

```
In [1]: import pandas as pd
```

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

```
Requirement already satisfied: bs4 in c:\users\hp\anaconda3\lib\site-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in c:\users\hp\anaconda3\lib\site-packages (from bs4) (4.11.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\hp\anaconda3\lib\site-packages (from beautifulsoup4->bs4) (2.3.2.post1)
```

```
In [3]: !pip install requests
```

```
Requirement already satisfied: requests in c:\users\hp\anaconda3\lib\site-packages (2.28.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\hp\anaconda3\lib\site-packages (from requests) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\hp\anaconda3\lib\site-packages (from requests) (1.26.14)
Requirement already satisfied: idna<4,>=2.5 in c:\users\hp\anaconda3\lib\site-packages (from requests) (3.4)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\hp\anaconda3\lib\site-packages (from requests) (2.0.4)
```

```
In [4]: #importing the libraries
from bs4 import BeautifulSoup
import requests
```

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

```
In [5]: #get request to the webpage from where we can take the content. make sure the r
page1=requests.get("https://en.wikipedia.org/wiki/Main_Page")
page1
```

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

```
In [6]: #page content
        soup1=BeautifulSoup(page1.content)
        soup1
```

```
Out[6]: <!DOCTYPE html>
<html class="client-nojs vector-feature-language-in-header-enabled vector-feature-language-in-main-page-header-disabled vector-feature-sticky-header-disabled vector-feature-page-tools-pinned-disabled vector-feature-toc-pinned-enabled vector-feature-main-menu-pinned-disabled vector-feature-limited-width-enabled vector-feature-limited-width-content-enabled vector-feature-zebra-design-disabled" dir="ltr" lang="en">
<head>
<meta charset="utf-8"/>
<title>Wikipedia, the free encyclopedia</title>
<script>document.documentElement.className="client-js vector-feature-language-in-header-enabled vector-feature-language-in-main-page-header-disabled vector-feature-sticky-header-disabled vector-feature-page-tools-pinned-disabled vector-feature-toc-pinned-enabled vector-feature-main-menu-pinned-disabled vector-feature-limited-width-enabled vector-feature-limited-width-content-enabled vector-feature-zebra-design-disabled";(function(){var cookie=document.cookie.match(/(?:^| )enwikimwclientprefs=([^;]+);/);if(cookie){var featureName=cookie[1];document.documentElement.className=document.documentElement.className.replace(featureName+'-enabled',featureName+'-disabled');}})();</script></head>
<body>
<div>The page you were looking for does not exist. It is either been moved or deleted. If you believe this is an error, please contact us.</div>
</body>
</html>
```

```
In [7]: #creating an array
header = []

#Loop through all those tags and add the text to an array
for i in soup1.find_all(['h1', 'h2', 'h3', 'h4', 'h5', 'h6']):
    header.append(i.text)

#create dataframe (Name of the content and the content)
head=pd.DataFrame({'HeaderTag': header})
head
```

Out[7]:	HeaderTag
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

Write a python program to display list of respected former presidents of India(i.e. Name , Term of office) from <https://presidentofindia.nic.in/former-presidents.htm>

```
In [8]: page2=requests.get("https://presidentofindia.nic.in/former-presidents.htm")
page2
```

```
Out[8]: <Response [200]>
```

```
In [9]: soup2=BeautifulSoup(page2.content)
soup2
```

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

```
In [10]: #to find the first element of the specific tag here its h3..
#split() helps to stop the flow of the statement/word after that element
first_title=soup2.find('h3')
first_title.text.split('(')[0]
```

```
Out[10]: 'Shri Ram Nath Kovind '
```

```
In [11]: first_term=soup2.find('p')
first_term.text.split(':')[ -1]
```

```
Out[11]: ' 25 July, 2017 to 25 July, 2022 '
```

In [12]: #create an array

```
names = []
terms = []

#Loop through the div where the class is as mentioned and find h3 and p from the
for n in soup2.find_all('div',class_='presidentListing'):
    name = n.find('h3').text.split('(')[0]
    names.append(name)

    term = n.find('p').text.split(':')[1]
    terms.append(term)

president = pd.DataFrame({'Name': names, 'Term of Office': terms})
```

Out[12]:

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

Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape and make data frame a) Top 10 ODI teams in men's cricket along with the records for matches, points and rating.

In [13]:

```
page3=requests.get("https://www.icc-cricket.com/rankings/mens/team-rankings/odi")
page3
```

Out[13]: <Response [200]>

```
In [14]: soup3=BeautifulSoup(page3.content)
soup3
```

```
Out[14]: <!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 [15]: #Men ODI teams
points = []
teams = []
matches = []
ratings = []

table = soup3.find("table", class_="table")

teams_tr_b = table.find_all("tr", class_="rankings-block__banner")

#as the first team which has been highlighted on the List had different table class
for n in teams_tr_b:
    td1_b = n.find("span", class_="u-hide-phablet")
    teamOne = td1_b.text.strip()
    teams.append(teamOne)

    td2_b = n.find("td", class_="rankings-block__banner--matches")
    matchOne = td2_b.text.strip()
    matches.append(matchOne)

    td3_b = n.find("td", class_="rankings-block__banner--points")
    pointsOne = td3_b.text.strip()
    points.append(pointsOne)

    td4_b = n.find("td", class_="rankings-block__banner--rating")
    ratingOne = td4_b.text.strip()
    ratings.append(ratingOne)

teams_tr = table.find_all("tr", class_="table-body")
for u in teams_tr:
    td2 = u.find("span", class_="u-hide-phablet")
    teamsName = td2.text.strip()
    teams.append(teamsName)

    td3 = u.find_all("td", class_="table-body__cell u-center-text")
    matchesNo = td3[0].text.strip()
    pointsNo = td3[1].text.strip()
    matches.append(matchesNo)
    points.append(pointsNo)

    td4 = u.find("td", class_="table-body__cell u-text-right rating")
    ratingsNo = td4.text.strip()
    ratings.append(ratingsNo)

odi_team = pd.DataFrame({'Teams': teams, 'Matches': matches, 'Points': points,
odi_team.head(10) #gets only the top 10 results
```

Out[15]:

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

Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape and make data frame b) Top 10 ODI Batsmen along with the records of their team and rating.

In [16]: `page4=requests.get("https://www.icc-cricket.com/rankings/mens/player-rankings/captain-rankings/batting")
page4`

Out[16]: <Response [200]>

In [17]: `soup4=BeautifulSoup(page4.content)
soup4`

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

```
In [18]: ##Men ODI Batsman
positions = []
players = []
teams = []
ratings = []

divBatting = soup4.find("div", attrs={"data-cricket-role": "batting"})
batsman_table = divBatting.find("table", class_="table rankings-card-table")

positionOne = divBatting.find("span", class_="rankings-block__pos-number")
position=positionOne.text.strip()
positions.append(position)

playerOne = divBatting.find("div", class_="rankings-block__banner--name")
player=playerOne.text.strip()
players.append(player)

teamNation = divBatting.find("div", class_="rankings-block__banner--nationality")
team=teamNation.text.split(" ")[0].strip()
teams.append(team)
rating=teamNation.text.split(" ")[-1].strip()
ratings.append(rating)

table_tr = batsman_table.find_all("tr", class_="table-body")
for b in table_tr:
    td1 = b.find("span", class_="rankings-table__pos-number")
    position = td1.text.strip()
    positions.append(position)

    td2 = b.find("td", class_="table-body__cell name")
    player = td2.text.strip()
    players.append(player)

    td3 = b.find("span", class_="table-body__logo-text")
    team = td3.text.strip()
    teams.append(team)

    td4 = b.find("td", class_="table-body__cell u-text-right rating")
    rating = td4.text.strip()
    ratings.append(rating)

df = pd.DataFrame({"Position": positions, "Player": players, "Team Nation": teams})
df
```

Out[18]:

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

Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape and make data frame c) Top 10 ODI bowlers along with the records of their team and rating

In [19]: *##Men ODI Bowlers*

```
positions = []
players = []
teams = []
ratings = []

divBowling = soup4.find("div", attrs={"data-cricket-role": "bowling"})
bowling_table = divBowling.find("table", class_="table rankings-card-table")

positionOne = divBowling.find("span", class_="rankings-block__pos-number")
position=positionOne.text.strip()
positions.append(position)

playerOne = divBowling.find("div", class_="rankings-block__banner--name")
player=playerOne.text.strip()
players.append(player)

teamNation = divBowling.find("div", class_="rankings-block__banner--nationality")
team=teamNation.text.split(" ")[0].strip()
teams.append(team)
rating=teamNation.text.split(" ")[-1].strip()
ratings.append(rating)

table_tr = bowling_table.find_all("tr", class_="table-body")
for b in table_tr:
    td1 = b.find("span", class_="rankings-table__pos-number")
    position = td1.text.strip()
    positions.append(position)

    td2 = b.find("td", class_="table-body__cell name")
    player = td2.text.strip()
    players.append(player)

    td3 = b.find("span", class_="table-body__logo-text")
    team = td3.text.strip()
    teams.append(team)

    td4 = b.find("td", class_="table-body__cell u-text-right rating")
    rating = td4.text.strip()
    ratings.append(rating)

df = pd.DataFrame({"Position": positions, "Player": players, "Team Nation": teams, "Rating": ratings})
```

Out[19]:

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

Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape and make data frame a) Top 10 ODI teams in women's cricket along with the records for matches, points and rating.

In [20]: `page5=requests.get("https://www.icc-cricket.com/rankings/womens/team-rankings/c
page5`

Out[20]: <Response [200]>

In [21]: `soup5=BeautifulSoup(page5.content)
soup5`

Out[21]: <!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 mat
ch cricket teams. Discover latest ICC rankings table, predict upcoming mat
ches, see points and ratings for all teams." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official International Cricket Council rankings for test mat
ch cricket teams. Discover latest ICC rankings table, predict upcoming mat
ches, see points and ratings for all teams." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default
-thumbnail.jpg" name="twitter:image"/>
<meta content="ICC Women's ODI Team Rankings | ICC" property="og:title"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default
-thumbnail.jpg" property="og:image"/>
<title>ICC Women's ODI Team Rankings | ICC</title>
<meta content="Official International Cricket Council rankings for test mat
ch cricket teams. Discover latest ICC rankings table, predict upcoming mat
ches, see points and ratings for all teams." name="description"/>

```
In [22]: #Women ODI teams
points = []
teams = []
matches = []
ratings = []

table = soup5.find("table", class_="table")

teams_tr_b = table.find_all("tr", class_="rankings-block__banner")
for n in teams_tr_b:
    td1_b = n.find("span", class_="u-hide-phablet")
    teamOne = td1_b.text.strip()
    teams.append(teamOne)

    td2_b = n.find("td", class_="rankings-block__banner--matches")
    matchOne = td2_b.text.strip()
    matches.append(matchOne)

    td3_b = n.find("td", class_="rankings-block__banner--points")
    pointsOne = td3_b.text.strip()
    points.append(pointsOne)

    td4_b = n.find("td", class_="rankings-block__banner--rating")
    ratingOne = td4_b.text.strip()
    ratings.append(ratingOne)

teams_tr = table.find_all("tr", class_="table-body")
for u in teams_tr:
    td2 = u.find("span", class_="u-hide-phablet")
    teamsName = td2.text.strip()
    teams.append(teamsName)

    td3 = u.find_all("td", class_="table-body__cell u-center-text")
    matchesNo = td3[0].text.strip()
    pointsNo = td3[1].text.strip()
    matches.append(matchesNo)
    points.append(pointsNo)

    td4 = u.find("td", class_="table-body__cell u-text-right rating")
    ratingsNo = td4.text.strip()
    ratings.append(ratingsNo)

odi_team = pd.DataFrame({'Teams': teams, 'Matches': matches, 'Points': points,
odi_team.head(10)
```

Out[22]:

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

Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape and make data frame b) Top 10 women's ODI Batting players along with the records of their team and rating.

In [23]: `page6=requests.get("https://www.icc-cricket.com/rankings/womens/player-rankings")
page6`

Out[23]: <Response [200]>

In [24]: `soup6=BeautifulSoup(page6.content)
soup6`

Out[24]: <!DOCTYPE html>
<html lang="en">
<head>
<meta content="ICC Women's ODI Player Rankings | ICC" name="twitter:title"/>
<meta content="website" property="og:type"/>
<meta content="summary_large_image" property="twitter:card"/>
<meta content="Official International Cricket Council rankings for ODI match cricket players. Discover latest ICC rankings table, predict upcoming matches, see points and ratings for all teams." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official International Cricket Council rankings for ODI match cricket players. Discover latest ICC rankings table, predict upcoming matches, see points and ratings for all teams." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-thumbnail.jpg" name="twitter:image"/>
<meta content="ICC Women's ODI Player Rankings | ICC" property="og:title"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-thumbnail.jpg" property="og:image"/>
<title>ICC Women's ODI Player Rankings | ICC</title>

```
In [25]: ##Women ODI Batsman
positions = []
players = []
teams = []
ratings = []

divBatting = soup6.find("div", attrs={"data-cricket-role": "batting"})
batsman_table = divBatting.find("table", class_="table rankings-card-table")

positionOne = divBatting.find("span", class_="rankings-block__pos-number")
possition=positionOne.text.strip()
positions.append(possition)

playerOne = divBatting.find("div", class_="rankings-block__banner--name")
player=playerOne.text.strip()
players.append(player)

teamNation = divBatting.find("div", class_="rankings-block__banner--nationality")
team=teamNation.text.split(" ")[0].strip()
teams.append(team)
rating=teamNation.text.split(" ")[-1].strip()
ratings.append(rating)

table_tr = batsman_table.find_all("tr", class_="table-body")
for b in table_tr:
    td1 = b.find("span", class_="rankings-table__pos-number")
    possition = td1.text.strip()
    positions.append(possition)

    td2 = b.find("td", class_="table-body__cell name")
    player = td2.text.strip()
    players.append(player)

    td3 = b.find("span", class_="table-body__logo-text")
    team = td3.text.strip()
    teams.append(team)

    td4 = b.find("td", class_="table-body__cell u-text-right rating")
    rating = td4.text.strip()
    ratings.append(rating)

df = pd.DataFrame({"Position": positions, "Player": players, "Team Nation": t
df
```

Out[25]:

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

Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape and make data frame c) Top 10 women's ODI all-rounder along with the records of their team and rating.

```
In [26]: ##Women ODI Allrounder
positions = []
players = []
teams = []
ratings = []

divBowling = soup6.find("div", attrs={"data-cricket-role": "all_round"})
bowling_table = divBowling.find("table", class_="table rankings-card-table")

positionOne = divBowling.find("span", class_="rankings-block__pos-number")
possition=positionOne.text.strip()
positions.append(possition)

playerOne = divBowling.find("div", class_="rankings-block__banner--name")
player=playerOne.text.strip()
players.append(player)

teamNation = divBowling.find("div", class_="rankings-block__banner--nationality")
team=teamNation.text.split(" ")[0].strip()
teams.append(team)
rating=teamNation.text.split(" ")[-1].strip()
ratings.append(rating)

table_tr = bowling_table.find_all("tr", class_="table-body")
for b in table_tr:
    td1 = b.find("span", class_="rankings-table__pos-number")
    possition = td1.text.strip()
    positions.append(possition)

    td2 = b.find("td", class_="table-body__cell name")
    player = td2.text.strip()
    players.append(player)

    td3 = b.find("span", class_="table-body__logo-text")
    team = td3.text.strip()
    teams.append(team)

    td4 = b.find("td", class_="table-body__cell u-text-right rating")
    rating = td4.text.strip()
    ratings.append(rating)

df = pd.DataFrame({"Position": positions, "Player": players, "Team Nation": t
df
```

Out[26]:

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

Write a python program to scrape mentioned news details from <https://www.cnbc.com/world/?region=world> (<https://www.cnbc.com/world/?region=world>) and make data frame i) Headline ii) Time iii) News Link

In [27]:

```
page7=requests.get("https://www.cnbc.com/world/?region=world")
page7
```

Out[27]:

```
<Response [200]>
```

In [28]:

```
soup7=BeautifulSoup(page7.content)
soup7
```

Out[28]:

```
<!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; margin-bottom:20px}
```

```
In [29]: headlines = []
times = []
newsLink = []

latestNews = soup7.find("div", class_="LatestNews-isHomePage LatestNews-isIntl")
newsList = latestNews.find("ul", class_="LatestNews-list")
listContent = newsList.find("li", class_="LatestNews-item")

for l in newsList:
    heading = l.find("a", class_="LatestNews-headline")
    hText = heading.text.strip()
    headlines.append(hText)

    time = l.find("span", class_="LatestNews-wrapper")
    hTime = time.text.strip()
    times.append(hTime)

    link = l.find("a", href=True)
    hLink = link['href']
    newsLink.append(hLink)

df = pd.DataFrame({"Headlines": headlines, "Time": times, "News Link": newsLink})
df
```

Out[29]:

	Headlines	Time	News Link
0	In Disney-DeSantis fight, former AG Bill Barr ...	10 Min Ago	https://www.cnbc.com/2023/05/24/in-disney-desa...
1	A Harvard brain expert never does these 5 thin...	12 Min Ago	https://www.cnbc.com/2023/05/24/harvard-medica...
2	Harvard career expert: 3 strategies to land a ...	20 Min Ago	https://www.cnbc.com/2023/05/24/harvard-career...
3	Deutsche Bank says this small-cap stock can ra...	22 Min Ago	/pro/
4	Share of high earners living paycheck to paych...	31 Min Ago	https://www.cnbc.com/2023/05/24/share-of-high-...
5	A.I. poses existential risk of people being 'h...	1 Hour Ago	https://www.cnbc.com/2023/05/24/ai-poses-exist...
6	Jim Cramer's top 10 things to watch in the sto...	1 Hour Ago	/investingclub/
7	Financial coach: Why an HSA is at least 17% be...	2 Hours Ago	https://www.cnbc.com/2023/05/24/financial-coac...
8	Wednesday's top analyst calls: Tesla, Amazon, ...	2 Hours Ago	/pro/
9	Citigroup to spin off its Mexico business, Ban...	2 Hours Ago	https://www.cnbc.com/2023/05/24/citigroup-to-s...
10	Airbnb CEO: Here's 'the best piece of advice I...	2 Hours Ago	https://www.cnbc.com/2023/05/24/airbnb-ceo-bri...
11	Meta has started its latest round of layoffs, ...	2 Hours Ago	https://www.cnbc.com/2023/05/24/meta-layoffs-l...
12	SoftBank hits back at S&P after it cuts credit...	2 Hours Ago	https://www.cnbc.com/2023/05/24/softbank-hits-...
13	Abercrombie & Fitch surges more than 20% after...	2 Hours Ago	https://www.cnbc.com/2023/05/24/abercrombie-fi...
14	Bank of America boosts price target on Eli Lil...	2 Hours Ago	/pro/
15	Stocks making the biggest moves premarket: Mod...	2 Hours Ago	https://www.cnbc.com/2023/05/24/stocks-making-...
16	Without a hint of irony, Putin rages against c...	2 Hours Ago	https://www.cnbc.com/2023/05/24/having-invaded...
17	Top EU regulator defends mega \$1.3 billion pri...	2 Hours Ago	https://www.cnbc.com/2023/05/24/irish-data-reg...
18	Psychology expert: 7 toxic signs of a passive-...	3 Hours Ago	https://www.cnbc.com/2023/05/24/psychology-exp...
19	First Astranis internet satellite working 'per...	3 Hours Ago	https://www.cnbc.com/2023/05/24/astranis-first...
20	Chinese Tesla rival shares dive 11% after it f...	3 Hours Ago	https://www.cnbc.com/2023/05/24/xpeng-xpev-q1...
21	The U.S. isn't the only country with a debt ce...	3 Hours Ago	https://www.cnbc.com/2023/05/24/the-us-isnt-th...
22	Password sharing crackdown can lead to big gai...	3 Hours Ago	/pro/

	Headlines	Time	News Link
23	France bans short-haul flights as it looks to ...	3 Hours Ago	https://www.cnbc.com/2023/05/24/france-bans-do...
24	Kohl's shares spike as retailer reports a surp...	3 Hours Ago	https://www.cnbc.com/2023/05/24/kohls-kss-earn...
25	Mortgage demand drops again as rates cross bac...	4 Hours Ago	https://www.cnbc.com/2023/05/24/mortgage-deman...
26	Buy this beat-down regional bank with capital ...	4 Hours Ago	/pro/
27	Goldman Sachs backs Japan's biggest taxi app w...	4 Hours Ago	https://www.cnbc.com/2023/05/24/goldman-sachs-...
28	5 things to know before the stock market opens...	4 Hours Ago	https://www.cnbc.com/2023/05/24/5-things-to-kn...
29	Jefferies says this housing cycle bellwether c...	4 Hours Ago	/pro/

Write a python program to scrape the details of most downloaded articles from AI in last 90 days.<https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-articles>
[\(https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-articles\)](https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-articles) Scrape below mentioned details and make data frame
i) Paper Title ii) Authors iii) Published Date iv) Paper URL

```
In [30]: page8=requests.get("https://www.journals.elsevier.com/artificial-intelligence/n  
page8
```

Out[30]: <Response [200]>

```
In [31]: soup8=BeautifulSoup(page8.content)
          soup8
```

```
Out[31]: <!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/X0043702.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"/>
```

```
In [32]: paper_titles = []
authors = []
publish_dates = []
paper_urls = []

latest_news = soup8.find("div", class_="sc-orwwe2-3 j0MrrY")
news_ul = latest_news.find("ul", class_="sc-9zxyh7-0 cMKaMj")
news_li = news_ul.find("li", class_="sc-9zxyh7-1 sc-9zxyh7-2 kOEIEO hvoVxs")

for a in news_ul:
    title = a.find("h2", class_="sc-1qrq3sd-1 gRGSUS sc-1nmom32-0 sc-1nmom32-1")
    hText = title.text.strip()
    paper_titles.append(hText)

    author = a.find("span", class_="sc-1w3fpd7-0 dnCnAO")
    aText = author.text.strip()
    authors.append(aText)

    date = a.find("span", class_="sc-1thf9ly-2 dvggwt")
    dText = date.text.strip()
    publish_dates.append(dText)

    url = a.find("a", href=True)
    uText = url['href']
    paper_urls.append(uText)

df = pd.DataFrame({"Paper Title": paper_titles, "Authors": authors, "Published": publish_dates})
df
```

Out[32]:

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

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

Write a python program to scrape mentioned details from dineout.co.in and make data frame
 i) Restaurant name ii) Cuisine iii) Location iv) Ratings v) Image UR

Hi, I am not able to access the website for DineOut it might be blocked for the country where I live. We have an assignment (Q7) where I couldn't get the access for the website. I had raised the ticket and the ticket no is 16508.

Thank You

Regards

Meghana Mouli