           '536',
           '529',
           '525',
           '521',
           '518',
           '514',
           '512',
           '510',
           '506',
           '505',
```

```
        '501',
        '500',
        '498',
        '496',
        '493',
        '493',
        '489',
        '487',
        '484',
        '480',
        '479',
        '479',
        '478',
        '476',
        '476',
        '475',
        '471',
        '465',
        '464',
        '464',
        '462',
        '461',
        '460',
        '458',
        '457',
        '453',
        '452',
        '446',
        '445',
        '444',
        '444',
        '443',
        '440',
        '437',
        '437',
        '437',
        '432',
        '431']
```

In [23]:
```python
df =pd.DataFrame({'Batsman':plname,'Team':team,'Rating':rate})
df.head(10)
```

Out[23]:

	Batsman	Team	Rating
0	Shubman Gill	IND	823
1	Quinton de Kock	SA	769
2	Heinrich Klaasen	SA	756
3	David Warner	AUS	747
4	Virat Kohli	IND	747
5	Harry Tector	IRE	729
6	Rohit Sharma	IND	725
7	Rassie van der Dussen	SA	716
8	Imam-ul-Haq	PAK	704
9	Dawid Malan	ENG	699

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

In [24]:
```python
response = requests.get('https://www.icc-cricket.com/rankings/mens/player-r
response
```

Out[24]: `<Response [200]>`

In [25]:
```python
page = BeautifulSoup(response.content)
page
```

Out[25]:
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta content="ICC Men's ODI Bowling | Player Rankings | ICC" name="twi
tter:title"/>
<meta content="website" property="og:type"/>
<meta content="summary_large_image" property="twitter:card"/>
<meta content="Official ICC Cricket website - live matches, scores, new
s, highlights, commentary, rankings, videos and fixtures from the Inter
national Cricket Council." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official ICC Cricket website - live matches, scores, new
s, highlights, commentary, rankings, videos and fixtures from the Inter
national Cricket Council." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/def
ault-thumbnail.jpg" name="twitter:image"/>
<meta content="ICC Men's ODI Bowling | Player Rankings | ICC" property
="og:title"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/def
```

In [26]:
```python
bowler = []

for i in page.find_all('td',class_="table-body__cell rankings-table__name n
    bowler.append(i.text.replace('\n',''))

bowler
```

```
Out[26]:  ['Mohammed Siraj',
           'Keshav Maharaj',
           'Rashid Khan',
           'Trent Boult',
           'Mohammad Nabi',
           'Adam Zampa',
           'Matt Henry',
           'Kuldeep Yadav',
           'Shaheen Afridi',
           'Mujeeb Ur Rahman',
           'Mitchell Starc',
           'Jasprit Bumrah',
           'Kagiso Rabada',
           'Mark Watt',
           'Lungi Ngidi',
           'Bernard Scholtz',
           'Maheesh Theekshana',
           'Saurabh Netravalkar',
           'Chris Woakes',
           'Shakib Al Hasan',
           'Mohammad Shami',
           'Akeal Hosein',
           'Mitchell Santner',
           'Reece Topley',
           'Alzarri Joseph',
           'Andy McBrine',
           'Haris Rauf',
           'Jofra Archer',
           'Sandeep Lamichhane',
           'Ravindra Jadeja',
           'Mustafizur Rahman',
           'Mark Adair',
           'Pat Cummins',
           'Adil Rashid',
           'Mehedi Hasan',
           'Richard Ngarava',
           'Tabraiz Shamsi',
           'Christopher Sole',
           'Ruben Trumpelmann',
           'Aayan Afzal Khan',
           'Marco Jansen',
           'Bilal Khan',
           'David Willey',
           'Logan van Beek',
           'Wanindu Hasaranga',
           'Lachlan Ferguson',
           'Mohammad Nawaz',
           'Mark Wood',
           'Craig Young',
           'Zeeshan Maqsood',
           'Blessing Muzarabani',
           'Kaleemullah',
           'Fazalhaq Farooqi',
           'Nosthush Kenjige',
           'Aryan Dutt',
           'Shardul Thakur',
           'Safyaan Sharif',
           'Rohan Mustafa',
           'Sean Abbott',
           'Sikandar Raza',
           'Joshua Little',
```

```
                'Assad Vala',
                'Taskin Ahmed',
                'Chad Soper',
                'Hardik Pandya',
                'Dipendra Airee',
                'Fred Klaassen',
                'Zahoor Khan',
                'Naseem Shah',
                'Moeen Ali',
                'Shadab Khan',
                'Junaid Siddique',
                'Michael Leask',
                'Jason Holder',
                'Dushmantha Chameera',
                'Ish Sodhi',
                'Yuzvendra Chahal',
                'Tim Southee',
                'Dhananjaya de Silva',
                'Mohammad Shoriful Islam',
                'Tangeni Lungameni',
                'Taijul Islam',
                'Dilshan Madushanka',
                'Hasan Ali',
                'Karan KC',
                'Brandon McMullen',
                'Anrich Nortje',
                'Simi Singh',
                'Sam Curran',
                'Dunith Wellalage',
                'Gudakesh Motie-Kanhai',
                'Glenn Maxwell',
                'Nisarg Patel',
                'Sompal Kami',
                'Kasun Rajitha',
                'Lalit Rajbanshi',
                'Ashton Agar',
                'Mohammad Wasim',
                'Gulbadin Naib']
```

In [27]:
```python
Tname = []

for i in page.find_all('span',class_="table-body__logo-text"):
    Tname.append(i.text)

Tname
```

```
Out[27]: ['IND',
          'SA',
          'AFG',
          'NZ',
          'AFG',
          'AUS',
          'NZ',
          'IND',
          'PAK',
          'AFG',
          'AUS',
          'IND',
          'SA',
          'SCO',
          'SA',
          'NAM',
          'SL',
          'USA',
          'ENG',
          'BAN',
          'IND',
          'WI',
          'NZ',
          'ENG',
          'WI',
          'IRE',
          'PAK',
          'ENG',
          'NEP',
          'IND',
          'BAN',
          'IRE',
          'AUS',
          'ENG',
          'BAN',
          'ZIM',
          'SA',
          'SCO',
          'NAM',
          'UAE',
          'SA',
          'OMA',
          'ENG',
          'NED',
          'SL',
          'NZ',
          'PAK',
          'ENG',
          'IRE',
          'OMA',
          'ZIM',
          'OMA',
          'AFG',
          'USA',
          'NED',
          'IND',
          'SCO',
          'UAE',
          'AUS',
          'ZIM',
          'IRE',
```

```
                          'PNG',
                          'BAN',
                          'PNG',
                          'IND',
                          'NEP',
                          'NED',
                          'UAE',
                          'PAK',
                          'ENG',
                          'PAK',
                          'UAE',
                          'SCO',
                          'WI',
                          'SL',
                          'NZ',
                          'IND',
                          'NZ',
                          'SL',
                          'BAN',
                          'NAM',
                          'BAN',
                          'SL',
                          'PAK',
                          'NEP',
                          'SCO',
                          'SA',
                          'IRE',
                          'ENG',
                          'SL',
                          'WI',
                          'AUS',
                          'USA',
                          'NEP',
                          'SL',
                          'NEP',
                          'AUS',
                          'PAK',
                          'AFG']
```

In [28]:
```python
Rating = []

for i in page.find_all('td',class_="table-body__cell rating"):
    Rating.append(i.text)

Rating
```

```
Out[28]: ['668',
          '656',
          '654',
          '653',
          '641',
          '635',
          '634',
          '632',
          '625',
          '625',
          '622',
          '620',
          '620',
          '611',
          '598',
          '589',
          '586',
          '586',
          '576',
          '573',
          '564',
          '563',
          '563',
          '558',
          '558',
          '545',
          '541',
          '539',
          '539',
          '537',
          '531',
          '529',
          '528',
          '526',
          '519',
          '518',
          '515',
          '509',
          '508',
          '500',
          '500',
          '499',
          '498',
          '489',
          '488',
          '488',
          '488',
          '488',
          '487',
          '483',
          '482',
          '481',
          '480',
          '479',
          '478',
          '477',
          '477',
          '473',
          '472',
          '468',
          '465',
```

```
        '464',
        '463',
        '462',
        '459',
        '458',
        '449',
        '446',
        '446',
        '441',
        '440',
        '439',
        '431',
        '429',
        '427',
        '427',
        '424',
        '424',
        '422',
        '421',
        '421',
        '421',
        '419',
        '418',
        '418',
        '417',
        '417',
        '416',
        '415',
        '413',
        '411',
        '411',
        '409',
        '409',
        '406',
        '399',
        '398',
        '396',
        '391']
```

In [29]:
```python
df1 = pd.DataFrame({'Bowlers Name':bowler,'Team Name':Tname,'Rating':Rating
df1.head(10)
```

Out[29]:

	Bowlers Name	Team Name	Rating
0	Mohammed Siraj	IND	668
1	Keshav Maharaj	SA	656
2	Rashid Khan	AFG	654
3	Trent Boult	NZ	653
4	Mohammad Nabi	AFG	641
5	Adam Zampa	AUS	635
6	Matt Henry	NZ	634
7	Kuldeep Yadav	IND	632
8	Shaheen Afridi	PAK	625
9	Mujeeb Ur Rahman	AFG	625

Question 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.

In [30]:
```python
page = requests.get('https://www.icc-cricket.com/rankings/womens/team-ranki
page
```

Out[30]: <Response [200]>

In [31]:
```python
soup2 = BeautifulSoup(page.content)
soup2
```

Out[31]:
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta content="ICC Women's ODI Team Rankings | ICC" name="twitter:titl
e"/>
<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 rankings table, predict upcomi
ng 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 rankings table, predict upcomi
ng matches, see points and ratings for all teams." name="twitter:descri
ption"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/def
ault-thumbnail.jpg" name="twitter:image"/>
<meta content="ICC Women's ODI Team Rankings | ICC" property="og:titl
e"/>
```

```
In [32]: Tname = []

         for i in soup2.find_all('span',class_="u-hide-phablet"):
             Tname.append(i.text)
         Tname
```

```
Out[32]: ['Australia',
          'England',
          'South Africa',
          'India',
          'New Zealand',
          'West Indies',
          'Sri Lanka',
          'Bangladesh',
          'Thailand',
          'Pakistan',
          'Ireland',
          'Netherlands',
          'Zimbabwe']
```

Top 10 women's ODI Batting players along with the records of their team and rating

```
In [33]: page = requests.get('https://www.icc-cricket.com/rankings/womens/player-ran
         page
```

```
Out[33]: <Response [200]>
```

```
In [34]: soup3 = BeautifulSoup(page.content)
         soup3
```

```
Out[34]: <!DOCTYPE html>
         <html lang="en">
         <head>
         <meta content="ICC Women's ODI Batting | Player Rankings | ICC" name="t
         witter:title"/>
         <meta content="website" property="og:type"/>
         <meta content="summary_large_image" property="twitter:card"/>
         <meta content="Official ICC Cricket website - live matches, scores, new
         s, highlights, commentary, rankings, videos and fixtures from the Inter
         national Cricket Council." name="description"/>
         <meta content="@icc" property="twitter:site"/>
         <meta content="Official ICC Cricket website - live matches, scores, new
         s, highlights, commentary, rankings, videos and fixtures from the Inter
         national Cricket Council." name="twitter:description"/>
         <meta content="https://www.icc-cricket.com/resources/ver/i/elements/def
         ault-thumbnail.jpg" name="twitter:image"/>
         <meta content="ICC Women's ODI Batting | Player Rankings | ICC" propert
         y="og:title"/>
         <meta content="https://www.icc-cricket.com/resources/ver/i/elements/def
```

In [35]:
```python
Pname = []

for i in soup3.find_all('td',class_="table-body__cell rankings-table__name
    Pname.append(i.text.replace('\n',''))
Pname
```

```
Out[35]: ['Beth Mooney',
          'Chamari Athapaththu',
          'Laura Wolvaardt',
          'Smriti Mandhana',
          'Alyssa Healy',
          'Ellyse Perry',
          'Harmanpreet Kaur',
          'Meg Lanning',
          'Marizanne Kapp',
          'Amelia Kerr',
          'Heather Knight',
          'Tammy Beaumont',
          'Chloe-Lesleigh Tryon',
          'Ashleigh Gardner',
          'Hayley Matthews',
          'Stafanie Taylor',
          'Fargana Hoque',
          'Danielle Wyatt',
          'Sophie Devine',
          'Bismah Maroof',
          'Gaby Lewis',
          'Nattakan Chantam',
          'Suzie Bates',
          'Sidra Ameen',
          'Deepti Sharma',
          'Sune Luus',
          'Amy Jones',
          'Aliya Riaz',
          'Laura Delany',
          'Maddie Green',
          'Nida Dar',
          'Nigar Sultana',
          'Tahlia McGrath',
          'Orla Prendergast',
          'Leah Paul',
          'Annabel Sutherland',
          'Sophia Dunkley',
          'Yastika Bhatia',
          'Amy Hunter',
          'Naruemol Chaiwai',
          'Umaima Sohail',
          'Phoebe Litchfield',
          'Shafali Verma',
          'Emma Lamb',
          'Rumana Ahmed',
          'Brooke Halliday',
          'Babette de Leede',
          'Harshitha Madavi',
          'Muneeba Ali',
          'Lara Goodall',
          'Pooja Vastrakar',
          'Nilakshi Silva',
          'Harleen Deol',
          'Kavisha Dilhari',
          'Jess Jonassen',
          'Lauren Down',
          'Jemimah Rodriques',
          'Hasini Perera',
          'Sharne Mayers',
          'Kathryn Bryce',
          'Georgia Wareham',
```

```
        'Nannapat Koncharoenkai',
        'Salma Khatun',
        'Tazmin Brits',
        'Trisha Chetty',
        'Sharmin Supta',
        'Shemaine Campbelle',
        'Richa Ghosh',
        'Nadine Klerk',
        'Andrie Steyn',
        'Charlotte Dean',
        'Heather Siegers',
        'Chinelle Henry',
        'Chedean Nation',
        'Lata Mondal',
        'Prasadani Weerakkody',
        'Shamima Sultana',
        'Rashada Williams',
        'Ritu Moni',
        'Robine Rijke',
        'Josephine Nkomo',
        'Kycia Knight',
        'Mary Waldron',
        'Arlene Kelly',
        'Sophie Ecclestone',
        'Alice Capsey',
        'Fatima Sana',
        'Kate Cross',
        'Hannah Rowe',
        'Rosenan Kanoh',
        'Georgina Dempsey',
        'Jess Kerr',
        'Eimear Richardson',
        'Frederique Overdijk',
        'Murshida Khatun',
        'Meghna Singh',
        'Aaliyah Alleyne',
        'Maia Bouchier',
        'Sadaf Shamas']
```

```python
In [36]: Tname = []

         for i in soup3.find_all('span',class_="table-body__logo-text"):
             Tname.append(i.text)
         Tname
```

```
Out[36]:  ['AUS',
           'SL',
           'SA',
           'IND',
           'AUS',
           'AUS',
           'IND',
           'AUS',
           'SA',
           'NZ',
           'ENG',
           'ENG',
           'SA',
           'AUS',
           'WI',
           'WI',
           'BAN',
           'ENG',
           'NZ',
           'PAK',
           'IRE',
           'THA',
           'NZ',
           'PAK',
           'IND',
           'SA',
           'ENG',
           'PAK',
           'IRE',
           'NZ',
           'PAK',
           'BAN',
           'AUS',
           'IRE',
           'IRE',
           'AUS',
           'ENG',
           'IND',
           'IRE',
           'THA',
           'PAK',
           'AUS',
           'IND',
           'ENG',
           'BAN',
           'NZ',
           'NED',
           'SL',
           'PAK',
           'SA',
           'IND',
           'SL',
           'IND',
           'SL',
           'AUS',
           'NZ',
           'IND',
           'SL',
           'ZIM',
           'SCO',
           'AUS',
```

```
'THA',
'BAN',
'SA',
'SA',
'BAN',
'WI',
'IND',
'SA',
'SA',
'ENG',
'NED',
'WI',
'WI',
'BAN',
'SL',
'BAN',
'WI',
'BAN',
'NED',
'ZIM',
'WI',
'IRE',
'IRE',
'ENG',
'ENG',
'PAK',
'ENG',
'NZ',
'THA',
'IRE',
'NZ',
'IRE',
'NED',
'BAN',
'IND',
'WI',
'ENG',
'PAK']
```

```python
In [37]: rate= []

         for i in soup3.find_all('td',class_="table-body__cell rating"):
             rate.append(i.text)
         rate
```

Out[37]: ['750',
          '736',
          '727',
          '708',
          '698',
          '697',
          '694',
          '662',
          '642',
          '628',
          '597',
          '596',
          '582',
          '579',
          '574',
          '569',
          '565',
          '553',
          '553',
          '541',
          '536',
          '535',
          '531',
          '528',
          '515',
          '503',
          '502',
          '493',
          '488',
          '463',
          '445',
          '439',
          '431',
          '425',
          '418',
          '417',
          '414',
          '413',
          '412',
          '412',
          '387',
          '384',
          '382',
          '380',
          '379',
          '377',
          '375',
          '375',
          '375',
          '366',
          '363',
          '352',
          '349',
          '348',
          '343',
          '326',
          '326',
          '325',
          '324',
          '324',
          '308',

```
 '304',
 '304',
 '296',
 '296',
 '294',
 '294',
 '293',
 '293',
 '293',
 '292',
 '288',
 '286',
 '286',
 '285',
 '284',
 '281',
 '274',
 '268',
 '263',
 '260',
 '256',
 '255',
 '253',
 '253',
 '247',
 '245',
 '245',
 '243',
 '240',
 '234',
 '228',
 '227',
 '225',
 '224',
 '215',
 '214',
 '211',
 '210']
```

In [38]:
```python
df = pd.DataFrame({'Player Name':Pname,'Team Name':Tname,'Rating':rate})
df.head(10)
```

Out[38]:

	Player Name	Team Name	Rating
0	Beth Mooney	AUS	750
1	Chamari Athapaththu	SL	736
2	Laura Wolvaardt	SA	727
3	Smriti Mandhana	IND	708
4	Alyssa Healy	AUS	698
5	Ellyse Perry	AUS	697
6	Harmanpreet Kaur	IND	694
7	Meg Lanning	AUS	662
8	Marizanne Kapp	SA	642
9	Amelia Kerr	NZ	628

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

In [39]:
```python
url = requests.get('https://www.icc-cricket.com/rankings/womens/player-rank
url
```

Out[39]: `<Response [200]>`

In [40]:
```python
soup4 = BeautifulSoup(url.content)
soup4
```

Out[40]:
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta content="ICC Women's ODI All Rounder | Player Rankings | ICC" nam
e="twitter:title"/>
<meta content="website" property="og:type"/>
<meta content="summary_large_image" property="twitter:card"/>
<meta content="Official ICC Cricket website - live matches, scores, new
s, highlights, commentary, rankings, videos and fixtures from the Inter
national Cricket Council." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official ICC Cricket website - live matches, scores, new
s, highlights, commentary, rankings, videos and fixtures from the Inter
national Cricket Council." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/def
ault-thumbnail.jpg" name="twitter:image"/>
<meta content="ICC Women's ODI All Rounder | Player Rankings | ICC" pro
perty="og:title"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/def
```

In [41]:
```python
Pname = []

for i in soup4.find_all('td',class_="table-body__cell rankings-table__name
    Pname.append(i.text.replace('\n',''))
Pname
```

Out[41]:
```
['Ashleigh Gardner',
 'Natalie Sciver-Brunt',
 'Hayley Matthews',
 'Amelia Kerr',
 'Deepti Sharma',
 'Ellyse Perry',
 'Jess Jonassen',
 'Sophie Devine',
 'Nida Dar',
 'Chloe-Lesleigh Tryon',
 'Chamari Athapaththu',
 'Sophie Ecclestone',
 'Laura Delany',
 'Charlotte Dean',
 'Rumana Ahmed',
 'Annabel Sutherland',
 'Georgia Wareham',
 'Kate Cross',
 'Salma Khatun']
```

In [42]:
```python
Tname = []

for i in soup4.find_all('span',class_="table-body__logo-text"):
    Tname.append(i.text)
Tname
```

Out[42]:
```
['AUS',
 'ENG',
 'WI',
 'NZ',
 'IND',
 'AUS',
 'AUS',
 'NZ',
 'PAK',
 'SA',
 'SL',
 'ENG',
 'IRE',
 'ENG',
 'BAN',
 'AUS',
 'AUS',
 'ENG',
 'BAN']
```

In [43]:
```python
rate = []

for i in soup4.find_all('td',class_="table-body__cell rating"):
    rate.append(i.text)
rate
```

Out[43]:
```
['377',
 '360',
 '358',
 '346',
 '312',
 '282',
 '227',
 '227',
 '224',
 '212',
 '190',
 '188',
 '172',
 '166',
 '166',
 '153',
 '153',
 '153',
 '153']
```

```
In [44]: df = pd.DataFrame({'Player Name':Pname,'Team Name':Tname,'Rating':rate})
         df.head(10)
```

Out[44]:

	Player Name	Team Name	Rating
0	Ashleigh Gardner	AUS	377
1	Natalie Sciver-Brunt	ENG	360
2	Hayley Matthews	WI	358
3	Amelia Kerr	NZ	346
4	Deepti Sharma	IND	312
5	Ellyse Perry	AUS	282
6	Jess Jonassen	AUS	227
7	Sophie Devine	NZ	227
8	Nida Dar	PAK	224
9	Chloe-Lesleigh Tryon	SA	212

Question 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 [45]: url = requests.get('https://www.cnbc.com/world/?region=world')
         url
```

Out[45]: <Response [200]>

In [46]:
```
soup =BeautifulSoup(url.content)
soup
```

Out[46]:
```
<!DOCTYPE html>
<html itemscope="" itemtype="https://schema.org/WebPage" lang="en" pref
ix="og=https://ogp.me/ns#"><head><meta content="telephone=no" name="for
mat-detection"/><style type="text/css">@charset "UTF-8";.Modal-modalBac
kground{background:#000000b3;height:100%;left:0;overflow-y:auto;positio
n:fixed;top:0;transition:background-color .4s;width:100%;z-index:10000
1}.Modal-bottomModal.Modal-modal{background:#f8f8f8;border-radius:3px;b
ottom:0;box-shadow:5px 5px 20px #1717171a;display:inline-block;height:5
28px;left:0;margin-top:0!important;max-width:100%;position:fixed;top:au
to;transform:none;width:100%}@media (max-width:1019px){.Modal-bottomMod
al.Modal-modal{height:642px}}@media (max-width:759px){.Modal-bottomModa
l.Modal-modal{height:100%;position:relative;top:0}}.Modal-modal{backgro
und-color:#fff;border-radius:3px;box-shadow:5px 5px 20px #1717171a;disp
lay:inline-block;left:50%;margin-top:10vh;max-width:100%;overflow:auto;
position:relative;transform:translateX(-50%)}@media (max-width:759px){.
Modal-modal{height:100%;left:auto;margin:0;transform:none;width:100%}}.
Modal-modalContents{overflow:auto}@media (max-width:759px){.Modal-modal
Contents{height:100%}}.Modal-closeButton{color:#a9a9a9;cursor:pointer;p
osition:absolute;right:12px;top:12px}.RecaptchaAcknowledgement-acknowle
```

In [47]:
```python
headline = []

for i in soup.find_all('a',class_="LatestNews-headline"):
    headline.append(i.text)
headline
```

Out[47]:
```
['5 things to know before the stock market opens Monday',
 "How 3 stocks tied to China's economy are navigating its uneven post-Covi
d recovery",
 'Canadian auto union reaches deal with Stellantis after brief labor strik
e',
 'Analyst calls: All the market-moving chatter from Wall Street on Monday
morning',
 "Biden unveils U.S. government's first-ever AI executive order",
 'Treasury yields rise as investors look to Fed meeting, key data due this
week',
 "Moscow claims the West is attempting to 'split Russian society' after ai
rport stormed by mob",
 'U.S. is working with Indo-Pacific partners to boost economic resilience,
says USTR',
 'Israeli defense forces expand Gaza ground incursion; thousands break int
o UN warehouses',
 'CNBC Daily Open: Are things about to get more difficult?',
 'European markets higher despite geopolitical tensions; German economy co
ntracts',
 'Is your hotel sustainable? Not if these two things are in your room',
 'WTO chief warns Israel-Hamas war could hurt global growth if conflict sp
ills over',
 "HSBC's after-tax profit surges over 235% year-on-year, announces $3 bill
ion share buyback ",
 'Evergrande shares fall 20% to all-time low as court adjourns winding-up
hearing',
 'McDonald's revenue climbs 14% as price hikes boost U.S. sales',
 'Oil prices slip even as Middle East tensions spike, as investors eye Fed
meeting',
 'UAW-Ford deal includes $8.1 billion in investment, $5,000 ratification b
onuses',
 'Standard Chartered-owned crypto firm Zodia launches in Hong Kong',
 'CNBC Daily Open: The perfect storm',
 "Asia markets mixed ahead of Japan's BOJ decision and South Korea inflati
on this week",
 "Amid an AI 'game of thrones,' these China tech giants offer real value:
Analyst",
 'Fund manager says this under-the-radar network stock is set for an AI bo
ost',
 'Stock futures rebound to start a big week with the S&P 500 mired in a co
rrection',
 'Chinese consumer stocks that analysts like for their global potential',
 'Cramer: This market has put investors in a box. This is how we break out
of it',
 'Google Bard asks Bill Nye if AI can help avoid end of world. Science Guy
said...',
 "What's a good price to get into a stock that has rallied, like Apple or
Nvidia?",
 "McDonald's HR exec: The No. 1 skill you get working in fast food",
 "The top 10 best states for millennials—New York, California didn't make
the list"]
```

In [48]:
```python
time = []

for i in soup.find_all('time',class_="LatestNews-timestamp"):
    time.append(i.text)
time
```

Out[48]:
```
['45 Min Ago',
 '2 Hours Ago',
 '2 Hours Ago',
 '2 Hours Ago',
 '3 Hours Ago',
 '3 Hours Ago',
 '3 Hours Ago',
 '4 Hours Ago',
 '5 Hours Ago',
 '6 Hours Ago',
 '6 Hours Ago',
 '6 Hours Ago',
 '6 Hours Ago',
 '8 Hours Ago',
 '8 Hours Ago',
 '8 Hours Ago',
 '9 Hours Ago',
 '11 Hours Ago',
 '11 Hours Ago',
 '12 Hours Ago',
 '12 Hours Ago',
 '13 Hours Ago',
 '13 Hours Ago',
 '13 Hours Ago',
 '14 Hours Ago',
 '20 Hours Ago',
 '21 Hours Ago',
 '21 Hours Ago',
 '22 Hours Ago',
 '22 Hours Ago']
```

In [49]:
```python
links = []

for i in soup.find_all('a',class_="LatestNews-headline"):
    links.append(i['href'])
links
```

Out[49]: ['https://www.cnbc.com/2023/10/30/5-things-to-know-before-the-stock-market
          -opens-monday-october-30.html',
           'https://www.cnbc.com/2023/10/30/how-3-stocks-tied-to-chinas-economy-are-
          navigating-its-uneven-post-covid-recovery.html',
           'https://www.cnbc.com/2023/10/30/canadian-auto-union-initiates-national-s
          trike-at-stellantis.html',
           'https://www.cnbc.com/2023/10/30/analyst-calls-all-the-market-moving-chat
          ter-from-wall-street-on-monday-morning.html',
           'https://www.cnbc.com/2023/10/30/biden-unveils-us-governments-first-ever-
          ai-executive-order.html',
           'https://www.cnbc.com/2023/10/30/us-treasurys-investors-look-to-fed-meeti
          ng-key-data-due-this-week.html',
           'https://www.cnbc.com/2023/10/30/ukraine-war-live-updates-latest-news-on-
          russia-and-the-war-in-ukraine.html',
           'https://www.cnbc.com/2023/10/30/us-working-with-indo-pacific-partners-to
          -boost-economic-resilience-ustr.html',
           'https://www.cnbc.com/2023/10/30/israel-hamas-war-live-updates-latest-new
          s-on-gaza-conflict.html',
           'https://www.cnbc.com/2023/10/30/cnbc-daily-open-are-things-about-to-get-
          more-difficult.html',
           'https://www.cnbc.com/2023/10/30/european-markets-live-updates-stocks-new
          s-data-and-earnings.html',
           'https://www.cnbc.com/2023/10/30/is-your-hotel-sustainable-not-if-these-t
          wo-things-are-in-your-room.html',
           'https://www.cnbc.com/2023/10/30/wto-chief-warns-israel-hamas-war-could-h
          urt-global-growth.html',
           'https://www.cnbc.com/2023/10/30/hsbc-profit-after-surges-over-235percent
          -year-on-year.html',
           'https://www.cnbc.com/2023/10/30/evergrande-shares-at-all-time-low-as-cou
          rt-adjourns-winding-up-hearing.html',
           'https://www.cnbc.com/2023/10/30/mcdonalds-mcd-q3-2023-earnings.html',
           'https://www.cnbc.com/2023/10/30/oil-prices-caution-ahead-of-fed-meeting-
          middle-east-tensions-spike.html',
           'https://www.cnbc.com/2023/10/30/uaw-ford-deal-includes-8point1b-in-inves
          tment-5000-ratification-bonus.html',
           'https://www.cnbc.com/2023/10/30/standard-chartered-owned-crypto-firm-zod
          ia-launches-in-hong-kong.html',
           'https://www.cnbc.com/2023/10/30/cnbc-daily-open-the-perfect-storm.html',
           'https://www.cnbc.com/2023/10/30/asia-stock-markets-today-live-updates-ba
          nk-of-japan-meeting-starts-.html',
           'https://www.cnbc.com/2023/10/30/these-chinese-tech-stocks-offer-a-lot-of
          -value-says-dan-ives-tech-analyst.html',
           'https://www.cnbc.com/2023/10/30/missed-nvidia-fund-manager-picks-under-t
          he-radar-networking-stock-for-ai-boost.html',
           'https://www.cnbc.com/2023/10/29/stock-market-today-live-updates.html',
           'https://www.cnbc.com/2023/10/29/chinese-consumer-stocks-that-analysts-li
          ke-for-their-global-potential.html',
           'https://www.cnbc.com/2023/10/29/cramer-this-market-put-investors-in-a-bo
          x-this-how-we-break-out-of-it.html',
           'https://www.cnbc.com/2023/10/29/how-bill-nye-educated-google-bard-on-how
          -ai-can-help-save-the-world.html',
           'https://www.cnbc.com/2023/10/29/what-is-a-good-price-to-get-into-a-stock
          -that-has-rallied-like-apple-nvidia.html',
           'https://www.cnbc.com/2023/10/29/mcdonalds-hr-exec-the-no-1-skill-you-get
          -working-in-fast-food.html',
           'https://www.cnbc.com/2023/10/29/best-states-millennials-scholaroo-study.
          html']

In [50]:
```python
df = pd.DataFrame({'Headlines':headline,'Time':time,'News Link':links})
df
```

Out[50]:

	Headlines	Time	News Link
0	5 things to know before the stock market opens...	45 Min Ago	https://www.cnbc.com/2023/10/30/5-things-to-kn...
1	How 3 stocks tied to China's economy are navig...	2 Hours Ago	https://www.cnbc.com/2023/10/30/how-3-stocks-t...
2	Canadian auto union reaches deal with Stellant...	2 Hours Ago	https://www.cnbc.com/2023/10/30/canadian-auto-...
3	Analyst calls: All the market-moving chatter f...	2 Hours Ago	https://www.cnbc.com/2023/10/30/analyst-calls-...
4	Biden unveils U.S. government's first-ever AI ...	3 Hours Ago	https://www.cnbc.com/2023/10/30/biden-unveils-...
5	Treasury yields rise as investors look to Fed ...	3 Hours Ago	https://www.cnbc.com/2023/10/30/us-treasurys-i...
6	Moscow claims the West is attempting to 'split...	3 Hours Ago	https://www.cnbc.com/2023/10/30/ukraine-war-li...
7	U.S. is working with Indo-Pacific partners to ...	4 Hours Ago	https://www.cnbc.com/2023/10/30/us-working-wit...
8	Israeli defense forces expand Gaza ground incu...	5 Hours Ago	https://www.cnbc.com/2023/10/30/israel-hamas-w...
9	CNBC Daily Open: Are things about to get more ...	6 Hours Ago	https://www.cnbc.com/2023/10/30/cnbc-daily-ope...
10	European markets higher despite geopolitical t...	6 Hours Ago	https://www.cnbc.com/2023/10/30/european-marke...
11	Is your hotel sustainable? Not if these two th...	6 Hours Ago	https://www.cnbc.com/2023/10/30/is-your-hotel-...
12	WTO chief warns Israel-Hamas war could hurt gl...	6 Hours Ago	https://www.cnbc.com/2023/10/30/wto-chief-warn...
13	HSBC's after-tax profit surges over 235% year-...	8 Hours Ago	https://www.cnbc.com/2023/10/30/hsbc-profit-af...
14	Evergrande shares fall 20% to all-time low as ...	8 Hours Ago	https://www.cnbc.com/2023/10/30/evergrande-sha...
15	McDonald's revenue climbs 14% as price hikes b...	8 Hours Ago	https://www.cnbc.com/2023/10/30/mcdonalds-mcd-...
16	Oil prices slip even as Middle East tensions s...	9 Hours Ago	https://www.cnbc.com/2023/10/30/oil-prices-cau...
17	UAW-Ford deal includes $8.1 billion in investm...	11 Hours Ago	https://www.cnbc.com/2023/10/30/uaw-ford-deal-...
18	Standard Chartered-owned crypto firm Zodia lau...	11 Hours Ago	https://www.cnbc.com/2023/10/30/standard-chart...
19	CNBC Daily Open: The perfect storm	12 Hours Ago	https://www.cnbc.com/2023/10/30/cnbc-daily-ope...
20	Asia markets mixed ahead of Japan's BOJ decisi...	12 Hours Ago	https://www.cnbc.com/2023/10/30/asia-stock-mar...
21	Amid an AI 'game of thrones,' these China tech...	13 Hours Ago	https://www.cnbc.com/2023/10/30/these-chinese-...
22	Fund manager says this under-the-radar network...	13 Hours Tago	https://www.cnbc.com/2023/10/30/missed-nvidia-...
23	Stock futures rebound to start a big week with...	13 Hours Ago	https://www.cnbc.com/2023/10/29/stock-market-t...
24	Chinese consumer stocks that analysts like for...	14 Hours Ago	https://www.cnbc.com/2023/10/29/chinese-consum...

	Headlines	Time	News Link
25	Cramer: This market has put investors in a box...	20 Hours Ago	https://www.cnbc.com/2023/10/29/cramer-this-ma...
26	Google Bard asks Bill Nye if AI can help avoid...	21 Hours Ago	https://www.cnbc.com/2023/10/29/how-bill-nye-e...
27	What's a good price to get into a stock that h...	21 Hours Ago	https://www.cnbc.com/2023/10/29/what-is-a-good...
28	McDonald's HR exec: The No. 1 skill you get wo...	22 Hours Ago	https://www.cnbc.com/2023/10/29/mcdonalds-hr-e...
29	The top 10 best states for millennials—New Yor...	22 Hours Ago	https://www.cnbc.com/2023/10/29/best-states-mi...

Question 6: Write a python program to scrape the details of most downloaded articles from AI in last 90 days.https://www.journals.elsevier.com/artificia intelligence/most-downloaded-articles (https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-articles) Scrape below mentioned details and make data frame

In [51]:
```python
response = requests.get('https://www.journals.elsevier.com/artificial-intel
response
```

Out[51]: <Response [200]>

In [52]:
```python
soup = BeautifulSoup(response.content)
soup
```

Out[52]:
```
<!DOCTYPE html>
<html><head><meta charset="utf-8"/><meta content="width=device-width" n
ame="viewport"/><meta content="en_US" name="og:locale"/><meta content
="Most Downloaded Articles - Artificial Intelligence - Journal - Elsevi
er" property="og:title"/><meta content="The journal of Artificial Intel
ligence (AIJ)  welcomes papers on broad aspects of AI that constitute a
dvances in the overall field including, but not limited …" property="o
g:description"/><meta content="http://ars.els-cdn.com/content/image/X00
043702.jpg" name="og:image" property="og:image"/><meta content="http://
ars.els-cdn.com/content/image/X00043702.jpg" name="og:image:url" proper
ty="og:image:url"/><meta content="https://ars.els-cdn.com/content/imag
e/X00043702.jpg" name="og:image:secure_url" property="og:image:secure_u
rl"/><meta content="journals.elsevier.com/artificial-intelligence/most-
downloaded-articles" name="og:url"/><meta content="website" property="o
g:type"/><link href="/apple-touch-icon.png" rel="apple-touch-icon" size
s="180x180"/><link href="/favicon-32x32.png" rel="icon" sizes="32x32" t
ype="image/png"/><link href="/favicon-16x16.png" rel="icon" sizes="16x1
6" type="image/png"/><link color="#ff6c00" href="/safari-pinned-tab.sv
g" rel="mask-icon"/><title>Most Downloaded Articles - Artificial Intell
```

In [53]:
```python
title =[]

for i in soup.find_all('h2',class_="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nm
    title.append(i.text)
title
```

Out[53]: ['Reward is enough',
 'Explanation in artificial intelligence: Insights from the social science
s',
 '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 rein
forcement learning',
 'Explaining individual predictions when features are dependent: More accu
rate approximations to Shapley values',
 'Multiple object tracking: A literature review',
 'A survey of inverse reinforcement learning: Challenges, methods and prog
ress',
 'Evaluating XAI: A comparison of rule-based and example-based explanation
s',
 'Explainable AI tools for legal reasoning about cases: A study on the Eur
opean Court of Human Rights',
 'Hard choices in artificial intelligence',
 'Assessing the communication gap between AI models and healthcare profess
ionals: Explainability, utility and trust in AI-driven clinical decision-m
aking',
 '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 implementati
on',
 'A review of possible effects of cognitive biases on interpretation of ru
le-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 stak
eholder perspective on XAI and a conceptual model guiding interdisciplinar
y XAI research']

```python
In [54]: authors =[]

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

```
Out[54]: ['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 [55]:
```python
publish_date =[]

for i in soup.find_all('span',class_="sc-1thf9ly-2 dvggWt"):
    publish_date.append(i.text)
publish_date
```

Out[55]: ['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 [56]:
```python
paper_url = []

for i in soup.find_all('a',class_="sc-5smygv-0 fIXTHm"):
    paper_url.append(i['href'])

paper_url
```

Out[56]: ['https://www.sciencedirect.com/science/article/pii/S0004370221000862',
 'https://www.sciencedirect.com/science/article/pii/S0004370218305988',
 'https://www.sciencedirect.com/science/article/pii/S0004370298000551',
 'https://www.sciencedirect.com/science/article/pii/S0004370214001386',
 'https://www.sciencedirect.com/science/article/pii/S0004370221001788',
 'https://www.sciencedirect.com/science/article/pii/S0004370215000910',
 'https://www.sciencedirect.com/science/article/pii/S0004370299000521',
 'https://www.sciencedirect.com/science/article/pii/S0004370221000539',
 'https://www.sciencedirect.com/science/article/pii/S0004370220301958',
 'https://www.sciencedirect.com/science/article/pii/S0004370221000515',
 'https://www.sciencedirect.com/science/article/pii/S0004370220301533',
 'https://www.sciencedirect.com/science/article/pii/S0004370223000073',
 'https://www.sciencedirect.com/science/article/pii/S0004370221001065',
 'https://www.sciencedirect.com/science/article/pii/S0004370222001795',
 'https://www.sciencedirect.com/science/article/pii/S0004370221000102',
 'https://www.sciencedirect.com/science/article/pii/S0004370219300116',
 'https://www.sciencedirect.com/science/article/pii/S000437029700043X',
 'https://www.sciencedirect.com/science/article/pii/S0004370216300790',
 'https://www.sciencedirect.com/science/article/pii/S0004370221000096',
 'https://www.sciencedirect.com/science/article/pii/S0004370216300224',
 'https://www.sciencedirect.com/science/article/pii/S0004370211000178',
 'https://www.sciencedirect.com/science/article/pii/S0004370222001692',
 'https://www.sciencedirect.com/science/article/pii/S00043702980023X',
 'https://www.sciencedirect.com/science/article/pii/S0004370221000242']

In [57]:
```python
df = pd.DataFrame({'Paper Title':title,'Authors':authors,'Publish Date':pub
df
```

Out[57]:

	Paper Title	Authors	Publish 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: Insigh...	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/...

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

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

```
In [58]: page = requests.get('https://www.dineout.co.in/delhi-restaurants/buffet-spe
         page
```

Out[58]: <Response [200]>

In [59]:
```python
soup1 = BeautifulSoup(page.content)
soup1
```

Out[59]:
```
<!DOCTYPE html>
<html lang="en"><head><meta charset="utf-8"/><meta content="IE=edge" ht
tp-equiv="X-UA-Compatible"/><meta content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no" name="viewport"/><link
href="/manifest.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;
```

In [60]:
```python
Name = []

for i in soup1.find_all('a',class_="restnt-name ellipsis"):
    Name.append(i.text)

Name
```

Out[60]:
```
['Castle Barbeque',
 'Cafe Knosh',
 'India Grill',
 'The Barbeque Company',
 'Delhi Barbeque',
 'The Monarch - Bar Be Que Village',
 'Indian Grill Room',
 'The Barbeque Times']
```

In [61]:
```python
loc = []

for i in soup1.find_all('div',class_="restnt-loc ellipsis"):
    loc.append(i.text)

loc
```

Out[61]:
```
['Connaught Place, Central Delhi',
 'The Leela Ambience Convention Hotel,Shahdara, East Delhi',
 'Hilton Garden Inn,Saket, South Delhi',
 'Gardens Galleria,Sector 38A, Noida',
 'Taurus Sarovar Portico,Mahipalpur, South Delhi',
 'Indirapuram Habitat Centre,Indirapuram, Ghaziabad',
 'Suncity Business Tower,Golf Course Road, Gurgaon',
 'M2K Corporate Park,Sector 51, Gurgaon']
```

In [62]:
```python
rating = []

for i in soup1.find_all('div',class_="restnt-rating rating-4"):
    rating.append(i.text)
rating
```

Out[62]: ['4', '4.3', '3.9', '3.9', '3.7', '3.8', '4.3', '4.1']

In [63]:
```python
cuisine = []

for i in soup1.find_all('span',class_="double-line-ellipsis"):
    cuisine.append(i.text)
cuisine
```

Out[63]: ['₹ 2,000 for 2 (approx) | Chinese, North Indian',
 '₹ 3,000 for 2 (approx) | Italian, Continental',
 '₹ 2,400 for 2 (approx) | North Indian, Italian',
 '₹ 1,700 for 2 (approx) | North Indian, Chinese',
 '₹ 1,800 for 2 (approx) | North Indian',
 '₹ 1,900 for 2 (approx) | North Indian',
 '₹ 2,200 for 2 (approx) | North Indian, Mughlai',
 '₹ 1,500 for 2 (approx) | North Indian, Continental, Chinese, South India
n']

In [64]:
```python
Image = []

for i in soup1.find_all('img',class_="no-img"):
    Image.append(i['data-src'])
Image
```

Out[64]: ['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/4/p/m/p406-1
5438184745c04ccea491bc.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/2/q/t/p2687-
169589385765154961ea87c.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/5/d/i/p52501
-1661855212630de5eceb6d2.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/n/o/p34822
-15599107305cfa594a13c24.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/y/f/p549-1
65000147262590640c0afc.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/1/u/r/p10642
8-166073786162fcd945925a9.jpg?tr=tr:n-medium']

In [65]:
```python
df = pd.DataFrame({'Restaurant name':Name,'Cuisine':cuisine,'Location':loc,
df
```

Out[65]:

	Restaurant name	Cuisine	Location	Rating	Image U
0	Castle Barbeque	₹ 2,000 for 2 (approx) | Chinese, North Indian	Connaught Place, Central Delhi	4	https://im1.dineout.co.in/images/uploads/rest
1	Cafe Knosh	₹ 3,000 for 2 (approx) | Italian, Continental	The Leela Ambience Convention Hotel,Shahdara, ...	4.3	https://im1.dineout.co.in/images/uploads/rest
2	India Grill	₹ 2,400 for 2 (approx) | North Indian, Italian	Hilton Garden Inn,Saket, South Delhi	3.9	https://im1.dineout.co.in/images/uploads/rest
3	The Barbeque Company	₹ 1,700 for 2 (approx) | North Indian, Chinese	Gardens Galleria,Sector 38A, Noida	3.9	https://im1.dineout.co.in/images/uploads/rest
4	Delhi Barbeque	₹ 1,800 for 2 (approx) | North Indian	Taurus Sarovar Portico,Mahipalpur, South Delhi	3.7	https://im1.dineout.co.in/images/uploads/rest
5	The Monarch - Bar Be Que Village	₹ 1,900 for 2 (approx) | North Indian	Indirapuram Habitat Centre,Indirapuram, Ghaziabad	3.8	https://im1.dineout.co.in/images/uploads/rest
6	Indian Grill Room	₹ 2,200 for 2 (approx) | North Indian, Mughlai	Suncity Business Tower,Golf Course Road, Gurgaon	4.3	https://im1.dineout.co.in/images/uploads/rest
7	The Barbeque Times	₹ 1,500 for 2 (approx) | North Indian, Contine...	M2K Corporate Park,Sector 51, Gurgaon	4.1	https://im1.dineout.co.in/images/uploads/rest

In [ ]:

In [ ]:

In [ ]:

In [ ]: