

```
In [1]: # Question-2 List of former President of INDIA. with name & term of office
```

```
In [2]: ! pip install bs4
! pip install requests
```

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

```
In [3]: from bs4 import BeautifulSoup
import requests
```

```
In [4]: # firstly we are checking the RESPONSE CODE website.
page = requests.get ('https://presidentofindia.nic.in/former-presidents.htm')
page
```

```
Out[4]: <Response [200]>
```

```
In [5]: # response code is in between [200-299] that means site is allowed us to scrap
```

```
In [ ]:
```

```
In [6]: # to download the whole content of the web page
```

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

```
Out[6]: <!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="type" />-->
<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>
<script type="text/javascript">
    ,
```

```
In [7]: # now to find the 'tag' and the 'class name' of each of the president
```

```
pd1 = soup.find('div', class_="presidentListing")
pd1.text.split('\n')[1]
```

```
Out[7]: 'Shri Ram Nath Kovind (birth - 1945)'
```

```
In [8]: pd_all = []
for i in soup.find_all('div', class_="presidentListing"):
    pd_all.append(i.text.split('\n')[1])
```

In [9]: pd_all

```
Out[9]: ['Shri Ram Nath Kovind (birth - 1945)',
'Shri Pranab Mukherjee (1935-2020)',
'Smt Pratibha Devisingh Patil (birth - 1934)',
'DR. A.P.J. Abdul Kalam (1931-2015)',
'Shri K. R. Narayanan (1920 - 2005)',
'Dr Shankar Dayal Sharma (1918-1999)',
'Shri R Venkataraman (1910-2009)',
'Giani Zail Singh (1916-1994)',
'Shri Neelam Sanjiva Reddy (1913-1996)',
'Dr. Fakhruddin Ali Ahmed (1905-1977)',
'Shri Varahagiri Venkata Giri (1894-1980)',
'Dr. Zakir Husain (1897-1969)',
'Dr. Sarvepalli Radhakrishnan (1888-1975)',
'Dr. Rajendra Prasad (1884-1963) ']
```

```
In [10]: pdt = []
for i in soup.find_all('div', class_="presidentListing"):
    pdt.append(i.text.split('\n')[2])
```

In [11]: pdt

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

In [12]: import pandas as pd

```
In [13]: df = pd.DataFrame ({'NAME OF PRESIDENT':pd_all, 'Term of Office':pdt})
df
```

```
Out[13]:
```

	NAME OF PRESIDENT	Term of Office
0	Shri Ram Nath Kovind (birth - 1945)	Term of Office: 25 July, 2017 to 25 July, 2022
1	Shri Pranab Mukherjee (1935-2020)	Term of Office: 25 July, 2012 to 25 July, 2017
2	Smt Pratibha Devisingh Patil (birth - 1934)	Term of Office: 25 July, 2007 to 25 July, 2012
3	DR. A.P.J. Abdul Kalam (1931-2015)	Term of Office: 25 July, 2002 to 25 July, 2007
4	Shri K. R. Narayanan (1920 - 2005)	Term of Office: 25 July, 1997 to 25 July, 2002
5	Dr Shankar Dayal Sharma (1918-1999)	Term of Office: 25 July, 1992 to 25 July, 1997
6	Shri R Venkataraman (1910-2009)	Term of Office: 25 July, 1987 to 25 July, 1992
7	Giani Zail Singh (1916-1994)	Term of Office: 25 July, 1982 to 25 July, 1987
8	Shri Neelam Sanjiva Reddy (1913-1996)	Term of Office: 25 July, 1977 to 25 July, 1982
9	Dr. Fakhruddin Ali Ahmed (1905-1977)	Term of Office: 24 August, 1974 to 11 February...
10	Shri Varahagiri Venkata Giri (1894-1980)	Term of Office: 3 May, 1969 to 20 July, 1969 a...
11	Dr. Zakir Husain (1897-1969)	Term of Office: 13 May, 1967 to 3 May, 1969
12	Dr. Sarvepalli Radhakrishnan (1888-1975)	Term of Office: 13 May, 1962 to 13 May, 1967
13	Dr. Rajendra Prasad (1884-1963)	Term of Office: 26 January, 1950 to 13 May, 1962

In []: =====

In []:

```
In [14]: # Question- 3 ICC-CRICKET.COM (MEN'S)

page = requests.get ('https://www.icc-cricket.com/rankings/mens/team-rankings/t20i')
page
```

Out[14]: <Response [200]>

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

```
Out[15]: <!DOCTYPE html>
<html lang="en">
<head>
<meta content="ICC Men's T20I Team Rankings | ICC" name="twitter:title"/>
<meta content="website" property="og:type"/>
<meta content="summary_large_image" property="twitter:card"/>
<meta content="Official ICC ranking for Twenty20 International cricket teams. Discover latest rankings, predict upcoming matches, see points and ratings for all teams." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official ICC ranking for Twenty20 International cricket teams. Discover latest rankings, 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 T20I 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 T20I Team Rankings | ICC</title>
<meta content="Official ICC ranking for Twenty20 International cricket teams. Discover latest rankings, predict upcoming matches, see points and ratings for all teams." property="og:description"/>
<!-- Designed and built by Pulselive - www.pulselive.com -->
<script>
```

```
In [16]: # top 10 odi teams:-
```

```
In [17]: odi10 = soup.find ('span',class_="u-hide-phablet")
odi10.text
```

```
Out[17]: 'India'
```

```
In [18]: odi_10 = []
for i in soup.find_all ('span',class_="u-hide-phablet"):
    odi_10.append (i.text)
```

```
In [19]: odi = odi_10[:10]
```

```
In [ ]:
```

```
In [20]: # from total 84 teams we shortlist top 10 teams
odi
```

```
Out[20]: ['India',
'England',
'New Zealand',
'Pakistan',
'South Africa',
'Australia',
'West Indies',
'Sri Lanka',
'Bangladesh',
'Afghanistan']
```

```
In [ ]:
```

```
In [21]: # MATCHES
```

```
In [22]: od= soup.find ('td', class_="table-body__cell u-center-text")
od.text
```

```
Out[22]: '39'
```

```
In [23]: od_all = []
for i in soup.find_all ('td', class_="table-body__cell u-center-text"):
    od_all.append(i.text)
```

```
In [24]: od1 = od_all[:-1:2]
```

```
In [25]: od1[:11]
```

```
Out[25]: ['39', '46', '50', '32', '31', '34', '37', '39', '27', '36', '37']
```

```
In [26]: od1[0]
```

```
Out[26]: '39'
```

```
In [27]: odi.insert(0,'52')
```

```
In [28]: matches = odi[:10]
```

```
In [29]: matches
```

```
Out[29]: ['52', '39', '46', '50', '32', '31', '34', '37', '39', '27']
```

```
In [ ]:
```

```
In [30]: # points
```

```
In [31]: od_all = []  
for i in soup.find_all ('td', class_="table-body__cell u-center-text"):  
    od_all.append(i.text)
```

```
In [32]: points = od_all[1::2]
```

```
In [33]: points.insert(0,'138889')
```

```
In [34]: score_points = points[:10]
```

```
In [35]: score_points
```

```
Out[35]: ['138889',  
          '10,117',  
          '11,794',  
          '12,719',  
          '8,080',  
          '7,681',  
          '8,086',  
          '8,774',  
          '8,656',  
          '5,915']
```

```
In [ ]:
```

```
In [36]: # rating
```

```
In [37]: rt = []  
for i in soup.find_all ('td', class_="table-body__cell u-text-right rating"):  
    rt.append(i.text)
```

```
In [38]: rt.insert(0,'267')
```

```
In [39]: rating = rt[:10]
```

```
In [40]: rating
```

```
Out[40]: ['267', '259', '256', '254', '253', '248', '238', '237', '222', '219']
```

```
In [41]: len(rating), len(matches), len(odi), len(score_points)
```

```
Out[41]: (10, 10, 10, 10)
```

```
In [42]: import pandas as pd
```

```
In [43]: df=pd.DataFrame({'Country':odi, 'Matches':matches, 'Ranking':rating, 'Points':score_points})
```

In [44]: df

Out[44]:

	Country	Matches	Ranking	Points
0	India	52	267	138889
1	England	39	259	10,117
2	New Zealand	46	256	11,794
3	Pakistan	50	254	12,719
4	South Africa	32	253	8,080
5	Australia	31	248	7,681
6	West Indies	34	238	8,086
7	Sri Lanka	37	237	8,774
8	Bangladesh	39	222	8,656
9	Afghanistan	27	219	5,915

In []:

```
=====
```

In [45]: # Question- 4 ICC-CRICKET.COM (WOMEN'S)

In [46]: page = requests.get ('https://www.icc-cricket.com/rankings/womens/team-rankings/odi')
page

Out[46]: <Response [200]>

In [48]: soup = BeautifulSoup (page.content)
soup

Out[48]:

```
<!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 rankings to
ble, 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 rankings to
ble, 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 rankings to
ble, predict upcoming matches, see points and ratings for all teams." property="og:description"/>
<!-- Designed and built by Pulselive - www.pulselive.com -->
<script>
```

In [49]: # we are finding here first team

```
tm = soup.find ('span', class_="u-hide-phablet")
tm.text
```

Out[49]: 'Australia'

In [50]: # now we found successfully the first, now we have to make list of all thirteen teams:-

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

In [51]: # we will creat a list of total thirteen teams with the variable name (tm_all)
tm_all

Out[51]:

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

```
In [115]: # here in the "tm_all" variable we have the complete list of top 10 countries :-  
tmn_all = tm_all [0:10]
```

```
In [116]: tmn_all
```

```
Out[116]: ['Australia',  
          'England',  
          'South Africa',  
          'India',  
          'New Zealand',  
          'West Indies',  
          'Thailand',  
          'Bangladesh',  
          'Pakistan',  
          'Sri Lanka']
```

```
In [ ]:
```

```
In [ ]:
```

```
In [55]: # here in this site we have a common class for the "matches" & "points", so we have to do that, we import the whole list  
# and we make seprate - seprate lists of "matches" and "points" from the combined list, with the help of 'SLICING'  
  
cmn = []  
for i in soup.find_all ('td', class_="table-body__cell u-center-text"):  
    cmn.append (i.text)
```

```
In [56]: # here we get the common list of 'matches' & 'points' in variable 'cmn'.  
cmn
```

```
Out[56]: ['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 [60]: match = cmn[::2]
```

```
In [66]: # here we have total match list from team2 to team10, team1 matches we have to add with the help of list function because  
# its is commonly combined with banner on the site, so it not shows with the common class,  
# therefore we have to add the 'matches', & 'points' seprately on [0th index] with the help of [.insert func]  
  
match[:9]
```

```
Out[66]: ['28', '26', '27', '25', '27', '11', '14', '27', '9']
```

```
In [77]: match.pop(1)
```

```
Out[77]: 21
```

```
In [78]: match
```

```
Out[78]: ['21', '28', '26', '27', '25', '27', '11', '14', '27', '9', '14', '9', '11']
```

```
In [80]: match_all = match[0:10]
```

```
In [81]: # now here in 'match_all' variable we have the complete List of matches of top 10 countries.  
match_all
```

```
Out[81]: ['21', '28', '26', '27', '25', '27', '11', '14', '27', '9']
```

```
In [ ]:
```

```
In [83]: # here we get the common List of 'matches' & 'points' in variable 'cmn'.  
cmn
```

```
Out[83]: ['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 [89]: # now here in the "cmn" variable we have the common List "matches" & "points", so we have separate points from here.  
# here with the help of "SLICING FUNCTIONS" which is [start from : end upto Last : skipped value] we separate all points from  
# the List.
```

```
pnt = cmn [1::2]
```

```
In [90]: pnt
```

```
Out[90]: ['3,342',  
          '3,098',  
          '2,820',  
          '2,553',  
          '2,535',  
          '821',  
          '977',  
          '1,678',  
          '479',  
          '548',  
          '0',  
          '0']
```

```
In [91]: pnt [:9]
```

```
Out[91]: ['3,342', '3,098', '2,820', '2,553', '2,535', '821', '977', '1,678', '479']
```

```
In [96]: # now in the above "pnt" variable we have points of countries from 2nd to 9th, so we have to insert 1st country points also.
```

```
pnt.insert(0,'3603')
```

```
In [97]: pnt
```

```
Out[97]: ['3603',  
          '3,342',  
          '3,098',  
          '2,820',  
          '2,553',  
          '2,535',  
          '821',  
          '977',  
          '1,678',  
          '479',  
          '548',  
          '0',  
          '0']
```

```
In [98]: points_all = pnt [:10]
```

```
In [99]: # now here in "points_all" variable we have the complete point list of all top 10 contries.  
points_all
```

```
Out[99]: ['3603',  
          '3,342',  
          '3,098',  
          '2,820',  
          '2,553',  
          '2,535',  
          '821',  
          '977',  
          '1,678',  
          '479']
```

```
In [ ]: # now we have all list of all top 10 [tmn_all][match_all][points_all]
```

```
In [100]: # now we have to find 'rating list' of all top 10 country teams.  
rtn = []  
for i in soup.find_all ('td',class_="table-body__cell u-text-right rating"):  
    rtn.append (i.text)
```

```
In [101]: rtn
```

```
Out[101]: ['119', '119', '104', '102', '94', '75', '70', '62', '53', '39', '0', '0']
```

```
In [104]: rtn.insert (0,'172')
```

```
In [110]: rtn
```

```
Out[110]: ['172',  
          '119',  
          '119',  
          '104',  
          '102',  
          '94',  
          '75',  
          '70',  
          '62',  
          '53',  
          '39',  
          '0',  
          '0']
```

```
In [112]: rtn_all = rtn[:10]
```

```
In [113]: # now we have all ratings of top10 teams in the variable "rtn_all"  
rtn_all
```

```
Out[113]: ['172', '119', '119', '104', '102', '94', '75', '70', '62', '53']
```

```
In [ ]: # now we have all list of all top 10 [TEAM NAME : tmn_all][MATCH : match_all][POINTS : points_all][RATING : rtn_all]
```

```
In [117]: # now we have to count the length of all list, which must be same  
len(tmn_all), len(match_all), len(points_all), len(rtn_all)
```

```
Out[117]: (10, 10, 10, 10)
```

```
In [ ]:
```

```
In [118]: # now all the above lengths are same and now we have to make dataframe from this  
# for which we have to import pandas  
import pandas as pd
```

```
In [119]: df = pd.DataFrame ({'TEAM NAME' : tmn_all, 'MATCH' : match_all, 'POINTS' : points_all, 'RATING' : rtn_all})
```


In [120]: df

Out[120]:

	TEAM NAME	MATCH	POINTS	RATING
0	Australia	21	3603	172
1	England	28	3,342	119
2	South Africa	26	3,098	119
3	India	27	2,820	104
4	New Zealand	25	2,553	102
5	West Indies	27	2,535	94
6	Thailand	11	821	75
7	Bangladesh	14	977	70
8	Pakistan	27	1,678	62
9	Sri Lanka	9	479	53

In []:

In []: # Question- 5 "CNBC.COM"

In [125]: page = requests.get ("https://www.cnn.com/world/?region=world")
page

Out[125]: <Response [200]>

In [126]: soup = BeautifulSoup (page.content)
soup

Out[126]: <!DOCTYPE html>
<html itemscope="" itemtype="https://schema.org/WebPage" lang="en" prefix="og=https://ogp.me/ns#"><head><meta content="teleph
hone=no" name="format-detection"/><style type="text/css">@charset "UTF-8";.RecaptchaAcknowledgement-acknowledgement{color:#7
474747;flex:1;font-size:11px;font-weight:600;line-height:15px;margin-bottom:7px;margin-top:24px;width:100%}.RecaptchaAcknowle
dgement-acknowledgement a{color:#747474;font-weight:500;text-decoration:none}.RecaptchaAcknowledgement-acknowledgement a:hov
er{color:#747474;text-decoration:underline}.RecaptchaAcknowledgement-acknowledgement a:active{color:#747474}.RecaptchaAcknow
ldgement-recaptchaPadding{margin-top:15px}.RecaptchaAcknowledgement-centerAligned{text-align:center}.RecaptchaAcknowledgemen
t-leftAligned{text-align:left}.RecaptchaAcknowledgement-rightAligned{text-align:right}.AuthForms-container{margin:0 auto;pa
dding:0 10px;width:458px}@media (max-width:759px){.AuthForms-container{max-width:458px;padding:20px 0 0;width:100%}.AuthFor
ms-container .AuthForms-signupContainer{margin:0 auto;padding:0 41px;text-align:center;width:458px}@media (max-width:759px)
{.AuthForms-container .AuthForms-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:ce
nter}.AuthForms-container .AuthForms-headerTitle{color:#071d39;font-size:20px;font-weight:700;text-align:center}.AuthForms-c
ontainer .AuthForms-signinContent{font-size:14px;font-weight:450;padding-top:20px}.AuthForms-container .AuthForms-toggleCta
{color:#424242;font-size:12px;font-weight:500;text-align:right}.AuthForms-container .AuthForms-instructions{margin-bottom:20
px;text-align:left}.AuthForms-container .AuthForms-instructions p{color:#171717;font-size:14px;line-height:16px}.AuthForms-c
ontainer .AuthForms-instructions p a{color:#005594;text-decoration:none}.AuthForms-container .AuthForms-instructions p a:hov
er{color:#002f6c}.AuthForms-container .AuthForms-formInputContainer{margin:0 auto 10px;max-width:350px}.AuthForms-container
.AuthForms-formInputContainer.AuthForms-withErrors{margin:0 auto 25px}.AuthForms-container .AuthForms-formInput{width:100%}.

In [128]: # here we find the individual news head lines
nws = soup.find ('a', class_="LatestNews-headline")
nws.text

Out[128]: "Greek stocks have climbed 40% over the last year – as the country shakes off 'problem child of Europe' label"

In [130]: nws_all = []
for i in soup.find_all ('a', class_="LatestNews-headline"):
nws_all.append(i.text)

In [131]: nws_all

Out[131]: ["Greek stocks have climbed 40% over the last year – as the country shakes off 'problem child of Europe' label",
 'Treasury yields steady as investors watch debt ceiling talks, economic reports',
 'Europe is producing more startup 'mafias' than ever despite tech rout',
 'Putin says Russia is going through 'difficult times'; Kremlin sees no path to peace in Ukraine yet',
 'European banks are stronger than their U.S. rivals, analysts say. Here's why',
 'CNBC Daily Open: A debt ceiling deal won't fix everything',
 'UK inflation sinks below 10% for the first time since August',
 'China's tech sector may see 'fewer surprises' in regulation: S&P report',
 'The West's oil war against Russia is starting to lose momentum',
 'European markets sharply lower as U.S. debt ceiling talks stumble; UK inflation falls to 8.7%',
 'China pledges new audit measures to curb risks and stabilize growth ',
 'China's new ambassador to the U.S. arrives to 'safeguard' Beijing's interests',
 'Palo Alto Networks CEO lauds generative AI as boon for efficiency ',
 'Here's the secret sauce that fueled Palo Alto's beat and raise despite tough times',
 "'Hurricane has landed:' Activist investor doubles down on office space short ",
 'Goldman Sachs loves Uber, Meta and Amazon, and ranks them in order of preference',
 'Fund manager reveals an overlooked corner of health care with big opportunities',
 'Cramer's Lightning Round: Get rid of Chimera',
 'CNBC Daily Open: A debt ceiling deal won't solve everything',
 'Target companies that put the customer first, Cramer says',
 'Asia markets slide: Shanghai shares hit 4-month low; Japan, HK post second-straight daily loss',
 'Elon Musk says his days are 'long and complicated' running SpaceX, Tesla and Twitter",
 'Apple's Eddy Cue says success starts by saying 'no' to almost everything',
 'Stock futures slip as investors eye debt ceiling clash in Washington',
 'Shares of Urban Outfitters spike after fiscal first-quarter earnings beat',
 'Stocks making the biggest moves after hours: Palo Alto Networks and more',
 'Citizens Bank fined for mismanaging fraud complaints',
 'Consumer spending is slowing, but these two Club retailers are well-positioned',
 'Nearly 50 AGs sue company that allegedly facilitated billions of spam calls',
 'Broadcom bolsters stock with new Apple deal. Here's why the experts like it"]

In []:

In [132]: *# here we find the time of occurrence of news.*
 tme_all = []
 for i in soup.find_all ('time', class_="LatestNews-timestamp"):
 tme_all.append(i.text)

In [133]: tme_all

Out[133]: ['33 Min Ago',
 '1 Hour Ago',
 '2 Hours Ago',
 '2 Hours Ago',
 '3 Hours Ago',
 '3 Hours Ago',
 '3 Hours Ago',
 '4 Hours Ago',
 '4 Hours Ago',
 '5 Hours Ago',
 '7 Hours Ago',
 '8 Hours Ago',
 '9 Hours Ago',
 '9 Hours Ago',
 '9 Hours Ago',
 '10 Hours Ago',
 '10 Hours Ago',
 '10 Hours Ago',
 '10 Hours Ago',
 '10 Hours Ago',
 '10 Hours Ago',
 '10 Hours Ago',
 '11 Hours Ago',
 '11 Hours Ago',
 '12 Hours Ago',
 '12 Hours Ago',
 '12 Hours Ago',
 '13 Hours Ago',
 '13 Hours Ago',
 '13 Hours Ago',
 '13 Hours Ago']

In []:

In [140]: lnk_all = []
 for i in soup.find_all ('a',class_="LatestNews-headline"):
 lnk_all.append(i.get('href'))

In [141]: lnk_all

Out[141]: ['https://www.cnbc.com/2023/05/24/greek-stocks-have-soared-as-it-shakes-off-problem-child-of-europe-tag.html',
 'https://www.cnbc.com/2023/05/24/us-treasuries-investors-watch-debt-ceiling-talks-economic-reports.html',
 'https://www.cnbc.com/2023/05/24/europe-and-israel-produce-more-startup-mafias-despite-tech-rout-accel.html',
 'https://www.cnbc.com/2023/05/24/ukraine-war-live-updates-latest-news-on-russia-and-the-war-in-ukraine.html',
 'https://www.cnbc.com/2023/05/24/european-banks-are-stronger-than-their-us-rivals-analysts-say-heres-why.html',
 'https://www.cnbc.com/2023/05/24/stock-markets-a-debt-ceiling-deal-wont-fix-everything.html',
 'https://www.cnbc.com/2023/05/24/uk-inflation-sinks-below-10percent-for-the-first-time-since-august.html',
 'https://www.cnbc.com/2023/05/24/chinas-tech-sector-may-see-fewer-surprises-in-regulation-sp-report.html',
 'https://www.cnbc.com/2023/05/24/ukraine-the-wests-oil-war-against-russia-is-losing-momentum.html',
 'https://www.cnbc.com/2023/05/24/european-markets-live-updates-stocks-news-data-us-debt-ceiling.html',
 'https://www.cnbc.com/2023/05/24/china-pledges-new-audit-measures-to-curb-risks-and-stabilize-growth.html',
 'https://www.cnbc.com/2023/05/24/chinas-new-us-ambassador-arrives-to-safeguard-beijings-interests.html',
 'https://www.cnbc.com/2023/05/23/palo-alto-networks-ceo-lauds-generative-ai-as-boon-for-efficiency.html',
 'https://www.cnbc.com/2023/05/23/secret-sauce-that-fueled-palo-alto-networks-q3-beat-and-raise.html',
 'https://www.cnbc.com/2023/05/23/hurricane-has-landed-jonathan-litt-shorts-entire-office-space-group.html',
 'https://www.cnbc.com/2023/05/24/goldman-sachs-loves-uber-meta-and-amazon-and-ranks-them-in-order-of-preference.html',
 'https://www.cnbc.com/2023/05/24/fund-manager-says-theres-investment-opportunities-in-femtech.html',
 'https://www.cnbc.com/2023/05/23/cramers-lightning-round-get-rid-of-chimera.html',
 'https://www.cnbc.com/2023/05/24/stock-markets-a-debt-ceiling-deal-wont-solve-everything.html',
 'https://www.cnbc.com/2023/05/23/target-companies-that-always-put-the-customer-first-cramer-says.html',
 'https://www.cnbc.com/2023/05/24/asia-markets-set-to-open-weaker-us-debt-ceiling-deal-still-elusive.html',
 'https://www.cnbc.com/2023/05/23/elon-musk-splits-time-across-spacex-tesla-and-twitter-heres-how.html',
 'https://www.cnbc.com/2023/05/23/apples-eddy-cue-says-success-starts-by-saying-no-to-almost-everything.html',
 'https://www.cnbc.com/2023/05/23/stock-market-today-live-updates.html',
 'https://www.cnbc.com/2023/05/23/urban-outfitters-urbn-q1-earnings-report-2023.html',
 'https://www.cnbc.com/2023/05/23/stocks-moving-big-after-hours-panw-urbn-intu-tol.html',
 'https://www.cnbc.com/2023/05/23/cfbp-slaps-citizens-bank-with-9-million-fine-over-fraud-complaints.html',
 'https://www.cnbc.com/2023/05/23/consumer-spending-is-weakening-as-economic-uncertainty-grows.html',
 'https://www.cnbc.com/2023/05/23/avid-telecom-facilitated-billions-of-spam-calls-state-ags-allege.html',
 'https://www.cnbc.com/2023/05/23/broadcom-bolsters-stock-with-new-apple-deal-why-the-experts-like-it.html']

In []:

In [142]: *# here we find the length of all variables are same, which is must required.*
 len(tme_all), len(lnk_all), len(nws_all)

Out[142]: (30, 30, 30)

In [143]: *# now we convert the above data into the dataframe*
for which we import pandas
 import pandas as pd

In [144]: df=pd.DataFrame({'Time':tme_all, 'Latest News':nws_all, 'News Link':lnk_all})

In [145]: df

Out[145]:

	Time	Latest News	News Link
0	33 Min Ago	Greek stocks have climbed 40% over the last ye...	https://www.cnbc.com/2023/05/24/greek-stocks-h...
1	1 Hour Ago	Treasury yields steady as investors watch debt...	https://www.cnbc.com/2023/05/24/us-treasurys-i...
2	2 Hours Ago	Europe is producing more startup 'mafias' than...	https://www.cnbc.com/2023/05/24/europe-and-isr...
3	2 Hours Ago	Putin says Russia is going through 'difficult ...	https://www.cnbc.com/2023/05/24/ukraine-war-li...
4	3 Hours Ago	European banks are stronger than their U.S. ri...	https://www.cnbc.com/2023/05/24/european-banks...
5	3 Hours Ago	CNBC Daily Open: A debt ceiling deal won't fix...	https://www.cnbc.com/2023/05/24/stock-markets-...
6	3 Hours Ago	UK inflation sinks below 10% for the first tim...	https://www.cnbc.com/2023/05/24/uk-inflation-s...
7	4 Hours Ago	China's tech sector may see 'fewer surprises' ...	https://www.cnbc.com/2023/05/24/chinas-tech-se...
8	4 Hours Ago	The West's oil war against Russia is starting ...	https://www.cnbc.com/2023/05/24/ukraine-the-we...
9	5 Hours Ago	European markets sharply lower as U.S. debt ce...	https://www.cnbc.com/2023/05/24/european-marke...
10	7 Hours Ago	China pledges new audit measures to curb risks...	https://www.cnbc.com/2023/05/24/china-pledges-...
11	8 Hours Ago	China's new ambassador to the U.S. arrives to ...	https://www.cnbc.com/2023/05/24/chinas-new-us-...
12	9 Hours Ago	Palo Alto Networks CEO lauds generative AI as ...	https://www.cnbc.com/2023/05/23/palo-alto-netw...
13	9 Hours Ago	Here's the secret sauce that fueled Palo Alto'...	https://www.cnbc.com/2023/05/23/secret-sauce-t...
14	9 Hours Ago	'Hurricane has landed:' Activist investor doub...	https://www.cnbc.com/2023/05/23/hurricane-has-...
15	10 Hours Ago	Goldman Sachs loves Uber, Meta and Amazon, and...	https://www.cnbc.com/2023/05/24/goldman-sachs-...
16	10 Hours Ago	Fund manager reveals an overlooked corner of h...	https://www.cnbc.com/2023/05/24/fund-manager-s...
17	10 Hours Ago	Cramer's Lightning Round: Get rid of Chimera	https://www.cnbc.com/2023/05/23/cramers-lightn...
18	10 Hours Ago	CNBC Daily Open: A debt ceiling deal won't sol...	https://www.cnbc.com/2023/05/24/stock-markets-...
19	10 Hours Ago	Target companies that put the customer first, ...	https://www.cnbc.com/2023/05/23/target-compani...
20	10 Hours Ago	Asia markets slide: Shanghai shares hit 4-mont...	https://www.cnbc.com/2023/05/24/asia-markets-s...
21	11 Hours Ago	Elon Musk says his days are 'long and complica...	https://www.cnbc.com/2023/05/23/elon-musk-spli...
22	11 Hours Ago	Apple's Eddy Cue says success starts by saying...	https://www.cnbc.com/2023/05/23/apples-eddy-cu...
23	12 Hours Ago	Stock futures slip as investors eye debt ceili...	https://www.cnbc.com/2023/05/23/stock-market-t...
24	12 Hours Ago	Shares of Urban Outfitters spike after fiscal ...	https://www.cnbc.com/2023/05/23/urban-outfite...
25	12 Hours Ago	Stocks making the biggest moves after hours: P...	https://www.cnbc.com/2023/05/23/stocks-moving-...
26	13 Hours Ago	Citizens Bank fined for mismanaging fraud comp...	https://www.cnbc.com/2023/05/23/cfbp-slaps-cit...
27	13 Hours Ago	Consumer spending is slowing, but these two Cl...	https://www.cnbc.com/2023/05/23/consumer-spend...
28	13 Hours Ago	Nearly 50 AGs sue company that allegedly facil...	https://www.cnbc.com/2023/05/23/avid-telecom-f...
29	13 Hours Ago	Broadcom bolsters stock with new Apple deal. H...	https://www.cnbc.com/2023/05/23/broadcom-bolst...

In []: =====

In []: # Question-6 JOURNALS

In [146]: page = requests.get ('https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-articles')
page

Out[146]: <Response [200]>

In [147]: soup = BeautifulSoup (page.content)
soup

Out[147]: <!DOCTYPE html>
<html><head><meta charset="utf-8"><meta content="width=device-width" name="viewport"/><meta content="en_US" name="og:local
e"/><meta content="Most Downloaded Articles - Artificial Intelligence - Journal - Elsevier" property="og:title"/><meta conte
nt="The journal of Artificial Intelligence (AIJ) welcomes papers on broad aspects of AI that constitute advances in the ove
rall field including, but not limited ..." property="og:description"/><meta content="http://ars.els-cdn.com/content/image/X000
43702.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:secur
e_url" property="og:image:secure_url"/><meta content="journals.elsevier.com/artificial-intelligence/most-downloaded-article
s" name="og:url"/><meta content="website" property="og:type"/><link href="/apple-touch-icon.png" rel="apple-touch-icon" size
s="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 Dow
nloaded 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 ..." na
me="description"/><meta content="Most Downloaded Articles - Artificial Intelligence - Journal - Elsevier" name="twitter:titl
e"/><meta content="The journal of Artificial Intelligence (AIJ) welcomes papers on broad aspects of AI that constitute adva
nces in the overall field including, but not limited ..." name="twitter:description"/><meta content="http://ars.els-cdn.com/co
ntent/image/X00043702.jpg" name="twitter:image"/><meta content="summary" name="twitter:card"/><script type="application/ld+j
son">{"@context":"https://schema.org";"@type":"Periodical";"issn":"00
04-3702";"name":"Artificial Intelligence";"publisher":"Elsevier";"acquirel

```
In [149]: pge = soup.find('h2',class_="sc-1qrq3sd-1 gRGUSUS sc-1nmom32-0 sc-1nmom32-1 btcbyu goSKRg")
pge.text
```

```
Out[149]: 'Reward is enough'
```

```
In [150]: pge_all = []
for i in soup.find_all('h2',class_="sc-1qrq3sd-1 gRGUSUS sc-1nmom32-0 sc-1nmom32-1 btcbyu goSKRg"):
    pge_all.append(i.text)
```

```
In [151]: pge_all
```

```
Out[151]: ['Reward is enough',
'Explanation in artificial intelligence: Insights from the social sciences',
'Creativity and artificial intelligence',
'Conflict-based search for optimal multi-agent pathfinding',
'Knowledge graphs as tools for explainable machine learning: A survey',
'Law and logic: A review from an argumentation perspective',
'Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning',
'Explaining individual predictions when features are dependent: More accurate approximations to Shapley values',
'Multiple object tracking: A literature review',
'A survey of inverse reinforcement learning: Challenges, methods and progress',
'Evaluating XAI: A comparison of rule-based and example-based explanations',
'Explainable AI tools for legal reasoning about cases: A study on the European Court of Human Rights',
'Hard choices in artificial intelligence',
'Assessing the communication gap between AI models and healthcare professionals: Explainability, utility and trust in AI-driven clinical decision-making',
'Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies',
'The Hanabi challenge: A new frontier for AI research',
'Wrappers for feature subset selection',
'Artificial cognition for social human-robot interaction: An implementation',
'A review of possible effects of cognitive biases on interpretation of rule-based machine learning models',
'The multifaceted impact of Ada Lovelace in the digital age',
'Robot ethics: Mapping the issues for a mechanized world',
'Reward (Mis)design for autonomous driving',
'Planning and acting in partially observable stochastic domains',
'What do we want from Explainable Artificial Intelligence (XAI)? - A stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research']
```

```
In [154]: auth_all = []
for i in soup.find_all('span', class_="sc-1w3fpd7-0 dnCnA0"):
    auth_all.append(i.text)
```

```
In [155]: auth_all
```

```
Out[155]: ['David Silver, Satinder Singh, Doina Precup, Richard S. Sutton ',
'Tim Miller ',
'Margaret A. Boden ',
'Guni Sharon, Roni Stern, Ariel Felner, Nathan R. Sturtevant ',
'Illaria 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 [156]: pub_dt = []
for i in soup.find_all('span',class_="sc-1thf9ly-2 dvggWt"):
    pub_dt.append(i.text)
```

```
In [157]: pub_dt
```

```
Out[157]: ['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 [158]: url = []  
for i in soup.find_all('a',class_="sc-5smylv-0 fIXTHm"):  
    url.append(i.get('href'))
```

```
In [159]: url
```

```
Out[159]: ['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/S000437029800023X',  
          'https://www.sciencedirect.com/science/article/pii/S0004370221000242']
```

```
In [160]: len(pge_all), len(auth_all), len(pub_dt), len(url)
```

```
Out[160]: (24, 24, 24, 24)
```

```
In [161]: import pandas as pd
```

```
In [162]: df=pd.DataFrame({'Paper Title':pge_all, 'Authers':auth_all, 'Published Date':pub_dt, 'Paper url':url})
```

In [163]: df

Out[163]:

	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: Insign...	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/...
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 Bahnik, 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/...

In []:

In []:

In []: # Question- 7 (DINEOUT)

In [164]: page = requests.get('https://www.dineout.co.in/indore-restaurants/welcome-back')
page

Out[164]: <Response [200]>

In [165]: soup = BeautifulSoup (page.content)
soup

```

Out[165]: <!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="/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;
  }
  .async-hide{
    display: none !important;
  }

```

```

In [166]: rst = []
for i in soup.find_all('div',class_="restnt-info cursor"):
    rst.append(i.text)

```

In [167]:

```
rst
```

Out[167]:

```
['KT- Sero Bar And RestaurantAirport Road, West Indore',
 'Vegtawa RestaurantSudama Nagar, South Indore',
 'Ranjeet's KitchenSudama Nagar, South Indore",
 'The ICF LoungeSudama Nagar, South Indore',
 'Cakes N CraftSudama Nagar, South Indore',
 'Food Zone Family RestaurantSudama Nagar, South Indore',
 'Boozer's BarDwarkapuri, South Indore",
 'Tonicue The BarHotel Malwa Inn,Rajendra Nagar, South Indore',
 'Rim Zim Club And ResortRajendra Nagar, South Indore',
 'Cheese FactoryManik Bagh Road, South Indore',
 'Waffles & MoreManik Bagh Road, South Indore',
 'Ministry Of GrubManik Bagh Road, South Indore',
 'Hotel Bee TownRajendra Nagar, South Indore',
 'Naivedyam RestaurantRajendra Nagar, South Indore',
 'Cakes N CraftBhanwar Kuan, South Indore',
 'ConstellationHotel WOW,Vijay Nagar, East Indore',
 'Koffee KingSnehnagar, East Indore',
 'Singh SahabSapna Sangeeta Mall,Navlakha, Central Indore',
 'The Cafe SquareBhanwar Kuan, South Indore',
 'Deluxe BarRNT Marg, East Indore',
 'Osmium LoungeSapna Sangeeta Mall,Navlakha, Central Indore']
```

In [201]:

```
rst_all = rst[:10]
```

In [203]:

```
# here we are taking list for top 10 restaurantsrst_all
rst_all
```

Out[203]:

```
['KT- Sero Bar And RestaurantAirport Road, West Indore',
 'Vegtawa RestaurantSudama Nagar, South Indore',
 'Ranjeet's KitchenSudama Nagar, South Indore",
 'The ICF LoungeSudama Nagar, South Indore',
 'Cakes N CraftSudama Nagar, South Indore',
 'Food Zone Family RestaurantSudama Nagar, South Indore',
 'Boozer's BarDwarkapuri, South Indore",
 'Tonicue The BarHotel Malwa Inn,Rajendra Nagar, South Indore',
 'Rim Zim Club And ResortRajendra Nagar, South Indore',
 'Cheese FactoryManik Bagh Road, South Indore']
```

In [182]:

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

In [183]:

```
# above we split the text by ('|' as a seprator)
cusn
```

Out[183]:

```
[' North Indian, Chinese',
 ' Chinese, North Indian',
 ' North Indian, Chinese',
 ' Chinese, North Indian, Desserts, Fast Food, Italian',
 ' Desserts',
 ' Chinese, North Indian, Fast Food',
 ' North Indian',
 ' Chinese, North Indian',
 ' Mughlai, North Indian, Chinese',
 ' Fast Food',
 ' Fast Food, Desserts',
 ' Chinese, North Indian, Italian, Fast Food',
 ' Fast Food, North Indian, Chinese',
 ' North Indian, South Indian, Fast Food, Beverages',
 ' Desserts',
 ' North Indian, Italian, Continental, Asian',
 ' Beverages, Desserts',
 ' North Indian',
 ' Fast Food, North Indian, Chinese',
 ' Mughlai, North Indian',
 ' North Indian, Chinese, Fast Food']
```

In [199]:

```
cusn_all = cusn[:10]
```



```
In [200]: # here we are taking List for top 10 restaurants
cusrn_all
```

```
Out[200]: [' North Indian, Chinese',
           ' Chinese, North Indian',
           ' North Indian, Chinese',
           ' Chinese, North Indian, Desserts, Fast Food, Italian',
           ' Desserts',
           ' Chinese, North Indian, Fast Food',
           ' North Indian',
           ' Chinese, North Indian',
           ' Mughlai, North Indian, Chinese',
           ' Fast Food']
```

```
In [ ]:
```

```
In [188]: loc = []
for i in soup.find_all('div',class_="restnt-loc ellipsis"):
    loc.append(i.text)
```

```
In [189]: loc
```

```
Out[189]: ['Airport Road, West Indore',
           'Sudama Nagar, South Indore',
           'Sudama Nagar, South Indore',
           'Sudama Nagar, South Indore',
           'Sudama Nagar, South Indore',
           'Sudama Nagar, South Indore',
           'Dwarkapuri, South Indore',
           'Hotel Malwa Inn,Rajendra Nagar, South Indore',
           'Rajendra Nagar, South Indore',
           'Manik Bagh Road, South Indore',
           'Manik Bagh Road, South Indore',
           'Manik Bagh Road, South Indore',
           'Rajendra Nagar, South Indore',
           'Rajendra Nagar, South Indore',
           'Bhanwar Kuan, South Indore',
           'Hotel WOW,Vijay Nagar, East Indore',
           'Snehnagar, East Indore',
           'Sapna Sangeeta Mall,Navlakha, Central Indore',
           'Bhanwar Kuan, South Indore',
           'RNT Marg, East Indore',
           'Sapna Sangeeta Mall,Navlakha, Central Indore']
```

```
In [196]: loc_all = loc[:10]
```

```
In [198]: loc_all
```

```
Out[198]: ['Airport Road, West Indore',
           'Sudama Nagar, South Indore',
           'Sudama Nagar, South Indore',
           'Sudama Nagar, South Indore',
           'Sudama Nagar, South Indore',
           'Sudama Nagar, South Indore',
           'Dwarkapuri, South Indore',
           'Hotel Malwa Inn,Rajendra Nagar, South Indore',
           'Rajendra Nagar, South Indore',
           'Manik Bagh Road, South Indore']
```

```
In [192]: ratings = []
for i in soup.find_all('div',class_="restnt-rating rating-4"):
    ratings.append(i.text)
```

```
In [193]: ratings
```

```
Out[193]: ['3.6',
           '3.6',
           '3.5',
           '3.5',
           '4.2',
           '3.9',
           '3.5',
           '4.1',
           '3.6',
           '4',
           '3.8',
           '3.8',
           '4.3']
```

```
In [194]: rtng_all = ratings[:10]
```

```
In [195]: rtng_all
```

```
Out[195]: ['3.6', '3.6', '3.5', '3.5', '4.2', '3.9', '3.5', '4.1', '3.6', '4']
```

```
In [205]: lnk = []
for i in soup.find_all('img', class_="no-img"):
    lnk.append(i.get('data-src'))
```

```
In [206]: lnk
```

```
Out[206]: ['https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/j/i/p54666-15559281915cbd947f34dca.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/6/c/t/p61685-15646373745d4278be1eb96.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/h/v/p5382-164182371561dc3de3c1418.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/v/g/p54615-164129704561d43495c25b4.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/p/t/p5336-163955242261b995a665d0a.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/6/d/o/p68466-15717416915daedffbc8e9d.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/a/x/p54670-15591971605cef75e82ac69.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/g/z/p54582-164129225261d421dc77491.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/t/d/p54604-15588749675cea8b577f749.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/6/a/n/p65740-15712096745da6c1ca2a259.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/6/p/s/p65732-15687179205d80bc60b8f96.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/9/q/z/p97598-163298844661556d1e1fcbb.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/a/d/p56670-15571463175cd02acd07ac9.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/7/q/b/p71059-15744966435dd8e983a8d1c.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/p/t/p5336-163955242261b995a665d0a.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/l/d/p54705-15583358405ce251608cdc9.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/i/k/p5321-15565995045cc7d2d0a10e4.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/a/l/p5342-15568614425ccbd202d08e0.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/6/b/z/p65763-15687186795d80bf57d25c1.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/6/r/t/p64434-15679501475d750543ec952.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/1/u/f/p17636-15590260335cecd9713aecf.jpg?tr=tr:n-medium']
```

```
In [207]: lnk_all = lnk[:10]
```

```
In [208]: lnk_all
```

```
Out[208]: ['https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/j/i/p54666-15559281915cbd947f34dca.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/6/c/t/p61685-15646373745d4278be1eb96.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/h/v/p5382-164182371561dc3de3c1418.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/v/g/p54615-164129704561d43495c25b4.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/p/t/p5336-163955242261b995a665d0a.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/6/d/o/p68466-15717416915daedffbc8e9d.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/a/x/p54670-15591971605cef75e82ac69.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/g/z/p54582-164129225261d421dc77491.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/t/d/p54604-15588749675cea8b577f749.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/6/a/n/p65740-15712096745da6c1ca2a259.jpg?tr=tr:n-medium']
```

```
In [ ]:
```

```
In [209]: len(rst_all),len(cusn_all),len(loc_all),len(rtng_all),len(lnk_all)
```

```
Out[209]: (10, 10, 10, 10, 10)
```

```
In [210]: import pandas as pd
```

```
In [211]: df=pd.DataFrame({'Restaurant Name':rst_all, 'Cuisine':cusn_all, 'Location':loc_all, 'Ratings':rtng_all, 'image url':lnk_all})
```

In [212]:

df

Out[212]:

	Restaurant Name	Cuisine	Location	Ratings	image url
0	KT- Sero Bar And RestaurantAirport Road, West ...	North Indian, Chinese	Airport Road, West Indore	3.6	https://im1.dineout.co.in/images/uploads/resta...
1	Vegtawa RestaurantSudama Nagar, South Indore	Chinese, North Indian	Sudama Nagar, South Indore	3.6	https://im1.dineout.co.in/images/uploads/resta...
2	Ranjeet's KitchenSudama Nagar, South Indore	North Indian, Chinese	Sudama Nagar, South Indore	3.5	https://im1.dineout.co.in/images/uploads/resta...
3	The ICF LoungeSudama Nagar, South Indore	Chinese, North Indian, Desserts, Fast Food, l...	Sudama Nagar, South Indore	3.5	https://im1.dineout.co.in/images/uploads/resta...
4	Cakes N CraftSudama Nagar, South Indore	Desserts	Sudama Nagar, South Indore	4.2	https://im1.dineout.co.in/images/uploads/resta...
5	Food Zone Family RestaurantSudama Nagar, South...	Chinese, North Indian, Fast Food	Sudama Nagar, South Indore	3.9	https://im1.dineout.co.in/images/uploads/resta...
6	Boozer's BarDwarkapuri, South Indore	North Indian	Dwarkapuri, South Indore	3.5	https://im1.dineout.co.in/images/uploads/resta...
7	Tonicue The BarHotel Malwa Inn,Rajendra Nagar,...	Chinese, North Indian	Hotel Malwa Inn,Rajendra Nagar, South Indore	4.1	https://im1.dineout.co.in/images/uploads/resta...
8	Rim Zim Club And ResortRajendra Nagar, South I...	Mughlai, North Indian, Chinese	Rajendra Nagar, South Indore	3.6	https://im1.dineout.co.in/images/uploads/resta...
9	Cheese FactoryManik Bagh Road, South Indore	Fast Food	Manik Bagh Road, South Indore	4	https://im1.dineout.co.in/images/uploads/resta...

In []:

=====

In [235]:

```
# Question- 1 (Wikipedia)

# Fetch the HTML content from Wikipedia.org
url = "https://en.wikipedia.org/wiki/Main_Page"
response = requests.get(url)
html_content = response.content

# Parse the HTML content with BeautifulSoup
soup = BeautifulSoup(html_content, "html.parser")

# Find all header tags (h1 to h6)
header_tags = soup.find_all(["h1", "h2", "h3", "h4", "h5", "h6"])

# Extract the text from header tags
header_texts = [header.get_text() for header in header_tags]

# Create a dataframe
df = pd.DataFrame({"Headers": header_texts})

# Display the dataframe
print(df)
```

	Headers
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

In []:

In []: