Web Scraping

In [1]:
```
# Importing Important Libraries...
import pandas as pd
from bs4 import BeautifulSoup as bs4
import requests
```

___Q1. Write a python program to display all the header tags from‘en.wikipedia.org/wiki/Main_Page’.

In [2]:
```
# Writing a python program to display all the header tags from‘en.wikipedia.org/wiki/Main_Page’.

page = requests.get('https://en.wikipedia.org/wiki/Main_Page')

# getting a page conten
soup = bs4 (page.content) #bs4 is used as BeautifulSoup

header_tags = []  # Empty List

#scraping the Headers
for header in soup.find_all(['h1','h2','h3','h4','h5','h6']):
    header_tags.append(header.name+" "+header.text.strip())

#printing final result
header_tags
```

Out[2]:
```
['h1 Main Page',
 "h2 From today's featured article",
 'h2 Did you know\xa0...',
 'h2 In the news',
 'h2 On this day',
 "h2 Today's featured picture",
 'h2 Other areas of Wikipedia',
 "h2 Wikipedia's sister projects",
 'h2 Wikipedia languages',
 'h2 Navigation menu',
 'h3 Personal tools',
 'h3 Namespaces',
 'h3 Variants\nexpanded\ncollapsed',
 'h3 Views',
 'h3 More\nexpanded\ncollapsed',
 'h3 Search',
 'h3 Navigation',
 'h3 Contribute',
 'h3 Tools',
 'h3 Print/export',
 'h3 In other projects',
 'h3 Languages']
```

___Q2. Write a python program to display IMDB’s Top rated 100 movies’ data (i.e. Name, IMDB rating, Year of release) and make data frame

In [3]:
```
# Writing a paython program to scrap top 100 IMDB Movies name, Year of Release and IMDB Rating

#requesting Url access
url = "https://www.imdb.com/list/ls091520106/"
page1 = requests.get(url)

#see page content
soup1 = bs4 (page1.content)

#scraping name
name=soup1.find_all('h3', class_='lister-item-header')

# scraping Movies name
movies_name = []

for i in name:
    for j in i.find_all('a'):
        movies_name.append(j.text)

#scraping Year

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

```
#Scraping IMDB rating

rating = soup1.find_all("div", class_="ipl-rating-star small")

IMDB_rating = []

for i in rating:
      IMDB_rating.append(float(i.text))

#preparing data frame for Movie name, Year of release and IMDB ratings
IMDB_top_100= pd.DataFrame({})
IMDB_top_100['movies_name']= movies_name
IMDB_top_100['year_of_release']= year_of_release
IMDB_top_100['IMDB_rating']= IMDB_rating

# printing the final result
IMDB_top_100
```

Out[3]:

	movies_name	year_of_release	IMDB_rating
0	The Shawshank Redemption	1994	9.3
1	The Godfather	1972	9.2
2	The Godfather: Part II	1974	9.0
3	The Dark Knight	2008	9.0
4	12 Angry Men	1957	9.0
...	...	...	...
95	North by Northwest	1959	8.3
96	A Clockwork Orange	1971	8.3
97	Snatch	2000	8.3
98	Le fabuleux destin d'Amélie Poulain	2001	8.3
99	The Kid	1921	8.3

100 rows × 3 columns

___Q3 Write a python program to display IMDB's Top rated 100 Indian movies' data (i.e. Name, IMDB rating, Year of release) and make data frame.

In [4]:

```
# Writing a paython program to scrap top 100 Indian Movies Name, Year of Release and IMDB Rating

#requesting url access
url = "https://www.imdb.com/list/ls009997493/"
page2 = requests.get(url)

#see page content
soup2 = bs4 (page2.content)

#scraping name
name=soup2.find_all('h3', class_='lister-item-header')

# scraping Movies name
movies_name = []

for i in name:
    for j in i.find_all('a'):
        movies_name.append(j.text)

#scraping Year

year= soup2. find_all("span",class_= "lister-item-year text-muted unbold")
year_of_release = []
for k in year:
    a=k.text.replace('(','')
    year_of_release.append(a.replace(')',''))

#Scraping IMDB rating

rating = soup2.find_all("div", class_="ipl-rating-star small")

IMDB_rating = []

for i in rating:
      IMDB_rating.append(float(i.text))

#preparing data frame for Movie name, Year of release and IMDB ratings
Indain_top_100_movies= pd.DataFrame({})
Indain_top_100_movies['movies_name']= movies_name
```

```
Indain_top_100_movies['year_of_release']= year_of_release
Indain_top_100_movies['IMDB_rating']= IMDB_rating

# Printing final result
Indain_top_100_movies
```

Out[4]:

	movies_name	year_of_release	IMDB_rating
0	Rang De Basanti	2006	8.1
1	3 Idiots	2009	8.4
2	Taare Zameen Par	2007	8.4
3	Dil Chahta Hai	2001	8.1
4	Swades: We, the People	2004	8.2
...	...	...	...
95	Wake Up Sid	2009	7.6
96	Rangeela	1995	7.5
97	Shatranj Ke Khilari	1977	7.7
98	Pyaar Ka Punchnama	2011	7.6
99	Ek Hasina Thi	2004	7.5

100 rows × 3 columns

___Q4 i) Top 10 ODI teams in men’s cricket along with the records for matches, points and rating

In [5]:

```
# writitng paython program to scrap Top 10 ODI teams in men’s cricket along with the records for matches, points

# requesting url access
url="https://www.icc-cricket.com/rankings/mens/team-rankings/odi"

page3= requests.get(url)

# getting cotent from URl
soup3=bs4(page3.content)

# scraping Team name
team=soup3.find_all("span",class_='u-hide-phablet')
team_name = []
for i in team:
    team_name.append(i.text)
#forming empty List
matches = []
points = []
ratings = []
new_list= []

# using for fundction to get pull the data for list

for i in soup3.find_all("td",class_='ranking-block__banner--matches'):
    matches.append(i.text)
for i in soup3.find_all("td",class_='ranking-block__banner--points'):
    points.append(i.text)
for i in soup3.find_all('td',class_='rankings-block__banner--rating u-text-right'):
    ratings.append(i.text.replace('\n',' '))
for i in soup3.find_all("td",class_='table-body__cell u-center-text'):
    new_list.append(i.text)
for i in range(0,len(new_list)-1,2):
    matches.append(new_list[i])
    points.append(new_list[i+1])
for i in soup3.find_all('td',class_='table-body__cell u-text-right rating'):
    ratings.append(i.text)

#forming data frame

ODI_Team=pd.DataFrame({})
ODI_Team['Team_name'] = team_name[:10]
ODI_Team['Matches'] = matches[:10]
ODI_Team['Points'] = points[:10]
ODI_Team['Rating'] = ratings[:10]


#printing the final result
ODI_Team
```

Out[5]:

	Team_name	Matches	Points	Rating
0		32	3,793	121 ...

	New Zealand			
1	England	28	3,244	119
2	Australia	32	3,624	116
3	India	25	2,459	113
4	South Africa	27	2,524	98
5	Pakistan	30	2,740	93
6	Bangladesh	30	2,523	91
7	West Indies	32	2,657	84
8	Sri Lanka	17	1,054	83
9	Afghanistan	7	336	62

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

In [7]:
```
# writing a paython programm to scrap Top 10 ODI Batsman with their records and rating.

#send get request to the webpage server to get the source code of the page
url = "https://www.icc-cricket.com/rankings/mens/player-rankings/odi/batting"

#requesting permission for yrl access
page4 = requests.get(url)

# see content in page6
soup4 = bs4(page4.content)

players = [] #empty list
team_name = [] #empty list
rating = [] #empty list

for i in soup4.find_all("div",class_='rankings-block__banner--name-large'): # first place player name
    players.append(i.text)
for i in soup4.find_all("div",class_='rankings-block__banner--nationality'): # first place player team name
    team_name.append(i.text.replace("\n",""))
for i in soup4.find_all("div",class_='rankings-block__banner--rating'): # first place player rating
    rating.append(i.text)
for i in soup4.find_all("td",class_='table-body__cell rankings-table__name name'):# players name
    for j in i.find_all('a'):
        players.append(j.text)
for i in soup4.find_all("span",class_='table-body__logo-text'): # players team name
    team_name.append(i.text)
for i in soup4.find_all("td",class_='table-body__cell rating'): # players rating
    rating.append(i.text)
# Make data frame of top 10 ICC Batsmen
Batsmen=pd.DataFrame({})
Batsmen['Player']=players[:10]
Batsmen['Team']=team_name[:10]
Batsmen['Rating']=rating[:10]
Batsmen
```

Out[7]:

	Player	Team	Rating
0	Babar Azam	PAK	873
1	Virat Kohli	IND	844
2	Rohit Sharma	IND	813
3	Ross Taylor	NZ	801
4	Aaron Finch	AUS	779
5	Jonny Bairstow	ENG	775
6	David Warner	AUS	762
7	Shai Hope	WI	758
8	Kane Williamson	NZ	754
9	Quinton de Kock	SA	747

(iii) Top 10 ODI bowlers along with the records of their team andrating.

In [9]:
```
# writing a paython programm to scrap Top 10 ODI Bowler with their records and rating.

#send get request to the webpage server to get the source code of the page
url = "https://www.icc-cricket.com/rankings/mens/player-rankings/odi/bowling"

#requesting permission for yrl access
```

```
page5 = requests.get(url)

# page content
soup5 = bs4(page5.content)

#empty list
players = []
team_name = []
rating = []

for i in soup5.find_all("div",class_='rankings-block__banner--name-large'): # first place player name
    players.append(i.text)
for i in soup5.find_all("div",class_='rankings-block__banner--nationality'): # first place player team name
    team_name.append(i.text.replace("\n",""))
for i in soup5.find_all("div",class_='rankings-block__banner--rating'): # first place player rating
    rating.append(i.text)
for i in soup5.find_all("td",class_='table-body__cell rankings-table__name name'):# players name
    for j in i.find_all('a'):
        players.append(j.text)
for i in soup5.find_all("span",class_='table-body__logo-text'): # players team name
    team_name.append(i.text)
for i in soup5.find_all("td",class_='table-body__cell rating'): # players rating
    rating.append(i.text)
# Make data frame of top 10 ICC Batsmen
Bowler=pd.DataFrame({})
Bowler['Player']=players[:10]
Bowler['Team']=team_name[:10]
Bowler['Rating']=rating[:10]

#printing final result
Bowler
```

Out[9]:

	Player	Team	Rating
0	Trent Boult	NZ	737
1	Josh Hazlewood	AUS	709
2	Mujeeb Ur Rahman	AFG	708
3	Chris Woakes	ENG	700
4	Mehedi Hasan	BAN	692
5	Matt Henry	NZ	691
6	Jasprit Bumrah	IND	679
7	Mitchell Starc	AUS	652
8	Shakib Al Hasan	BAN	650
9	Kagiso Rabada	SA	646

___Q5 (i) Top 10 ODI teams in women's cricket along with the records for matches, points and rating.

In [11]:

```
# writitng paython program to scrap Top 10 ODI teams in women's cricket along with the records for matches, point

url="https://www.icc-cricket.com/rankings/womens/team-rankings/odi"

page6= requests.get(url)    #requesting permission from URL

# getting cotent from URl
soup6=bs4(page6.content)

# scraping Team name
team=soup6.find_all("span",class_='u-hide-phablet')
team_name = []
for i in team:
    team_name.append(i.text)

#forming empty List
matches = []
points = []
ratings = []
new_list= []

# scraping data

for i in soup6.find_all("td",class_='rankings-block__banner--matches'):
    matches.append(i.text)
for i in soup6.find_all("td",class_='rankings-block__banner--points'):
    points.append(i.text)
for i in soup6.find_all("td",class_='rankings-block__banner--rating u-text-right'):
    ratings.append(i.text.replace("\n",""))
for i in soup6.find_all("td",class_='table-body__cell u-center-text'):
    new_list.append(i.text,)
for i in range(0,len(new_list)-1,2):
```

```
    matches.append(new_list[i])
    points.append(new_list[i+1])
for i in soup6.find_all('td',class_='table-body__cell u-text-right rating'):
    ratings.append(i.text)

#forming data frame

ODI_women_Team=pd.DataFrame({})
ODI_women_Team['Team_name'] = team_name[:10]
ODI_women_Team['Matches'] = matches[:10]
ODI_women_Team['Points'] = points[:10]
ODI_women_Team['Rating'] = ratings[:10]

#printing the final result
ODI_women_Team
```

Out[11]:

	Team_name	Matches	Points	Rating
0	Australia	21	3,379	161 ...
1	England	25	2,983	119
2	South Africa	29	3,390	117
3	India	26	2,934	113
4	New Zealand	26	2,392	92
5	West Indies	22	1,872	85
6	Pakistan	20	1,496	75
7	Bangladesh	5	306	61
8	Sri Lanka	11	519	47
9	Ireland	2	25	13

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

In [12]:

```
#writitng paython program to scrap Top 10 ODI women along with the records of their team and rating(on the bases

url="https://www.icc-cricket.com/rankings/womens/player-rankings/odi/batting"

page7= requests.get(url)    #requesting permission from URL

# getting cotent from URl
soup7=bs4(page7.content)

#empty list
players = []
team_name = []
rating = []

# using for fundction to get the data for list

for i in soup7.find_all("div",class_='rankings-block__banner--name-large'):
    players.append(i.text)
for i in soup7.find_all("div",class_='ranking-block__banner--nationality'):
    team_name.append(i.text.replace("\n",""))
for i in soup7.find_all("div",class_='rankings-block__banner--rating'):
    rating.append(i.text)
for i in soup7.find_all("td",class_='table-body__cell rankings-table__name name'):
    for j in i.find_all('a'):
        players.append(j.text)
for i in soup7.find_all("span",class_='table-body__logo-text'):
    team_name.append(i.text)
for i in soup7.find_all("td",class_='table-body__cell rating'):
    rating.append(i.text)

#forming data frame

Women_players=pd.DataFrame({})
Women_players['Player'] = players[:10]
Women_players['Team'] = team_name[:10]
Women_players['Rating'] = rating[:10]

#printing the final result
Women_players
```

Out[12]:

	Player	Team	Rating
0	Lizelle Lee	AUS	761
1	Alyssa Healy	IND	750
2	Mithali Raj	ENG	738
3		NZ	728

	Tammy Beaumont		
4	Amy Satterthwaite	IND	717
5	Smriti Mandhana	AUS	710
6	Meg Lanning	AUS	699
7	Beth Mooney	ENG	690
8	Heather Knight	SA	674
9	Laura Wolvaardt	AUS	672

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

In [13]:
```
#writitng paython program to scrap Top 10 women's ODI all-rounder along with the records of their team and rating

# using url
url="https://www.icc-cricket.com/rankings/womens/player-rankings/odi/all-rounder"

page8= requests.get(url)    #requesting permission from URL

# getting cotent from URl
soup8=bs4(page8.content)

#empty list
players = []
team_name = []
rating = []

# using for fundction to get the data for list

for i in soup8.find_all("div",class_='rankings-block__banner--name-large'):
    players.append(i.text)
for i in soup8.find_all("div",class_='rankings-block__banner--nationality'):
    team_name.append(i.text.replace("\n",""))
for i in soup8.find_all("div",class_='rankings-block__banner--rating'):
    rating.append(i.text)
for i in soup8.find_all("td",class_='table-body__cell rankings-table__name name'):
    for j in i.find_all('a'):
        players.append(j.text)
for i in soup8.find_all("span",class_='table-body__logo-text'):
    team_name.append(i.text)
for i in soup8.find_all("td",class_='table-body__cell rating'):
    rating.append(i.text)

#forming data frame

Women_allrounder_player=pd.DataFrame({})
Women_allrounder_player['Player'] = players[:10]
Women_allrounder_player['Team'] = team_name[:10]
Women_allrounder_player['Rating'] = rating[:10]

#printing the final result
Women_allrounder_player
```

Out[13]:

	Player	Team	Rating
0	Marizanne Kapp	SA	384
1	Natalie Sciver	ENG	372
2	Ellyse Perry	AUS	365
3	Stafanie Taylor	WI	322
4	Deepti Sharma	IND	299
5	Ashleigh Gardner	AUS	275
6	Dane van Niekerk	SA	274
7	Jess Jonassen	AUS	272
8	Katherine Brunt	ENG	272
9	Jhulan Goswami	IND	251

Q6___Write a python program to scrape details of all the mobile phones under Rs. 20,000 listed on Amazon.in. The scraped data should include Product Name, Price, Image URL and Average Rating.

In [24]:
```
#writing paython program to scrape details of all the mobile phones under Rs. 20,000 listed on Amazon.in

url="https://www.amazon.in/s?k=best+mobile+under+20000&hvadid=72224314750395&hvbmt=be&hvdev=c&hvqmt=e&tag=msndesk
```

```
#getting url
page9 = requests.get(url)

#page content
soup9 = bs4(page9.content)

#Empty list
product_name = []
price = []
rating = []
img_url = []

for i in soup9.find_all("span", class_="a-size-medium a-color-base a-text-normal"):
    product_name.append(i.text)

for i in soup9.find_all("span", class_="a-price-whole"):
    price.append(i.text)
for i in soup9.find_all("span", class_="a-icon-alt"):
    rating.append(i.text)

for i in soup9.find_all("img", class_="s-image"):
    img_url.append(i.get("src"))

#Data Frame

mobile_phones=pd.DataFrame({})
mobile_phones['product_name']= product_name[:16]
mobile_phones['price']= price[:16]
mobile_phones['Rating']=rating[:16]
mobile_phones['img_url'] = img_url[:16]


#Printing the final result
mobile_phones
```

Out[24]:

	product_name	price	Rating	img_url
0	Samsung Galaxy M12 (Blue,4GB RAM, 64GB Storage...	9,499	4.1 out of 5 stars	https://m.media-amazon.com/images/I/71r69Y7BSe...
1	OPPO A74 5G (Fantastic Purple,6GB RAM,128GB St...	15,990	4.2 out of 5 stars	https://m.media-amazon.com/images/I/71geVdy6-O...
2	Redmi 9A (Nature Green, 2GB RAM, 32GB Storage)...	6,999	4.2 out of 5 stars	https://m.media-amazon.com/images/I/71sxlhYhKW...
3	Redmi 9 (Sky Blue, 4GB RAM, 64GB Storage) | 2....	20,500	4.3 out of 5 stars	https://images-eu.ssl-images-amazon.com/images...
4	OPPO A31 (Fantasy White, 6GB RAM, 128GB Storag...	20,000	4.4 out of 5 stars	https://m.media-amazon.com/images/I/51IkqCGcIQ...
5	Samsung Galaxy M31 (Ocean Blue, 8GB RAM, 128GB...	29,990	4.3 out of 5 stars	https://m.media-amazon.com/images/I/71iWvC-Sk1...
6	Redmi 10 Prime (Phantom Black 4GB RAM 64GB | H...	8,499	4.2 out of 5 stars	https://m.media-amazon.com/images/I/71FringFil...
7	Samsung Galaxy M32 5G (Slate Black, 6GB RAM, 1...	11,490	4.2 out of 5 stars	https://m.media-amazon.com/images/I/71A9Vo1Bat...
8	Redmi Note 10S (Frost White, 6GB RAM, 64GB Sto...	15,999	4.2 out of 5 stars	https://m.media-amazon.com/images/I/61CnyJ-IbM...
9	Samsung Galaxy M51 (Celestial Black, 6GB RAM, ...	16,999	3.6 out of 5 stars	https://m.media-amazon.com/images/I/71-Su4Wr0H...
10	Tecno Spark 7T(Jewel Blue, 4GB RAM, 64GB Stora...	13,999	3.8 out of 5 stars	https://m.media-amazon.com/images/I/817clKAKcq...
11	Samsung Galaxy M12 (Black,4GB RAM, 64GB Storag...	19,999	4.0 out of 5 stars	https://m.media-amazon.com/images/I/71QT7dSK4B...
12	OPPO A31 (Mystery Black, 6GB RAM, 128GB Storag...	8,499	4.4 out of 5 stars	https://m.media-amazon.com/images/I/81SJHWh-df...
13	realme narzo 30 (Racing Blue, 6GB RAM, 128GB S...	9,499	4.0 out of 5 stars	https://m.media-amazon.com/images/I/713AhSUtbH...
14	realme narzo 50A (Oxygen Blue, 4GB RAM + 128GB...	11,490	4.1 out of 5 stars	https://m.media-amazon.com/images/I/81aWyRY67S...
15	OPPO A74 5G (Fluid Black,6GB RAM,128GB Storage...	15,499	4.2 out of 5 stars	https://m.media-amazon.com/images/I/7162Y5fPdk...

Q6___Write a python program to scrape house details from mentioned url. It should include house title, location, area, emi and price
https://www.nobroker.in/property/sale/bangalore/Electronic%20City?type=BHK4&searchParam=W3sibGF0IjoxMi44N
DUyMTQ1LCJsb24iOjc3LjY2MDE2OTUsInBsYWNlSWQiOiJDaElKdy1GUWQ0cHNyanNSSGZkYXpnXzhYRW8
iLCJwbGFjZU5hbWUiOiJFbGVjdHJvbmljIENpdHkifV0=&propertyAge=0&radius=2.0"

In [26]:

```
# writing paython program to scrape house details which include house title, location,area, emi and price

url="https://www.nobroker.in/property/sale/bangalore/Electronic%20City?type=BHK4&searchParam=W3sibGF0IjoxMi44NDUy

#grtting an url

page10= requests.get(url)

#see content in
soup10= bs4(page10.content)

#empty list
house = []
location= []
```

```
Area= []
EMI= []
price= []

houses= soup10.find_all('a', class_="nb__3CnI6")
for i in houses:
    house.append(i.text)

loc=soup10.find_all('div',class_='nb__2CMjv')
for i in loc:
    location.append(i.text)

area= soup10.find_all('div',class_='nb__3oNyC')
for i in area:
    Area.append(i.text)

full_info = []
detail= soup10.find_all('div',class_='font-semi-bold heading-6')
for i in detail:
    full_info.append(i.text)

for i in range(1,len(full_info),3):
    EMI.append(full_info[i])

for i in range(2,len(full_info),3):
    price.append(full_info[i])


nobroker=pd.DataFrame({})
nobroker['House']=house
nobroker['Area']= area
nobroker['location']=location
nobroker['EMI']=EMI
nobroker['prince']=price
nobroker
```

Out[26]:

	House	Area	location	EMI	prince
0	4 BHK In Independent House For Sale In Hebba...	[1,800 sqft]	Independent House, Bangalore - Hosur Road, Nea...	₹77,374/Month	₹1.35 Crores
1	4 BHK Apartment For Sale In Nisarga Residenc...	[2,000 sqft]	Nisarga Residency Near Thali Resturant, Anant...	₹45,851/Month	₹80 Lacs
2	4 BHK Flat For Sale In Sobha Silicon Oasis ...	[1,879 sqft]	Sobha Silicon Oasis Naganathapura, Rayasandra ...	₹9,170/Month	₹16 Lacs
3	4 BHK For Sale In Daadys Garden In Electronic...	[2,600 sqft]	Daadys Garden Kammasandra Rd, Kammasandra, El...	₹85,971/Month	₹1.5 Crores
4	4 BHK Flat For Sale In , Electronic City	[2,000 sqft]	Standalone Building, 16th Cross Road Neeladri ...	₹39,546/Month	₹69 Lacs
5	4 BHK Flat For Sale In Hosa Road, Parappana ...	[3,000 sqft]	Standalone Building, 11th cross.anjanadri lay out	₹71,643/Month	₹1.25 Crores
6	4 BHK In Independent House For Sale In Elect...	[3,000 sqft]	Independent House, surya nagar face 1	₹1.43 Lacs/Month	₹2.5 Crores
7	4 BHK In Independent House For Sale In Elect...	[1,200 sqft]	Independent House, Hosur Rd,Near Infosys Limited	₹42,985/Month	₹75 Lacs
8	4 BHK Apartment For Sale In Gopalan Gardenia...	[2,650 sqft]	Gopalan Gardenia Gopalan gardenia, Veerasandr...	₹68,777/Month	₹1.2 Crores
9	4 BHK For Sale In Gpr Royale In Gpr Royale	[3,100 sqft]	6th Cross	₹85,971/Month	₹1.5 Crores

Q8___Write a python program to scrape mentioned details from 'https://www.dineout.co.in/delhi-restaurants/buffetspecial' : i) Restaurant name ii) Cuisine iii) Location

In [55]:

```
#send get request to the webpage server to get the source code of the page

url = "https://www.dineout.co.in/delhi-restaurants/buffet"

#getting url
page11 = requests.get(url)

# see content in page14
soup11 = bs4(page11.content)


name = []  #empty list
location = []  #empty list
cusinine = []  #empty list


# scrape name title
names = soup11.find_all("a", class_="restnt-name ellipsis")
for i in names:
    name.append(i.text)

# scrape location title
loc = soup11.find_all("div", class_="restnt-loc ellipsis")
for i in loc:
    location.append(i.text)
```

```
# scrape cusinine title
cus = soup11.find_all("span", class_="double-line-ellipsis")
for i in cus:
    cusinine.append(i.text)



# making data frame
restaurant=pd.DataFrame({})
restaurant['name']=name
restaurant['location']=location
restaurant['cusinine']=cusinine


#printing the final result
restaurant
```

Out[55]:

	name	location	cusinine
0	The G.T. ROAD	M-Block,Connaught Place, Central Delhi	₹ 1,800 for 2 (approx) | North Indian
1	Tamra	Shangri La Eros New Delhi,Janpath, Central Delhi	₹ 4,000 for 2 (approx) | Multi-Cuisine, Europe...
2	Barbeque Nation	Connaught Place, Central Delhi	₹ 1,700 for 2 (approx) | North Indian, Barbecu...
3	Lazeez Affaire	Connaught Place, Central Delhi	₹ 2,600 for 2 (approx) | North Indian, Mughlai
4	Castle Barbeque	Connaught Place, Central Delhi	₹ 2,000 for 2 (approx) | Chinese, North Indian
5	Khandani Rajdhani	Scindia House,Connaught Place, Central Delhi	₹ 1,000 for 2 (approx) | Gujarati, Rajasthani
6	Fifty9	Radisson Blu Marina,Connaught Place, Central D...	₹ 2,800 for 2 (approx) | North Indian, Contine...
7	Shang Palace	Shangri La Eros New Delhi,Janpath, Central Delhi	₹ 4,000 for 2 (approx) | Chinese, Asian
8	MIST	The Park Hotel,Connaught Place, Central Delhi	₹ 4,000 for 2 (approx) | Multi-Cuisine
9	24/7	The Lalit New Delhi,Connaught Place, Central D...	₹ 5,000 for 2 (approx) | Multi-Cuisine, Asian,...
10	The One	Le Meridien New Delhi,Janpath, Central Delhi	₹ 5,000 for 2 (approx) | Continental, European
11	Yellow Brick Road	Ambassador,Khan Market, Central Delhi	₹ 3,500 for 2 (approx) | Continental, North In...
12	Varq	The Taj Mahal,Mansingh Road, Central Delhi	₹ 4,000 for 2 (approx) | North Indian, Mughlai
13	Jungle Jamboree	3CS Mall,Lajpat Nagar - 3, South Delhi	₹ 1,400 for 2 (approx) | North Indian, Barbecu...
14	Suruchi	Karol Bagh, Central Delhi	₹ 800 for 2 (approx) | Rajasthani, North India...
15	Gram by Taksim	Ansal Plaza Mall,Khel Gaon, South Delhi	₹ 2,200 for 2 (approx) | Asian, North Indian, ...
16	Barbeque Nation	Unity One Mall,Janakpuri, West Delhi	₹ 1,700 for 2 (approx) | Barbecue, North India...
17	Paatra	Jaypee Siddharth,Rajendra Place, West Delhi	₹ 3,000 for 2 (approx) | Mughlai, North Indian...
18	Barbeque Nation	Community Centre - New Friends Colony,New Frie...	₹ 1,700 for 2 (approx) | North Indian, Barbecu...
19	Castle Barbeque	Pacific Mall,Tagore Garden, West Delhi	₹ 2,000 for 2 (approx) | North Indian, Chinese
20	The Barbeque Company	Karkardooma, East Delhi	₹ 2,100 for 2 (approx) | North Indian, Barbecu...

Q9___Write a python program to scrape weather details for last 24 hours from 'https://en.tutiempo.net/delhi.html?data=last-24- hours' :

In [31]:

```
#send get request to the webpage server to get the source code of the page
url = "https://en.tutiempo.net/delhi.html?data=last-24-%20hours"
page12 = requests.get(url)
# see content in page14
soup12 = bs4(page12.content)


Hour = []  #empty list
Temperature = []  #empty list
wind = []  #empty list
Humidity = []   #empty list

#scrape wind title
winds = soup12.find_all("span", class_="wind")
for i in winds:
    wind.append(i.text)

#scrape temperature title
temp = soup12.find_all("span", class_="t temp")
for i in temp:
    Temperature.append(i.text)

#scrape Hour title
hr = soup12.find_all("span",class_="h")
for i in hr:
    Hour.append(i.text)
```

```
# make data frame
weather=pd.DataFrame({})
weather['wind']=wind
weather['Temperature']=Temperature
weather['Hour']=Hour
weather
```

Out[31]:

	wind	Temperature	Hour
0	Calm	19°	Now
1	6 km/h	23°	03:00
2	6 km/h	22°	06:00
3	8 km/h	25°	09:00
4	9 km/h	29°	12:00
5	10 km/h	30°	15:00
6	6 km/h	28°	18:00
7	9 km/h	25°	21:00
8	9 km/h	24°	00:00
9	9 km/h	23°	03:00
10	9 km/h	22°	06:00
11	12 km/h	25°	09:00
12	10 km/h	29°	12:00
13	7 km/h	30°	15:00
14	11 km/h	28°	18:00
15	12 km/h	25°	21:00
16	14 km/h	23°	00:00
17	12 km/h	22°	03:00
18	13 km/h	21°	06:00
19	12 km/h	24°	09:00
20	10 km/h	29°	12:00
21	7 km/h	30°	15:00
22	11 km/h	28°	18:00
23	10 km/h	26°	21:00
24	10 km/h	25°	00:00
25	10 km/h	23°	03:00
26	4 km/h	22°	06:00
27	4 km/h	26°	09:00
28	9 km/h	29°	12:00
29	12 km/h	30°	15:00
30	11 km/h	27°	18:00
31	4 km/h	26°	21:00
32	4 km/h	25°	00:00
33	6 km/h	23°	03:00
34	5 km/h	22°	06:00
35	8 km/h	25°	09:00
36	11 km/h	29°	12:00
37	13 km/h	29°	15:00
38	8 km/h	27°	18:00
39	5 km/h	26°	21:00
40	7 km/h	24°	00:00
41	6 km/h	23°	03:00
42	7 km/h	23°	06:00
43	9 km/h	25°	09:00
44	13 km/h	29°	12:00
45	14 km/h	30°	15:00
46	12 km/h	28°	18:00
47	10 km/h	27°	21:00
48	8 km/h	25°	00:00

49	7 km/h	24°	03:00
50	8 km/h	23°	06:00
51	9 km/h	25°	09:00
52	13 km/h	29°	12:00
53	14 km/h	30°	15:00
54	10 km/h	28°	18:00
55	8 km/h	27°	21:00

In [ ]: