

## Answer No.01

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

```
Requirement already satisfied: bs4 in c:\users\asus\anaconda3\lib\site-packages (0.0.1)  
Requirement already satisfied: beautifulsoup4 in c:\users\asus\anaconda3\lib\site-packages (from bs4) (4.11.1)  
Requirement already satisfied: soupsieve>1.2 in c:\users\asus\anaconda3\lib\site-packages (from beautifulsoup4->bs4) (2.3.1)  
Requirement already satisfied: requests in c:\users\asus\anaconda3\lib\site-packages (2.27.1)  
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\asus\anaconda3\lib\site-packages (from requests) (2.0.4)  
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\asus\anaconda3\lib\site-packages (from requests) (1.26.9)  
Requirement already satisfied: certifi>=2017.4.17 in c:\users\asus\anaconda3\lib\site-packages (from requests) (2021.10.8)  
Requirement already satisfied: idna<4,>=2.5 in c:\users\asus\anaconda3\lib\site-packages (from requests) (3.3)
```

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

```
In [5]: # Send the request to the webpage server  
page=requests.get('https://en.wikipedia.org/wiki/Main_Page')  
page
```

```
Out[5]: <Response [200]>
```

In [6]: # Page Content

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

Out[6]: <!DOCTYPE html>

```
<html class="client-nojs vector-feature-language-in-header-enabled vector-
feature-language-in-main-page-header-disabled vector-feature-language-aler-
t-in-sidebar-enabled vector-feature-sticky-header-disabled vector-feature-
page-tools-disabled vector-feature-page-tools-pinned-disabled vector-fea-
ture-main-menu-pinned-disabled vector-feature-limited-width-enabled vector-f
eature-limited-width-content-enabled" dir="ltr" lang="en">
<head>
<meta charset="utf-8"/>
<title>Wikipedia, the free encyclopedia</title>
<script>document.documentElement.className="client-js vector-feature-langu
age-in-header-enabled vector-feature-language-in-main-page-header-disabled
vector-feature-language-alert-in-sidebar-enabled vector-feature-sticky-hea
der-disabled vector-feature-page-tools-disabled vector-feature-page-tools-
pinned-disabled vector-feature-main-menu-pinned-disabled vector-feature-li
mited-width-enabled vector-feature-limited-width-content-enabled";(functio
n(){var cookie=document.cookie.match(/(?:^| ; )enwikimwclientprefs=([^;]
+);/);if(cookie){var featureName=cookie[1];document.documentElement.classNa
me=document.documentElement.className.replace(featureName+'-enabled',featu
reName+'-disabled');}})();</script>
```

```
In [10]: from urllib.request import urlopen
from bs4 import BeautifulSoup
html = urlopen('https://en.wikipedia.org/wiki/Main_Page')
bs = BeautifulSoup(html, "html.parser")
titles = bs.find_all(['h1', 'h2', 'h3', 'h4', 'h5', 'h6'])
print('List all the header tags :', *titles, sep='\n\n')
```

List all the header tags :

```
<h1 class="firstHeading mw-first-heading" id="firstHeading" style="display: none"><span class="mw-page-title-main">Main Page</span></h1>

<h1><span class="mw-headline" id="Welcome_to_Wikipedia">Welcome to <a href="/wiki/Wikipedia" title="Wikipedia">Wikipedia</a></span></h1>

<h2 class="mp-h2" id="mp-tfa-h2"><span id="From_today.27s_featured_article"></span><span class="mw-headline" id="From_today's_featured_article">From today's featured article</span></h2>

<h2 class="mp-h2" id="mp-dyk-h2"><span class="mw-headline" id="Did_you_know_...">Did you know ...</span></h2>

<h2 class="mp-h2" id="mp-itn-h2"><span class="mw-headline" id="In_the_news">In the news</span></h2>

<h2 class="mp-h2" id="mp-otd-h2"><span class="mw-headline" id="On_this_day">On this day</span></h2>

<h2 class="mp-h2" id="mp-tfp-h2"><span id="Today.27s_featured_picture"></span><span class="mw-headline" id="Today's_featured_picture">Today's featured picture</span></h2>

<h2 class="mp-h2" id="mp-other"><span class="mw-headline" id="Other_areas_of_Wikipedia">Other areas of Wikipedia</span></h2>

<h2 class="mp-h2" id="mp-sister"><span id="Wikipedia.27s_sister_projects"></span><span class="mw-headline" id="Wikipedia's_sister_projects">Wikipedia's sister projects</span></h2>

<h2 class="mp-h2" id="mp-lang"><span class="mw-headline" id="Wikipedia_languages">Wikipedia languages</span></h2>
```

```
In [12]: import pandas as pd
df=pd.DataFrame({'Header tags':titles})
df
```

Out[12]:

	Header tags
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]]

## Answer No.02

```
In [16]: url=requests.get('https://www.imdb.com/list/ls031732693/')
url
```

Out[16]: <Response [200]>

```
In [20]: story=BeautifulSoup(url.content,'html.parser') #Because page is based on html
print(story.prettify()) # Prettify will remove all the new lines.
```

```
<!DOCTYPE html>
<html xmlns:fb="http://www.facebook.com/2008/fbml" xmlns:og="http://ogp.me/ns#">
  <head>
    <meta charset="utf-8"/>
    <script type="text/javascript">
      var IMDbTimer={starttime: new Date().getTime(),pt:'java'};
    </script>
    <script>
      if (typeof uet == 'function') {
        uet("bb", "LoadTitle", {wb: 1});
      }
    </script>
    <script>
      (function(t){ (t.events = t.events || {})[ "csm_head_pre_title" ] = new Date().getTime(); })(IMDbTimer);
    </script>
    <title>
      Top 50 -Best Hollywood Movies Of All Time - IMDb
    </title>
  </head>
  <body>
    <div>
      <h1>Top 50 -Best Hollywood Movies Of All Time - IMDb</h1>
      <p>This is a list of the top 50 best Hollywood movies of all time, ranked by user rating on IMDb. The list includes classic films like "The Godfather" and "Star Wars", as well as more recent hits like "The Dark Knight" and "Avengers: Endgame".</p>
      <ul>
        <li>1. The Godfather</li>
        <li>2. Star Wars: Episode IV - A New Hope</li>
        <li>3. The Shawshank Redemption</li>
        <li>4. The Godfather: Part II</li>
        <li>5. The Dark Knight</li>
        <li>6. The Empire Strikes Back</li>
        <li>7. One Flew Over the Cuckoo's Nest</li>
        <li>8. The Godfather: Part III</li>
        <li>9. The Thin Red Line</li>
        <li>10. The Dark Knight Rises</li>
        <li>11. The Lord of the Rings: The Return of the King</li>
        <li>12. The Empire Strikes Back</li>
        <li>13. The Shawshank Redemption</li>
        <li>14. The Godfather: Part II</li>
        <li>15. The Dark Knight</li>
        <li>16. The Thin Red Line</li>
        <li>17. The Empire Strikes Back</li>
        <li>18. The Shawshank Redemption</li>
        <li>19. The Godfather: Part II</li>
        <li>20. The Dark Knight</li>
        <li>21. The Thin Red Line</li>
        <li>22. The Empire Strikes Back</li>
        <li>23. The Shawshank Redemption</li>
        <li>24. The Godfather: Part II</li>
        <li>25. The Dark Knight</li>
        <li>26. The Thin Red Line</li>
        <li>27. The Empire Strikes Back</li>
        <li>28. The Shawshank Redemption</li>
        <li>29. The Godfather: Part II</li>
        <li>30. The Dark Knight</li>
        <li>31. The Thin Red Line</li>
        <li>32. The Empire Strikes Back</li>
        <li>33. The Shawshank Redemption</li>
        <li>34. The Godfather: Part II</li>
        <li>35. The Dark Knight</li>
        <li>36. The Thin Red Line</li>
        <li>37. The Empire Strikes Back</li>
        <li>38. The Shawshank Redemption</li>
        <li>39. The Godfather: Part II</li>
        <li>40. The Dark Knight</li>
        <li>41. The Thin Red Line</li>
        <li>42. The Empire Strikes Back</li>
        <li>43. The Shawshank Redemption</li>
        <li>44. The Godfather: Part II</li>
        <li>45. The Dark Knight</li>
        <li>46. The Thin Red Line</li>
        <li>47. The Empire Strikes Back</li>
        <li>48. The Shawshank Redemption</li>
        <li>49. The Godfather: Part II</li>
        <li>50. The Dark Knight</li>
      </ul>
    </div>
  </body>
</html>
```

```
In [22]: # Scrape the movie names
scraped_movies=story.find_all('h3',class_='lister-item-header')
scraped_movies
```

```
Out[22]: [<h3 class="lister-item-header">
   <span class="lister-item-index unbold text-primary">1.</span>
   <a href="/title/tt0468569/">The Dark Knight</a>
   <span class="lister-item-year text-muted unbold">(2008)</span>
</h3>,
<h3 class="lister-item-header">
   <span class="lister-item-index unbold text-primary">2.</span>
   <a href="/title/tt0111161/">The Shawshank Redemption</a>
   <span class="lister-item-year text-muted unbold">(1994)</span>
</h3>,
<h3 class="lister-item-header">
   <span class="lister-item-index unbold text-primary">3.</span>
   <a href="/title/tt0110912/">Pulp Fiction</a>
   <span class="lister-item-year text-muted unbold">(1994)</span>
</h3>,
<h3 class="lister-item-header">
   <span class="lister-item-index unbold text-primary">4.</span>
   <a href="/title/tt1375666/">Inception</a>
   <span class="lister-item-year text-muted unbold">(2010)</span>
   ...]
```

```
In [23]: # parse the movie names

movies=[]
for movie in scraped_movies:
    movies.append(movie.get_text().strip()) # get_text is the inbuilt function
movies
```

```
Out[23]: ['1.\nThe Dark Knight\n(2008)',  
'2.\nThe Shawshank Redemption\n(1994)',  
'3.\nPulp Fiction\n(1994)',  
'4.\nInception\n(2010)',  
'5.\nThe Usual Suspects\n(1995)',  
'6.\nMemento\n(2000)',  
'7.\nThe Silence of the Lambs\n(1991)',  
'8.\nFight Club\n(1999)',  
'9.\nThe Godfather\n(1972)',  
'10.\n12 Angry Men\n(1957)',  
'11.\nAmerican History X\n(1998)',  
'12.\nThe Dark Knight Rises\n(2012)',  
'13.\nThe Pursuit of Happyness\n(2006)',  
'14.\nShutter Island\n(2010)',  
'15.\n2001: A Space Odyssey\n(1968)',  
'16.\nThe Machinist\n(2004)',  
"17.\nSchindler's List\n(1993)",  
'18.\nSe7en\n(1995)',  
'19.\nGone Girl\n(2014)',  
'20.\nThe Man from Earth\n(2007)',  
'21.\nThe Prestige\n(2006)',  
'22.\nBatman Begins\n(2005)',  
'23.\nInterstellar\n(2014)',  
'24.\nThe Grand Budapest Hotel\n(2014)',  
'25.\nRequiem for a Dream\n(2000)',  
'26.\nGisaengchung\n(2019)',  
'27.\nThe Shining\n(1980)',  
'28.\nGoodFellas\n(1990)',  
'29.\nThe Pianist\n(2002)',  
'30.\nThe Godfather Part II\n(1974)',  
'31.\nDjango Unchained\n(2012)',  
'32.\nDr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb\n(1964)',  
'33.\nReservoir Dogs\n(1992)',  
'34.\nEternal Sunshine of the Spotless Mind\n(2004)',  
'35.\nA Clockwork Orange\n(1971)',  
'36.\n1917\n(2019)',  
'37.\nUp\n(2009)',  
'38.\nKill Bill: Vol. 1\n(2003)',  
'39.\nA Beautiful Mind\n(2001)',  
'40.\nThe Wolf of Wall Street\n(2013)',  
'41.\nNo Country for Old Men\n(2007)',  
'42.\nFargo\n(1996)',  
'43.\nGone Girl\n(2014)',  
'44.\nMad Max: Fury Road\n(2015)',  
'45.\nGangs of Wasseypur\n(2012)',  
'46.\nWhiplash\n(2014)',  
'47.\nAmerican Beauty\n(1999)',  
'48.\nTrainspotting\n(1996)',  
'49.\nBoogie Nights\n(1997)',  
'50.\nJoker\n(I) (2019)']
```

```
In [24]: movies=[]
for movie in scraped_movies:
    movie=movie.get_text().replace('\n',"") # \n is for next line
    movie=movie.strip(" ") # for remove spaces
    movies.append(movie)
movies
```

```
Out[24]: ['1.The Dark Knight(2008)',  
          '2.The Shawshank Redemption(1994)',  
          '3.Pulp Fiction(1994)',  
          '4.Inception(2010)',  
          '5.The Usual Suspects(1995)',  
          '6.Memento(2000)',  
          '7.The Silence of the Lambs(1991)',  
          '8.Fight Club(1999)',  
          '9.The Godfather(1972)',  
          '10.12 Angry Men(1957)',  
          '11.American History X(1998)',  
          '12.The Dark Knight Rises(2012)',  
          '13.The Pursuit of Happyness(2006)',  
          '14.Shutter Island(2010)',  
          '15.2001: A Space Odyssey(1968)',  
          '16.The Machinist(2004)',  
          "17.Schindler's List(1993)",  
          '18.Se7en(1995)',  
          '19.Gone Girl(2014)',  
          '20.The Man from Earth(2007)',  
          '21.The Prestige(2006)',  
          '22.Batman Begins(2005)',  
          '23.Interstellar(2014)',  
          '24.The Grand Budapest Hotel(2014)',  
          '25.Requiem for a Dream(2000)',  
          '26.Gisaengchung(2019)',  
          '27.The Shining(1980)',  
          '28.GoodFellas(1990)',  
          '29.The Pianist(2002)',  
          '30.The Godfather Part II(1974)',  
          '31.Django Unchained(2012)',  
          '32.Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb(1964)',  
          '33.Reservoir Dogs(1992)',  
          '34.Eternal Sunshine of the Spotless Mind(2004)',  
          '35.A Clockwork Orange(1971)',  
          '36.1917(2019)',  
          '37.Up(2009)',  
          '38.Kill Bill: Vol. 1(2003)',  
          '39.A Beautiful Mind(2001)',  
          '40.The Wolf of Wall Street(2013)',  
          '41.No Country for Old Men(2007)',  
          '42.Fargo(1996)',  
          '43.Gone Girl(2014)',  
          '44.Mad Max: Fury Road(2015)',  
          '45.Gangs of Wasseypur(2012)',  
          '46.Whiplash(2014)',  
          '47.American Beauty(1999)',  
          '48.Trainspotting(1996)',  
          '49.Boogie Nights(1997)',  
          '50.Joker(I) (2019)']
```

```
In [27]: # get rating for movies
scraped_ratings=story.find_all('div',class_='ipl-rating-star small')
scraped_ratings
```

```
Out[27]: [<div class="ipl-rating-star small">
    <span class="ipl-rating-star__star">
        <svg class="ipl-icon ipl-star-icon" fill="#000000" height="24" viewBox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg">
            <path d="M0 0h24v24H0z" fill="none"></path>
            <path d="M12 17.27L18.18 21l-1.64-7.03L22 9.24l-7.19-.61L12 2 9.19 8.63 2 9.24l5.46 4.73L5.82 21z"></path>
            <path d="M0 0h24v24H0z" fill="none"></path>
        </svg>
    </span>
    <span class="ipl-rating-star__rating">9</span>
</div>,
<div class="ipl-rating-star small">
    <span class="ipl-rating-star__star">
        <svg class="ipl-icon ipl-star-icon" fill="#000000" height="24" viewBox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg">
            <path d="M0 0h24v24H0z" fill="none"></path>
            <path d="M12 17.27L18.18 21l-1.64-7.03L22 9.24l-7.19-.61L12 2 9.19 8.63 2 9.24l5.46 4.73L5.82 21z"></path>
            <path d="M0 0h24v24H0z" fill="none"></path>
        </svg>
    </span>
    <span class="ipl-rating-star__rating">9</span>
</div>]
```

```
In [28]: # parse ratings
ratings=[]
for rating in scraped_ratings:
    rating=rating.get_text().replace('\n','')
    ratings.append(rating)
ratings
```

```
Out[28]: ['9',
 '9.3',
 '8.9',
 '8.8',
 '8.5',
 '8.4',
 '8.6',
 '8.8',
 '9.2',
 '9',
 '8.5',
 '8.4',
 '8',
 '8.2',
 '8.3',
 '7.7',
 '9',
 '8.6',
 '8.1',
 '7.9',
 '8.5',
 '8.2',
 '8.6',
 '8.1',
 '8.3',
 '8.5',
 '8.4',
 '8.7',
 '8.5',
 '9',
 '8.4',
 '8.4',
 '8.3',
 '8.3',
 '8.3',
 '8.2',
 '8.3',
 '8.2',
 '8.2',
 '8.2',
 '8.2',
 '8.2',
 '8.2',
 '8.1',
 '8.1',
 '8.1',
 '8.1',
 '8.2',
 '8.5',
 '8.4',
 '8.1',
 '7.9',
 '8.4']
```

```
In [31]: # store the scraped data
import pandas as pd
data=pd.DataFrame()
data['Movie Names']=movies
data['Ratings']=ratings
data
```

Out[31]:

	Movie Names	Ratings
0	1.The Dark Knight(2008)	9
1	2.The Shawshank Redemption(1994)	9.3
2	3.Pulp Fiction(1994)	8.9
3	4.Inception(2010)	8.8
4	5.The Usual Suspects(1995)	8.5
5	6.Memento(2000)	8.4
6	7.The Silence of the Lambs(1991)	8.6
7	8.Fight Club(1999)	8.8
8	9.The Godfather(1972)	9.2
9	10.12 Angry Men(1957)	9
10	11.American History X(1998)	8.5
11	12.The Dark Knight Rises(2012)	8.4
12	13.The Pursuit of Happyness(2006)	8
13	14.Shutter Island(2010)	8.2
14	15.2001: A Space Odyssey(1968)	8.3
15	16.The Machinist(2004)	7.7
16	17.Schindler's List(1993)	9
17	18.Se7en(1995)	8.6
18	19.Gone Girl(2014)	8.1
19	20.The Man from Earth(2007)	7.9
20	21.The Prestige(2006)	8.5
21	22.Batman Begins(2005)	8.2
22	23.Interstellar(2014)	8.6
23	24.The Grand Budapest Hotel(2014)	8.1
24	25.Requiem for a Dream(2000)	8.3
25	26.Gisaengchung(2019)	8.5
26	27.The Shining(1980)	8.4
27	28.GoodFellas(1990)	8.7
28	29.The Pianist(2002)	8.5
29	30.The Godfather Part II(1974)	9
30	31.Django Unchained(2012)	8.4
31	32.Dr. Strangelove or: How I Learned to Stop W...	8.4
32	33.Reservoir Dogs(1992)	8.3
33	34.Eternal Sunshine of the Spotless Mind(2004)	8.3
34	35.A Clockwork Orange(1971)	8.3
35	36.1917(2019)	8.2

	Movie Names	Ratings
36	37.Up(2009)	8.3
37	38.Kill Bill: Vol. 1(2003)	8.2
38	39.A Beautiful Mind(2001)	8.2
39	40.The Wolf of Wall Street(2013)	8.2
40	41.No Country for Old Men(2007)	8.2
41	42.Fargo(1996)	8.1
42	43.Gone Girl(2014)	8.1
43	44.Mad Max: Fury Road(2015)	8.1
44	45.Gangs of Wasseypur(2012)	8.2
45	46.Whiplash(2014)	8.5
46	47.American Beauty(1999)	8.4
47	48.Trainspotting(1996)	8.1
48	49.Boogie Nights(1997)	7.9
49	50.Joker(l) (2019)	8.4

In [32]: # get release year for movies

```
scraped_release_years=story.find_all('span',class_='lister-item-year text-mute')
scraped_release_years
```

Out[32]: [<span class="lister-item-year text-muted unbold">(2008)</span>,

```
<span class="lister-item-year text-muted unbold">(1994)</span>,
<span class="lister-item-year text-muted unbold">(1994)</span>,
<span class="lister-item-year text-muted unbold">(2010)</span>,
<span class="lister-item-year text-muted unbold">(1995)</span>,
<span class="lister-item-year text-muted unbold">(2000)</span>,
<span class="lister-item-year text-muted unbold">(1991)</span>,
<span class="lister-item-year text-muted unbold">(1999)</span>,
<span class="lister-item-year text-muted unbold">(1972)</span>,
<span class="lister-item-year text-muted unbold">(1957)</span>,
<span class="lister-item-year text-muted unbold">(1998)</span>,
<span class="lister-item-year text-muted unbold">(2012)</span>,
<span class="lister-item-year text-muted unbold">(2006)</span>,
<span class="lister-item-year text-muted unbold">(2010)</span>,
<span class="lister-item-year text-muted unbold">(1968)</span>,
<span class="lister-item-year text-muted unbold">(2004)</span>,
<span class="lister-item-year text-muted unbold">(1993)</span>,
<span class="lister-item-year text-muted unbold">(1995)</span>,
<span class="lister-item-year text-muted unbold">(2014)</span>,
<span class="lister-item-year text-muted unbold">(2007)</span>,
<span class="lister-item-year text-muted unbold">(2006)</span>,
<span class="lister-item-year text-muted unbold">(2005)</span>,
<span class="lister-item-year text-muted unbold">(2014)</span>,
<span class="lister-item-year text-muted unbold">(2014)</span>,
<span class="lister-item-year text-muted unbold">(2000)</span>,
<span class="lister-item-year text-muted unbold">(2019)</span>,
<span class="lister-item-year text-muted unbold">(1980)</span>,
<span class="lister-item-year text-muted unbold">(1990)</span>,
<span class="lister-item-year text-muted unbold">(2002)</span>,
<span class="lister-item-year text-muted unbold">(1974)</span>,
<span class="lister-item-year text-muted unbold">(2012)</span>,
<span class="lister-item-year text-muted unbold">(1964)</span>,
<span class="lister-item-year text-muted unbold">(1992)</span>,
<span class="lister-item-year text-muted unbold">(2004)</span>,
<span class="lister-item-year text-muted unbold">(1971)</span>,
<span class="lister-item-year text-muted unbold">(2019)</span>,
<span class="lister-item-year text-muted unbold">(2009)</span>,
<span class="lister-item-year text-muted unbold">(2003)</span>,
<span class="lister-item-year text-muted unbold">(2001)</span>,
<span class="lister-item-year text-muted unbold">(2013)</span>,
<span class="lister-item-year text-muted unbold">(2007)</span>,
<span class="lister-item-year text-muted unbold">(1996)</span>,
<span class="lister-item-year text-muted unbold">(2014)</span>,
<span class="lister-item-year text-muted unbold">(2015)</span>,
<span class="lister-item-year text-muted unbold">(2012)</span>,
<span class="lister-item-year text-muted unbold">(2014)</span>,
<span class="lister-item-year text-muted unbold">(1999)</span>,
<span class="lister-item-year text-muted unbold">(1996)</span>,
<span class="lister-item-year text-muted unbold">(1997)</span>,
<span class="lister-item-year text-muted unbold">(I) (2019)</span>]
```

```
In [33]: # parse release years
years_of_release=[]
for release_year in scraped_release_years:
    release_year=release_year.get_text().replace('\n','')
    years_of_release.append(release_year)
years_of_release
```

```
Out[33]: [ '(2008)',  
          '(1994)',  
          '(1994)',  
          '(2010)',  
          '(1995)',  
          '(2000)',  
          '(1991)',  
          '(1999)',  
          '(1972)',  
          '(1957)',  
          '(1998)',  
          '(2012)',  
          '(2006)',  
          '(2010)',  
          '(1968)',  
          '(2004)',  
          '(1993)',  
          '(1995)',  
          '(2014)',  
          '(2007)',  
          '(2006)',  
          '(2005)',  
          '(2014)',  
          '(2014)',  
          '(2000)',  
          '(2019)',  
          '(1980)',  
          '(1990)',  
          '(2002)',  
          '(1974)',  
          '(2012)',  
          '(1964)',  
          '(1992)',  
          '(2004)',  
          '(1971)',  
          '(2019)',  
          '(2009)',  
          '(2003)',  
          '(2001)',  
          '(2013)',  
          '(2007)',  
          '(1996)',  
          '(2014)',  
          '(2015)',  
          '(2012)',  
          '(2014)',  
          '(1999)',  
          '(1996)',  
          '(1997)',  
          '(I) (2019)']
```

```
In [35]: # store the scraped data
import pandas as pd
data=pd.DataFrame()
data['Movie Names']=movies
data['Ratings']=ratings
data['Year of Release']=years_of_release
data
```

Out[35]:

	Movie Names	Ratings	Year of Release
0	1.The Dark Knight(2008)	9	(2008)
1	2.The Shawshank Redemption(1994)	9.3	(1994)
2	3.Pulp Fiction(1994)	8.9	(1994)
3	4.Inception(2010)	8.8	(2010)
4	5.The Usual Suspects(1995)	8.5	(1995)
5	6.Memento(2000)	8.4	(2000)
6	7.The Silence of the Lambs(1991)	8.6	(1991)
7	8.Fight Club(1999)	8.8	(1999)
8	9.The Godfather(1972)	9.2	(1972)
9	10.12 Angry Men(1957)	9	(1957)
10	11.American History X(1998)	8.5	(1998)
11	12.The Dark Knight Rises(2012)	8.4	(2012)
12	13.The Pursuit of Happyness(2006)	8	(2006)
13	14.Shutter Island(2010)	8.2	(2010)
14	15.2001: A Space Odyssey(1968)	8.3	(1968)
15	16.The Machinist(2004)	7.7	(2004)
16	17.Schindler's List(1993)	9	(1993)
17	18.Se7en(1995)	8.6	(1995)
18	19.Gone Girl(2014)	8.1	(2014)
19	20.The Man from Earth(2007)	7.9	(2007)
20	21.The Prestige(2006)	8.5	(2006)
21	22.Batman Begins(2005)	8.2	(2005)
22	23.Interstellar(2014)	8.6	(2014)
23	24.The Grand Budapest Hotel(2014)	8.1	(2014)
24	25.Requiem for a Dream(2000)	8.3	(2000)
25	26.Gisaengchung(2019)	8.5	(2019)
26	27.The Shining(1980)	8.4	(1980)
27	28.GoodFellas(1990)	8.7	(1990)
28	29.The Pianist(2002)	8.5	(2002)
29	30.The Godfather Part II(1974)	9	(1974)
30	31.Django Unchained(2012)	8.4	(2012)
31	32.Dr. Strangelove or: How I Learned to Stop W...	8.4	(1964)
32	33.Reservoir Dogs(1992)	8.3	(1992)
33	34.Eternal Sunshine of the Spotless Mind(2004)	8.3	(2004)
34	35.A Clockwork Orange(1971)	8.3	(1971)
35	36.1917(2019)	8.2	(2019)

	Movie Names	Ratings	Year of Release
36	37.Up(2009)	8.3	(2009)
37	38.Kill Bill: Vol. 1(2003)	8.2	(2003)
38	39.A Beautiful Mind(2001)	8.2	(2001)
39	40.The Wolf of Wall Street(2013)	8.2	(2013)
40	41.No Country for Old Men(2007)	8.2	(2007)
41	42.Fargo(1996)	8.1	(1996)
42	43.Gone Girl(2014)	8.1	(2014)
43	44.Mad Max: Fury Road(2015)	8.1	(2015)
44	45.Gangs of Wasseypur(2012)	8.2	(2012)
45	46.Whiplash(2014)	8.5	(2014)
46	47.American Beauty(1999)	8.4	(1999)
47	48.Trainspotting(1996)	8.1	(1996)
48	49.Boogie Nights(1997)	7.9	(1997)
49	50.Joker(I) (2019)	8.4	(I) (2019)

Answer No.03

```
In [84]: url1=requests.get('https://www.imdb.com/list/ls025646548/')
url1
```

Out[84]: <Response [200]>

```
In [90]: story1=BeautifulSoup(url1.content) #Because page is based on html that's why we  
print(story1.prettify()) # Prettify will remove all the new lines.
```

```
<!DOCTYPE html>
<html xmlns:fb="http://www.facebook.com/2008/fbml" xmlns:og="http://ogp.me/ns#">
<head>
  <meta charset="utf-8"/>
  <script type="text/javascript">
    var IMDbTimer={starttime: new Date().getTime(),pt:'java'};
  </script>
  <script>
    if (typeof uet == 'function') {
      uet("bb", "LoadTitle", {wb: 1});
    }
  </script>
  <script>
    (function(t){ (t.events = t.events || {})[ "csm_head_pre_title" ] = new Date().getTime(); })(IMDbTimer);
  </script>
  <title>
    Top 50 Hindi Movies of all time - IMDb
  </title>
```

```
In [86]: # Scrape the movie names
scraped_movies1=story1.find_all('h3',class_='lister-item-header')
scraped_movies1
```

```
Out[86]: [<h3 class="lister-item-header">
   <span class="lister-item-index unbold text-primary">1.</span>
   <a href="/title/tt0043306/">Awaara</a>
   <span class="lister-item-year text-muted unbold">(1951)</span>
</h3>,
<h3 class="lister-item-header">
   <span class="lister-item-index unbold text-primary">2.</span>
   <a href="/title/tt0048613/">Shree 420</a>
   <span class="lister-item-year text-muted unbold">(1955)</span>
</h3>,
<h3 class="lister-item-header">
   <span class="lister-item-index unbold text-primary">3.</span>
   <a href="/title/tt0050870/">Pyaasa</a>
   <span class="lister-item-year text-muted unbold">(1957)</span>
</h3>,
<h3 class="lister-item-header">
   <span class="lister-item-index unbold text-primary">4.</span>
   <a href="/title/tt0050665/">Madhumati</a>
   <span class="lister-item-year text-muted unbold">(1958)</span>
   ...]
```

```
In [87]: # parse the movie names

movies1=[]
for movie1 in scraped_movies1:
    movies1.append(movie1.get_text().strip()) # get_text is the inbuilt function
movies1
```

```
Out[87]: ['1.\nAwaara\n(1951)',  
'2.\nShree 420\n(1955)',  
'3.\nPyaasa\n(1957)',  
'4.\nMadhumati\n(1958)',  
'5.\nKanoon\n(1960)',  
'6.\nMughal-E-Azam\n(1960)',  
'7.\nKala Bazar\n(1960)',  
'8.\nWoh Kaun Thi?\n(1964)',  
'9.\nGuide\n(1965)',  
'10.\nAmarapali\n(1966)',  
'11.\nAnand\n(1971)',  
'12.\nRajnigandha\n(1974)',  
'13.\nArvind Desai Ki Ajeeb Dastaan\n(1978)',  
'14.\nSatte Pe Satta\n(1982)',  
'15.\nDisco Dancer\n(1982)',  
'16.\nJaane Bhi Do Yaaro\n(1983)',  
'17.\nArdh Satya\n(1983)',  
'18.\nKatha\n(1983)',  
'19.\nParinda\n(1989)',  
'20.\nBaazigar\n(1993)',  
'21.\nDarr\n(1993)',  
'22.\nRangeela\n(1995)',  
'23.\nDilwale Dulhania Le Jayenge\n(1995)',  
'24.\nBorder\n(I) (1997)',  
'25.\nSatya\n(1998)',  
'26.\nKaun?\n(1999)',  
'27.\nHera Pheri\n(2000)',  
'28.\nLagaan: Once Upon a Time in India\n(2001)',  
'29.\nCompany\n(2002)',  
'30.\nDarna Mana Hai\n(2003)',  
'31.\nMunna Bhai M.B.B.S.\n(2003)',  
'32.\nKhakee\n(2004)',  
'33.\nVeer-Zaara\n(2004)',  
'34.\nSwades: We, the People\n(2004)',  
'35.\nMain, Meri Patni... Aur Woh!\n(2005)',  
'36.\nGaram Masala\n(2005)',  
'37.\nDon\n(I) (2006)',  
'38.\nGuru\n(2007)',  
'39.\nChak De! India\n(2007)',  
'40.\nNo Smoking\n(2007)',  
'41.\nA Wednesday\n(2008)',  
'42.\nn3 Idiots\n(2009)',  
'43.\nMy Name Is Khan\n(2010)',  
'44.\nGuzaarish\n(2010)',  
'45.\nZindagi Na Milegi Dobara\n(2011)',  
'46.\nRockstar\n(2011)',  
'47.\nDon 2\n(2011)',  
'48.\nPiku\n(2015)',  
'49.\nAirlift\n(2016)',  
'50.\nAndhadhun\n(2018)']
```

```
In [88]: movies1=[]
for movie1 in scraped_movies1:
    movie1=movie1.get_text().replace('\n',"") # \n is for next line
    movie1=movie1.strip(" ") # for remove spaces
    movies1.append(movie1)
movies1
```

```
Out[88]: ['1.Awaara(1951)',  
          '2.Shree 420(1955)',  
          '3.Pyaasa(1957)',  
          '4.Madhumati(1958)',  
          '5.Kanoon(1960)',  
          '6.Mughal-E-Azam(1960)',  
          '7.Kala Bazar(1960)',  
          '8.Woh Kaun Thi?(1964)',  
          '9.Guide(1965)',  
          '10.Amrupali(1966)',  
          '11.Anand(1971)',  
          '12.Rajnigandha(1974)',  
          '13.Ar vind Desai Ki Ajeeb Dastaan(1978)',  
          '14.Satte Pe Satta(1982)',  
          '15.Disco Dancer(1982)',  
          '16.Jaane Bhi Do Yaaro(1983)',  
          '17.Ardh Satya(1983)',  
          '18.Katha(1983)',  
          '19.Parinda(1989)',  
          '20.Baazigar(1993)',  
          '21.Darr(1993)',  
          '22.Rangeela(1995)',  
          '23.Dilwale Dulhania Le Jayenge(1995)',  
          '24.Border(I) (1997)',  
          '25.Satya(1998)',  
          '26.Kaun?(1999)',  
          '27.Hera Pheri(2000)',  
          '28.Lagaan: Once Upon a Time in India(2001)',  
          '29.Company(2002)',  
          '30.Darna Mana Hai(2003)',  
          '31.Munna Bhai M.B.B.S.(2003)',  
          '32.Khakee(2004)',  
          '33.Veer-Zaara(2004)',  
          '34.Swades: We, the People(2004)',  
          '35.Main, Meri Patni... Aur Woh!(2005)',  
          '36.Garam Masala(2005)',  
          '37.Don(I) (2006)',  
          '38.Guru(2007)',  
          '39.Chak De! India(2007)',  
          '40.No Smoking(2007)',  
          '41.A Wednesday(2008)',  
          '42.3 Idiots(2009)',  
          '43.My Name Is Khan(2010)',  
          '44.Guzaarish(2010)',  
          '45.Zindagi Na Milegi Dobara(2011)',  
          '46.Rockstar(2011)',  
          '47.Don 2(2011)',  
          '48.Piku(2015)',  
          '49.Airlift(2016)',  
          '50.Andhadhun(2018)']
```

```
In [101]: # get rating for movies
scraped_ratings1=story1.find_all('div',class_='ipl-rating-star small')
scraped_ratings1
```

```
Out[101]: [<div class="ipl-rating-star small">
    <span class="ipl-rating-star__star">
        <svg class="ipl-icon ipl-star-icon" fill="#000000" height="24" viewBox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg">
            <path d="M0 0h24v24H0z" fill="none"></path>
            <path d="M12 17.27L18.18 21l-1.64-7.03L22 9.24l-7.19-.61L12 2 9.19 8.63 2 9.24l5.46 4.73L5.82 21z"></path>
            <path d="M0 0h24v24H0z" fill="none"></path>
        </svg>
    </span>
    <span class="ipl-rating-star__rating">7.8</span>
</div>,
<div class="ipl-rating-star small">
    <span class="ipl-rating-star__star">
        <svg class="ipl-icon ipl-star-icon" fill="#000000" height="24" viewBox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg">
            <path d="M0 0h24v24H0z" fill="none"></path>
            <path d="M12 17.27L18.18 21l-1.64-7.03L22 9.24l-7.19-.61L12 2 9.19 8.63 2 9.24l5.46 4.73L5.82 21z"></path>
            <path d="M0 0h24v24H0z" fill="none"></path>
        </svg>
    </span>
    <span class="ipl-rating-star__rating">7.8</span>
</div>]
```

```
In [102]: # parse ratings
ratings1=[]
for rating1 in scraped_ratings1:
    rating1=rating1.get_text().replace('\n','')
    ratings1.append(rating1)
ratings1
```

```
Out[102]: ['7.8',
 '7.9',
 '8.3',
 '7.8',
 '7.9',
 '8.1',
 '7.1',
 '7.5',
 '8.4',
 '6.7',
 '8.1',
 '7.3',
 '7.2',
 '7.2',
 '6.4',
 '8.3',
 '8.1',
 '7.7',
 '7.8',
 '7.6',
 '7.6',
 '7.4',
 '8',
 '7.9',
 '8.3',
 '7.8',
 '8.1',
 '8.1',
 '8',
 '6.3',
 '8.1',
 '7.4',
 '7.8',
 '8.2',
 '7.2',
 '6.7',
 '7.1',
 '7.7',
 '8.1',
 '7.3',
 '8.1',
 '8.4',
 '7.9',
 '7.4',
 '8.2',
 '7.7',
 '7',
 '7.6',
 '7.9',
 '8.2']
```

In [97]: # get release year for movies

```
scraped_release_years1=story1.find_all('span',class_='lister-item-year text-mu  
scraped_release_years1
```

Out[97]: [<span class="lister-item-year text-muted unbold">(1951)</span>,

```
<span class="lister-item-year text-muted unbold">(1955)</span>,  
<span class="lister-item-year text-muted unbold">(1957)</span>,  
<span class="lister-item-year text-muted unbold">(1958)</span>,  
<span class="lister-item-year text-muted unbold">(1960)</span>,  
<span class="lister-item-year text-muted unbold">(1960)</span>,  
<span class="lister-item-year text-muted unbold">(1960)</span>,  
<span class="lister-item-year text-muted unbold">(1964)</span>,  
<span class="lister-item-year text-muted unbold">(1965)</span>,  
<span class="lister-item-year text-muted unbold">(1966)</span>,  
<span class="lister-item-year text-muted unbold">(1971)</span>,  
<span class="lister-item-year text-muted unbold">(1974)</span>,  
<span class="lister-item-year text-muted unbold">(1978)</span>,  
<span class="lister-item-year text-muted unbold">(1982)</span>,  
<span class="lister-item-year text-muted unbold">(1982)</span>,  
<span class="lister-item-year text-muted unbold">(1983)</span>,  
<span class="lister-item-year text-muted unbold">(1983)</span>,  
<span class="lister-item-year text-muted unbold">(1983)</span>,  
<span class="lister-item-year text-muted unbold">(1983)</span>,  
<span class="lister-item-year text-muted unbold">(1993)</span>,  
<span class="lister-item-year text-muted unbold">(1993)</span>,  
<span class="lister-item-year text-muted unbold">(1995)</span>,  
<span class="lister-item-year text-muted unbold">(1995)</span>,  
<span class="lister-item-year text-muted unbold">(I) (1997)</span>,  
<span class="lister-item-year text-muted unbold">(1998)</span>,  
<span class="lister-item-year text-muted unbold">(1999)</span>,  
<span class="lister-item-year text-muted unbold">(2000)</span>,  
<span class="lister-item-year text-muted unbold">(2001)</span>,  
<span class="lister-item-year text-muted unbold">(2002)</span>,  
<span class="lister-item-year text-muted unbold">(2003)</span>,  
<span class="lister-item-year text-muted unbold">(2003)</span>,  
<span class="lister-item-year text-muted unbold">(2004)</span>,  
<span class="lister-item-year text-muted unbold">(2004)</span>,  
<span class="lister-item-year text-muted unbold">(2004)</span>,  
<span class="lister-item-year text-muted unbold">(2005)</span>,  
<span class="lister-item-year text-muted unbold">(2005)</span>,  
<span class="lister-item-year text-muted unbold">(I) (2006)</span>,  
<span class="lister-item-year text-muted unbold">(2007)</span>,  
<span class="lister-item-year text-muted unbold">(2007)</span>,  
<span class="lister-item-year text-muted unbold">(2007)</span>,  
<span class="lister-item-year text-muted unbold">(2008)</span>,  
<span class="lister-item-year text-muted unbold">(2009)</span>,  
<span class="lister-item-year text-muted unbold">(2010)</span>,  
<span class="lister-item-year text-muted unbold">(2010)</span>,  
<span class="lister-item-year text-muted unbold">(2011)</span>,  
<span class="lister-item-year text-muted unbold">(2011)</span>,  
<span class="lister-item-year text-muted unbold">(2011)</span>,  
<span class="lister-item-year text-muted unbold">(2015)</span>,  
<span class="lister-item-year text-muted unbold">(2016)</span>,  
<span class="lister-item-year text-muted unbold">(2018)</span>]
```

```
In [98]: # parse release years
years_of_release1=[]
for release_year1 in scraped_release_years1:
    release_year1=release_year1.get_text().replace('\n','')
    years_of_release1.append(release_year1)
years_of_release1
```

```
Out[98]: [ '(1951)',  
          '(1955)',  
          '(1957)',  
          '(1958)',  
          '(1960)',  
          '(1960)',  
          '(1960)',  
          '(1964)',  
          '(1965)',  
          '(1966)',  
          '(1971)',  
          '(1974)',  
          '(1978)',  
          '(1982)',  
          '(1982)',  
          '(1983)',  
          '(1983)',  
          '(1983)',  
          '(1989)',  
          '(1993)',  
          '(1993)',  
          '(1995)',  
          '(1995)',  
          '(I) (1997)',  
          '(1998)',  
          '(1999)',  
          '(2000)',  
          '(2001)',  
          '(2002)',  
          '(2003)',  
          '(2003)',  
          '(2004)',  
          '(2004)',  
          '(2004)',  
          '(2004)',  
          '(2005)',  
          '(2005)',  
          '(I) (2006)',  
          '(2007)',  
          '(2007)',  
          '(2007)',  
          '(2008)',  
          '(2009)',  
          '(2010)',  
          '(2010)',  
          '(2011)',  
          '(2011)',  
          '(2011)',  
          '(2015)',  
          '(2016)',  
          '(2018)']
```

```
In [103]: import pandas as pd
data1=pd.DataFrame()
data1[ 'Movie Names']=movies1
data1[ 'Year of Release']=years_of_release1
data1[ 'Rating']=ratings1
data1
```

Out[103]:

	Movie Names	Year of Release	Rating
0	1.Awaara(1951)	(1951)	7.8
1	2.Shree 420(1955)	(1955)	7.9
2	3.Pyaasa(1957)	(1957)	8.3
3	4.Madhumati(1958)	(1958)	7.8
4	5.Kanoon(1960)	(1960)	7.9
5	6.Mughal-E-Azam(1960)	(1960)	8.1
6	7.Kala Bazar(1960)	(1960)	7.1
7	8.Woh Kaun Thi?(1964)	(1964)	7.5
8	9.Guide(1965)	(1965)	8.4
9	10.Amrapali(1966)	(1966)	6.7
10	11.Anand(1971)	(1971)	8.1
11	12.Rajnigandha(1974)	(1974)	7.3
12	13.Ar vind Desai Ki Ajeeb Dastaan(1978)	(1978)	7.2
13	14.Satte Pe Satta(1982)	(1982)	7.2
14	15.Disco Dancer(1982)	(1982)	6.4
15	16.Jaane Bhi Do Yaaro(1983)	(1983)	8.3
16	17.Ardh Satya(1983)	(1983)	8.1
17	18.Katha(1983)	(1983)	7.7
18	19.Parinda(1989)	(1989)	7.8
19	20.Baazigar(1993)	(1993)	7.6
20	21.Darr(1993)	(1993)	7.6
21	22.Rangeela(1995)	(1995)	7.4
22	23.Dilwale Dulhania Le Jayenge(1995)	(1995)	8
23	24.Border(I) (1997)	(I) (1997)	7.9
24	25.Satya(1998)	(1998)	8.3
25	26.Kaun?(1999)	(1999)	7.8
26	27.Hera Pheri(2000)	(2000)	8.1
27	28.Lagaan: Once Upon a Time in India(2001)	(2001)	8.1
28	29.Company(2002)	(2002)	8
29	30.Darna Mana Hai(2003)	(2003)	6.3
30	31.Munna Bhai M.B.B.S.(2003)	(2003)	8.1
31	32.Khakee(2004)	(2004)	7.4
32	33.Veer-Zaara(2004)	(2004)	7.8
33	34.Swades: We, the People(2004)	(2004)	8.2
34	35.Main, Meri Patni... Aur Woh!(2005)	(2005)	7.2
35	36.Garam Masala(2005)	(2005)	6.7

	Movie Names	Year of Release	Rating
36	37.Don(I) (2006)	(I) (2006)	7.1
37	38.Guru(2007)	(2007)	7.7
38	39.Chak De! India(2007)	(2007)	8.1
39	40.No Smoking(2007)	(2007)	7.3
40	41.A Wednesday(2008)	(2008)	8.1
41	42.3 Idiots(2009)	(2009)	8.4
42	43.My Name Is Khan(2010)	(2010)	7.9
43	44.Guzaarish(2010)	(2010)	7.4
44	45.Zindagi Na Milegi Dobara(2011)	(2011)	8.2
45	46.Rockstar(2011)	(2011)	7.7
46	47.Don 2(2011)	(2011)	7
47	48.Piku(2015)	(2015)	7.6
48	49.Airlift(2016)	(2016)	7.9
49	50.Andhadhun(2018)	(2018)	8.2

## Answer No.04

```
In [4]: page2=requests.get('https://presidentofindia.nic.in/former-presidents.htm')
page2
```

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

```
In [5]: soup2=BeautifulSoup(page2.content,'html.parser') # It shows the source code of
print(soup2.prettify())
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<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 content="no-cache" http-equiv="pragma"/>
<!--<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=EmulateIE10" 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 Viewport -->
<!--<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" />-->
    <!-- Start JS -->
```

```
In [15]: scraped_names2=soup2.find_all('div',class_='presidentListing')
scraped_names2
```

```
Out[15]: [<div class="presidentListing">
    <h3>Shri Ram Nath Kovind (birth - 1945)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 2017 to 25 July, 2022
  </p>
  <p><a href="https://ramnathkovind.nic.in" target="_blank">https://ramnathkov
ind.nic.in</a></p>
</div>,
<div class="presidentListing">
    <h3>Shri Pranab Mukherjee (1935-2020)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 2012 to 25 July, 2017
  </p>
  <p><a href="http://pranabmukherjee.nic.in" target="_blank">http://pranabmukh
erjee.nic.in</a></p>
</div>,
<div class="presidentListing">
    <h3>Smt Pratibha Devi Singh Patil (birth - 1934)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 2007 to 25 July, 2012
  </p>
  <p><a href="http://pratibhapatil.nic.in" target="_blank">http://pratibhapati
l.nic.in</a></p>
</div>,
<div class="presidentListing">
    <h3>DR. A.P.J. Abdul Kalam (1931-2015)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 2002 to 25 July, 2007
  </p>
  <p><a href="http://abdulkalam.nic.in" target="_blank">http://abdulkalam.nic.
in</a></p>
</div>,
<div class="presidentListing">
    <h3>Shri K. R. Narayanan (1920 - 2005)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 1997 to 25 July, 2002
  </p>
</div>,
<div class="presidentListing">
    <h3>Dr Shankar Dayal Sharma (1918-1999)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 1992 to 25 July, 1997
  </p>
</div>,
<div class="presidentListing">
    <h3>Shri R Venkataraman (1910-2009)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 1987 to 25 July, 1992
  </p>
</div>,
<div class="presidentListing">
    <h3>Giani Zail Singh (1916-1994)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 1982 to 25 July, 1987
  </p>
</div>,
<div class="presidentListing">
    <h3>Shri Neelam Sanjiva Reddy (1913-1996)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 1977 to 25 July, 1982
  </p>
</div>,
<div class="presidentListing">
    <h3>Dr. Fakhruddin Ali Ahmed (1905-1977)</h3>
    <p><span class="terms">Term of Office:</span> 24 August, 1974 to 11 Februar
y, 1977</p>
```

```

</div>,
<div class="presidentListing">
<h3>Shri Varahagiri Venkata Giri (1894-1980)</h3>
<p><span class="terms">Term of Office:</span> 3 May, 1969 to 20 July, 1969 a
nd 24 August, 1969 to 24 August, 1974</p>
</div>,
<div class="presidentListing">
<h3>Dr. Zakir Husain (1897-1969)</h3>
<p><span class="terms">Term of Office:</span> 13 May, 1967 to 3 May, 1969</p
>
</div>,
<div class="presidentListing">
<h3>Dr. Sarvepalli Radhakrishnan (1888-1975)</h3>
<p><span class="terms">Term of Office:</span> 13 May, 1962 to 13 May, 1967</p
>
</div>,
<div class="presidentListing">
<h3>Dr. Rajendra Prasad (1884-1963) </h3>
<p><span class="terms">Term of Office:</span> 26 January, 1950 to 13 May, 19
62</p>
</div>]

```

In [16]:

```

scraped_president_name2=[]
for scraped_name2 in scraped_names2:
    scraped_name2=scraped_name2.get_text().replace('\n',"")
    scraped_president_name2.append(scraped_name2)
scraped_president_name2

```

Out[16]:

```

['Shri Ram Nath Kovind (birth - 1945)Term of Office: 25 July, 2017 to 25 Jul
y, 2022 https://ramnathkovind.nic.in', (https://ramnathkovind.nic.in,),
 'Shri Pranab Mukherjee (1935-2020)Term of Office: 25 July, 2012 to 25 July,
2017 http://pranabmukherjee.nic.in', (http://pranabmukherjee.nic.in,),
 'Smt Pratibha Devi Singh Patil (birth - 1934)Term of Office: 25 July, 2007 to
25 July, 2012 http://pratibhapatil.nic.in', (http://pratibhapatil.nic.in,),
 'DR. A.P.J. Abdul Kalam (1931-2015)Term of Office: 25 July, 2002 to 25 July,
2007 http://abdulkalam.nic.in', (http://abdulkalam.nic.in,),
 'Shri K. R. Narayanan (1920 - 2005)Term of Office: 25 July, 1997 to 25 July,
2002 ',
 'Dr Shankar Dayal Sharma (1918-1999)Term of Office: 25 July, 1992 to 25 Jul
y, 1997 ',
 'Shri R Venkataraman (1910-2009)Term of Office: 25 July, 1987 to 25 July, 19
92 ',
 'Giani Zail Singh (1916-1994)Term of Office: 25 July, 1982 to 25 July, 1987
',
 'Shri Neelam Sanjiva Reddy (1913-1996)Term of Office: 25 July, 1977 to 25 Ju
ly, 1982 ',
 'Dr. Fakhruddin Ali Ahmed (1905-1977)Term of Office: 24 August, 1974 to 11 F
ebruary, 1977',
 'Shri Varahagiri Venkata Giri (1894-1980)Term of Office: 3 May, 1969 to 20 J
uly, 1969 and 24 August, 1969 to 24 August, 1974',
 'Dr. Zakir Husain (1897-1969)Term of Office: 13 May, 1967 to 3 May, 1969',
 'Dr. Sarvepalli Radhakrishnan (1888-1975)Term of Office: 13 May, 1962 to 13
May, 1967',
 'Dr. Rajendra Prasad (1884-1963) Term of Office: 26 January, 1950 to 13 May,
1962']

```

```
In [17]: scraped_names2=soup2.find_all('h3')
scraped_names2
```

```
Out[17]: [<h3>Shri Ram Nath Kovind (birth - 1945)</h3>,
<h3>Shri Pranab Mukherjee (1935-2020)</h3>,
<h3>Smt Pratibha Devi Singh Patil (birth - 1934)</h3>,
<h3>DR. A.P.J. Abdul Kalam (1931-2015)</h3>,
<h3>Shri K. R. Narayanan (1920 - 2005)</h3>,
<h3>Dr Shankar Dayal Sharma (1918-1999)</h3>,
<h3>Shri R Venkataraman (1910-2009)</h3>,
<h3>Giani Zail Singh (1916-1994)</h3>,
<h3>Shri Neelam Sanjiva Reddy (1913-1996)</h3>,
<h3>Dr. Fakhruddin Ali Ahmed (1905-1977)</h3>,
<h3>Shri Varahagiri Venkata Giri (1894-1980)</h3>,
<h3>Dr. Zakir Husain (1897-1969)</h3>,
<h3>Dr. Sarvepalli Radhakrishnan (1888-1975)</h3>,
<h3>Dr. Rajendra Prasad (1884-1963) </h3>]
```

```
In [18]: scraped_president_name2=[]
for scraped_name2 in scraped_names2:
    scraped_name2=scraped_name2.get_text().replace('\n','')
    scraped_president_name2.append(scraped_name2)
scraped_president_name2
```

```
Out[18]: ['Shri Ram Nath Kovind (birth - 1945)',
 'Shri Pranab Mukherjee (1935-2020)',
 'Smt Pratibha Devi Singh 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 [19]: len(scraped_president_name2)
```

```
Out[19]: 14
```

```
In [20]: scraped_termofoffice2=soup2.find_all('div',class_='presidentListing')
scraped_termofoffice2
```

```
Out[20]: [<div class="presidentListing">
    <h3>Shri Ram Nath Kovind (birth - 1945)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 2017 to 25 July, 2022
  </p>
  <p><a href="https://ramnathkovind.nic.in" target="_blank">https://ramnathkov
ind.nic.in</a></p>
</div>,
<div class="presidentListing">
    <h3>Shri Pranab Mukherjee (1935-2020)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 2012 to 25 July, 2017
  </p>
  <p><a href="http://pranabmukherjee.nic.in" target="_blank">http://pranabmukh
erjee.nic.in</a></p>
</div>,
<div class="presidentListing">
    <h3>Smt Pratibha Devi Singh Patil (birth - 1934)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 2007 to 25 July, 2012
  </p>
  <p><a href="http://pratibhapatil.nic.in" target="_blank">http://pratibhapati
l.nic.in</a></p>
</div>,
<div class="presidentListing">
    <h3>DR. A.P.J. Abdul Kalam (1931-2015)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 2002 to 25 July, 2007
  </p>
  <p><a href="http://abdulkalam.nic.in" target="_blank">http://abdulkalam.nic.
in</a></p>
</div>,
<div class="presidentListing">
    <h3>Shri K. R. Narayanan (1920 - 2005)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 1997 to 25 July, 2002
  </p>
</div>,
<div class="presidentListing">
    <h3>Dr Shankar Dayal Sharma (1918-1999)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 1992 to 25 July, 1997
  </p>
</div>,
<div class="presidentListing">
    <h3>Shri R Venkataraman (1910-2009)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 1987 to 25 July, 1992
  </p>
</div>,
<div class="presidentListing">
    <h3>Giani Zail Singh (1916-1994)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 1982 to 25 July, 1987
  </p>
</div>,
<div class="presidentListing">
    <h3>Shri Neelam Sanjiva Reddy (1913-1996)</h3>
    <p><span class="terms">Term of Office:</span> 25 July, 1977 to 25 July, 1982
  </p>
</div>,
<div class="presidentListing">
    <h3>Dr. Fakhruddin Ali Ahmed (1905-1977)</h3>
    <p><span class="terms">Term of Office:</span> 24 August, 1974 to 11 Februar
y, 1977</p>
```

```
</div>,
<div class="presidentListing">
<h3>Shri Varahagiri Venkata Giri (1894-1980)</h3>
<p><span class="terms">Term of Office:</span> 3 May, 1969 to 20 July, 1969 a
nd 24 August, 1969 to 24 August, 1974</p>
</div>,
<div class="presidentListing">
<h3>Dr. Zakir Husain (1897-1969)</h3>
<p><span class="terms">Term of Office:</span> 13 May, 1967 to 3 May, 1969</p
>
</div>,
<div class="presidentListing">
<h3>Dr. Sarvepalli Radhakrishnan (1888-1975)</h3>
<p><span class="terms">Term of Office:</span> 13 May, 1962 to 13 May, 1967</p
>
</div>,
<div class="presidentListing">
<h3>Dr. Rajendra Prasad (1884-1963) </h3>
<p><span class="terms">Term of Office:</span> 26 January, 1950 to 13 May, 19
62</p>
</div>]
```

In [21]: scraped\_termofoffice2=soup2.find\_all('p')  
scraped\_termofoffice2

Out[21]: [<p><span class="terms">Term of Office:</span> 25 July, 2017 to 25 July, 2022</p>, <p><a href="https://ramnathkovind.nic.in" target="\_blank">https://ramnathkovind.nic.in</a></p>, <p><span class="terms">Term of Office:</span> 25 July, 2012 to 25 July, 2017</p>, <p><a href="http://pranabmukherjee.nic.in" target="\_blank">http://pranabmukherjee.nic.in</a></p>, <p><span class="terms">Term of Office:</span> 25 July, 2007 to 25 July, 2012</p>, <p><a href="http://pratibhapatil.nic.in" target="\_blank">http://pratibhapatil.nic.in</a></p>, <p><span class="terms">Term of Office:</span> 25 July, 2002 to 25 July, 2007</p>, <p><a href="http://abdulkalam.nic.in" target="\_blank">http://abdulkalam.nic.in</a></p>, <p><span class="terms">Term of Office:</span> 25 July, 1997 to 25 July, 2002</p>, <p><span class="terms">Term of Office:</span> 25 July, 1992 to 25 July, 1997</p>, <p><span class="terms">Term of Office:</span> 25 July, 1987 to 25 July, 1992</p>, <p><span class="terms">Term of Office:</span> 25 July, 1982 to 25 July, 1987</p>, <p><span class="terms">Term of Office:</span> 25 July, 1977 to 25 July, 1982</p>, <p><span class="terms">Term of Office:</span> 24 August, 1974 to 11 Februar y, 1977</p>, <p><span class="terms">Term of Office:</span> 3 May, 1969 to 20 July, 1969 a nd 24 August, 1969 to 24 August, 1974</p>, <p><span class="terms">Term of Office:</span> 13 May, 1967 to 3 May, 1969</p>, <p><span class="terms">Term of Office:</span> 13 May, 1962 to 13 May, 1967</p>, <p><span class="terms">Term of Office:</span> 26 January, 1950 to 13 May, 19 62</p>, <p class="copyright">Copyright © 2022 The Rashtrapati Bhavan.</p>, <p class="lastUpdated"><span id="ContentPlaceHolder1\_UserPageLastUpdateDate1 \_lblPageLastUpdateDate">Page last updated on: <strong>22-July-2022 15:58 PM</strong></span></p>, <p class="govNic"><a href="http://www.nic.in/" target="\_blank"></a> Website hosted by National Informatics Centre. Website Content provided by the President's Secretariat.</p>]

In [65]: scraped\_termofoffice2=soup2.find\_all('p')  
scraped\_termofoffice2

Out[65]: [

<span class="terms">Term of Office:</span> 25 July, 2017 to 25 July, 2022</p>,  
<p><a href="https://ramnathkovind.nic.in" target="\_blank">https://ramnathkovind.nic.in</a></p>,  
<p><span class="terms">Term of Office:</span> 25 July, 2012 to 25 July, 2017</p>,  
<p><a href="http://pranabmukherjee.nic.in" target="\_blank">http://pranabmukherjee.nic.in</a></p>,  
<p><span class="terms">Term of Office:</span> 25 July, 2007 to 25 July, 2012</p>,  
<p><a href="http://pratibhapatil.nic.in" target="\_blank">http://pratibhapatil.nic.in</a></p>,  
<p><span class="terms">Term of Office:</span> 25 July, 2002 to 25 July, 2007</p>,  
<p><a href="http://abdulkalam.nic.in" target="\_blank">http://abdulkalam.nic.in</a></p>,  
<p><span class="terms">Term of Office:</span> 25 July, 1997 to 25 July, 2002</p>,  
<p><span class="terms">Term of Office:</span> 25 July, 1992 to 25 July, 1997</p>,  
<p><span class="terms">Term of Office:</span> 25 July, 1987 to 25 July, 1992</p>,  
<p><span class="terms">Term of Office:</span> 25 July, 1982 to 25 July, 1987</p>,  
<p><span class="terms">Term of Office:</span> 25 July, 1977 to 25 July, 1982</p>,  
<p><span class="terms">Term of Office:</span> 24 August, 1974 to 11 Februar y, 1977</p>,  
<p><span class="terms">Term of Office:</span> 3 May, 1969 to 20 July, 1969 a nd 24 August, 1969 to 24 August, 1974</p>,  
<p><span class="terms">Term of Office:</span> 13 May, 1967 to 3 May, 1969</p>,  
<p><span class="terms">Term of Office:</span> 13 May, 1962 to 13 May, 1967</p>,  
<p><span class="terms">Term of Office:</span> 26 January, 1950 to 13 May, 19 62</p>,  
<p class="copyright">Copyright © 2022 The Rashtrapati Bhavan.</p>,  
<p class="lastUpdated"><span id="ContentPlaceHolder1\_UserPageLastUpdateDate1 \_lblPageLastUpdateDate">Page last updated on: <strong>22-July-2022 15:58 PM</strong></span></p>,  
<p class="govNic"><a href="http://www.nic.in/" target="\_blank"></a>  
Website hosted by National Informatics Centre. Website Content provided by the President's Secretariat.</p>]

```
In [36]: scraped_tof2=[]
for scraped_f2 in scraped_termofoffice2:
    scraped_f2=scraped_f2.get_text().replace('\n','')
    scraped_f2=scraped_f2.replace('https://','')
    scraped_tof2.append(scraped_f2)
scraped_tof2
```

```
Out[36]: ['Term of Office: 25 July, 2017 to 25 July, 2022 ',
 'ramnathkovind.nic.in',
 'Term of Office: 25 July, 2012 to 25 July, 2017 ',
 'http://pranabmukherjee.nic.in',
 'Term of Office: 25 July, 2007 to 25 July, 2012 ',
 'http://pratibhapatil.nic.in',
 'Term of Office: 25 July, 2002 to 25 July, 2007 ',
 'http://abdulkalam.nic.in',
 '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',
 'Copyright © 2022 The Rashtrapati Bhavan.',
 'Page last updated on: 22-July-2022 15:58 PM',
 '\r Website hosted by National Informatics Centre. Website Content provided by the President's Secretariat.]
```

```
In [189]: scraped_tof2=[]
for scraped_f2 in scraped_termofoffice2:
    scraped_f2=scraped_f2.get_text().replace('\n','')
    scraped_tof2.append(scraped_f2)
scraped_tof2
```

```
Out[189]: ['Term of Office: 25 July, 2017 to 25 July, 2022 ',
'https://ramnathkovind.nic.in',
'Term of Office: 25 July, 2012 to 25 July, 2017 ',
'http://pranabmukherjee.nic.in',
'Term of Office: 25 July, 2007 to 25 July, 2012 ',
'http://pratibhapatil.nic.in',
'Term of Office: 25 July, 2002 to 25 July, 2007 ',
'http://abdulkalam.nic.in',
'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',
'Copyright © 2022 The Rashtrapati Bhavan.',
'Page last updated on: 22-July-2022 15:58 PM',
"\r Website hosted by National Informatics Centre. Website Content provided by the President's Secretariat."]
```

```
In [57]: len(scraped_tof2)
```

```
Out[57]: 21
```

```
In [34]: tf=scraped_tof2[0:14]
tf
```

```
Out[34]: ['Term of Office: 25 July, 2017 to 25 July, 2022 ',
'https://ramnathkovind.nic.in',
'Term of Office: 25 July, 2012 to 25 July, 2017 ',
'http://pranabmukherjee.nic.in',
'Term of Office: 25 July, 2007 to 25 July, 2012 ',
'http://pratibhapatil.nic.in',
'Term of Office: 25 July, 2002 to 25 July, 2007 ',
'http://abdulkalam.nic.in',
'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']
```

In [31]: `len(tf)`

Out[31]: 14

In [32]: `import pandas as pd`  
`data2=pd.DataFrame()`  
`data2['Name']=scraped_president_name2`  
`data2['Term of office']=tf`  
`data2`

Out[32]:

	Name	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)	<a href="https://ramnathkovind.nic.in">https://ramnathkovind.nic.in</a>
2	Smt Pratibha Devi Singh Patil (birth - 1934)	Term of Office: 25 July, 2012 to 25 July, 2017
3	DR. A.P.J. Abdul Kalam (1931-2015)	<a href="http://pranabmukherjee.nic.in">http://pranabmukherjee.nic.in</a>
4	Shri K. R. Narayanan (1920 - 2005)	Term of Office: 25 July, 2007 to 25 July, 2012
5	Dr Shankar Dayal Sharma (1918-1999)	<a href="http://pratibhapatil.nic.in">http://pratibhapatil.nic.in</a>
6	Shri R Venkataraman (1910-2009)	Term of Office: 25 July, 2002 to 25 July, 2007
7	Giani Zail Singh (1916-1994)	<a href="http://abdulkalam.nic.in">http://abdulkalam.nic.in</a>
8	Shri Neelam Sanjiva Reddy (1913-1996)	Term of Office: 25 July, 1997 to 25 July, 2002
9	Dr. Fakhruddin Ali Ahmed (1905-1977)	Term of Office: 25 July, 1992 to 25 July, 1997
10	Shri Varahagiri Venkata Giri (1894-1980)	Term of Office: 25 July, 1987 to 25 July, 1992
11	Dr. Zakir Husain (1897-1969)	Term of Office: 25 July, 1982 to 25 July, 1987
12	Dr. Sarvepalli Radhakrishnan (1888-1975)	Term of Office: 25 July, 1977 to 25 July, 1982
13	Dr. Rajendra Prasad (1884-1963)	Term of Office: 24 August, 1974 to 11 February...

## Answer N0.05

In [48]: `page3=requests.get('https://www.icc-cricket.com/rankings/mens/team-rankings/odi')`  
`page3`

Out[48]: <Response [200]>

In [49]: `soup3=BeautifulSoup(page3.content) # It shows the source code of the web page  
soup3`

Out[49]: `<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta content="ICC Men's ODI Team Rankings | ICC" name="twitter:title"/>  
 <meta content="website" property="og:type"/>  
 <meta content="summary_large_image" property="twitter:card"/>  
 <meta content="Official International Cricket Council ranking for One Day International (ODI) cricket teams. Discover latest ICC rankings table, predict upcoming matches, see points and ratings for all teams." name="description"/>  
 <meta content="@icc" property="twitter:site"/>  
 <meta content="Official International Cricket Council ranking for One Day International (ODI) cricket teams. Discover latest ICC rankings table, predict upcoming matches, see points and ratings for all teams." name="twitter:description"/>  
 <meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-thumbnail.jpg" name="twitter:image"/>  
 <meta content="ICC Men's ODI Team Rankings | ICC" property="og:title"/>  
 <meta content="https://www.icc-cricket.com/resources/ver/i/elements/default...>`

In [50]: `scraped_team3=soup3.find_all('span',class_='u-hide-phablet')  
scraped_team3`

Out[50]: `[<span class="u-hide-phablet">India</span>,<span class="u-hide-phablet">Australia</span>,<span class="u-hide-phablet">New Zealand</span>,<span class="u-hide-phablet">England</span>,<span class="u-hide-phablet">Pakistan</span>,<span class="u-hide-phablet">South Africa</span>,<span class="u-hide-phablet">Bangladesh</span>,<span class="u-hide-phablet">Sri Lanka</span>,<span class="u-hide-phablet">Afghanistan</span>,<span class="u-hide-phablet">West Indies</span>,<span class="u-hide-phablet">Ireland</span>,<span class="u-hide-phablet">Scotland</span>,<span class="u-hide-phablet">Zimbabwe</span>,<span class="u-hide-phablet">Namibia</span>,<span class="u-hide-phablet">Netherlands</span>,<span class="u-hide-phablet">Oman</span>,<span class="u-hide-phablet">UAE</span>,<span class="u-hide-phablet">United States</span>,<span class="u-hide-phablet">Nepal</span>,<span class="u-hide-phablet">Papua New Guinea</span>]`

```
In [51]: scraped_team_name3=[]
for scraped_3 in scraped_team3:
    scraped_3=scraped_3.get_text().replace('\n',"")
    scraped_team_name3.append(scraped_3)
scraped_team_name3
```

```
Out[51]: ['India',
'Australia',
'New Zealand',
'England',
'Pakistan',
'South Africa',
'Bangladesh',
'Sri Lanka',
'Afghanistan',
'West Indies',
'Ireland',
'Scotland',
'Zimbabwe',
'Namibia',
'Netherlands',
'Oman',
'UAE',
'United States',
'Nepal',
'Papua New Guinea']
```

```
In [52]: scraped_matches3=soup3.find_all('table',class_='table')
scraped_matches3
```

```
Out[52]: [<table class="table">
<thead>
<tr class="table-head">
<th class="table-head__cell u-text-right">Pos</th>
<th class="table-head__cell rankings-table__team">
<span class="u-hide-mobile">Team</span>
<span class="u-show-mobile">T</span>
</th>
<th class="table-head__cell u-center-text">
<span class="u-hide-mobile">Matches</span>
<span class="u-show-mobile">M</span>
</th>
<th class="table-head__cell u-center-text">
<span class="u-hide-mobile">Points</span>
<span class="u-show-mobile">P</span>
</th>
<th class="table-head__cell u-text-right rating">
<span class="u-hide-mobile">Rating</span>
<span class="u-show-mobile">R</span>
</th>
```

```
In [38]: print(len(scraped_teammatches3 ),len(scraped_teammatches3))
```

```
100 100
```

```
In [39]: scraped_points3=soup3.find_all('td',class_='rankings-block--points')
scraped_points3
```

```
Out[39]: [<td class="rankings-block__points">5,010</td>]
```

```
In [119]: scraped_point3=soup3.find_all('tr',class_='table-body')
scraped_point3
```

```
Out[119]: [<tr class="table-body" data-team-id="15">
    <td class="table-body__cell table-body__cell--position u-text-right">2</td>
    <td class="table-body__cell rankings-table__team">
        <span class="flag-15 table-body_logo AUS"></span>
        <span class="u-hide-phablet">Australia</span>
        <span class="u-show-phablet">AUS</span>
    </td>
    <td class="table-body__cell u-center-text">32</td>
    <td class="table-body__cell u-center-text">3,572</td>
    <td class="table-body__cell u-text-right rating">112</td>
</tr>,
<tr class="table-body" data-team-id="16">
    <td class="table-body__cell table-body__cell--position u-text-right">3</td>
    <td class="table-body__cell rankings-table__team">
        <span class="flag-15 table-body_logo NZ"></span>
        <span class="u-hide-phablet">New Zealand</span>
        <span class="u-show-phablet">NZ</span>
    </td>
```

```
In [120]: scraped_team_point3=[]
for scraped_p_3 in scraped_point3:
    scraped_p_3=scraped_p_3.get_text().replace('\n','')
    scraped_team_point3.append(scraped_p_3)
scraped_team_point3
```

```
Out[120]: ['2AustraliaAUS323,572112',
'3New ZealandNZ293,229111',
'4EnglandENG333,656111',
'5PakistanPAK252,649106',
'6South AfricaSA272,775103',
'7BangladeshBAN333,12995',
'8Sri LankaSL342,97688',
'9AfghanistanAFG201,41971',
'10West IndiesWI412,90271',
'11IrelandIRE251,30052',
'12ScotlandSCO311,45947',
'13ZimbabweZIM281,20143',
'14NamibiaNAM2697137',
'15NetherlandsNED2167332',
'16OmanOMA3091931',
'17UAEUAE2569328',
'18United StatesUSA3182126',
'19NepalNEP2845616',
'20Papua New GuineaPNG301284']
```

```
In [123]: scraped_team_points3=[]
for scraped_po_3 in scraped_point3:
    scraped_po_3=scraped_po_3.get_text().replace('\n',' ')
    scraped_team_points3.append(scraped_po_3)
scraped_team_points3
```

```
Out[123]: [' 2  Australia AUS 32 3,572 112 ',
' 3  New Zealand NZ 29 3,229 111 ',
' 4  England ENG 33 3,656 111 ',
' 5  Pakistan PAK 25 2,649 106 ',
' 6  South Africa SA 27 2,775 103 ',
' 7  Bangladesh BAN 33 3,129 95 ',
' 8  Sri Lanka SL 34 2,976 88 ',
' 9  Afghanistan AFG 20 1,419 71 ',
' 10  West Indies WI 41 2,902 71 ',
' 11  Ireland IRE 25 1,300 52 ',
' 12  Scotland SCO 31 1,459 47 ',
' 13  Zimbabwe ZIM 28 1,201 43 ',
' 14  Namibia NAM 26 971 37 ',
' 15  Netherlands NED 21 673 32 ',
' 16  Oman OMA 30 919 31 ',
' 17  UAE UAE 25 693 28 ',
' 18  United States USA 31 821 26 ',
' 19  Nepal NEP 28 456 16 ',
' 20  Papua New Guinea PNG 30 128 4 ]
```

```
In [124]: len(scraped_team_points3)
```

```
Out[124]: 19
```

```
In [125]: scraped_team_points3[0:11]
```

```
Out[125]: [' 2  Australia AUS 32 3,572 112 ',
' 3  New Zealand NZ 29 3,229 111 ',
' 4  England ENG 33 3,656 111 ',
' 5  Pakistan PAK 25 2,649 106 ',
' 6  South Africa SA 27 2,775 103 ',
' 7  Bangladesh BAN 33 3,129 95 ',
' 8  Sri Lanka SL 34 2,976 88 ',
' 9  Afghanistan AFG 20 1,419 71 ',
' 10  West Indies WI 41 2,902 71 ',
' 11  Ireland IRE 25 1,300 52 ',
' 12  Scotland SCO 31 1,459 47 ']
```

```
In [127]: import pandas as pd  
df3=pd.DataFrame({'team,matches, points and rating':scraped_team_points3})  
df3
```

Out[127]: team,matches, points and rating

0	2 Australia AUS 32 3,572 112
1	3 New Zealand NZ 29 3,229 111
2	4 England ENG 33 3,656 111
3	5 Pakistan PAK 25 2,649 106
4	6 South Africa SA 27 2,775 103
5	7 Bangladesh BAN 33 3,129 95
6	8 Sri Lanka SL 34 2,976 88
7	9 Afghanistan AFG 20 1,419 71
8	10 West Indies WI 41 2,902 71
9	11 Ireland IRE 25 1,300 52
10	12 Scotland SCO 31 1,459 47
11	13 Zimbabwe ZIM 28 1,201 43
12	14 Namibia NAM 26 971 37
13	15 Netherlands NED 21 673 32
14	16 Oman OMA 30 919 31
15	17 UAE UAE 25 693 28
16	18 United States USA 31 821 26
17	19 Nepal NEP 28 456 16
18	20 Papua New Guinea PNG 30 128 4

## Answer No.05

```
In [75]: page10=requests.get('https://www.icc-cricket.com/rankings/mens/player-rankings/  
page10
```

Out[75]: <Response [200]>

```
In [76]: soup10=BeautifulSoup(page10.content,'html.parser') # It shows the source code  
soup10
```

```
Out[76]: <!DOCTYPE html>
```

```
<html lang="en">  
<head>  
    <meta content="ICC Men's ODI Bowling | Player Rankings | ICC" name="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, news, highlights, commentary, rankings, videos and fixtures from the International Cricket Council." name="description"/>  
    <meta content="@icc" property="twitter:site"/>  
    <meta content="Official ICC Cricket website - live matches, scores, news, highlights, commentary, rankings, videos and fixtures from the International Cricket Council." name="twitter:description"/>  
    <meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-thumbnail.jpg" name="twitter:image"/>  
    <meta content="ICC Men's ODI Bowling | Player Rankings | ICC" property="og:title"/>
```

```
In [79]: table10 = soup10.find("table",class_="table rankings-table")  
table10
```

```
Out[79]: <table class="table rankings-table">  
    <thead>  
        <tr class="table-head">  
            <th class="table-head__cell u-text-right">Pos</th>  
            <th class="table-head__cell">Player</th>  
            <th class="table-head__cell">Team</th>  
            <th class="table-head__cell table-head__cell--rating">Rating</th>  
            <th class="table-head__cell u-text-right u-hide-phablet">Career Best Rating</th>  
        </tr>  
    </thead>  
    <tr class="rankings-block__banner">  
        <td class="rankings-block__position">  
            <div class="rankings-block__pos-container">  
                <span class="rankings-block__pos-number">  
                    1  
                </span>  
            </div>  
        </td>  
        <div class="ranking-pos no-change">  
            <span class="ranking-pos__no-change-line"></span>  
        </div>  
    </tr>
```

```
In [81]: f_bowler = table10.find("div",class_="rankings-block__banner--name-large").text  
f_bowler
```

```
Out[81]: 'Mohammed Siraj'
```

```
In [82]: b_team= table10.find("div",class_="rankings-block__banner--nationality").text.  
b_team
```

```
Out[82]: 'IND'
```

```
In [83]: b_rating = table10.find("div",class_="rankings-block__banner--rating").text  
b_rating
```

```
Out[83]: '729'
```

```
In [84]: bowlers = table10.find_all("td",class_="table-body__cell rankings-table__name  
bowlers
```

```
Out[84]: [<td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/857">Josh Hazlewood</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/969">Trent Boult</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/490">Mitchell Starc</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/2885">Rashid Khan</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/958">Adam Zampa</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/201">Shakib Al Hasan</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/4530">Shaheen Afridi</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/1594">Mustafizur Rahman</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/4572">Mujeeb Ur Rahman</a>  
    </td>]
```

```
In [86]: bowler = [f_bowler]
for i in bowlers:
    bowler.append(i.text.replace('\n',''))
bowler
```

```
Out[86]: ['Mohammed Siraj',
 'Josh Hazlewood',
 'Trent Boult',
 'Mitchell Starc',
 'Rashid Khan',
 'Adam Zampa',
 'Shakib Al Hasan',
 'Shaheen Afridi',
 'Mustafizur Rahman',
 'Mujeeb Ur Rahman']
```

```
In [88]: teams = table10.find_all("span",class_="table-body__logo-text")[0:9]
teams
```

```
Out[88]: [<span class="table-body__logo-text">AUS</span>,
<span class="table-body__logo-text">NZ</span>,
<span class="table-body__logo-text">AUS</span>,
<span class="table-body__logo-text">AFG</span>,
<span class="table-body__logo-text">AUS</span>,
<span class="table-body__logo-text">BAN</span>,
<span class="table-body__logo-text">PAK</span>,
<span class="table-body__logo-text">BAN</span>,
<span class="table-body__logo-text">AFG</span>]
```

```
In [89]: team=[b_team]
for i in teams:
    team.append(i.text)
team
```

```
Out[89]: ['IND', 'AUS', 'NZ', 'AUS', 'AFG', 'AUS', 'BAN', 'PAK', 'BAN', 'AFG']
```

```
In [91]: ratings = table10.find_all("td",class_="table-body__cell rating")[0:9]
ratings
```

```
Out[91]: [<td class="table-body__cell rating">727</td>,
<td class="table-body__cell rating">708</td>,
<td class="table-body__cell rating">665</td>,
<td class="table-body__cell rating">659</td>,
<td class="table-body__cell rating">655</td>,
<td class="table-body__cell rating">652</td>,
<td class="table-body__cell rating">641</td>,
<td class="table-body__cell rating">638</td>,
<td class="table-body__cell rating">637</td>]
```

```
In [93]: rating = [b_rating]
for i in ratings:
    rating.append(i.text)
rating
```

```
Out[93]: ['729', '727', '708', '665', '659', '655', '652', '641', '638', '637']
```

```
In [94]: data = list(zip(bowler,team, rating))
data
```

```
Out[94]: [('Mohammed Siraj', 'IND', '729'),
('Josh Hazlewood', 'AUS', '727'),
('Trent Boult', 'NZ', '708'),
('Mitchell Starc', 'AUS', '665'),
('Rashid Khan', 'AFG', '659'),
('Adam Zampa', 'AUS', '655'),
('Shakib Al Hasan', 'BAN', '652'),
('Shaheen Afridi', 'PAK', '641'),
('Mustafizur Rahman', 'BAN', '638'),
('Mujeeb Ur Rahman', 'AFG', '637')]
```

```
In [95]: import pandas as pd
df10 = pd.DataFrame(data, columns = ["Bowler Name", "Team", "Ratings"])
df10
```

```
Out[95]:
```

	Bowler Name	Team	Ratings
0	Mohammed Siraj	IND	729
1	Josh Hazlewood	AUS	727
2	Trent Boult	NZ	708
3	Mitchell Starc	AUS	665
4	Rashid Khan	AFG	659
5	Adam Zampa	AUS	655
6	Shakib Al Hasan	BAN	652
7	Shaheen Afridi	PAK	641
8	Mustafizur Rahman	BAN	638
9	Mujeeb Ur Rahman	AFG	637

```
In [96]: page11=requests.get('https://www.icc-cricket.com/rankings/mens/player-rankings')
page11
```

```
Out[96]: <Response [200]>
```

```
In [97]: soup11=BeautifulSoup(page11.content,'html.parser') # It shows the source code  
soup11
```

Out[97]: <!DOCTYPE html>

```
<html lang="en">
<head>
<meta content="ICC Men's ODI Batting | Player Rankings | ICC" name="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, news, highlights, commentary, rankings, videos and fixtures from the International Cricket Council." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official ICC Cricket website - live matches, scores, news, highlights, commentary, rankings, videos and fixtures from the International Cricket Council." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-thumbnail.jpg" name="twitter:image"/>
<meta content="ICC Men's ODI Batting | Player Rankings | ICC" property="og:title"/>
```

```
In [98]: table11 = soup11.find("table",class_="table rankings-table")
table11
```

```
Out[98]: <table class="table rankings-table">
<thead>
<tr class="table-head">
<th class="table-head__cell u-text-right">Pos</th>
<th class="table-head__cell">Player</th>
<th class="table-head__cell">Team</th>
<th class="table-head__cell table-head__cell--rating">Rating</th>
<th class="table-head__cell u-text-right u-hide-phablet">Career Best Rating</th>
</tr>
</thead>
<tr class="rankings-block__banner">
<td class="rankings-block__position">
<div class="rankings-block__pos-container">
<span class="rankings-block__pos-number">
1
</span>
<div class="ranking-pos no-change">
<span class="ranking-pos__no-change-line"></span>

```

```
In [99]: f_batsman = table11.find("div",class_="rankings-block__banner--name-large").text
```

Out[99]: 'Babar Azam'

```
In [100]: bat_team= table11.find("div",class_="rankings-block__banner--nationality").text  
bat_team
```

```
Out[100]: 'PAK'
```

```
In [101]: bat_rating = table11.find("div",class_="rankings-block__banner--rating").text  
bat_rating
```

```
Out[101]: '887'
```

```
In [102]: batsman = table11.find_all("td",class_="table-body__cell rankings-table__name")  
batsman
```

```
Out[102]: [<td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/1277">Rassie van der Dussen</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/170">David Warner</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/834">Quinton de Kock</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/1568">Imam-ul-Haq</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/3761">Shubman Gill</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/164">Virat Kohli</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/271">Steve Smith</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/107">Rohit Sharma</a>  
    </td>,  
    <td class="table-body__cell rankings-table__name name">  
    <a href="/rankings/mens/player-rankings/440">Kane Williamson</a>  
    </td>]
```

```
In [195]: batsmans = []
for bt in batsman:
    batsmans.append(bt.text.replace('\n',''))
batsmans
```

```
AttributeError                                     Traceback (most recent call last)
Input In [195], in <cell line: 2>()
      1 batsmans = []
      2 for bt in batsman:
----> 3     batsmans.append(bt.text.replace('\n', ''))
      4 batsmans

AttributeError: 'str' object has no attribute 'text'
```

```
In [125]: page12=requests.get('https://www.icc-cricket.com/rankings/mens/team-rankings/c  
page12
```

Out[125]: <Response [200]>

```
In [126]: soup12=BeautifulSoup(page12.content,'html.parser') # It shows the source code  
soup12
```

Out[126]: <!DOCTYPE html>

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

```
In [128]: table12 = soup12.find("table",class_="table")
table12
```

```
Out[128]: <table class="table">
<thead>
<tr class="table-head">
<th class="table-head__cell u-text-right">Pos</th>
<th class="table-head__cell rankings-table__team">
<span class="u-hide-mobile">Team</span>
<span class="u-show-mobile">T</span>
</th>
<th class="table-head__cell u-center-text">
<span class="u-hide-mobile">Matches</span>
<span class="u-show-mobile">M</span>
</th>
<th class="table-head__cell u-center-text">
<span class="u-hide-mobile">Points</span>
<span class="u-show-mobile">P</span>
</th>
<th class="table-head__cell u-text-right rating">
<span class="u-hide-mobile">Rating</span>
<span class="u-show-mobile">R</span>
<th>
```

```
In [129]: team_names = table12.find("span",class_="u-hide-phablet").text
team_names
```

```
Out[129]: 'India'
```

```
In [130]: team_matches= table12.find("td",class_="rankings-block__banner--matches").text
team_matches
```

```
Out[130]: '44'
```

```
In [131]: team_points = table12.find("td",class_="rankings-block__banner--points").text
team_points
```

```
Out[131]: '5,010'
```

```
In [135]: team_ratings = table12.find("td",class_="rankings-block__banner--rating u-text")
team_ratings
```

```
Out[135]: ''
```

114

```
In [183]: whole_team=table12.find_all("span",class_="u-hide-phablet")[0:10]
whole_team
```

```
Out[183]: [<span class="u-hide-phablet">India</span>,
<span class="u-hide-phablet">Australia</span>,
<span class="u-hide-phablet">New Zealand</span>,
<span class="u-hide-phablet">England</span>,
<span class="u-hide-phablet">Pakistan</span>,
<span class="u-hide-phablet">South Africa</span>,
<span class="u-hide-phablet">Bangladesh</span>,
<span class="u-hide-phablet">Sri Lanka</span>,
<span class="u-hide-phablet">Afghanistan</span>,
<span class="u-hide-phablet">West Indies</span>]
```

```
In [184]: all_team = []
for i in whole_team:
    all_team.append(i.text.replace('\n',''))
all_team
```

```
Out[184]: ['India',
'Australia',
'New Zealand',
'England',
'Pakistan',
'South Africa',
'Bangladesh',
'Sri Lanka',
'Afghanistan',
'West Indies']
```

```
In [185]: len(all_team)
```

```
Out[185]: 10
```

```
In [168]: whole_matches=table12.find_all("td",class_="table-body__cell u-center-text")[0:10]
whole_matches
```

```
Out[168]: [<td class="table-body__cell u-center-text">32</td>,
<td class="table-body__cell u-center-text">3,572</td>,
<td class="table-body__cell u-center-text">29</td>,
<td class="table-body__cell u-center-text">3,229</td>,
<td class="table-body__cell u-center-text">33</td>,
<td class="table-body__cell u-center-text">3,656</td>,
<td class="table-body__cell u-center-text">25</td>,
<td class="table-body__cell u-center-text">2,649</td>,
<td class="table-body__cell u-center-text">27</td>]
```

```
In [158]: all_matches = []
for mt in whole_matches:
    all_matches.append(mt.text.replace('\n',''))
all_matches
```

```
Out[158]: ['32', '3,572', '29', '3,229', '33', '3,656', '25', '2,649', '27', '2,775']
```

```
In [174]: len(all_matches)
```

```
Out[174]: 10
```

```
In [167]: whole_points=table12.find_all("td",class_="table-body__cell u-center-text")[0:whole_points]
```

```
Out[167]: [<td class="table-body__cell u-center-text">32</td>,
             <td class="table-body__cell u-center-text">3,572</td>,
             <td class="table-body__cell u-center-text">29</td>,
             <td class="table-body__cell u-center-text">3,229</td>,
             <td class="table-body__cell u-center-text">33</td>,
             <td class="table-body__cell u-center-text">3,656</td>,
             <td class="table-body__cell u-center-text">25</td>,
             <td class="table-body__cell u-center-text">2,649</td>,
             <td class="table-body__cell u-center-text">27</td>]
```

```
In [161]: all_points = []
for mt in whole_points:
    all_points.append(mt.text.replace('\n', ''))
```

```
all_points
```

```
Out[161]: ['32', '3,572', '29', '3,229', '33', '3,656', '25', '2,649', '27', '2,775']
```

```
In [173]: len(all_points)
```

```
Out[173]: 10
```

```
In [165]: whole_ratings=table12.find_all("td",class_="table-body__cell u-text-right rating")whole_ratings
```

```
Out[165]: [<td class="table-body__cell u-text-right rating">112</td>,
             <td class="table-body__cell u-text-right rating">111</td>,
             <td class="table-body__cell u-text-right rating">111</td>,
             <td class="table-body__cell u-text-right rating">106</td>,
             <td class="table-body__cell u-text-right rating">103</td>,
             <td class="table-body__cell u-text-right rating">95</td>,
             <td class="table-body__cell u-text-right rating">88</td>,
             <td class="table-body__cell u-text-right rating">71</td>,
             <td class="table-body__cell u-text-right rating">71</td>]
```

```
In [170]: all_ratings = [team_ratings]
for mt in whole_ratings:
    all_ratings.append(mt.text.replace('\n',''))
all_ratings
```

```
Out[170]: ['114',
'112',
'111',
'111',
'106',
'103',
'95',
'88',
'71',
'71']
```

```
In [171]: len(all_ratings)
```

```
Out[171]: 10
```

```
In [186]: import pandas as pd
df12=pd.DataFrame({'Team':all_team , 'Match':all_matches, 'Point':all_points , 'Rating':all_ratings})
df12
```

```
Out[186]:
```

	Team	Match	Point	Rating
0	India	32	32	114 ...
1	Australia	3,572	3,572	112
2	New Zealand	29	29	111
3	England	3,229	3,229	111
4	Pakistan	33	33	106
5	South Africa	3,656	3,656	103
6	Bangladesh	25	25	95
7	Sri Lanka	2,649	2,649	88
8	Afghanistan	27	27	71
9	West Indies	2,775	2,775	71

```
In [ ]:
```

```
In [ ]:
```

## Answer No.06

```
In [66]: page9=requests.get('https://www.icc-cricket.com/rankings/womens/team-rankings/')
page9
```

```
Out[66]: <Response [200]>
```

```
In [68]: soup9=BeautifulSoup(page9.content,'html.parser') # It shows the source code of
soup9
```

```
Out[68]: <!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 ma
tch cricket teams. Discover latest ICC rankings table, predict upcoming ma
tches, see points and ratings for all teams." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official International Cricket Council rankings for test ma
tch cricket teams. Discover latest ICC rankings table, predict upcoming ma
tches, see points and ratings for all teams." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/defaul
t-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/defaul
t-thumbnail.jpg" property="og:image"/>
<!--> ICC Women's ODI Team Rankings | ICC /<!-->
```

```
In [69]: scraped_team_match4=soup9.find_all('table',class_='table')
scraped_team_match4
```

```
Out[69]: [<table class="table">
<thead>
<tr class="table-head">
<th class="table-head__cell u-text-right">Pos</th>
<th class="table-head__cell rankings-table__team">
<span class="u-hide-mobile">Team</span>
<span class="u-show-mobile">T</span>
</th>
<th class="table-head__cell u-center-text">
<span class="u-hide-mobile">Matches</span>
<span class="u-show-mobile">M</span>
</th>
<th class="table-head__cell u-center-text">
<span class="u-hide-mobile">Points</span>
<span class="u-show-mobile">P</span>
</th>
<th class="table-head__cell u-text-right rating">
<span class="u-hide-mobile">Rating</span>
<span class="u-show-mobile">R</span>
</th>
```

```
In [190]: show_teams = table12.find_all("span",class_="u-hide-phablet")[0:10]
show_teams
```

```
Out[190]: [<span class="u-hide-phablet">India</span>,
<span class="u-hide-phablet">Australia</span>,
<span class="u-hide-phablet">New Zealand</span>,
<span class="u-hide-phablet">England</span>,
<span class="u-hide-phablet">Pakistan</span>,
<span class="u-hide-phablet">South Africa</span>,
<span class="u-hide-phablet">Bangladesh</span>,
<span class="u-hide-phablet">Sri Lanka</span>,
<span class="u-hide-phablet">Afghanistan</span>,
<span class="u-hide-phablet">West Indies</span>]
```

```
In [191]: see_teams=[]
for team15 in show_teams:
    see_teams.append(team15.text.replace('\n',''))
see_teams
```

```
Out[191]: ['India',
'Australia',
'New Zealand',
'England',
'Pakistan',
'South Africa',
'Bangladesh',
'Sri Lanka',
'Afghanistan',
'West Indies']
```

```
In [194]: one_bats = table12.find("div",class_="rankings-block__banner--name-large").tex
one_bats
```

```
-----
AttributeError                                     Traceback (most recent call last)
Input In [194], in <cell line: 1>()
----> 1 one_bats = table12.find("div",class_="rankings-block__banner--name-large").text
      2 one_bats
```

```
AttributeError: 'NoneType' object has no attribute 'text'
```

## Answer No.07

```
In [35]: page4=requests.get('https://www.cnbc.com/world/?region=world')
page4
```

```
Out[35]: <Response [200]>
```

In [36]: `soup4=BeautifulSoup(page4.content) # It shows the source code of the web page  
soup4`

Out[36]: `<!DOCTYPE html>  
<html itemscope="" itemtype="https://schema.org/WebPage" lang="en" prefix="og=https://ogp.me/ns#><head><meta content="telephone=no" name="format-detection"/><style type="text/css">@charset "UTF-8"; .RecaptchaAcknowledgement-acknowledgement{color:#747474;flex:1;font-size:11px;font-weight:600;line-height:15px;margin-bottom:7px;margin-top:24px;width:100%}.RecaptchaAcknowledgement-acknowledgement a{color:#747474;font-weight:500;text-decoration:none}.RecaptchaAcknowledgement-acknowledgement a:hover{color:#747474;text-decoration:underline}.RecaptchaAcknowledgement-acknowledgement a:active{color:#747474}.RecaptchaAcknowledgement-reCaptchaPadding{margin-top:15px}.RecaptchaAcknowledgement-centerAligned{text-align:center}.RecaptchaAcknowledgement-leftAligned{text-align:left}.RecaptchaAcknowledgement-rightAligned{text-align:right}.AuthForms-container{margin:0 auto;padding:0 10px;width:458px}@media (max-width:759px){.AuthForms-container{max-width:458px;padding:20px 0 0;width:100%}}.AuthForms-container .AuthForms-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}`

In [37]: `scraped_headline4=soup4.find_all('div',class_='LatestNews-headlineWrapper')  
scraped_headline4`

Out[37]: `[<div class="LatestNews-headlineWrapper"><span class="LatestNews-wrapper"><time class="LatestNews-timestamp">7 Hours Ago</time></span><a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/cramers-lightning-round-triton-international-is-not-a-buy.html" title="Cramer's lightning round: Triton International is not a buy">Cramer's lightning round: Triton International is not a buy</a></div>,  
<div class="LatestNews-headlineWrapper"><span class="LatestNews-wrapper"><time class="LatestNews-timestamp">8 Hours Ago</time></span><a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/cramers-week-ahead-there-are-plenty-of-high-quality-stocks-to-buy.html" title="Cramer's week ahead: There are plenty of high-quality stocks to buy">Cramer's week ahead: There are plenty of high-quality stocks to buy</a></div>,  
<div class="LatestNews-headlineWrapper"><span class="LatestNews-wrapper"><time class="LatestNews-timestamp">8 Hours Ago</time></span><a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/goldman-sachs-digital-asset-chief-lays-out-blockchain-strategy.html" title="Crypto markets are in a 'flight to quality,' says Goldman digital assets chief ">Crypto markets are in a 'flight to quality,' says Goldman digital assets chief </a></div>,`

```
In [38]: scraped_news_headline4=[]
for scraped_h_4 in scraped_headline4:
    scraped_h_4=scraped_h_4.get_text().replace('\n',"")
    scraped_news_headline4.append(scraped_h_4)
scraped_news_headline4
```

Out[38]: ["7 Hours AgoCramer's lightning round: Triton International is not a buy",  
'8 Hours AgoCramer's week ahead: There are plenty of high-quality stocks to  
buy',  
"8 Hours AgoCrypto markets are in a 'flight to quality,' says Goldman digital  
assets chief ",  
'9 Hours AgoWhat it feels like when your mental health condition becomes a punc  
hline',  
'9 Hours AgoAlec Baldwin attorneys seek to reduce possible penalty in Rust movie set case',  
"9 Hours AgoPro Picks: Watch all of Friday's big stock calls on CNBC",  
'10 Hours AgoU.S. sanctions six Chinese tech companies for supporting spy ba  
lloon programs',  
'11 Hours AgoWall Street: These 2 fairly valued Club stocks will extend stro  
ng starts to 2023',  
"12 Hours AgoThis week's best-performing stocks include a pharma name up mor  
e than 20%",  
'12 Hours AgoWhat to watch in the markets in the week ahead',  
'12 Hours AgoU.S. shoots down second 'high altitude object' days after Chine  
se spy balloon',  
'12 Hours AgoDemocrat and GOP attorneys general face off on abortion pill la  
wsuit',  
"13 Hours AgoTaylor Swift says fans will 'get on it' to reduce egg prices. 3  
things to know",  
'13 Hours AgoGoogle employees slam CEO Sundar Pichai\x80for 'rushed, botche  
d' announcement of GPT rival',  
'14 Hours AgoWall Street's favorite sports-betting stocks ahead of the Super  
Bowl',  
'14 Hours AgoHere's how much money Americans saved at every age in 2022',  
'14 Hours AgoBill Gates: Nuclear waste is not a reason to avoid nuclear ener  
gy',  
'14 Hours AgoStocks making the biggest moves midday: Lyft, Spotify, Expedia,  
Yelp and more ',  
'14 Hours Ago3 investing lessons from the Club's meeting Friday',  
"14 Hours AgoAstronaut Sen. Kelly touts 'stunning' space business growth",  
"14 Hours AgoAir Force's aging E-3 Sentry presents airborne battle readiness  
issues",  
"14 Hours AgoFBI finds another classified document in search of Mike Pence's  
home",  
"14 Hours AgoWe're buying the dip in a beauty stock that stands to benefit f  
rom China reopening",  
'14 Hours AgoNational Archives, museum sued for telling visitors to remove p  
ro-life messages',  
"14 Hours AgoHere's what's happening with home prices right now",  
'14 Hours AgoAmazon is the latest threat to Facebook as ad targeting suffer  
s',  
"15 Hours AgoThis is how Microsoft and Alphabet stack up against each other  
in the A.I. war",  
'15 Hours AgoThese are the best deals on TVs ahead of Super Bowl Sunday',  
'15 Hours AgoTop House Republicans scrutinize SEC investigation of crypto ex  
change FTX',  
"15 Hours AgoSteve Wozniak warns: ChatGPT is 'useful,' but it can 'make horr  
ible mistakes'" ]

```
In [39]: scraped_headlines4=soup4.find_all('a',class_='LatestNews-headline')
scraped_headlines4
```

Out[39]: [<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/cramers-lightning-round-triton-international-is-not-a-buy.html" title="Cramer's lightning round: Triton International is not a buy">Cramer's lightning round: Triton International is not a buy</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/cramers-week-ahead-there-are-plenty-of-high-quality-stocks-to-buy.html" title="Cramer's week ahead: There are plenty of high-quality stocks to buy">Cramer's week ahead: There are plenty of high-quality stocks to buy</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/goldman-sachs-digital-asset-chief-lays-out-blockchain-strategy.html" title="Crypto markets are in a 'flight to quality,' says Goldman digital assets chief ">Crypto markets are in a 'flight to quality,' says Goldman digital assets chief </a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/what-it-feels-like-when-your-condition-becomes-a-punchline.html" title="What it feels like when your mental health condition becomes a punchline">What it feels like when your mental health condition becomes a punchline</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/rust-shooting-alec-baldwin-possible-penalty.html" title="Alec Baldwin attorneys seek to reduce possible penalty in Rust movie set case">Alec Baldwin attorneys seek to reduce possible penalty in Rust movie set case</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/pro-picks-watch-all-of-fridays-big-stock-calls-on-cnbc.html" title="Pro Picks: Watch all of Friday's big stock calls on CNBC">Pro Picks: Watch all of Friday's big stock calls on CNBC</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/us-sanctions-six-chinese-tech-companies-for-supporting-spy-balloon-programs.html" title="U.S. sanctions six Chinese tech companies for supporting spy balloon programs">U.S. sanctions six Chinese tech companies for supporting spy balloon programs</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/2-fairly-valued-club-stocks-will-continue-their-strong-starts-in-2023.html" title="Wall Street: These 2 fairly valued Club stocks will extend strong starts to 2023">Wall Street: These 2 fairly valued Club stocks will extend strong starts to 2023</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/this-weeks-best-performing-stocks-include-this-pharmaceutical-stock.html" title="This week's best-performing stocks include a pharma name up more than 20%">This week's best-performing stocks include a pharma name up more than 20%</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/investors-will-have-their-eyes-on-the-consumer-price-index-in-the-week-ahead.html" title="What to watch in the markets in the week ahead">What to watch in the markets in the week ahead</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/us-shoots-down-second-high-altitude-object-on-bidens-orders.html" title="U.S. shoots down second 'high altitude object' days after Chinese spy balloon">U.S. shoots down second 'high altitude object' days after Chinese spy balloon</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/abortion-pill-states-says-pulling-medication-would-have-devastating-consequences.html" title="Democrat and GOP attorneys general face off on abortion pill lawsuit">Democrat and GOP attorneys general face off on abortion pill lawsuit</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/high-egg-prices-taylor-swift-says-fans-will-reduce-cost.html" title="Taylor Swift says fans will 'get on it' to reduce egg prices. 3 things to know">Taylor Swift says fans will 'get on it' to reduce egg prices. 3 things to know</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/google-employees-slam-ceo-sundar-pichai-for-rushed-bard-announcement.html" title="Go

ogle employees slam CEO Sundar Pichai for 'rushed, botched' announcement of GPT rival">Google employees slam CEO Sundar Pichai for 'rushed, botched' announcement of GPT rival</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/wall-streets-favorite-gambling-stocks-ahead-of-the-super-bowl.html" title="Wall Street's favorite sports-betting stocks ahead of the Super Bowl">Wall Street's favorite sports-betting stocks ahead of the Super Bowl</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/average-amount-of-money-americans-saved-in-2022.html" title="Here's how much money Americans saved at every age in 2022">Here's how much money Americans saved at every age in 2022</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/bill-gates-nuclear-waste-is-not-a-reason-avoid-nuclear-energy.html" title="Bill Gates: Nuclear waste is not a reason to avoid nuclear energy">Bill Gates: Nuclear waste is not a reason to avoid nuclear energy</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/stocks-moving-big-midday-lyft-spot-expe-yelp.html" title="Stocks making the biggest moves midday: Lyft, Spotify, Expedia, Yelp and more ">Stocks making the biggest moves midday: Lyft, Spotify, Expedia, Yelp and more </a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/jim-cramers-investing-club-meeting-friday-be-selective-buying-in-this-market.html" title="3 investing lessons from the Club's meeting Friday">3 investing lessons from the Club's meeting Friday</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/astronaut-sen-mark-kelly-praises-space-business.html" title="Astronaut Sen. Kelly touts 'stunning' space business growth">Astronaut Sen. Kelly touts 'stunning' space business growth</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/air-forces-aging-awacs-stirs-questions-of-airborne-battle-readiness.html" title="Air Force's aging E-3 Sentry presents airborne battle readiness issues">Air Force's aging E-3 Sentry presents airborne battle readiness issues</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/fbi-searching-ex-trump-vp-pence-home-for-classified-docs.html" title="FBI finds another classified document in search of Mike Pence's home">FBI finds another classified document in search of Mike Pence's home</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/buying-the-dip-in-a-beauty-stock-that-stands-to-benefit-from-china-reopening.html" title="We're buying the dip in a beauty stock that stands to benefit from China reopening">We're buying the dip in a beauty stock that stands to benefit from China reopening</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/national-archives-sued-for-abortion-opponent-clothing-flap.html" title="National Archives, museum sued for telling visitors to remove pro-life messages">National Archives, museum sued for telling visitors to remove pro-life messages</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/home-prices-mortgage-rates-fall.html" title="Here's what's happening with home prices right now">Here's what's happening with home prices right now</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/amazon-is-the-latest-threat-to-facebook-as-ad-targeting-suffers.html" title="Amazon is the latest threat to Facebook as ad targeting suffers">Amazon is the latest threat to Facebook as ad targeting suffers</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/heres-how-microsoft-and-alphabet-compare-in-the-ai-war.html" title="This is how Microsoft and Alphabet stack up against each other in the A.I. war">This is how Microsoft and Alphabet stack up against each other in the A.I. war</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/tv-prices-sink-ahead-of-the-super-bowl-here-are-the-best-deals.html" title="These ar

e the best deals on TVs ahead of Super Bowl Sunday">These are the best deals on TVs ahead of Super Bowl Sunday</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/ftx-collapse-top-house-republicans-scrutinize-sec-investigation-of-failed-crypto-exchange.html" title="Top House Republicans scrutinize SEC investigation of crypto exchange FTX">Top House Republicans scrutinize SEC investigation of crypto exchange FTX</a>,  
<a class="LatestNews-headline" href="https://www.cnbc.com/2023/02/10/steve-wozniak-warns-about-ai-chatgpt-can-make-horrible-mistakes.html" title="Steve Wozniak warns: ChatGPT is 'useful,' but it can 'make horrible mistakes'">Steve Wozniak warns: ChatGPT is 'useful,' but it can 'make horrible mistakes'</a>]

In [40]: # scraped headlines

```
scraped_newsheadline4=[]
for scraped_h4 in scraped_headlines4:
    scraped_h4=scraped_h4.get_text().replace('\n','')
    scraped_newsheadline4.append(scraped_h4)
scraped_newsheadline4
```

Out[40]: ["Cramer's lightning round: Triton International is not a buy",  
 'Cramer's week ahead: There are plenty of high-quality stocks to buy',  
 "Crypto markets are in a 'flight to quality,' says Goldman digital assets chie",  
 'What it feels like when your mental health condition becomes a punchline',  
 'Alec Baldwin attorneys seek to reduce possible penalty in Rust movie set ca",  
 "Pro Picks: Watch all of Friday's big stock calls on CNBC",  
 'U.S. sanctions six Chinese tech companies for supporting spy balloon progra",  
 'Wall Street: These 2 fairly valued Club stocks will extend strong starts to 2023',  
 "This week's best-performing stocks include a pharma name up more than 20%",  
 'What to watch in the markets in the week ahead',  
 'U.S. shoots down second 'high altitude object' days after Chinese spy ballo",  
 'Democrat and GOP attorneys general face off on abortion pill lawsuit',  
 "Taylor Swift says fans will 'get on it' to reduce egg prices. 3 things to k",  
 'Google employees slam CEO Sundar Pichai\x00for \'rushed, botched\' announceme",  
 'Wall Street's favorite sports-betting stocks ahead of the Super Bowl',  
 'Here's how much money Americans saved at every age in 2022',  
 'Bill Gates: Nuclear waste is not a reason to avoid nuclear energy',  
 'Stocks making the biggest moves midday: Lyft, Spotify, Expedia, Yelp and mo",  
 '3 investing lessons from the Club's meeting Friday',  
 "Astronaut Sen. Kelly touts 'stunning' space business growth",  
 "Air Force's aging E-3 Sentry presents airborne battle readiness issues",  
 "FBI finds another classified document in search of Mike Pence's home",  
 "We're buying the dip in a beauty stock that stands to benefit from China re",  
 'National Archives, museum sued for telling visitors to remove pro-life mess",  
 "Here's what's happening with home prices right now",  
 'Amazon is the latest threat to Facebook as ad targeting suffers',  
 'This is how Microsoft and Alphabet stack up against each other in the A.I.",  
 'These are the best deals on TVs ahead of Super Bowl Sunday',  
 'Top House Republicans scrutinize SEC investigation of crypto exchange FTX',  
 "Steve Wozniak warns: ChatGPT is 'useful,' but it can 'make horrible mistake",  
 s'"]]

```
In [41]: scraped_time4=soup4.find_all('time',class_='LatestNews-timestamp')
scraped_time4
```

```
Out[41]: [<time class="LatestNews-timestamp">7 Hours Ago</time>,
<time class="LatestNews-timestamp">8 Hours Ago</time>,
<time class="LatestNews-timestamp">8 Hours Ago</time>,
<time class="LatestNews-timestamp">9 Hours Ago</time>,
<time class="LatestNews-timestamp">9 Hours Ago</time>,
<time class="LatestNews-timestamp">10 Hours Ago</time>,
<time class="LatestNews-timestamp">11 Hours Ago</time>,
<time class="LatestNews-timestamp">12 Hours Ago</time>,
<time class="LatestNews-timestamp">12 Hours Ago</time>,
<time class="LatestNews-timestamp">12 Hours Ago</time>,
<time class="LatestNews-timestamp">12 Hours Ago</time>,
<time class="LatestNews-timestamp">13 Hours Ago</time>,
<time class="LatestNews-timestamp">13 Hours Ago</time>,
<time class="LatestNews-timestamp">14 Hours Ago</time>]
```

```
In [42]: # scraped time
scraped_newstime4=[]
for scraped_th4 in scraped_time4:
    scraped_th4=scraped_th4.get_text().replace('\n','')
    scraped_newstime4.append(scraped_th4)
scraped_newstime4
```

```
Out[42]: ['7 Hours Ago',
          '8 Hours Ago',
          '8 Hours Ago',
          '9 Hours Ago',
          '9 Hours Ago',
          '9 Hours Ago',
          '10 Hours Ago',
          '11 Hours Ago',
          '12 Hours Ago',
          '12 Hours Ago',
          '12 Hours Ago',
          '12 Hours Ago',
          '13 Hours Ago',
          '13 Hours Ago',
          '14 Hours Ago',
          '15 Hours Ago',
          '15 Hours Ago',
          '15 Hours Ago',
          '15 Hours Ago']
```

```
In [44]: # scraped url
url4=[]
for i in soup4.find_all('a',class_="LatestNews-headline"):
    url4.append(i.get("href")) # links is stored in data-src ,by get we will g
url4
```

```
Out[44]: ['https://www.cnbc.com/2023/02/10/cramers-lightning-round-triton-internationa  
l-is-not-a-buy.html',  
 'https://www.cnbc.com/2023/02/10/cramers-week-ahead-there-are-plenty-of-high  
-quality-stocks-to-buy.html',  
 'https://www.cnbc.com/2023/02/10/goldman-sachs-digital-asset-chief-lays-out  
blockchain-strategy.html',  
 'https://www.cnbc.com/2023/02/10/what-it-feels-like-when-your-condition-becomes-a-punchline.html',  
 'https://www.cnbc.com/2023/02/10/rust-shooting-alec-baldwin-possible-penalty.html',  
 'https://www.cnbc.com/2023/02/10/pro-picks-watch-all-of-fridays-big-stock-ca  
lls-on-cnbc.html',  
 'https://www.cnbc.com/2023/02/10/us-sanctions-six-chinese-tech-companies-for  
-supporting-spy-balloon-programs.html',  
 'https://www.cnbc.com/2023/02/10/2-fairly-valued-club-stocks-will-continue-t  
heir-strong-starts-in-2023.html',  
 'https://www.cnbc.com/2023/02/10/this-weeks-best-performing-stocks-include-t  
his-pharmaceutical-stock.html',  
 'https://www.cnbc.com/2023/02/10/investors-will-have-their-eyes-on-the-consu  
mer-price-index-in-the-week-ahead.html',  
 'https://www.cnbc.com/2023/02/10/us-shoots-down-second-high-altitude-object  
on-bidens-orders.html',  
 'https://www.cnbc.com/2023/02/10/abortion-pill-states-says-pulling-medication  
n-would-have-devastating-consequences.html',  
 'https://www.cnbc.com/2023/02/10/high-egg-prices-taylor-swift-says-fans-will  
-reduce-cost.html',  
 'https://www.cnbc.com/2023/02/10/google-employees-slam-ceo-sundar-pichai-for  
-rushed-bard-announcement.html',  
 'https://www.cnbc.com/2023/02/10/wall-streets-favorite-gambling-stocks-ahead  
-of-the-super-bowl.html',  
 'https://www.cnbc.com/2023/02/10/average-amount-of-money-americans-saved-in  
2022.html',  
 'https://www.cnbc.com/2023/02/10/bill-gates-nuclear-waste-is-not-a-reason-av  
oid-nuclear-energy.html',  
 'https://www.cnbc.com/2023/02/10/stocks-moving-big-midday-lyft-spot-expe-yel  
p.html',  
 'https://www.cnbc.com/2023/02/10/jim-cramers-investing-club-meeting-friday-b  
e-selective-buying-in-this-market.html',  
 'https://www.cnbc.com/2023/02/10/astronaut-sen-mark-kelly-praises-space-busi  
ness.html',  
 'https://www.cnbc.com/2023/02/10/air-forces-aging-awacs-stirs-questions-of-a  
irborne-battle-readiness.html',  
 'https://www.cnbc.com/2023/02/10/fbi-searching-ex-trump-vp-pence-home-for-cl  
assified-docs.html',  
 'https://www.cnbc.com/2023/02/10/buying-the-dip-in-a-beauty-stock-that-stand  
s-to-benefit-from-china-reopening.html',  
 'https://www.cnbc.com/2023/02/10/national-archives-sued-for-abortion-opponen  
t-clothing-flap.html',  
 'https://www.cnbc.com/2023/02/10/home-prices-mortgage-rates-fall.html',  
 'https://www.cnbc.com/2023/02/10/amazon-is-the-latest-threat-to-facebook-as  
ad-targeting-suffers.html',  
 'https://www.cnbc.com/2023/02/10/heres-how-microsoft-and-alphabet-compare-in  
-the-ai-war.html',  
 'https://www.cnbc.com/2023/02/10/tv-prices-sink-ahead-of-the-super-bowl-here  
-are-the-best-deals.html',  
 'https://www.cnbc.com/2023/02/10/ftx-collapse-top-house-republicans-scrutini  
ze-sec-investigation-of-failed-crypto-exchange.html',
```

```
'https://www.cnbc.com/2023/02/10/steve-wozniak-warns-about-ai-chatgpt-can-make-horrible-mistakes.html']
```

```
In [45]: import pandas as pd  
df4=pd.DataFrame({'Headline':scraped_newsheadline4,'Time':scraped_newstime4,'N  
df4
```

Out[45]:

	Headline	Time	News Link
0	Cramer's lightning round: Triton International...	7 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/cramers-lightn...">https://www.cnbc.com/2023/02/10/cramers-lightn...</a>
1	Cramer's week ahead: There are plenty of high-...	8 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/cramers-week-a...">https://www.cnbc.com/2023/02/10/cramers-week-a...</a>
2	Crypto markets are in a 'flight to quality,' s...	8 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/goldman-sachs-...">https://www.cnbc.com/2023/02/10/goldman-sachs-...</a>
3	What it feels like when your mental health con...	9 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/what-it-feels-...">https://www.cnbc.com/2023/02/10/what-it-feels-...</a>
4	Alec Baldwin attorneys seek to reduce possible...	9 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/rust-shooting-...">https://www.cnbc.com/2023/02/10/rust-shooting-...</a>
5	Pro Picks: Watch all of Friday's big stock cal...	9 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/pro-picks-watc...">https://www.cnbc.com/2023/02/10/pro-picks-watc...</a>
6	U.S. sanctions six Chinese tech companies for ...	10 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/us-sanctions-s...">https://www.cnbc.com/2023/02/10/us-sanctions-s...s...</a>
7	Wall Street: These 2 fairly valued Club stocks...	11 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/2-fairly-value...">https://www.cnbc.com/2023/02/10/2-fairly-value...</a>
8	This week's best-performing stocks include a p...	12 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/this-weeks-bes...">https://www.cnbc.com/2023/02/10/this-weeks-bes...</a>
9	What to watch in the markets in the week ahead	12 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/investors-will...">https://www.cnbc.com/2023/02/10/investors-will...</a>
10	U.S. shoots down second 'high altitude object'...	12 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/us-shoots-down...">https://www.cnbc.com/2023/02/10/us-shoots-down...</a>
11	Democrat and GOP attorneys general face off on...	12 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/abortion-pill-...">https://www.cnbc.com/2023/02/10/abortion-pill-...</a>
12	Taylor Swift says fans will 'get on it' to red...	13 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/high-egg-price...">https://www.cnbc.com/2023/02/10/high-egg-price...</a>
13	Google employees slam CEO Sundar Pichai for 'r...	13 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/google-employe...">https://www.cnbc.com/2023/02/10/google-employe...</a>
14	Wall Street's favorite sports-betting stocks a...	14 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/wall-streets-f...">https://www.cnbc.com/2023/02/10/wall-streets-f...</a>
15	Here's how much money Americans saved at every...	14 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/average-amount...">https://www.cnbc.com/2023/02/10/average-amount...</a>
16	Bill Gates: Nuclear waste is not a reason to a...	14 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/bill-gates-nuc...">https://www.cnbc.com/2023/02/10/bill-gates-nuc...</a>
17	Stocks making the biggest moves midday: Lyft, ...	14 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/stocks-moving-...">https://www.cnbc.com/2023/02/10/stocks-moving-...</a>
18	3 investing lessons from the Club's meeting Fr...	14 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/jim-cramers-in...">https://www.cnbc.com/2023/02/10/jim-cramers-in...</a>
19	Astronaut Sen. Kelly touts 'stunning' space bu...	14 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/astronaut-sen-...">https://www.cnbc.com/2023/02/10/astronaut-sen-...</a>
20	Air Force's aging E-3 Sentry presents airborne...	14 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/air-forces-ag...">https://www.cnbc.com/2023/02/10/air-forces-ag...</a>
21	FBI finds another classified document in searc...	14 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/fbi-searching-...">https://www.cnbc.com/2023/02/10/fbi-searching-...</a>
22	We're buying the dip in a beauty stock that st...	14 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/buying-the-dip...">https://www.cnbc.com/2023/02/10/buying-the-dip...</a>

	Headline	Time	News Link
23	National Archives, museum sued for telling vis...	14 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/national-archi...">https://www.cnbc.com/2023/02/10/national-archi...</a>
24	Here's what's happening with home prices right...	14 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/home-prices-mo...">https://www.cnbc.com/2023/02/10/home-prices-mo...</a>
25	Amazon is the latest threat to Facebook as ad ...	14 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/amazon-is-the-...">https://www.cnbc.com/2023/02/10/amazon-is-the-...</a>
26	This is how Microsoft and Alphabet stack up ag...	15 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/heres-how-micr...">https://www.cnbc.com/2023/02/10/heres-how-micr...</a>
27	These are the best deals on TVs ahead of Super...	15 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/tv-prices-sink...">https://www.cnbc.com/2023/02/10/tv-prices-sink...</a>
28	Top House Republicans scrutinize SEC investiga...	15 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/ftx-collapse-t...">https://www.cnbc.com/2023/02/10/ftx-collapse-t...</a>
29	Steve Wozniak warns: ChatGPT is 'useful,' but ...	15 Hours Ago	<a href="https://www.cnbc.com/2023/02/10/steve-wozniak-...">https://www.cnbc.com/2023/02/10/steve-wozniak-...</a>

## Answer No.08

```
In [3]: page5=requests.get('https://www.journals.elsevier.com/artificial-intelligence/  
page5')
```

Out[3]: <Response [200]>

```
In [4]: soup5=BeautifulSoup(page5.content) # It shows the source code of the web page  
soup5
```

```
Out[4]: <!DOCTYPE html>
<html><head><meta charset="utf-8"/><meta content="width=device-width" name="viewport"/><meta content="en_US" name="og:locale"/><meta content="Most Downloaded Articles - Artificial Intelligence - Journal - Elsevier" property="og:title"/><meta content="The journal of Artificial Intelligence (AIJ) welcomes papers on broad aspects of AI that constitute advances in the overall field including, but not limited ..." property="og:description"/><meta content="http://ars.els-cdn.com/content/image/X00043702.jpg" name="og:image" property="og:image"/><meta content="http://ars.els-cdn.com/content/image/X00043702.jpg" name="og:image:url" property="og:image:url"/><meta content="https://ars.els-cdn.com/content/image/X00043702.jpg" name="og:image:secure_url" property="og:image:secure_url"/><meta content="journals.elsevier.com/artificial-intelligence/most-downloaded-articles" name="og:url"/><meta content="website" property="og:type"/><link href="/apple-touch-icon.png" rel="apple-touch-icon" sizes="180x180"/><link href="/favicon-32x32.png" rel="icon" sizes="32x32" type="image/png"/><link href="/favicon-16x16.png" rel="icon" sizes="16x16" type="image/png"/><link color="#ff6c00" href="/safari-pinned-tab.svg" rel="mask-icon"/><title>Most Downloaded Articles - Artificial Intelligence - Journal - Elsevier</title><meta content="The journal of Artificial Intelligence (AIJ) welcomes papers on broad aspects of AI that constitute advances in the overall field including, but not limited ..." property="og:description"/><meta content="http://ars.els-cdn.com/content/image/X00043702.jpg" name="og:image" property="og:image"/><meta content="http://ars.els-cdn.com/content/image/X00043702.jpg" name="og:image:url" property="og:image:url"/><meta content="https://ars.els-cdn.com/content/image/X00043702.jpg" name="og:image:secure_url" property="og:image:secure_url"/><meta content="journals.elsevier.com/artificial-intelligence/most-downloaded-articles" name="og:url"/><meta content="website" property="og:type"/><link href="/apple-touch-icon.png" rel="apple-touch-icon" sizes="180x180"/><link href="/favicon-32x32.png" rel="icon" sizes="32x32" type="image/png"/><link href="/favicon-16x16.png" rel="icon" sizes="16x16" type="image/png"/><link color="#ff6c00" href="/safari-pinned-tab.svg" rel="mask-icon"/><title>Most Downloaded Articles - Artificial Intelligence - Journal - Elsevier</title>
```

```
In [5]: scraped_title5=soup5.find_all('h2',class_='sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc')
```

**Out[5]:** [<h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Reward is enough</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Making sense of raw input</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Law and logic: A review from an argumentation perspective</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Creativity and artificial intelligence</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Artificial cognition for social human-robot interaction: An implementation</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Explanation in artificial intelligence: Insights from the social sciences</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Making sense of sensory input</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Conflict-based search for optimal multi-agent pathfinding</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">The Hanabi challenge: A new frontier for AI research</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Evaluating XAI: A comparison of rule-based and example-based explanations</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Argumentation in artificial intelligence</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Algorithms for computing strategies in two-player simultaneous move games</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Multiple object tracking: A literature review</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Selection of relevant features and examples in machine learning</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">A survey of inverse reinforcement learning: Challenges, methods and progress</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Explaining individual predictions when features are dependent: More accurate approximations to Shapley values</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">A review of possible effects of cognitive biases on interpretation of rule-based machine learning models</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Integrating social power into the decision-making of cognitive agents</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">"That's (not) the output I expected!" On the role of end user expectations in creating explanations of AI systems</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Algorithm runtime prediction: Methods & evaluation</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Wrap-ups for feature subset selection</h2>, <h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Commonsense visual sensemaking for autonomous driving – On generalised neurosymbolic online abduction integrating vision and semantics</h2>,

```
<h2 class="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1 btcbYu goSKRg">Quantum computation, quantum theory and AI</h2>]
```

```
In [6]: # scraped titles
scraped_article_title5=[]
for scraped_t5 in scraped_title5:
    scraped_t5=scraped_t5.get_text().replace('\n',"")
    scraped_article_title5.append(scraped_t5)
scraped_article_title5
```

```
Out[6]: ['Reward is enough',
'Making sense of raw input',
'Law and logic: A review from an argumentation perspective',
'Creativity and artificial intelligence',
'Artificial cognition for social human-robot interaction: An implementation',
'Explanation in artificial intelligence: Insights from the social sciences',
'Making sense of sensory input',
'Conflict-based search for optimal multi-agent pathfinding',
'Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning',
'The Hanabi challenge: A new frontier for AI research',
'Evaluating XAI: A comparison of rule-based and example-based explanations',
'Argumentation in artificial intelligence',
'Algorithms for computing strategies in two-player simultaneous move games',
'Multiple object tracking: A literature review',
'Selection of relevant features and examples in machine learning',
'A survey of inverse reinforcement learning: Challenges, methods and progress',
'Explaining individual predictions when features are dependent: More accurate approximations to Shapley values',
'A review of possible effects of cognitive biases on interpretation of rule-based machine learning models',
'Integrating social power into the decision-making of cognitive agents',
'"That's (not) the output I expected!" On the role of end user expectations in creating explanations of AI systems',
'Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies',
'Algorithm runtime prediction: Methods & evaluation',
'Wrappers for feature subset selection',
'Commonsense visual sensemaking for autonomous driving - On generalised neurosymbolic online abduction integrating vision and semantics',
'Quantum computation, quantum theory and AI']
```

```
In [7]: scraped_authors5=soup5.find_all('span',class_='sc-1w3fpd7-0 dnCnAO')  
scraped_authors5
```

```
Out[7]: [<span class="sc-1w3fpd7-0 dnCnAO">Silver, David, Singh, Satinder, Precup, Doina, Sutton, Richard S. </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Evans, Richard, Bošnjak, Matko and 5 more </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Prakken, Henry, Sartor, Giovanni </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Boden, Margaret A. </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Lemaignan, Séverin, Warnier, Mathieu and 3 more</span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Miller, Tim </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Evans, Richard, Hernández-Orallo, José and 3 more</span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Sharon, Guni, Stern, Roni, Felner, Ariel, Sturtevant, Nathan R. </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Sutton, Richard S., Precup, Doina, Singh, Satinder </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Bard, Nolan, Foerster, Jakob N. and 13 more</span>,  
<span class="sc-1w3fpd7-0 dnCnAO">van der Waa, Jasper, Nieuwburg, Elisabeth, Cremers, Anita, Neerincx, Mark </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Bench-Capon, T.J.M., Dunne, Paul E. </span>,<br>,  
<span class="sc-1w3fpd7-0 dnCnAO">Bošanský, Branislav, Lisý, Viliam and 3 more</span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Luo, Wenhan, Xing, Junliang and 4 more</span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Blum, Avrim L., Langley, Pat </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Arora, Saurabh, Doshi, Prashant </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Aas, Kjersti, Jullum, Martin, Løland, Anders </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Kliegr, Tomáš, Bahník, Štěpán, Fürnkranz, Johannes </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Pereira, Gonçalo, Prada, Rui, Santos, Pedro A. </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Riveiro, Maria, Thill, Serge </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Kenny, Eoin M., Ford, Courtney, Quinn, Molony, Keane, Mark T. </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Hutter, Frank, Xu, Lin, Hoos, Holger H., Leyton-Brown, Kevin </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Kohavi, Ron, John, George H. </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Suchan, Jakob, Bhatt, Mehul, Varadarajan, Srikrishna </span>,  
<span class="sc-1w3fpd7-0 dnCnAO">Ying, Mingsheng </span>]
```

In [8]: # scraped authors

```
scraped_article_author5=[]
for scraped_a5 in scraped_authors5:
    scraped_a5=scraped_a5.get_text().replace('\n','')
    scraped_article_author5.append(scraped_a5)
scraped_article_author5
```

Out[8]: ['Silver, David, Singh, Satinder, Precup, Doina, Sutton, Richard S. ',

'Evans, Richard, Bošnjak, Matko and 5 more',  
'Prakken, Henry, Sartor, Giovanni ',  
'Boden, Margaret A. ',  
'Lemaignan, Séverin, Warnier, Mathieu and 3 more',  
'Miller, Tim ',  
'Evans, Richard, Hernández-Orallo, José and 3 more',  
'Sharon, Guni, Stern, Roni, Felner, Ariel, Sturtevant, Nathan R. ',  
'Sutton, Richard S., Precup, Doina, Singh, Satinder ',  
'Bard, Nolan, Foerster, Jakob N. and 13 more',  
'van der Waa, Jasper, Nieuwburg, Elisabeth, Cremers, Anita, Neerincx, Mark  
',  
'Bench-Capon, T.J.M., Dunne, Paul E. ',  
'Bošanský, Branislav, Lisý, Viliam and 3 more',  
'Luo, Wenhan, Xing, Junliang and 4 more',  
'Blum, Avrim L., Langley, Pat ',  
'Arora, Saurabh, Doshi, Prashant ',  
'Aas, Kjersti, Jullum, Martin, Løland, Anders ',  
'Kliegr, Tomáš, Bahník, Štěpán, Fürnkranz, Johannes ',  
'Pereira, Gonçalo, Prada, Rui, Santos, Pedro A. ',  
'Riveiro, Maria, Thill, Serge ',  
'Kenny, Eoin M., Ford, Courtney, Quinn, Molly, Keane, Mark T. ',  
'Hutter, Frank, Xu, Lin, Hoos, Holger H., Leyton-Brown, Kevin ',  
'Kohavi, Ron, John, George H. ',  
'Suchan, Jakob, Bhatt, Mehul, Varadarajan, Srikrishna ',  
'Ying, Mingsheng ']

```
In [9]: scraped_date5=soup5.find_all('span',class_='sc-1thf9ly-2 dvggwt')  
scraped_date5
```

```
Out[9]: [<span class="sc-1thf9ly-2 dvggwt"><span>October 2021</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>October 2021</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>October 2015</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>August 1998</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>June 2017</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>February 2019</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>April 2021</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>February 2015</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>August 1999</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>March 2020</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>February 2021</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>October 2007</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>August 2016</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>April 2021</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>December 1997</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>August 2021</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>September 2021</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>June 2021</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>December 2016</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>September 2021</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>May 2021</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>January 2014</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>December 1997</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>October 2021</span></span>,  
<span class="sc-1thf9ly-2 dvggwt"><span>February 2010</span></span>]
```

In [11]: # scraped dates

```
scraped_article_date5=[]
for scraped_d5 in scraped_date5:
    scraped_d5=scraped_d5.get_text().replace('\n','')
    scraped_article_date5.append(scraped_d5)
scraped_article_date5
```

Out[11]: ['October 2021',

```
'October 2021',
'October 2015',
'August 1998',
'June 2017',
'February 2019',
'April 2021',
'February 2015',
'August 1999',
'March 2020',
'February 2021',
'October 2007',
'August 2016',
'April 2021',
'December 1997',
'August 2021',
'September 2021',
'June 2021',
'December 2016',
'September 2021',
'May 2021',
'January 2014',
'December 1997',
'October 2021',
'February 2010']
```

```
In [33]: url5=[]
for i in soup5.find_all('a',class_="sc-5smygv-0 fIXTHm"):
    url5.append(i.get("href")) # Links is stored in data-src ,by get we will g
url5
```

```
Out[33]: ['https://www.sciencedirect.com/science/article/pii/S0004370221000862',
'https://www.sciencedirect.com/science/article/pii/S0004370221000722',
'https://www.sciencedirect.com/science/article/pii/S0004370215000910',
'https://www.sciencedirect.com/science/article/pii/S0004370298000551',
'https://www.sciencedirect.com/science/article/pii/S0004370216300790',
'https://www.sciencedirect.com/science/article/pii/S0004370218305988',
'https://www.sciencedirect.com/science/article/pii/S0004370220301855',
'https://www.sciencedirect.com/science/article/pii/S0004370214001386',
'https://www.sciencedirect.com/science/article/pii/S0004370299000521',
'https://www.sciencedirect.com/science/article/pii/S0004370219300116',
'https://www.sciencedirect.com/science/article/pii/S0004370220301533',
'https://www.sciencedirect.com/science/article/pii/S0004370207000793',
'https://www.sciencedirect.com/science/article/pii/S0004370216300285',
'https://www.sciencedirect.com/science/article/pii/S0004370220301958',
'https://www.sciencedirect.com/science/article/pii/S0004370297000635',
'https://www.sciencedirect.com/science/article/pii/S0004370221000515',
'https://www.sciencedirect.com/science/article/pii/S0004370221000539',
'https://www.sciencedirect.com/science/article/pii/S0004370221000096',
'https://www.sciencedirect.com/science/article/pii/S0004370216300868',
'https://www.sciencedirect.com/science/article/pii/S0004370221000588',
'https://www.sciencedirect.com/science/article/pii/S0004370221000102',
'https://www.sciencedirect.com/science/article/pii/S0004370213001082',
'https://www.sciencedirect.com/science/article/pii/S000437029700043X',
'https://www.sciencedirect.com/science/article/pii/S0004370221000734',
'https://www.sciencedirect.com/science/article/pii/S0004370209001398']
```

```
In [34]: import pandas as pd  
df5=pd.DataFrame({'Paper Title':scraped_article_title5,'Authors':scraped_artic  
df5
```

Out[34]:

	Paper Title	Authors	Published Date	Paper URL
0	Reward is enough	Silver, David, Singh, Satinder, Precup, Doina,...	October 2021	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
1	Making sense of raw input	Evans, Richard, Bošnjak, Matko and 5 more	October 2021	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
2	Law and logic: A review from an argumentation ...	Prakken, Henry, Sartor, Giovanni	October 2015	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
3	Creativity and artificial intelligence	Boden, Margaret A.	August 1998	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
4	Artificial cognition for social human–robot in...	Lemaignan, Séverin, Warnier, Mathieu and 3 more	June 2017	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
5	Explanation in artificial intelligence: Insights...	Miller, Tim	February 2019	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
6	Making sense of sensory input	Evans, Richard, Hernández-Orallo, José and 3 more	April 2021	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
7	Conflict-based search for optimal multi-agent ...	Sharon, Guni, Stern, Roni, Felner, Ariel, Stur...	February 2015	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
8	Between MDPs and semi-MDPs: A framework for te...	Sutton, Richard S., Precup, Doina, Singh, Satish...	August 1999	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
9	The Hanabi challenge: A new frontier for AI re...	Bard, Nolan, Foerster, Jakob N. and 13 more	March 2020	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
10	Evaluating XAI: A comparison of rule-based and...	van der Waa, Jasper, Nieuwburg, Elisabeth, Cre...	February 2021	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
11	Argumentation in artificial intelligence	Bench-Capon, T.J.M., Dunne, Paul E.	October 2007	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
12	Algorithms for computing strategies in two-pla...	Bošanský, Branislav, Lisý, Viliam and 3 more	August 2016	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
13	Multiple object tracking: A literature review	Luo, Wenhan, Xing, Junliang and 4 more	April 2021	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
14	Selection of relevant features and examples in...	Blum, Avrim L., Langley, Pat	December 1997	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>

	Paper Title	Authors	Published Date	Paper URL
15	A survey of inverse reinforcement learning: Ch...	Arora, Saurabh, Doshi, Prashant	August 2021	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
16	Explaining individual predictions when feature...	Aas, Kjersti, Jullum, Martin, Løland, Anders	September 2021	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
17	A review of possible effects of cognitive bias...	Kliegr, Tomáš, Bahník, Štěpán, Fürnkranz, Joha...	June 2021	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
18	Integrating social power into the decision-mak...	Pereira, Gonçalo, Prada, Rui, Santos, Pedro A.	December 2016	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
19	"That's (not) the output I expected!" On the r...	Riveiro, Maria, Thill, Serge	September 2021	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
20	Explaining black-box classifiers using post-ho...	Kenny, Eoin M., Ford, Courtney, Quinn, Molly, ...	May 2021	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
21	Algorithm runtime prediction: Methods & evalua...	Hutter, Frank, Xu, Lin, Hoos, Holger H., Leyto...	January 2014	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
22	Wrappers for feature subset selection	Kohavi, Ron, John, George H.	December 1997	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
23	Commonsense visual sensemaking for autonomous ...	Suchan, Jakob, Bhatt, Mehul, Varadarajan, Srik...	October 2021	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>
24	Quantum computation, quantum theory and AI	Ying, Mingsheng	February 2010	<a href="https://www.sciencedirect.com/science/article/...">https://www.sciencedirect.com/science/article/...</a>

## Answer No.09

```
In [12]: page6=requests.get('https://www.dineout.co.in/delhi-restaurants/buffet-special
page6
```

```
Out[12]: <Response [200]>
```

In [14]: `soup6=BeautifulSoup(page6.content) # It shows the source code of the web page  
soup6`

Out[14]: `<!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;  
 }</style>`

In [15]:

```
scraped_res_name6=soup6.find_all('a',class_='restnt-name ellipsis')
scraped_res_name6
```

Out[15]:

```
[<a analytics-action="RestaurantCardClick" analytics-label="86792_Castle Barbeque" class="restnt-name ellipsis" data-w-onclick="sendAnalyticsCommon|w1-restaurant" href="/delhi/castle-barbeque-connaught-place-central-delhi-86792">Castle Barbeque</a>,
 <a analytics-action="RestaurantCardClick" analytics-label="59633_Jungle Jamboree" class="restnt-name ellipsis" data-w-onclick="sendAnalyticsCommon|w1-restaurant" href="/delhi/jungle-jamboree-lajpat-nagar-3-south-delhi-59633">Jungle Jamboree</a>,
 <a analytics-action="RestaurantCardClick" analytics-label="406_Cafe Knosh" class="restnt-name ellipsis" data-w-onclick="sendAnalyticsCommon|w1-restaurant" href="/delhi/cafe-knosh-shahdara-east-delhi-406">Cafe Knosh</a>,
 <a analytics-action="RestaurantCardClick" analytics-label="38113_Castle Barbeque" class="restnt-name ellipsis" data-w-onclick="sendAnalyticsCommon|w1-restaurant" href="/delhi/castle-barbeque-tagore-garden-west-delhi-38113">Castle B arbeque</a>,
 <a analytics-action="RestaurantCardClick" analytics-label="79307_The Barbeque Company" class="restnt-name ellipsis" data-w-onclick="sendAnalyticsCommon|w1-restaurant" href="/delhi/the-barbeque-company-sector-38a-noida-79307">The Barbeque Company</a>,
 <a analytics-action="RestaurantCardClick" analytics-label="2687_India Grill" class="restnt-name ellipsis" data-w-onclick="sendAnalyticsCommon|w1-restaran t" href="/delhi/india-grill-saket-south-delhi-2687">India Grill</a>,
 <a analytics-action="RestaurantCardClick" analytics-label="52501_Delhi Barbeque" class="restnt-name ellipsis" data-w-onclick="sendAnalyticsCommon|w1-rest arant" href="/delhi/delhi-barbeque-mahipalpur-south-delhi-52501">Delhi Barbeque</a>,
 <a analytics-action="RestaurantCardClick" analytics-label="34822_The Monarch - Bar Be Que Village" class="restnt-name ellipsis" data-w-onclick="sendAnalyt icsCommon|w1-restarant" href="/delhi/the-monarch-bar-be-que-village-indirapur am-ghaziabad-34822">The Monarch - Bar Be Que Village</a>,
 <a analytics-action="RestaurantCardClick" analytics-label="549_Indian Grill Room" class="restnt-name ellipsis" data-w-onclick="sendAnalyticsCommon|w1-res tarant" href="/delhi/indian-grill-room-golf-course-road-gurgaon-549">Indian Grill Room</a>]
```

In [17]:

```
# scraped Names
scraped_rest_name6=[]
for scraped_rn6 in scraped_res_name6:
    scraped_rn6=scraped_rn6.get_text().replace('\n',"")
    scraped_rest_name6.append(scraped_rn6)
scraped_rest_name6
```

Out[17]:

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

```
In [104]: scraped_res_cui6=soup6.find_all('div',class_='detail-info')
scraped_res_cui6
```

```
Out[104]: [<div class="detail-info"><ul><li><span class="double-line-ellipsis"><span>₹  
2,000 for 2 (approx)</span><span> | </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/chinese-cuisine">Chinese</a><span>, </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/north-indian-cuisine">North Indian</a></span></li><li class="ellipsis"></li></ul></div>,  
<div class="detail-info"><ul><li><span class="double-line-ellipsis"><span>₹  
1,680 for 2 (approx)</span><span> | </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/north-indian-cuisine">North Indian</a><span>, </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/asian-cuisine">Asian</a><span>, </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/italian-cuisine">Italian</a></span></li><li class="ellipsis"></li></ul></div>,  
<div class="detail-info"><ul><li><span class="double-line-ellipsis"><span>₹  
3,000 for 2 (approx)</span><span> | </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/italian-cuisine">Italian</a><span>, </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/continental-cuisine">Continental</a></span></li><li class="ellipsis"></li></ul></div>,  
<div class="detail-info"><ul><li><span class="double-line-ellipsis"><span>₹  
2,000 for 2 (approx)</span><span> | </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/chinese-cuisine">Chinese</a><span>, </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/north-indian-cuisine">North Indian</a></span></li><li class="ellipsis"></li></ul></div>,  
<div class="detail-info"><ul><li><span class="double-line-ellipsis"><span>₹  
1,700 for 2 (approx)</span><span> | </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/north-indian-cuisine">North Indian</a><span>, </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/chinese-cuisine">Chinese</a></span></li><li class="ellipsis"></li></ul></div>,  
<div class="detail-info"><ul><li><span class="double-line-ellipsis"><span>₹  
2,400 for 2 (approx)</span><span> | </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/north-indian-cuisine">North Indian</a><span>, </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/italian-cuisine">Italian</a></span></li><li class="ellipsis"></li></ul></div>,  
<div class="detail-info"><ul><li><span class="double-line-ellipsis"><span>₹  
1,800 for 2 (approx)</span><span> | </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/north-indian-cuisine">North Indian</a></span></li><li class="ellipsis"></li></ul></div>,  
<div class="detail-info"><ul><li><span class="double-line-ellipsis"><span>₹  
1,900 for 2 (approx)</span><span> | </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/north-indian-cuisine">North Indian</a></span></li><li class="ellipsis"></li></ul></div>,  
<div class="detail-info"><ul><li><span class="double-line-ellipsis"><span>₹  
2,200 for 2 (approx)</span><span> | </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/north-indian-cuisine">North Indian</a><span>, </span><a data-w-onclick="stopClickPropagation|w1-restaurant" href="/delhi-restaurants/mughlai-cuisine">Mughlai</a></span></li><li class="ellipsis"></li></ul></div>]
```

```
In [105]: # scraped Cuisine
scraped_rest_cuisine6=[]
for scraped_rc6 in scraped_res_cui6:
    scraped_rc6=scraped_rc6.get_text().replace('\n',"")
    scraped_rest_cuisine6.append(scraped_rc6)
scraped_rest_cuisine6
```

```
Out[105]: ['₹ 2,000 for 2 (approx) | Chinese, North Indian',
'₹ 1,680 for 2 (approx) | North Indian, Asian, Italian',
'₹ 3,000 for 2 (approx) | Italian, Continental',
'₹ 2,000 for 2 (approx) | Chinese, North Indian',
'₹ 1,700 for 2 (approx) | North Indian, Chinese',
'₹ 2,400 for 2 (approx) | North Indian, Italian',
'₹ 1,800 for 2 (approx) | North Indian',
'₹ 1,900 for 2 (approx) | North Indian',
'₹ 2,200 for 2 (approx) | North Indian, Mughlai']
```

In [21]: `scraped_res_location6=soup6.find_all('div',class_='restnt-loc ellipsis')  
scraped_res_location6`

Out[21]: [`<div class="restnt-loc ellipsis" data-w-onclick="stopClickPropagation|w1-res  
tarant"><a data-name="Connaught Place" data-type="LocalityClick" href="/delhi  
-restaurants/central-delhi/connaught-place">Connaught Place</a>, <a data-name  
="Central Delhi" data-type="AreaClick" href="/delhi-restaurants/central-delh  
i">Central Delhi</a></div>,  
<div class="restnt-loc ellipsis" data-w-onclick="stopClickPropagation|w1-res  
tarant"><a href="/delhi-restaurants/3cs-mall-landmark">3CS Mall,</a><a data-n  
ame="Lajpat Nagar - 3" data-type="LocalityClick" href="/delhi-restaurants/sou  
th-delhi/lajpat-nagar-3">Lajpat Nagar - 3</a>, <a data-name="South Delhi" dat  
a-type="AreaClick" href="/delhi-restaurants/south-delhi">South Delhi</a></div  
>,  
<div class="restnt-loc ellipsis" data-w-onclick="stopClickPropagation|w1-res  
tarant"><a href="/delhi-restaurants/the-leela-ambience-convention-hotel-landm  
ark">The Leela Ambience Convention Hotel,</a><a data-name="Shahdara" data-typ  
e="LocalityClick" href="/delhi-restaurants/east-delhi/shahdara">Shahdara</a>,  
<a data-name="East Delhi" data-type="AreaClick" href="/delhi-restaurants/east  
-delhi">East Delhi</a></div>,  
<div class="restnt-loc ellipsis" data-w-onclick="stopClickPropagation|w1-res  
tarant"><a href="/delhi-restaurants/pacific-mall-landmark">Pacific Mall,</a><  
a data-name="Tagore Garden" data-type="LocalityClick" href="/delhi-restaurant  
s/west-delhi/tagore-garden">Tagore Garden</a>, <a data-name="West Delhi" data  
-type="AreaClick" href="/delhi-restaurants/west-delhi">West Delhi</a></div>,  
<div class="restnt-loc ellipsis" data-w-onclick="stopClickPropagation|w1-res  
tarant"><a href="/delhi-restaurants/gardens-galleria-landmark">Gardens Galler  
ia,</a><a data-name="Sector 38A" data-type="LocalityClick" href="/delhi-resta  
urants/noida/sector-38a">Sector 38A</a>, <a data-name="Noida" data-type="Area  
Click" href="/delhi-restaurants/noida">Noida</a></div>,  
<div class="restnt-loc ellipsis" data-w-onclick="stopClickPropagation|w1-res  
tarant"><a href="/delhi-restaurants/hilton-garden-inn-landmark">Hilton Garden  
Inn,</a><a data-name="Saket" data-type="LocalityClick" href="/delhi-restauran  
ts/south-delhi/saket">Saket</a>, <a data-name="South Delhi" data-type="AreaCl  
ick" href="/delhi-restaurants/south-delhi">South Delhi</a></div>,  
<div class="restnt-loc ellipsis" data-w-onclick="stopClickPropagation|w1-res  
tarant"><a href="/delhi-restaurants/taurus-sarovar-portico-landmark">Taurus S  
arovar Portico,</a><a data-name="Mahipalpur" data-type="LocalityClick" href  
="/delhi-restaurants/south-delhi/mahipalpur">Mahipalpur</a>, <a data-name="So  
uth Delhi" data-type="AreaClick" href="/delhi-restaurants/south-delhi">South  
Delhi</a></div>,  
<div class="restnt-loc ellipsis" data-w-onclick="stopClickPropagation|w1-res  
tarant"><a href="/delhi-restaurants/indirapuram-habitat-centre-landmark">Indi  
rapuram Habitat Centre,</a><a data-name="Indirapuram" data-type="LocalityClic  
k" href="/delhi-restaurants/ghaziabad/indirapuram">Indirapuram</a>, <a data-n  
ame="Ghaziabad" data-type="AreaClick" href="/delhi-restaurants/ghaziabad">Gha  
ziabad</a></div>,  
<div class="restnt-loc ellipsis" data-w-onclick="stopClickPropagation|w1-res  
tarant"><a href="/delhi-restaurants/suncity-business-tower-landmark">Suncity  
Business Tower,</a><a data-name="Golf Course Road" data-type="LocalityClick"  
href="/delhi-restaurants/gurgaon/golf-course-road">Golf Course Road</a>, <a d  
ata-name="Gurgaon" data-type="AreaClick" href="/delhi-restaurants/gurgaon">Gu  
rgaon</a></div>]`

```
In [22]: # scraped location
scraped_rest_loc6=[]
for scraped_rl6 in scraped_res_location6:
    scraped_rl6=scraped_rl6.get_text().replace('\n',"")
    scraped_rest_loc6.append(scraped_rl6)
scraped_rest_loc6
```

```
Out[22]: ['Connaught Place, Central Delhi',
          '3CS Mall,Lajpat Nagar - 3, South Delhi',
          'The Leela Ambience Convention Hotel,Shahdara, East Delhi',
          'Pacific Mall,Tagore Garden, West Delhi',
          'Gardens Galleria,Sector 38A, Noida',
          'Hilton Garden Inn,Saket, South Delhi',
          'Taurus Sarovar Portico,Mahipalpur, South Delhi',
          'Indirapuram Habitat Centre,Indirapuram, Ghaziabad',
          'Suncity Business Tower,Golf Course Road, Gurgaon']
```

```
In [23]: scraped_res_rating6=soup6.find_all('div',class_='restnt-rating rating-4')
scraped_res_rating6
```

```
Out[23]: [<div class="restnt-rating rating-4">4.1</div>,
           <div class="restnt-rating rating-4">3.9</div>,
           <div class="restnt-rating rating-4">4.3</div>,
           <div class="restnt-rating rating-4">3.9</div>,
           <div class="restnt-rating rating-4">4</div>,
           <div class="restnt-rating rating-4">3.9</div>,
           <div class="restnt-rating rating-4">3.6</div>,
           <div class="restnt-rating rating-4">3.8</div>,
           <div class="restnt-rating rating-4">4.3</div>]
```

```
In [24]: # scraped rating
scraped_rest_rate6=[]
for scraped_rrate6 in scraped_res_rating6:
    scraped_rrate6=scraped_rrate6.get_text().replace('\n',"")
    scraped_rest_rate6.append(scraped_rrate6)
scraped_rest_rate6
```

```
Out[24]: ['4.1', '3.9', '4.3', '3.9', '4', '3.9', '3.6', '3.8', '4.3']
```

In [26]: `scraped_res_image6=soup6.find_all('img',class_='no-img')  
scraped_res_image6`

Out[26]: [,  
,  
,  
,  
,  
,  
,  
,  
]

```
In [31]: # scraped url
images=[]
for i in soup6.find_all('img',class_="no-img"):
    images.append(i.get("data-src")) # links is stored in data-src ,by get we
images
```

```
Out[31]: ['https://im1.dineout.co.in/images/uploads/restaurant/sharpen/8/k/b/p86792-16
062953735fbe1f4d3fb7e.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/p/m/p59633-16
6088382462ff137009010.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/4/p/m/p406-1543
8184745c04cce491bc.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/j/o/p38113-15
959192065f1fc666130c.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/7/p/k/p79307-16
051787755fad1597f2bf9.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/2/v/t/p2687-148
2477169585cce712b90f.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/d/i/p52501-16
61855212630de5eceb6d2.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/n/o/p34822-15
599107305cfa594a13c24.jpg?tr=tr:n-medium',
'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/y/f/p549-1650
00147262590640c0afc.jpg?tr=tr:n-medium']
```

In [32]:

```
import pandas as pd
df6=pd.DataFrame({'Restaurant name':scraped_rest_name6,'Cuisine':scraped_rest_cuisin..
```

Out[32]:

	Restaurant name	Cuisine	Location	Ratings	Image URL
0	Castle Barbeque	₹ 2,000 for 2 (approx)   Chinese, North Indian	Connaught Place, Central Delhi	4.1	<a href="https://im1.dineout.co.in/images/uploads/resta..">https://im1.dineout.co.in/images/uploads/resta..</a>
1	Jungle Jamboree	₹ 1,680 for 2 (approx)   North Indian, Asian, ...	3CS Mall,Lajpat Nagar - 3, South Delhi	3.9	<a href="https://im1.dineout.co.in/images/uploads/resta..">https://im1.dineout.co.in/images/uploads/resta..</a>
2	Cafe Knosh	₹ 3,000 for 2 (approx)   Italian, Continental	The Leela Ambience Convention Hotel,Shahdara, ...	4.3	<a href="https://im1.dineout.co.in/images/uploads/resta..">https://im1.dineout.co.in/images/uploads/resta..</a>
3	Castle Barbeque	₹ 2,000 for 2 (approx)   Chinese, North Indian	Pacific Mall,Tagore Garden, West Delhi	3.9	<a href="https://im1.dineout.co.in/images/uploads/resta..">https://im1.dineout.co.in/images/uploads/resta..</a>
4	The Barbeque Company	₹ 1,700 for 2 (approx)   North Indian, Chinese	Gardens Galleria,Sector 38A, Noida	4	<a href="https://im1.dineout.co.in/images/uploads/resta..">https://im1.dineout.co.in/images/uploads/resta..</a>
5	India Grill	₹ 2,400 for 2 (approx)   North Indian, Italian	Hilton Garden Inn,Saket, South Delhi	3.9	<a href="https://im1.dineout.co.in/images/uploads/resta..">https://im1.dineout.co.in/images/uploads/resta..</a>
6	Delhi Barbeque	₹ 1,800 for 2 (approx)   North Indian	Taurus Sarovar Portico,Mahipalpur, South Delhi	3.6	<a href="https://im1.dineout.co.in/images/uploads/resta..">https://im1.dineout.co.in/images/uploads/resta..</a>
7	The Monarch - Bar Be Que Village	₹ 1,900 for 2 (approx)   North Indian	Indirapuram Habitat Centre,Indirapuram, Ghaziabad	3.8	<a href="https://im1.dineout.co.in/images/uploads/resta..">https://im1.dineout.co.in/images/uploads/resta..</a>
8	Indian Grill Room	₹ 2,200 for 2 (approx)   North Indian, Mughlai	Suncity Business Tower,Golf Course Road, Gurgaon	4.3	<a href="https://im1.dineout.co.in/images/uploads/resta..">https://im1.dineout.co.in/images/uploads/resta..</a>

In [ ]:

