In [10]:

```
#QUESTION 1:Write a python program to display IMDB’s Top rated 100 Indian movi
# https://www.imdb.com/list/ls056092300/ (i.e. name, rating, year of release)
```

```
Requirement already satisfied: bs4 in c:\users\microsoft\anaconda3\lib\site-p
ackages (0.0.2)
Requirement already satisfied: beautifulsoup4 in c:\users\microsoft\anaconda3
\lib\site-packages (from bs4) (4.12.2)
Requirement already satisfied: soupsieve>1.2 in c:\users\microsoft\anaconda3
\lib\site-packages (from beautifulsoup4->bs4) (2.4)
Requirement already satisfied: requests in c:\users\microsoft\anaconda3\lib\s
ite-packages (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\microsoft
\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\microsoft\anaconda3\l
ib\site-packages (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\microsoft\anaco
nda3\lib\site-packages (from requests) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\microsoft\anaco
nda3\lib\site-packages (from requests) (2023.7.22)
```

In [13]:

```
! pip install bs4
! pip install request

from bs4 import BeautifulSoup
import requests
import pandas as pd

url='https://www.imdb.com/list/ls056092300/'
response= requests.get(url)
if response.status_code == 200:
    soup = BeautifulSoup(response.text,'html.parser')
    names=[]
    rating=[]
    years=[]

    movies=soup.find_all('div',class_='lister-item mode-detail')


    for movie in movies:
        name=movie.find('h3',class_='lister-item-header').find('a').text.strip
        names.append(name)
        rating=movie.find('div',class_='ratings-bar').find('strong')
        rating=ratings.text.strip() if rating else 'N/A'
        ratings.append(rating)
        year=movie.find('span',class_='lister-item-year').text.strip()
        year=year.replace('(','').replace(')', '')
        years.append(years)

        df=pd.DataFrame ({
        'Name':names,
        'Rating':ratings,
        'Year':years
         })

    print(df)

    #df.to_csv('top_100_indian_movies.csv,index=False')
```

```
Requirement already satisfied: bs4 in c:\users\microsoft\anaconda3\lib\sit
e-packages (0.0.2)
Requirement already satisfied: beautifulsoup4 in c:\users\microsoft\anacon
da3\lib\site-packages (from bs4) (4.12.2)
Requirement already satisfied: soupsieve>1.2 in c:\users\microsoft\anacond
a3\lib\site-packages (from beautifulsoup4->bs4) (2.4)

ERROR: Could not find a version that satisfies the requirement request (fr
om versions: none)
ERROR: No matching distribution found for request
```

In [ ]:

```
##QUESTION 2.Write a python program to scrape details of all the posts from ht
#heading, date, content and the likes for the video from the link for the yout

! pip install requests
! pip install bs4
! pip install pytube

import requests
from bs4 import BeautifulSoup
from pytube import Youtube

def get_youtube_likes(video_url):
    try:
        yt=Youtube(video_url)
        return yt.likes
    except Exception as e:
        print(f"Error fetching likes for {video_url}:{e}")
        return None
    url='https://www.patreon.com/coreyms'
    response=requests.get(url)
    if response.status_code!=200:
        print(f"failed to retrieve page:Status_code{response.status_code}")
        exit()
        soup = BeautifulSoup(reponse.text,'html.parser')
        posts = soup.find_all('div',class_='post')

        for post in posts:
            try:
                heading=post.find('h2').text.strip()
                date=post.find('time').text.strip()
                content=post.find('a',herf=True,text='Watch on Youtube')
            if video_tag:
                    video_url=video_tag['herf']
                    likes=get_youtube_likes(video_url)
            else:
                        video_url=None
                        Likes=None
            print(f"Heading:{'heading'}")
            print(f"Date:{'date'}")
            print(f"Content:{'content'}")
            if video_url:
                  print(f"video.URL:{video_url}")
                  print("\n"+"-"*40 + "\n")

            except Exception as e:

                        print(f"Error processing a post:{e}")
```

In [ ]:

```
# QUESSTION:Write pythonn code to scrape house details from mentioned URL.It s
#https://www.nobroker.in , enter three localaties which are Indira Nagar,Jayan

! pip install bs4
! pip install requests
import requests
from bs4 import BeautifulSoup

base_url = 'https://www.nobroker.in'
localities = [" Indira Nagar, Jayanagar, Rajaji Nagar"]

def scrape_details(url):
    response = requests.get(url)
    soup=BeautifulSoup(response.text,'html.parser')

    listings= soup.find_all('div',class_='listing')
    details=[]

    for listing in listings:
        title=listing.find('h2',class_='title').text.strip()if listing.find('h
        location=listing.find('span',class_='location').text.strip()if listing
        area=listing.find('span',class_='area').text.strip()if listing.find('s
        emi=listing.find('span',class_='emi').text.strip()if listing.find('spa
        princ=listing.find('span',class_='price').text.strip()if listing.find(

        details.append({
            'title':title,
            'location':location,
            'area':area,
            'emi':emi,
            'price':price
        })

        return details
        def main():
            for locality in localities:
                search_url=f"{base_url}/property-for-rent-in{locality:replace(
                print(f"scraping details from:{search_url}")

                details=scrape.details(search_url)

                for detail in details:
                    print(f"Title:{detail['title']}")
                    print(f"Location:{detail['location']}")
                    print(f"Area:{detail['area']}")
                    print(f"Price:{detail['price']}")
                    print("-" * 40 )
```

In [ ]:

```
#QUESTION:Write Python code to scrape first 10 product details which include p
#https://www.bewakoof.com/bestseller?sor=popular
! pip install bs4
! pip install requests

import requests
from bs4 import BeautifulSoup
url='https://www.bewakoof.com/bestseller?sort=popular'
response=requests.get(url)
if response.status_code !=200:
    print('Failed to retrieve the web page')
    exist()
    soup=BeautifulSoup(response.text,'html.parser')
    product_container = soup.find_all('div',class_='productCard')
    product=[]
    for container in product_container[:10]:
        try:
            name=container.find('h3',class_='productCardName').text.strip()
            price=container.find('span',class_='productCardPrice').text.strip(
            image_url=container.find('img',class_='productCardImage')['src']
            products.append({
                'name':name,
                'price':price,
                'image_url':image_url
            })

    except AttributeError:

                continue,
                for product in products:
                    print(f"Product_Name:{product['name']}")
                    print(f"Price={product['price']}")
                    print(f"Image_Url={product['image_url']}")
                    print(f"---"* 40)
```

In [ ]:

```
## QUESTION .Please visit https://www.cnbc.com/world/?region=world and scrap-

! pip install bs4
! pip install requests

import requests
from bs4 import BeautifulSoup

url='https://www.cnbc.com/world/?region=world'
response=requests.get(url)
soup=Beautifulsoup(response.text,'html.parser')
articles=soup.find_all('div',class_='Card')
for article in articles:
    heading=article.find('a',class_='Card-title').get_text(strip=True) if arti
    date=article.find('time').get_text(strip=True) if article.find('time')else
    link=article.find('a',class_='Card-title')['herf'] if article.find('a',cla
print(f'Heading:{heading}')
print(f'Date:{date}')
print(f'Link:{link}')
print('---')
```

In [ ]:

```
#Please visit https://www.keaipublishing.com/en/journals/artificial-intelligen
#/ and scrap a) Paper title b) date c) Author

! pip install bs4
! pip install requests

import requests
from bs4 import BeautifulSoup

url='https://www.keaipublishing.com/en/journals/artificial-intelligence-in-agr
response=request.get(url)

if response.status_code == 200:
    soup=BeautifulSoup(response.text,'html.parser')
    articles=soup.find_all('div',class_='article-list-item')

    for article in articles:
        title=article.find('h3').get_text(strip=True) if article.find('h3')els
        date= date.find('time').get_text(strip=True) if article.find('time') e
        authors= article.find('div',class_='authors').get_text(strip=True)if a

        print(f'Title:{title}')
        print(f'Date:{date}')
        print(f'Authors:{authors}')
        print('---')
    else:
            print('Failed to retrieve the page.status_code:{response.status.co
```