WEB SCRAPING ASSIGNMENT_1

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

In [1]:
```
#importing requires libraries
from bs4 import BeautifulSoup
import requests
```

In [2]:
```
page=requests.get('https://en.wikipedia.org/wiki/Main_Page')
```

In [3]:
```
page
```

Out[3]:
```
<Response [200]>
```

In [4]:
```
soup=BeautifulSoup(page.content)
soup
```

```
mw-parser-output .mp-h2{margin:0.5em;padding:0.2em 0.4em;font-size:120%;font-
weight:bold;font-family:inherit}.mw-parser-output h2.mp-h2::after{border:non
e}.mw-parser-output .mp-later{font-size:85%;font-weight:normal}.mw-parser-out
put #mp-upper{width:100%;margin-top:4px;margin-bottom:0;border-spacing:0;bord
er-collapse:separate}.mw-parser-output #mp-upper .mid-table{border-color:tran
sparent}.mw-parser-output #mp-left{width:55%;border-color:#cef2e0;background:
#f5fffa}.mw-parser-output #mp-right{width:45%;border-color:#cedff2;backgroun
d:#f5faff}.mw-parser-output #mp-left,.mw-parser-output #mp-right{padding:0;ve
rtical-align:top}.mw-parser-output #mp-left .mp-h2{background:#cef2e0;border-
color:#a3bfb1}.mw-parser-output #mp-right .mp-h2{background:#cedff2;border-co
lor:#a3b0bf}.mw-parser-output #mp-tfa,.mw-parser-output #mp-dyk,.mw-parser-ou
tput #mp-itn,.mw-parser-output #mp-otd,.mw-parser-output #mp-other-lower>div
{padding:0.1em 0.6em}.mw-parser-output #mp-dyk-h2,.mw-parser-output #mp-otd-h

2{clear:both}.mw-parser-output #mp-middle{margin-top:4px;border-color:#f2ced
d;background:#fff5fa}.mw-parser-output #mp-middle,.mw-parser-output #mp-lowe
r,.mw-parser-output #mp-other-lower{overflow:auto}.mw-parser-output #mp-tfl-h
2{background:#f2cedd;border-color:#bfa3af}.mw-parser-output #mp-tfl{padding:
0.3em 0.7em}.mw-parser-output #mp-lower{margin-top:4px;border-color:#ddcef2;b
ackground:#faf5ff}.mw-parser-output #mp-tfp-h2{background:#ddcef2;border-colo
```

In [5]:
```
titles=[]
for i in soup.find_all('span',class_="mw-headline"):
    titles.append(i.text)

titles
```

Out[5]:
```
["From today's featured article",
 'Did you know\xa0...',
 'In the news',
 'On this day',
 "Today's featured picture",
 'Other areas of Wikipedia',
 "Wikipedia's sister projects",
 'Wikipedia languages']
```

2)Write a python program to display IMDB's Top rated 100 movies' data (i.e. name, rating, year of release)and make data frame.

In [6]:
```
movies=requests.get('https://www.imdb.com/list/ls055592025/')
movies
```

Out[6]:
```
<Response [200]>
```

In [7]:
```
soup1=BeautifulSoup(movies.content)
soup1
```

```
</script>

<link href="https://www.imdb.com/list/ls055592025/" rel="canonical"/>
<meta content="http://www.imdb.com/list/ls055592025/" property="og:url"/>
<script>
    if (typeof uet == 'function') {
      uet("bb", "LoadIcons", {wb: 1});
    }
</script>
<script>(function(t){ (t.events = t.events || {})["csm_head_pre_icon"] = new
Date().getTime(); })(IMDbTimer);</script>
<link href="https://m.media-amazon.com/images/G/01/imdb/images-ANDW73HA/favic
on_desktop_32x32._CB1582158068_.png" rel="icon" sizes="32x32"/>
<link href="https://m.media-amazon.com/images/G/01/imdb/images-ANDW73HA/favic
on_iPad_retina_167x167._CB1582158068_.png" rel="icon" sizes="167x167"/>
<link href="https://m.media-amazon.com/images/G/01/imdb/images-ANDW73HA/favic
on_iPhone_retina_180x180._CB1582158069_.png" rel="icon" sizes="180x180"/>
<link href="https://m.media-amazon.com/images/G/01/imdb/images-ANDW73HA/apple
-touch-icon-mobile._CB479963088_.png" rel="apple-touch-icon"/>
<link href="https://m.media-amazon.com/images/G/01/imdb/images-ANDW73HA/apple
```

In [8]:
```
names=[]
for i in soup1.find_all('h3',class_="lister-item-header"):
    names.append(i.find('a').text)

names
```

Out[8]:
```
['The Godfather',
 'The Shawshank Redemption',
 "Schindler's List",
 'Raging Bull',
 'Casablanca',
 'Citizen Kane',
 'Gone with the Wind',
 'The Wizard of Oz',
 "One Flew Over the Cuckoo's Nest",
 'Lawrence of Arabia',
 'Vertigo',
 'Psycho',
 'The Godfather: Part II',
 'On the Waterfront',
 'Sunset Blvd.',
 'Forrest Gump',
 'The Sound of Music',
 '12 Angry Men',
 'West Side Story',
 'Star Wars',
 '2001: A Space Odyssey',
 'E.T. the Extra-Terrestrial',
 'The Silence of the Lambs',
 'Chinatown',
 'The Bridge on the River Kwai',
 "Singin' in the Rain",
 "It's a Wonderful Life",
 'Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb',
 'Some Like It Hot',
 'Ben-Hur',
 'Apocalypse Now',
 'Amadeus',
 'The Lord of the Rings: The Return of the King',
 'Gladiator',
 'Titanic',
 'From Here to Eternity',
 'Saving Private Ryan',
 'Unforgiven',
 'Raiders of the Lost Ark',
 'Rocky',
 'A Streetcar Named Desire',
 'The Philadelphia Story',
 'To Kill a Mockingbird',
 'An American in Paris',
 'The Best Years of Our Lives',
 'My Fair Lady',
 'A Clockwork Orange',
 'Doctor Zhivago',
 'The Searchers',
 'Jaws',
```

```
 'Patton',
 'Butch Cassidy and the Sundance Kid',
 'The Treasure of the Sierra Madre',
 'Il buono, il brutto, il cattivo',
 'The Apartment',
 'Platoon',
 'High Noon',
 'Braveheart',
 'Dances with Wolves',
 'Jurassic Park',
 'The Exorcist',
 'The Pianist',
 'Goodfellas',
 'The Deer Hunter',
 'All Quiet on the Western Front',
 'Bonnie and Clyde',
 'The French Connection',
 'City Lights',
 'It Happened One Night',
 'A Place in the Sun',
 'Midnight Cowboy',
 'Mr. Smith Goes to Washington',
 'Rain Man',
 'Annie Hall',
 'Fargo',
 'Giant',
 'Shane',
 'The Grapes of Wrath',
 'The Green Mile',
 'Close Encounters of the Third Kind',
 'Nashville',
 'Network',
 'The Graduate',
 'American Graffiti',
 'Pulp Fiction',
 'Terms of Endearment',
 'Good Will Hunting',
 'The African Queen',
 'Stagecoach',
 'Mutiny on the Bounty',
 'The Great Dictator',
 'Double Indemnity',
 'The Maltese Falcon',
 'Wuthering Heights',
 'Taxi Driver',
 'Rear Window',
 'The Third Man',
 'Rebel Without a Cause',
 'North by Northwest',
 'Yankee Doodle Dandy']
```

In [9]:
```
rating=[]
for i in soup1.find_all('div',class_="ipl-rating-star small"):
   rating.append(i.text.replace('\n',''))

rating
```

Out[9]:
```
['9.2',
 '9.3',
 '8.9',
 '8.2',
 '8.5',
 '8.3',
 '8.1',
 '8',
 '8.7',
 '8.3',
 '8.3',
 '8.5',
 '9',
 '8.1',
 '8.4',
 '8.8',
 '8',
 '9',
 '7.5',
 '8.6',
 '8.3',
 '7.8',
 '8.6',
 '8.2',
 '8.1',
 '8.3',
 '8.6',
 '8.4',
 '8.2',
 '8.1',
 '8.4',
 '8.3',
 '8.9',
 '8.5',
 '7.8',
 '7.6',
 '8.6',
 '8.2',
 '8.4',
 '8.1',
 '8',
 '7.9',
 '8.2',
 '7.2',
 '8.1',
 '7.8',
 '8.3',
 '8',
 '7.9',
 '8',
```

```
'7.9',
'8',
'8.2',
'8.8',
'8.3',
'8.1',
'8',
'8.3',
'8',
'8.1',
'8',
'8.5',
'8.7',
'8.1',
'8.1',
'7.8',
'7.7',
'8.5',
'8.1',
'7.8',
'7.8',
'8.1',
'8',
'8',
'8.1',
'7.6',
'7.6',
'8.1',
'8.6',
'7.6',
'7.7',
'8.1',
'8',
'7.4',
'8.9',
'7.4',
'8.3',
'7.8',
'7.9',
'7.7',
'8.4',
'8.3',
'8',
'7.6',
'8.2',
'8.5',
'8.1',
'7.7',
'8.3',
'7.7']
```

In [10]:
```
year=[]
for i in soup1.find_all("span",class_="lister-item-year text-muted unbold"):
    year.append(i.text.replace('(','').replace(')',''))
year
```

Out[10]:
```
['1972',
 '1994',
 '1993',
 '1980',
 '1942',
 '1941',
 '1939',
 '1939',
 '1975',
 '1962',
 '1958',
 '1960',
 '1974',
 '1954',
 '1950',
 '1994',
 '1965',
 '1957',
 '1961',
 '1977',
 '1968',
 '1982',
 '1991',
 '1974',
 '1957',
 '1952',
 '1946',
 '1964',
 '1959',
 '1959',
 '1979',
 '1984',
 '2003',
 '2000',
 '1997',
 '1953',
 '1998',
 '1992',
 '1981',
 '1976',
 '1951',
 '1940',
 '1962',
 '1951',
 '1946',
 '1964',
 '1971',
 '1965',
 '1956',
 '1975',
 '1970',
```

```
 '1969',
 '1948',
 '1966',
 '1960',
 '1986',
 '1952',
 '1995',
 '1990',
 '1993',
 '1973',
 '2002',
 '1990',
 '1978',
 '1930',
 '1967',
 '1971',
 '1931',
 '1934',
 '1951',
 '1969',
 '1939',
 '1988',
 '1977',
 '1996',
 '1956',
 '1953',
 '1940',
 '1999',
 '1977',
 '1975',
 '1976',
 '1967',
 '1973',
 '1994',
 '1983',
 '1997',
 '1951',
 '1939',
 '1935',
 '1940',
 '1944',
 '1941',
 '1939',
 '1976',
 '1954',
 '1949',
 '1955',
 '1959',
 '1942']
```

In [11]:
```
#Printing Length
print(len(names),len(rating),len(year))
```

```
100 100 100
```

In [12]:
```
#Making dataframe
import pandas as pd
df=pd.DataFrame({'MovieName':names,'MovieRating':rating,'YearOfRelease':year})
df
```

Out[12]:

	MovieName	MovieRating	YearOfRelease
0	The Godfather	9.2	1972
1	The Shawshank Redemption	9.3	1994
2	Schindler's List	8.9	1993
3	Raging Bull	8.2	1980
4	Casablanca	8.5	1942
...	...	...	...
95	Rear Window	8.5	1954
96	The Third Man	8.1	1949
97	Rebel Without a Cause	7.7	1955
98	North by Northwest	8.3	1959
99	Yankee Doodle Dandy	7.7	1942

100 rows × 3 columns

3)Write a python program to display IMDB's Top rated 100 Indian movies' data (i.e. name, rating, year of release) and make data frame.

In [13]:
```
Imovies=requests.get('https://www.imdb.com/list/ls056092300/')
Imovies
```

Out[13]:
```
<Response [200]>
```

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

Out[14]:
```
<!DOCTYPE html>
<html xmlns:fb="http://www.facebook.com/2008/fbml" xmlns:og="http://ogp.me/ns
#">
<head>
<meta charset="utf-8"/>
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<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>100 Best Indian Films Till Date - IMDb</title>
<script>(function(t){ (t.events = t.events || {})["csm_head_post_title"] = ne
w Date().getTime(); })(IMDbTimer);</script>
<script>
```

In [15]:
```
Inames=[]
for i in soup3.find_all('h3',class_="lister-item-header"):
    Inames.append(i.find('a').text)

Inames
```

Out[15]:
```
['Ship of Theseus',
 'Iruvar',
 'Kaagaz Ke Phool',
 'Lagaan: Once Upon a Time in India',
 'Pather Panchali',
 'Charulata',
 'Rang De Basanti',
 'Dev.D',
 '3 Idiots',
 'Awaara',
 'Nayakan',
 'Aparajito',
 'Pushpaka Vimana',
 'Pyaasa',
 'Ghatashraddha',
 'Sholay',
 'Aradhana',
 'Do Ankhen Barah Haath',
 'Bombay',
 'Neecha Nagar',
 'Do Bigha Zamin',
 'Garm Hava',
 'Piravi',
 'Mughal-E-Azam',
 'Amma Ariyan',
 'Madhumati',
 'Goopy Gyne Bagha Byne',
 'Gangs of Wasseypur',
 'Guide',
 'Satya',
 'Roja',
 'Mr. India',
 'The Cloud-Capped Star',
 'Harishchandrachi Factory',
 'Masoom',
 'Agneepath',
 'Tabarana Kathe',
 'Zakhm',
 'Dil Chahta Hai',
 'Bhaag Milkha Bhaag',
 'Chupke Chupke',
 'Dilwale Dulhania Le Jayenge',
 'Taare Zameen Par',
 'Ardh Satya',
 'Bhumika',
 'Enthiran',
 'Sadma',
 'Shwaas',
 'Lamhe',
 'Haqeeqat',
```

```
 'Shree 420',
 'Kannathil Muthamittal',
 'Hum Aapke Hain Koun..!',
 'Ustad Hotel',
 'Bandit Queen',
 'Lakshya',
 'Black Friday',
 'Manthan',
 'Apoorva Raagangal',
 'English Vinglish',
 'Jewel Thief',
 'Pakeezah',
 'Maqbool',
 'Jis Desh Men Ganga Behti Hai',
 'Sahib Bibi Aur Ghulam',
 'Shatranj Ke Khilari',
 'Narthanasala',
 'Chandni Bar',
 'Vaaranam Aayiram',
 'Mr. and Mrs. Iyer',
 'Chandni',
 'English, August',
 'Celluloid',
 'Sagara Sangamam',
 'Munna Bhai M.B.B.S.',
 'Saaransh',
 'Guddi',
 'Vanaja',
 'Vazhakku Enn 18/9',
 'Gangaajal',
 'Angoor',
 'Guru',
 'Andaz Apna Apna',
 'Sangam',
 'Oka Oori Katha',
 'Bhuvan Shome',
 'Border',
 'Parineeta',
 'Devdas',
 'Abohomaan',
 'Kuch Kuch Hota Hai',
 'Pithamagan',
 'Veyyil',
 'Chemmeen',
 'Jaane Bhi Do Yaaro',
 'Apur Sansar',
 'Kanchivaram',
 'Monsoon Wedding',
 'Black',
 'Deewaar']
```

In [16]:
```
Irating=[]
for i in soup3.find_all('div',class_="ipl-rating-star small"):
   Irating.append(i.text.replace('\n',''))

Irating
```

Out[16]:
```
['8.1',
 '8.5',
 '8',
 '8.1',
 '8.4',
 '8.2',
 '8.1',
 '8',
 '8.4',
 '7.9',
 '8.7',
 '8.3',
 '8.6',
 '8.5',
 '7.3',
 '8.2',
 '7.7',
 '8.4',
 '8.1',
 '6.9',
 '8.3',
 '8.1',
 '7.7',
 '8.2',
 '7.1',
 '8',
 '8.8',
 '8.2',
 '8.4',
 '8.3',
 '8.2',
 '7.8',
 '8',
 '8.4',
 '8.4',
 '7.7',
 '8',
 '7.9',
 '8.1',
 '8.2',
 '8.3',
 '8.1',
 '8.4',
 '8.2',
 '7.4',
 '7.1',
 '8.4',
 '8.3',
 '7.4',
 '7.8',
```

```
 '8',
 '8.4',
 '7.5',
 '8.2',
 '7.6',
 '7.9',
 '8.5',
 '7.7',
 '7.5',
 '7.8',
 '8',
 '7.4',
 '8.1',
 '7.4',
 '8.3',
 '7.7',
 '7.7',
 '7.6',
 '8.2',
 '7.9',
 '6.8',
 '6.9',
 '7.7',
 '8.8',
 '8.1',
 '8.2',
 '7.2',
 '7.2',
 '8.3',
 '7.8',
 '8.3',
 '7.7',
 '8.1',
 '7.5',
 '7.7',
 '7.4',
 '7.9',
 '7.2',
 '7.9',
 '7.4',
 '7.6',
 '8.4',
 '7.8',
 '7.8',
 '8.4',
 '8.5',
 '8.1',
 '7.4',
 '8.2',
 '8.1']
```

In [17]:
```
Iyear=[]
for i in soup3.find_all("span",class_="lister-item-year text-muted unbold"):
    Iyear.append(i.text.replace('(','').replace(')',''))
Iyear
```

Out[17]:
```
['2012',
 '1997',
 '1959',
 '2001',
 '1955',
 '1964',
 '2006',
 '2009',
 '2009',
 '1951',
 '1987',
 '1956',
 '1987',
 '1957',
 '1977',
 '1975',
 '1969',
 '1957',
 '1995',
 '1946',
 '1953',
 '1974',
 '1989',
 '1960',
 '1986',
 '1958',
 '1969',
 '2012',
 '1965',
 '1998',
 '1992',
 '1987',
 '1960',
 '2009',
 '1983',
 '1990',
 '1986',
 '1998',
 '2001',
 '2013',
 '1975',
 '1995',
 '2007',
 '1983',
 '1977',
 '2010',
 '1983',
 '2004',
 '1991',
 '1964',
 '1955',
```

```
'2002',
'1994',
'2012',
'1994',
'2004',
'2004',
'1976',
'1975',
'2012',
'1967',
'1972',
'2003',
'1960',
'1962',
'1977',
'1963',
'2001',
'2008',
'2002',
'1989',
'1994',
'2013',
'1983',
'2003',
'1984',
'1971',
'2006',
'2012',
'2003',
'1982',
'2007',
'1994',
'I 1964',
'1978',
'1969',
'I 1997',
'2005',
'1955',
'2009',
'1998',
'2003',
'2006',
'1965',
'1983',
'1959',
'2008',
'2001',
'2005',
'1975']
```

In [18]:
```
#Printing Length
print(len(Inames),len(Irating),len(Iyear))
```

```
100 100 100
```

In [19]:
```
#Making dataframe
import pandas as pd
df=pd.DataFrame({'MovieName':Inames,'MovieRating':Irating,'YearOfRelease':Iyear})
df
```

Out[19]:

	MovieName	MovieRating	YearOfRelease
0	Ship of Theseus	8.1	2012
1	Iruvar	8.5	1997
2	Kaagaz Ke Phool	8	1959
3	Lagaan: Once Upon a Time in India	8.1	2001
4	Pather Panchali	8.4	1955
...	...	...	...
95	Apur Sansar	8.5	1959
96	Kanchivaram	8.1	2008
97	Monsoon Wedding	7.4	2001
98	Black	8.2	2005
99	Deewaar	8.1	1975

100 rows × 3 columns

4)Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape:

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

In [20]:
```
Mteams=requests.get('https://www.icc-cricket.com/rankings/mens/team-rankings/odi'
Mteams
```

Out[20]:
```
<Response [200]>
```

In [21]:
```
soup4a=BeautifulSoup(Mteams.content)
soup4a
```

Out[21]:
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta content="ICC Ranking for ODI teams International Cricket Council" 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 Int
ernational (ODI) cricket teams. Discover latest ICC rankings table, predict u
pcoming 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 Int
ernational (ODI) cricket teams. Discover latest ICC rankings table, predict u
pcoming matches, see points and ratings for all teams." name="twitter:descrip
tion"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-t
humbnail.jpg" name="twitter:image"/>
<meta content="ICC Ranking for ODI teams International Cricket Council" prope
rty="og:title"/>
```

In [22]:
```
Mteam=[]
for i in soup4a.find_all("span",class_="u-hide-phablet")[0:10]:
    Mteam.append(i.text)

Mteam
```

Out[22]:
```
['New Zealand',
 'England',
 'Australia',
 'India',
 'South Africa',
 'Pakistan',
 'Bangladesh',
 'West Indies',
 'Sri Lanka',
 'Afghanistan']
```

In [23]:
```
Mrates=[]
for i in soup4a.find_all("td",class_="rankings-block__banner--rating u-text-right
    Mrates.append(i.text.replace('\n                                ','').replace('\r
for i in soup4a.find_all("td",class_="table-body__cell u-text-right rating")[0:10
        Mrates.append(i.text)
Mrates
```

Out[23]:
```
['121', '119', '116', '113', '98', '93', '91', '84', '83', '62', '48']
```

In [24]:
```
Mmatches=[]
Mpoints=[]
Mnewlist=[]
for i in soup4a.find_all("td",class_="rankings-block__banner--matches"):
    Mmatches.append(i.text)
for i in soup4a.find_all("td",class_="rankings-block__banner--points"):
    Mpoints.append(i.text)
for i in soup4a.find_all("td",class_="table-body__cell u-center-text")[0:20]:
    Mnewlist.append(i.text)
for i in range(0,len(Mnewlist)-1,2):
    Mmatches.append(Mnewlist[i])
    Mpoints.append(Mnewlist[i+1])
Mmatches
```

Out[24]:
```
['17', '32', '28', '32', '25', '27', '30', '30', '32', '17', '7']
```

In [25]:
```
Mpoints
```

Out[25]:
```
['2,054',
 '3,793',
 '3,244',
 '3,624',
 '2,459',
 '2,524',
 '2,740',
 '2,523',
 '2,657',
 '1,054',
 '336']
```

4b)Top 10 ODI Batsmen in men along with the records of their team and rating

In [26]:
```
MBat=requests.get('https://www.icc-cricket.com/rankings/mens/player-rankings/odi/
MBat
```

Out[26]:
```
<Response [200]>
```

In [27]:
```
soup4b=BeautifulSoup(MBat.content)
soup4b
```

```
    }(document, 'script', 'facebook-jssdk'));
</script>
<div class="user-account-overlay js-user-account-overlay"></div>
<section class="user-account js-user-account" data-script="sso_user-account"
 data-widget="user-account">
<div class="user-account__wrapper js-panel-account u-hide">
<div class="user-account__header">
<div class="user-account__close js-account-slider-close-btn" role="button">
<svg class="icon"><use xlink:href="/resources/prod/v8.20.18/i/svg-output/icon
s.svg#icn-close" xmlns:xlink="http://www.w3.org/1999/xlink"></use></svg>
</div>
<div class="user-account__user-wrapper">
<div class="user-account__image">
<img alt="User Image" class="u-hide js-account-user-img" src="#"/>
</div>
<div class="user-account__user-info">
<p class="user-account__user-info__text js-account-user">Name Lastname</p>
<p class="user-account__user-info__profile js-profile-completion-info u-hid
e">
```

In [28]:
```
batsmen=[]
for i in soup4b.find_all("div",class_="rankings-block__banner--name-large"):
    batsmen.append(i.text)
for i in soup4b.find_all("td",class_="table-body__cell rankings-table__name name"
        batsmen.append(i.find('a').text)
batsmen
```

Out[28]:
```
['Babar Azam',
 'Virat Kohli',
 'Rohit Sharma',
 'Ross Taylor',
 'Aaron Finch',
 'Jonny Bairstow',
 'David Warner',
 'Shai Hope',
 'Kane Williamson',
 'Quinton de Kock',
 'Fakhar Zaman']
```

In [29]:
```
batteam=[]
for i in soup4b.find_all("div",class_="rankings-block__banner--nationality"):
    batteam.append(i.text.replace('\n\n','').replace('\n',''))
for i in soup4b.find_all("span",class_="table-body__logo-text")[0:10]:
    batteam.append(i.text)
batteam
```

Out[29]:
```
['PAK                    ',
 'IND',
 'IND',
 'NZ',
 'AUS',
 'ENG',
 'AUS',
 'WI',
 'NZ',
 'SA',
 'PAK']
```

In [30]:
```
batrate=[]
for i in soup4b.find_all("div",class_="rankings-block__banner--rating"):
    batrate.append(i.text)
for i in soup4b.find_all("td",class_="table-body__cell rating")[0:10]:
    batrate.append(i.text)
batrate
```

Out[30]:
```
['873', '844', '813', '801', '779', '775', '762', '758', '754', '743', '741']
```

4c)Top 10 ODI bowlers along with the records of their team and rating.

In [31]:
```
MBall=requests.get("https://www.icc-cricket.com/rankings/mens/player-rankings/odi
MBall
```

Out[31]:
```
<Response [200]>
```

In [32]:
```
soup4c=BeautifulSoup(MBall.content)
soup4c
```

Out[32]:
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta content="Live Cricket Scores &amp;amp; News International Cricket Council"
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, hig
hlights, commentary, rankings, videos and fixtures from the International Cri
cket Council." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official ICC Cricket website - live matches, scores, news, hig
hlights, commentary, rankings, videos and fixtures from the International Cri
cket Council." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-t
humbnail.jpg" name="twitter:image"/>
<meta content="Live Cricket Scores &amp;amp; News International Cricket Council"
property="og:title"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-t
```

In [33]:
```
bowlsmen=[]
for i in soup4c.find_all("div",class_="rankings-block__banner--name-large"):
    bowlsmen.append(i.text)
for i in soup4c.find_all("td",class_="table-body__cell rankings-table__name name"
    bowlsmen.append(i.find('a').text)
bowlsmen
```

Out[33]:
```
['Trent Boult',
 'Josh Hazlewood',
 'Mujeeb Ur Rahman',
 'Chris Woakes',
 'Mehedi Hasan',
 'Matt Henry',
 'Jasprit Bumrah',
 'Mitchell Starc',
 'Shakib Al Hasan',
 'Kagiso Rabada',
 'Mustafizur Rahman']
```

In [34]:
```
bowlteam=[]
for i in soup4c.find_all("div",class_="rankings-block__banner--nationality"):
    bowlteam.append(i.text.replace('\n\n','').replace('\n',''))
for i in soup4c.find_all("span",class_="table-body__logo-text")[0:10]:
    bowlteam.append(i.text)
bowlteam
```

Out[34]:
```
['NZ                    ',
 'AUS',
 'AFG',
 'ENG',
 'BAN',
 'NZ',
 'IND',
 'AUS',
 'BAN',
 'SA',
 'BAN']
```

In [35]:
```
bowlrate=[]
for i in soup4c.find_all("div",class_="rankings-block__banner--rating"):
    bowlrate.append(i.text)
for i in soup4c.find_all("td",class_="table-body__cell rating")[0:10]:
    bowlrate.append(i.text)
bowlrate
```

Out[35]:
```
['737', '709', '708', '700', '692', '691', '679', '652', '650', '643', '640']
```

5) Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape:

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

In [36]:
```
Wteams=requests.get("https://www.icc-cricket.com/rankings/womens/team-rankings/oc
Wteams
```

Out[36]:
```
<Response [200]>
```

In [37]:
```
soup5=BeautifulSoup(Wteams.content)
soup5
```

```
    };
</script>
<link href="https://fonts.googleapis.com/css?family=Hind+Siliguri:300,400,500,600" rel="stylesheet"/>
<!-- Polyfill service provided by the FT - https://github.com/Financial-Times/polyfill-service (https://github.com/Financial-Times/polyfill-service) -->
<script src="https://cdn.polyfill.io/v2/polyfill.min.js?features=default,fetch"></script>
<link href="/resources/prod/v8.20.18/styles/screen.css" rel="stylesheet"/>
<script>
                window.RESOURCE_VERSION = 'prod/v8.20.18';
        </script>
</head>
<body>
<!-- Google Tag Manager (noscript) -->
<noscript><iframe height="0" src="https://www.googletagmanager.com/ns.html?id=GTM-T7DPH24" style="display:none;visibility:hidden" width="0"></iframe></noscript>
<!-- End Google Tag Manager (noscript) -->
```

In [38]:
```
Wteam=[]
for i in soup5.find_all("span",class_="u-hide-phablet"):
    Wteam.append(i.text.replace("'',",''))
Wteam
```

Out[38]:
```
['Australia',
 'South Africa',
 'England',
 'India',
 'Bangladesh',
 'New Zealand',
 'West Indies',
 'Pakistan',
 'Ireland',
 'Sri Lanka',
 'Zimbabwe',
 '',
 '',
 '',
 '',
 '',
 '']
```

In [39]:
```
Wmatches=[]
Wnewlist=[]
Wpoints=[]
for i in soup5.find_all("td",class_="rankings-block__banner--matches"):
    Wmatches.append(i.text)
for i in soup5.find_all("td",class_="rankings-block__banner--points"):
    Wpoints.append(i.text)
for i in soup5.find_all("td",class_="table-body__cell u-center-text"):
    Wnewlist.append(i.text)
for i in range(0,len(Wnewlist)-1,2):
    Wmatches.append(Wnewlist[i])
    Wpoints.append(Wnewlist[i+1])
Wmatches
```

Out[39]:
```
['17', '19', '18', '17', '5', '19', '19', '18', '5', '5', '7']
```

In [40]:
```
Wpoints
```

Out[40]:
```
['2,746',
 '2,307',
 '2,148',
 '1,899',
 '475',
 '1,668',
 '1,658',
 '1,226',
 '240',
 '233',
 '0']
```

In [41]:
```
Wrate=[]
for i in soup5.find_all("td",class_="rankings-block__banner--rating u-text-right'
    Wrate.append(i.text.replace('\n                                ','').replace('\n
for i in soup5.find_all("td",class_="table-body__cell u-text-right rating"):
    Wrate.append(i.text)
Wrate
```

Out[41]:
```
['162', '121', '119', '112', '95', '88', '87', '68', '48', '47', '0']
```

5b) Top 10 women's ODI players along with the records of their team and rating.

In [42]:
```
Wbat=requests.get('https://www.icc-cricket.com/rankings/womens/player-rankings/oc
Wbat
```

Out[42]:
```
<Response [200]>
```

In [43]:
```
soup5b1=BeautifulSoup(Wbat.content)
soup5b1
```

Out[43]:
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta content="Live Cricket Scores &amp;amp; News International Cricket Council"
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, hig
hlights, commentary, rankings, videos and fixtures from the International Cri
cket Council." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official ICC Cricket website - live matches, scores, news, hig
hlights, commentary, rankings, videos and fixtures from the International Cri
cket Council." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-t
humbnail.jpg" name="twitter:image"/>
<meta content="Live Cricket Scores &amp;amp; News International Cricket Council"
property="og:title"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-t
```

In [44]:
```
WB=[]
for i in soup5b1.find_all("div",class_="rankings-block__banner--name-large"):
    WB.append(i.text)
for i in soup5b1.find_all("td",class_="table-body__cell rankings-table__name name
    WB.append(i.find('a').text)
WB
```

Out[44]:
```
['Lizelle Lee',
 'Alyssa Healy',
 'Mithali Raj',
 'Tammy Beaumont',
 'Amy Satterthwaite',
 'Smriti Mandhana',
 'Meg Lanning',
 'Beth Mooney',
 'Stafanie Taylor',
 'Heather Knight',
 'Laura Wolvaardt']
```

In [45]:
```
WBteam=[]
for i in soup5b1.find_all("div",class_="rankings-block__banner--nationality"):
    WBteam.append(i.text.replace('\n\n','').replace('\n',''))
for i in soup5b1.find_all("span",class_="table-body__logo-text")[0:10]:
    WBteam.append(i.text)
WBteam
```

Out[45]:
```
['SA                    ',
 'AUS',
 'IND',
 'ENG',
 'NZ',
 'IND',
 'AUS',
 'AUS',
 'WI',
 'ENG',
 'SA']
```

In [46]:
```
WBrate=[]
for i in soup5b1.find_all("div",class_="rankings-block__banner--rating"):
    WBrate.append(i.text)
for i in soup5b1.find_all("td",class_="table-body__cell rating")[0:10]:
    WBrate.append(i.text)
WBrate
```

Out[46]:
```
['761', '750', '738', '728', '717', '710', '699', '690', '676', '674', '672']
```

In [47]:
```
Wbowl=requests.get("https://www.icc-cricket.com/rankings/womens/player-rankings/c
Wbowl
```

Out[47]:
```
<Response [200]>
```

In [48]:
```
soup5b2=BeautifulSoup(Wbowl.content)
soup5b2
```

Out[48]:
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta content="Live Cricket Scores &amp;amp; News International Cricket Council"
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, hig
hlights, commentary, rankings, videos and fixtures from the International Cri
cket Council." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official ICC Cricket website - live matches, scores, news, hig
hlights, commentary, rankings, videos and fixtures from the International Cri
cket Council." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-t
humbnail.jpg" name="twitter:image"/>
<meta content="Live Cricket Scores &amp;amp; News International Cricket Council"
property="og:title"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-t
```

In [49]:
```
Wbowlers=[]
for i in soup5b2.find_all("div",class_="rankings-block__banner--name-large"):
    Wbowlers.append(i.text)
for i in soup5b2.find_all("td",class_="table-body__cell rankings-table__name name
    Wbowlers.append(i.find('a').text)
Wbowlers
```

Out[49]:
```
['Jess Jonassen',
 'Jhulan Goswami',
 'Megan Schutt',
 'Marizanne Kapp',
 'Sophie Ecclestone',
 'Shabnim Ismail',
 'Katherine Brunt',
 'Ayabonga Khaka',
 'Anya Shrubsole',
 'Kate Cross',
 'Natalie Sciver']
```

In [50]:
```
Wbteam=[]
for i in soup5b2.find_all("div",class_="rankings-block__banner--nationality"):
    Wbteam.append(i.text.replace('\n\n','').replace('\n',''))
for i in soup5b2.find_all("span",class_="table-body__logo-text")[0:10]:
    Wbteam.append(i.text)
Wbteam
```

Out[50]:
```
['AUS                    ',
 'IND',
 'AUS',
 'SA',
 'ENG',
 'SA',
 'ENG',
 'SA',
 'ENG',
 'ENG',
 'ENG']
```

In [51]:
```
Wbrate=[]
for i in soup5b2.find_all("div",class_="rankings-block__banner--rating"):
    Wbrate.append(i.text)
for i in soup5b2.find_all("td",class_="table-body__cell rating")[0:10]:
    Wbrate.append(i.text)
Wbrate
```

Out[51]:
```
['760', '727', '717', '715', '701', '688', '666', '643', '598', '589', '580']
```

5c) Top 10 women's ODI all-rounder along with the records of their team and rating

In [52]:
```
Wall=requests.get("https://www.icc-cricket.com/rankings/womens/player-rankings/oc
Wall
```

Out[52]:
```
<Response [200]>
```

In [53]:
```
soup5c=BeautifulSoup(Wall.content)
soup5c
```

Out[53]:
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta content="Live Cricket Scores &amp;amp; News International Cricket Council"
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, hig
hlights, commentary, rankings, videos and fixtures from the International Cri
cket Council." name="description"/>
<meta content="@icc" property="twitter:site"/>
<meta content="Official ICC Cricket website - live matches, scores, news, hig
hlights, commentary, rankings, videos and fixtures from the International Cri
cket Council." name="twitter:description"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-t
humbnail.jpg" name="twitter:image"/>
<meta content="Live Cricket Scores &amp;amp; News International Cricket Council"
property="og:title"/>
<meta content="https://www.icc-cricket.com/resources/ver/i/elements/default-t
```

In [54]:
```
wa=[]
for i in soup5c.find_all("div",class_="rankings-block__banner--name-large"):
    wa.append(i.text)
for i in soup5c.find_all("td",class_="table-body__cell rankings-table__name name"
    wa.append(i.find('a').text)
wa
```

Out[54]:
```
['Marizanne Kapp',
 'Natalie Sciver',
 'Ellyse Perry',
 'Stafanie Taylor',
 'Deepti Sharma',
 'Ashleigh Gardner',
 'Dane van Niekerk',
 'Hayley Matthews',
 'Jess Jonassen',
 'Katherine Brunt',
 'Jhulan Goswami']
```

In [55]:
```
wat=[]
for i in soup5c.find_all("div",class_="rankings-block__banner--nationality"):
    wat.append(i.text.replace('\n\n','').replace('\n',''))
for i in soup5c.find_all("span",class_="table-body__logo-text")[0:10]:
    wat.append(i.text)
wat
```

Out[55]:
```
['SA                    ',
 'ENG',
 'AUS',
 'WI',
 'IND',
 'AUS',
 'SA',
 'WI',
 'AUS',
 'ENG',
 'IND']
```

In [56]:
```
war=[]
for i in soup5c.find_all("div",class_="rankings-block__banner--rating"):
    war.append(i.text)
for i in soup5c.find_all("td",class_="table-body__cell rating")[0:10]:
    war.append(i.text)
war
```

Out[56]:
```
['384', '372', '365', '319', '299', '275', '274', '272', '272', '272', '251']
```

6) Write a python program to scrape details of all the posts from coreyms.com. Scrape the heading, date, content and the code for the video from the link for the youtube video from the post.

In [57]:
```
post=requests.get('https://coreyms.com/')
post
```

Out[57]:
```
<Response [200]>
```

In [58]:
```
soup6=BeautifulSoup(post.content,'html.parser')
soup6
```

Out[58]:
```
<!DOCTYPE html>

<html lang="en-US">
<head>
<meta charset="utf-8"/>
<meta content="width=device-width, initial-scale=1" name="viewport"/>
<!-- This site is optimized with the Yoast SEO plugin v15.4 - https://yoast.c
om/wordpress/plugins/seo/ (https://yoast.com/wordpress/plugins/seo/) -->
<title>CoreyMS - Development, Design, DIY, and more</title>
<meta content="Development, Design, DIY, and more" name="description"/>
<meta content="index, follow, max-snippet:-1, max-image-preview:large, max-vi
deo-preview:-1" name="robots"/>
<link href="https://coreyms.com/" rel="canonical"/>
<link href="https://coreyms.com/page/2" rel="next"/>
<meta content="en_US" property="og:locale"/>
<meta content="website" property="og:type"/>
<meta content="CoreyMS - Development, Design, DIY, and more" property="og:tit
le"/>
<meta content="Development, Design, DIY, and more" property="og:description"/
```

In [59]:
```
heading=[]
for i in soup6.find_all("h2",class_='entry-title'):
    heading.append(i.text)
heading
```

Out[59]:
```
['Python Tutorial: Zip Files – Creating and Extracting Zip Archives',
 'Python Data Science Tutorial: Analyzing the 2019 Stack Overflow Developer Sur
vey',
 'Python Multiprocessing Tutorial: Run Code in Parallel Using the Multiprocessi
ng Module',
 'Python Threading Tutorial: Run Code Concurrently Using the Threading Module',
 'Update (2019-09-03)',
 'Python Quick Tip: The Difference Between “==” and “is” (Equality vs Identit
y)',
 'Python Tutorial: Calling External Commands Using the Subprocess Module',
 'Visual Studio Code (Windows) – Setting up a Python Development Environment an
d Complete Overview',
 'Visual Studio Code (Mac) – Setting up a Python Development Environment and Co
mplete Overview',
 'Clarifying the Issues with Mutable Default Arguments']
```

In [60]:
```
date=[]
for i in soup6.find_all("time",class_="entry-time"):
    date.append(i.text)
date
```

Out[60]:
```
['November 19, 2019',
 'October 17, 2019',
 'September 21, 2019',
 'September 12, 2019',
 'September 3, 2019',
 'August 6, 2019',
 'July 24, 2019',
 'May 1, 2019',
 'May 1, 2019',
 'April 24, 2019']
```

In [61]:
```
content=[]
for i in soup6.find_all("div",class_="entry-content"):
    content.append(i.find('p').text)
content
```

Out[61]:
```
['In this video, we will be learning how to create and extract zip archives. We will start by using the zipfile module, and then we will see how to do thi s using the shutil module. We will learn how to do this with single files and directories, as well as learning how to use gzip as well. Let’s get starte d…',
 'In this Python Programming video, we will be learning how to download and a nalyze real-world data from the 2019 Stack Overflow Developer Survey. This is terrific practice for anyone getting into the data science field. We will lea rn different ways to analyze this data and also some best practices. Let’s ge t started…',
 'In this Python Programming video, we will be learning how to run code in pa rallel using the multiprocessing module. We will also look at how to process multiple high-resolution images at the same time using a ProcessPoolExecutor from the concurrent.futures module. Let’s get started…',
 'In this Python Programming video, we will be learning how to run threads co ncurrently using the threading module. We will also look at how to download m ultiple high-resolution images online using a ThreadPoolExecutor from the con current.futures module. Let’s get started…',
 'Hey everyone. I wanted to give you an update on my videos. I will be releas ing videos on threading and multiprocessing within the next week. Thanks so m uch for your patience. I currently have a temporary recording studio setup at my Airbnb that will allow me to record and edit the threading/multiprocessing videos. I am going to be moving into my new house in 10 days and once I have my recording studio setup then you can expect much faster video releases. I r eally appreciate how patient everyone has been while I go through this move, especially those of you who are contributing monthly through YouTube ',
 'In this Python Programming Tutorial, we will be learning the difference bet ween using “==” and the “is” keyword when doing comparisons. The difference b etween these is that “==” checks to see if values are equal, and the “is” key word checks their identity, which means it’s going to check if the values are identical in terms of being the same object in memory. We’ll learn more in th e video. Let’s get started…',
 'In this Python Programming Tutorial, we will be learning how to run externa l commands using the subprocess module from the standard library. We will lea rn how to run commands, capture the output, handle errors, and also how to pi pe output into other commands. Let’s get started…',
 'In this Python Programming Tutorial, we will be learning how to set up a Py thon development environment in VSCode on Windows. VSCode is a very nice free editor for writing Python applications and many developers are now switching over to this editor. In this video, we will learn how to install VSCode, get the Python extension installed, how to change Python interpreters, create vir tual environments, format/lint our code, how to use Git within VSCode, how to debug our programs, how unit testing works, and more. We have a lot to cover, so let’s go ahead and get started…',
 'In this Python Programming Tutorial, we will be learning how to set up a Py thon development environment in VSCode on MacOS. VSCode is a very nice free e ditor for writing Python applications and many developers are now switching o ver to this editor. In this video, we will learn how to install VSCode, get t he Python extension installed, how to change Python interpreters, create virt ual environments, format/lint our code, how to use Git within VSCode, how to debug our programs, how unit testing works, and more. We have a lot to cover,
```

```
so let's go ahead and get started…',
 'In this Python Programming Tutorial, we will be clarifying the issues with
mutable default arguments. We discussed this in my last video titled "5 Commo
n Python Mistakes and How to Fix Them", but I received many comments from peo
ple who were still confused. So we will be doing a deeper dive to explain exa
ctly what is going on here. Let's get started…']
```

In [62]:
```
links_list=[]
for i in soup6.find_all("iframe",class_="youtube-player"):
        links_list.append(i.attrs['src'])
links_list
```

Out[62]:
```
['https://www.youtube.com/embed/z0gguhEmWiY?version=3&rel=1&showsearch=0&showin
fo=1&iv_load_policy=1&fs=1&hl=en-US&autohide=2&wmode=transparent',
 'https://www.youtube.com/embed/_P7X8tMplsw?version=3&rel=1&showsearch=0&showin
fo=1&iv_load_policy=1&fs=1&hl=en-US&autohide=2&wmode=transparent',
 'https://www.youtube.com/embed/fKl2JW_qrso?version=3&rel=1&showsearch=0&showin
fo=1&iv_load_policy=1&fs=1&hl=en-US&autohide=2&wmode=transparent',
 'https://www.youtube.com/embed/IEEhzQoKtQU?version=3&rel=1&showsearch=0&showin
fo=1&iv_load_policy=1&fs=1&hl=en-US&autohide=2&wmode=transparent',
 'https://www.youtube.com/embed/mO_dS3rXDIs?version=3&rel=1&showsearch=0&showin
fo=1&iv_load_policy=1&fs=1&hl=en-US&autohide=2&wmode=transparent',
 'https://www.youtube.com/embed/2Fp1N6dof0Y?version=3&rel=1&showsearch=0&showin
fo=1&iv_load_policy=1&fs=1&hl=en-US&autohide=2&wmode=transparent',
 'https://www.youtube.com/embed/-nh9rCzPJ20?version=3&rel=1&showsearch=0&showin
fo=1&iv_load_policy=1&fs=1&hl=en-US&autohide=2&wmode=transparent',
 'https://www.youtube.com/embed/06I63_p-2A4?version=3&rel=1&showsearch=0&showin
fo=1&iv_load_policy=1&fs=1&hl=en-US&autohide=2&wmode=transparent',
 'https://www.youtube.com/embed/_JGmemuINww?version=3&rel=1&showsearch=0&showin
fo=1&iv_load_policy=1&fs=1&hl=en-US&autohide=2&wmode=transparent']
```

7) Write a python program to scrape house details from mentioned URL. It should include house title, location, area, EMI and price from nobroker.in.

In [63]:
```
house=requests.get("https://www.nobroker.in/property/sale/bangalore/Electronic%26
house
```

Out[63]:
```
<Response [200]>
```

In [64]:
```
soup7=BeautifulSoup(house.content)
soup7
```

Out[64]:
```
<!DOCTYPE html>
<html lang="en"><head>
<meta content="794951570520699" property="fb:pages"/>
<link href="https://www.nobroker.in" rel="canonical"/>
<link href="//www.googletagmanager.com" rel="dns-prefetch"/>
<link href="//www.google-analytics.com" rel="dns-prefetch"/>
<link href="//assets.nobroker.in" rel="dns-prefetch"/>
<link href="//images.nobroker.in" rel="dns-prefetch"/>
<link href="//assets.nobroker.in/static/img/favicon.png" id="favicon" rel="sh
ortcut icon"/>
<link href="https://images.nobroker.in/static/img/fav64.png" rel="apple-touch
-icon"/>
<meta charset="utf-8"/><meta charset="utf-8"/>
<meta content="app-id=com.nobroker.app&amp;referrer=utm_source%3Dnobroker%26u
tm_medium%3DmobileWeb" name="google-play-app"/>
<meta content="app-id=1200507100, app-argument=nobrokerapp://" name="apple-it
unes-app"/>
<meta content="#fd3752" name="theme-color"/>
<meta content="4 BHK  flats for sale in Electronic City, 4 BHK  apartments fo
```

In [65]:
```
house_title=[]
for i in soup7.find_all("h2",class_="heading-6 font-semi-bold nb__25Cl7"):
    house_title.append(i.find('span').text)
house_title
```

Out[65]:
```
['4 BHK In Independent House  For Sale  In Hebbagodi',
 '4 BHK In Independent House  For Sale  In Electronic City',
 '4 BHK In Independent House  For Sale  In Electronics City Phase 1, Electronic
City',
 '4 BHK Flat  For Sale  In Electronic City',
 '4 BHK Apartment  For Sale  In Nisarga Residency In Electronic City Phase Ii',
 '4 BHK Flat  For Sale  In Sobha Silicon Oasis  In Hosa Road',
 '4 BHK For Sale  In Daadys Garden In Electronic City',
 '4 BHK Flat  For Sale  In , Electronic City',
 '4 BHK Flat  For Sale  In Hosa Road, Parappana Agrahara',
 '4 BHK In Independent House  For Sale  In Electronic City',
 '4 BHK In Independent House  For Sale  In Electronic City',
 '4 BHK Apartment  For Sale  In Gopalan Gardenia In Electronic City',
 '4 BHK For Sale  In Gpr Royale In Gpr Royale',
 '4 BHK Flat  For Sale  In Heena Enclave In Electronic City',
 '4 BHK For Sale  In Deccan Palms Park In Electronic City',
 '4 BHK In Independent House  For Sale  In Electronic City Phase Ii']
```

In [66]:
```
location=[]
for i in soup7.find_all("div",class_="nb__1EwQz"):
    location.append(i.text)
location
```

Out[66]:
```
['Independent House, Bangalore - Hosur Road, Near  National Public School',
 'Independent House, Shanthi Pura, Electronic City Phase II,Near Sallapuriyamma Temple Shanthipura',
 'Independent House, brand factory',
 'Standalone Building, YOUNG LIFE PG FOR LADIES, Konappana Agrahara, Electronic City, Bengaluru, Karnataka, India',
 'Nisarga Residency\xa0 Near Thali Resturant, Ananth Nagar, Electronic City Phase II, Bangalore, Karnataka, INDIA.',
 'Sobha Silicon Oasis Naganathapura, Rayasandra Bengaluru, Karnataka 560100 India',
 'Daadys Garden\xa0 Kammasandra Rd, Kammasandra, Electronic City, Bengaluru, Karnataka 560100, India',
 'Standalone Building, 16th Cross Road Neeladri Nagar, near by brand factory',
 'Standalone Building, 11th cross.anjanadri lay out',
 'Independent House, surya nagar face 1',
 'Independent House, Hosur Rd,Near Infosys Limited',
 'Gopalan Gardenia\xa0 Gopalan gardenia, Veerasandra Main Rd, Veer Sandra, Electronic City, Bengaluru, Karnataka 560100, India',
 '6th Cross',
 ' Neeladri Nagar,Near Pioneer Sun Blossom',
 'Deccan Palms Park\xa0 Deccan Palms Villas, Deccan Palms Road, Shree Ananth Nagar Layout, Glass Factory Layout, Electronic City, Bengaluru, Karnataka 560100, India',
 'Independent House, Industrial Area Near Tech Mahindra']
```

In [67]:
```
emi=[]
for i in soup7.find_all("div",class_="font-semi-bold heading-6",id="roomType"):
    emi.append(i.text)
emi
```

Out[67]:
```
['₹77,374/Month',
 '₹31,522/Month',
 '₹39,546/Month',
 '₹28,657/Month',
 '₹45,851/Month',
 '₹91,703/Month',
 '₹85,971/Month',
 '₹39,546/Month',
 '₹71,643/Month',
 '₹1.43 Lacs/Month',
 '₹42,985/Month',
 '₹68,777/Month',
 '₹85,971/Month',
 '₹71,643/Month',
 '₹85,971/Month',
 '₹57,314/Month']
```

In [68]:
```
list=[]
sq=[]
pm=[]
price=[]
for i in soup7.find_all("div",class_="font-semi-bold heading-6"):
    list.append(i.text)
for i in range(0,len(list)-1,3):
    sq.append(list[i])
    pm.append(list[i+1])
    price.append(list[i+2])
price
```

Out[68]:
```
['₹1.35 Crores',
 '₹55 Lacs',
 '₹69 Lacs',
 '₹50 Lacs',
 '₹80 Lacs',
 '₹1.6 Crores',
 '₹1.5 Crores',
 '₹69 Lacs',
 '₹1.25 Crores',
 '₹2.5 Crores',
 '₹75 Lacs',
 '₹1.2 Crores',
 '₹1.5 Crores',
 '₹1.25 Crores',
 '₹1.5 Crores',
 '₹1 Crore']
```

8) Write a python program to scrape mentioned details from dineout.co.in :

i) Restaurant name

ii) Cuisine

iii) Location

iv) Ratings

v) Image URL

In [69]:
```
hotel=requests.get("https://www.dineout.co.in/delhi-restaurants/buffet-special")
hotel
```

Out[69]: <Response [200]>

In [70]:
```
soup8=BeautifulSoup(hotel.content)
soup8
```

Out[70]:
```
<!DOCTYPE html>
<html lang="en"><head><meta charset="utf-8"/><meta content="IE=edge" http-equ
iv="X-UA-Compatible"/><meta content="width=device-width, initial-scale=1.0, m
aximum-scale=1.0, user-scalable=no" name="viewport"/><link href="/manifest.js
on" rel="manifest"/><style type="text/css">
            @font-face {
                font-family: 'dineicon';
                src:  url('/fonts/dineicon.eot');
                src:  url('/fonts/dineicon.eot#iefix') format('embedded-opent
ype'),
                url('/fonts/dineicon.ttf') format('truetype'),
                url('/fonts/dineicon.woff') format('woff'),
                url('/fonts/dineicon.svg#dineicon') format('svg');
                font-weight: normal;
                                font-style: normal;
                                font-display: swap;
            }
            .hide {
                display: none !important;
            }
```

In [71]:
```
rest=[]
for i in soup8.find_all('a',class_="restnt-name ellipsis"):
    rest.append(i.text)
rest
```

Out[71]:
```
['Castle Barbeque',
 'Jungle Jamboree',
 'Castle Barbeque',
 'Cafe Knosh',
 'The Barbeque Company',
 'India Grill',
 'Delhi Barbeque',
 'The Monarch - Bar Be Que Village',
 'World Cafe',
 'Indian Grill Room',
 'Mad 4 Bar B Que',
 'Barbeque 29',
 'Glasshouse']
```

In [72]:
```
cui=[]
for i in soup8.find_all('span',class_="double-line-ellipsis"):
    cui.append(i.find('a').text)
cui
```

Out[72]:
```
['Chinese',
 'North Indian',
 'North Indian',
 'Multi-Cuisine',
 'Barbecue',
 'North Indian',
 'Barbecue',
 'North Indian',
 'North Indian',
 'North Indian',
 'North Indian',
 'North Indian',
 'Multi-Cuisine']
```

In [73]:
```
loc=[]
for i in soup8.find_all('div',class_="restnt-loc ellipsis"):
    loc.append(i.find('a').text)
loc
```

Out[73]:
```
['Connaught Place',
 '3CS Mall,',
 'Pacific Mall,',
 'The Leela Ambience Convention Hotel,',
 'Gardens Galleria,',
 'Hilton Garden Inn,',
 'Taurus Sarovar Portico,',
 'Indirapuram Habitat Centre,',
 'Vibe by The Lalit Traveller,',
 'Suncity Business Tower,',
 'Sector 29',
 'NIT',
 'DoubleTree By Hilton Gurugram Baani Square,']
```

In [74]:
```
Rrate=[]
for i in soup8.find_all('div',class_="restnt-rating rating-4"):
    Rrate.append(i.text)
Rrate
```

Out[74]:
```
['3.5',
 '3.9',
 '4',
 '4.3',
 '4.1',
 '3.9',
 '3.6',
 '3.9',
 '4.2',
 '4.3',
 '3.9',
 '4.2',
 '4.1']
```

In [75]:
```
img=[]
for i in soup8.find_all('img',class_="no-img"):
    img.append(i.attrs['data-src'])
img
```

Out[75]:
```
['https://im1.dineout.co.in/images/uploads/restaurant/sharpen/8/k/b/p86792-1606
2953735fbe1f4d3fb7e.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/a/k/p59633-1604
6474755fa4fa33c0e92.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/j/o/p38113-1595
9192065f1fcb666130c.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/4/j/v/p406-163401
663361651d79326d0.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/7/q/d/p79307-1605
1787075fad15532bd7c.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/2/v/t/p2687-14824
77169585cce712b90f.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/v/f/p52501-1600
6856545f68865616659.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/n/o/p34822-1559
9107305cfa594a13c24.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/1/p/y/p12366-1466
935020576fa6ecdc359.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/y/y/p549-151437
67525a438e30b3e19.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/4/j/e/p43488-1529
5778165b2b8158ceeef.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/w/r/p58842-1562
4171585d209806d9143.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/9/m/f/p9875-16057
921085fb6716cc44f8.jpg?tr=tr:n-medium']
```

9) Write a python program to scrape weather details for last 24 hours from Tutiempo.net :

In [73]:
```
img=[]
for i in soup8.find_all('img',class_="no-img"):
    img.append(i.attrs['data-src'])
img
```

Out[73]:
```
['https://im1.dineout.co.in/images/uploads/restaurant/sharpen/8/k/b/p86792-1606
2953735fbe1f4d3fb7e.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/a/k/p59633-1604
6474755fa4fa33c0e92.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/j/o/p38113-1595
9192065f1fcb666130c.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/4/j/v/p406-163401
663361651d79326d0.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/7/q/d/p79307-1605
1787075fad15532bd7c.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/2/v/t/p2687-14824
77169585cce712b90f.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/v/f/p52501-1600
6856545f68865616659.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/n/o/p34822-1559
9107305cfa594a13c24.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/1/p/y/p12366-1466
935020576fa6ecdc359.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/y/y/p549-151437
67525a438e30b3e19.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/4/j/e/p43488-1529
5778165b2b8158ceeef.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/w/r/p58842-1562
4171585d209806d9143.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/9/m/f/p9875-16057
921085fb6716cc44f8.jpg?tr=tr:n-medium']
```

9) Write a python program to scrape weather details for last 24 hours from Tutiempo.net :

i) Hour

ii) Temperature

iii) Wind

iv) Weather condition

v) Humidity

vi) Pressure

In [74]:
```
w=requests.get('https://en.tutiempo.net/delhi.html?data=last-24-hours')
w
```

Out[74]:
```
<Response [200]>
```

In [75]:
```
soup9=BeautifulSoup(w.content)
soup9
```

Out[75]:
```
<!DOCTYPE html>
<html class="pretty-scrollbar" lang="en"><head><meta content="text/html; char
set=utf-8" http-equiv="content-type"/><meta content="width=device-width, init
ial-scale=1.0, maximum-scale=1.0, user-scalable=no" name="viewport"/><meta co
ntent="Tutiempo Network, S.L." name="author"/><meta content="global" name="di
stribution"/><meta content="app-id=526005265" name="apple-itunes-app"/><meta
content="yes" name="mobile-web-app-capable"/><meta content="yes" name="apple-
mobile-web-app-capable"/><meta content="black" name="apple-mobile-web-app-sta
tus-bar-style"/><link href="/favicon.ico" rel="shortcut icon" type="image/ic
o"/><link href="/Browser/apple-icon-57x57.png" rel="apple-touch-icon" sizes
="57x57"/><link href="/Browser/apple-icon-60x60.png" rel="apple-touch-icon" s
izes="60x60"/><link href="/Browser/apple-icon-72x72.png" rel="apple-touch-ico
n" sizes="72x72"/><link href="/Browser/apple-icon-76x76.png" rel="apple-touch
-icon" sizes="76x76"/><link href="/Browser/apple-icon-114x114.png" rel="apple
-touch-icon" sizes="114x114"/><link href="/Browser/apple-icon-120x120.png" re
l="apple-touch-icon" sizes="120x120"/><link href="/Browser/apple-icon-144x14
4.png" rel="apple-touch-icon" sizes="144x144"/><link href="/Browser/apple-ico
n-152x152.png" rel="apple-touch-icon" sizes="152x152"/><link href="/Browser/a
pple-icon-180x180.png" rel="apple-touch-icon" sizes="180x180"/><link href="/B
```

In [76]:
```
hour=[]
for i in soup9.find_all('div',class_="last24 thh"):
    hour.append(i.find('td').text)
hour
```

Out[76]:
```
['09:30']
```

In [77]:
```
temp=[]
for i in soup9.find_all('td',class_="t Temp"):
    temp.append(i.text)
temp
```

Out[77]:
```
['17°C',
 '16°C',
 '16°C',
 '15°C',
 '15°C',
 '15°C',
 '15°C',
 '15°C',
 '15°C',
 '15°C',
 '15°C',
 '16°C',
 '16°C',
 '16°C',
 '16°C',
 '16°C',
 '16°C',
 '16°C',
 '16°C',
 '17°C',
 '17°C',
```

In [78]:
```
wind=[]
for i in soup9.find_all('td',class_="wind"):
    wind.append(i.text)
wind
```

Out[78]:
```
['7 km/h',
 '7 km/h',
 '7 km/h',
 'Calm',
 '6 km/h',
 '9 km/h',
 '6 km/h',
 '7 km/h',
 '6 km/h',
 '6 km/h',
 '7 km/h',
 '6 km/h',
 '6 km/h',
 '6 km/h',
 '6 km/h',
 '6 km/h',
 '6 km/h',
 '6 km/h',
 'Calm',
 'Calm',
```

In [79]:
```
wc=[]
for i in soup9.find_all("span",class_="thhip ico i0530 u3012n"):
    wc.append(i.text)
wc
```

Out[79]:
```
['Widespread Fog']
```

In [80]:
```
hum=[]
for i in soup9.find_all("td",class_="hr"):
    hum.append(i.text)
hum
```

Out[80]:
```
['82%',
 '88%',
 '88%',
 '88%',
 '88%',
 '88%',
 '88%',
 '88%',
 '88%',
 '88%',
 '82%',
 '82%',
 '82%',
 '82%',
 '82%',
 '82%',
 '82%',
 '82%',
 '82%',
 '82%',
```

In [81]:
```
pre=[]
for i in soup9.find_all("td",class_="prob"):
    pre.append(i.text)
pre
```

Out[81]:
```
['1020 hPa',
 '1019 hPa',
 '1019 hPa',
 '1018 hPa',
 '1018 hPa',
 '1018 hPa',
 '1017 hPa',
 '1017 hPa',
 '1016 hPa',
 '1016 hPa',
 '1016 hPa',
 '1016 hPa',
 '1016 hPa',
 '1016 hPa',
 '1017 hPa',
 '1017 hPa',
 '1017 hPa',
 '1017 hPa',
 '1017 hPa',
```

10) Write a python program to scrape monument name, monument description, image URL about top 10 monuments from puredestinations.co.uk.

In [82]:
```
place=requests.get("https://www.puredestinations.co.uk/top-10-famous-monuments-tc
place
```

Out[82]:
```
<Response [200]>
```

In [83]:
```
soup10=BeautifulSoup(place.content,'html.parser')
soup10
```

Out[83]:
```
<!DOCTYPE html>

<!--[if IE 7]>
<html class="ie ie7" lang="en">
<![endif]-->
<!--[if IE 8]>
<html class="ie ie8" lang="en">
<![endif]-->
<!--[if IE 9]>
<html class="ie ie9" lang="en">
<![endif]-->
<!--[if !(IE 7) | !(IE 8) | !(IE 9) ]><!-->
<html data-useragent="Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; Tri
dent/6.0)" lang="en">
<!--<![endif]-->
<head>
<title>Top 10 Famous Monuments to Visit In India | Pure Destinations</title>
<meta content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user
-scalable=no" name="viewport"/>
```

In [84]:
```
mname=[]
for i in soup10.find_all("div",class_="blog--single__content column--3-4 u-spacir
    for j in i.find_all('strong'):
        mname.append(j.text)
mname
```

Out[84]:
```
['Taj Mahal, Agra',
 'Golden Temple (Harmandir Sahib), Amritsar ',
 'Meenakshi Temple, Madurai',
 'Mysore Palace, Mysore',
 'Gateway of India, Mumbai',
 'Red Fort, New Delhi',
 'Hawa Mahal, Jaipur',
 'Qutub Minar, New Delhi',
 'Sanchi Stupa, Sanchi',
 'Charminar, Hyderabad',
 'Things to know before planning your trip to India']
```

In [85]:
```
mdesc=[]
for i in soup10.find_all("div",class_="blog--single__content column--3-4 u-spacir
    for j in i.find_all('p'):
        mdesc.append(j.text)
mdesc
```

Out[85]:
```
['Rich in culture and diversity, India is home to some of the finest historic
al monuments in the world. Most recognised by the UNESCO World Heritage Site,
the famous Indian monuments include the beautiful Taj Mahal, the sacred Golde
n Temple and the cultural site, Hawa Mahal. Discover and experience the magni
ficent insights into India’s rich heritage and ancient architecture. Read on
for our list of the top must see historical monuments in India below.',
 'Taj Mahal, Agra',
 'Enlisted in the Seven Wonders of the World, The Taj Mahal is one of the mos
t beautiful and famous buildings located in the city of Agra. This white marb
le monument was built by a Mughal Emperor called Shahajahan in memory of his
beloved wife. Due to its amazing architecture and the history behind it, this
world heritage site has become very popular to visit by all travellers and ro
mantics from all over the world.',
 '',
 'Golden Temple (Harmandir Sahib), Amritsar ',
 'The holiest shrine and pilgrimage place located in Amritsar is The Golden T
emple known as the Harmandir Sahib. This is the most famous and sacred Sikh G
urdwara in Punjab, India, adorned with rich history and gold gilded exterior.
If you are interested in culture and history, be sure to visit this popular a
```

In [86]:
```
imgurl=[]
for i in soup10.find_all("div",class_="blog--single__content column--3-4 u-spacir
    for j in i.find_all('img'):
        imgurl.append(j.attrs['src'])
imgurl
```

Out[86]:
```
['',
 'http://www.puredestinations.co.uk/wp-content/uploads/2016/11/TAJ-MAHAL-PD-BLO
G.jpg',
 '',
 'http://www.puredestinations.co.uk/wp-content/uploads/2016/11/Golden-Temple-PD
-BLOG.jpg',
 '',
 'http://www.puredestinations.co.uk/wp-content/uploads/2016/11/meenakshi-temple
-PD-BLOG.jpg',
 '',
 'http://www.puredestinations.co.uk/wp-content/uploads/2016/11/Mysore-palace-PD
-BLOG.jpg',
 '',
 'http://www.puredestinations.co.uk/wp-content/uploads/2016/11/gateway-to-india
-PD-BLOG.jpg',
 '',
 'http://www.puredestinations.co.uk/wp-content/uploads/2016/11/red-fort-PD-BLO
G.jpg',
 '',
 'http://www.puredestinations.co.uk/wp-content/uploads/2016/11/HAWA-MAHAL-PD-BL
OG.jpg',
 '',
 'http://www.puredestinations.co.uk/wp-content/uploads/2016/11/qutub-minar-PD-B
LOG.jpg',
 '',
 'http://www.puredestinations.co.uk/wp-content/uploads/2016/11/Sanchi-Stupi-PD-
BLOG.jpg',
 '',
 'http://www.puredestinations.co.uk/wp-content/uploads/2016/11/Charminar-PD-BLO
G-1.jpg']
```

In [ ]: