In [2]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [3]:
```
needed_headers = {'User-Agent': "Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWe
response = requests.get("https://www.themoviedb.org/movie")
```

In [4]:
```
response.status_code
```

Out[4]:
```
403
```

In [5]:
```
dwn_content = response.text
len(dwn_content)
```

Out[5]:
```
3167
```

In [7]:
```
dwn_content[:500]
```

Out[7]:
```
'<!DOCTYPE html>\n<html lang="en" class="no-js">\n  <head>\n    <title>Request
Error (403) - The Movie Database (TMDb)</title>\n    <meta http-equiv="X-UA-Com
patible" content="IE=edge" />\n    <meta http-equiv="cleartype" content="on">\n
<meta charset="utf-8">\n    <meta name="robots" content="noindex">\n    <meta n
ame="mobile-web-app-capable" content="yes">\n    <meta name="apple-mobile-web-a
pp-capable" content="yes">\n    <meta name="HandheldFriendly" content="True">\n
<meta name="MobileOptimized" c'
```

In [8]:
```
test_doc = BeautifulSoup(response.text, 'html.parser')
```

In [9]:
```
type(test_doc)
```

Out[9]:
```
bs4.BeautifulSoup
```

In [10]:
```
test_doc.find('title')
```

Out[10]:
```
<title>Request Error (403) - The Movie Database (TMDb)</title>
```

In [11]:
```
test_doc.find('img')
```

Out[11]:
```
<img height="60" src="https://www.themoviedb.org/assets/2/apple-touch-icon-cfba
7699efe7a742de25c28e08c38525f19381d31087c69e89d6bcb8e3c0ddfa.png" width="60"/>
```

In [12]:
```
def get_page_content(url):
    # In this case , we are going to give request.get function headers to avoid t

    get_headers = {'User-Agent': "Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleW
    response_page = requests.get(url, headers = get_headers )
    # we are going to raise exception here if status code gives any value other t
    if not response_page.ok:
        raise Exception ("Failed to request the data. Status Code:- {}".format(re
    else:
        page_content = response_page.text
        doc_page = BeautifulSoup(page_content, "html.parser")
        return doc_page
```

In [13]:
```
popular_shows_url = "https://www.themoviedb.org/movie"
doc = get_page_content(popular_shows_url)
```

In [14]:
```
doc.title.text
```

Out[14]:
```
'Popular Movies — The Movie Database (TMDB)'
```

In [15]:
```
doc.find_all('div', {'class': 'card style_1'})[0].h2.text
```

Out[15]:
```
'Black Adam'
```

In [18]:
```
doc.find_all('div', {'class': 'user_score_chart'})[0]['data-percent']
```

Out[18]:
```
'73.0'
```

In [23]:
```
def empty_dict():
    scraped_dict = {
                    'Title': [],
                    'User_rating': [],
                    'Release_date':[],
                    'Genre': [],
                    'Director': [],
                     'Cast': []
                    }
    return scraped_dict
```

In [24]:
```
def user_score_info(tag_user_score, i, scraped_dict):
    if tag_user_score[i]['data-percent'] == '0':
        scraped_dict['User_rating'].append('Not rated yet')
    else:
        scraped_dict['User_rating'].append(tag_user_score[i]['data-percent'])
```

In [25]:
```
doc.find_all('div', {'class': 'card style_1'})[0].h2.a['href']
```

Out[25]:
```
'/movie/436270'
```

In [26]:
```
def get_show_info(doc_page):
    base_link_1 = "https://www.themoviedb.org"
    tag_title = tag_premired_date = tag_shows_page = doc_page.find_all('div', {'c
    tag_user_score = doc_page.find_all('div', {"user_score_chart"})

    doc_2_list = []
    for link in tag_shows_page:
        # here we are creating the list of all the individual pages of the shows
        doc_2_list.append(get_page_content("https://www.themoviedb.org" + link.h2
       # we are going to have the function to return the list of all the informat
    return tag_title, tag_user_score, doc_2_list
```

In [27]:
```
len(get_show_info(doc))
```

Out[27]:
```
3
```

In [28]:
```
doc_2 = get_page_content("https://www.themoviedb.org/movie/436270")
```

In [29]:
```
tag_genre = doc_2.find('span', {"class": "genres"})
tag_genre_list = tag_genre.find_all('a')

check_genre =[]
for tag in tag_genre_list:
    check_genre.append(tag.text)

check_genre
```

Out[29]:
```
['Action', 'Fantasy', 'Science Fiction']
```

In [30]:
```
# lets create a function to get the genres for the movie.
# i here denotes the element of the list vairable ``doc2_page`` that contains dif
def get_genres(doc2_page, i):
    genres_tags = doc2_page[i].find('span', {"class": "genres"}).find_all('a')
    check_genre =[]

    for tag in genres_tags:
        check_genre.append(tag.text)
    return check_genre
```

In [31]:
```
# i here denotes the the element of the list type variable``doc2_page`` that cont

def get_show_Director(doc2_page, i):
    director_tags = doc2_page[i].find_all('li', {'class': 'Director'})
    director_list = []

    for t in director_tags:
         director_list.append(t.p.text)

    return director_list
```

In [32]:
```
def get_show_cast(doc2_page, i):
    cast_tags = doc2_page[i].find_all('li', {'class': 'card'})
    cast_lis = []

    for t in cast_tags:
         cast_lis.append(t.p.text)

    return cast_lis
```

In [33]:
```
import pandas as pd

def get_show_details(t_title, t_user_score, docs_2_list):
    # excuting a function here that empties the dictionary every time the functio
    scraped_dict =  empty_dict()
    for i in range (0, len(t_title)):
        scraped_dict['Title'].append(t_title[i].h2.text)
        user_score_info(t_user_score, i, scraped_dict)
        scraped_dict['Release_date'].append(t_title[i].p.text)
        scraped_dict['Genre'].append(get_genres(docs_2_list, i))
        scraped_dict['Director'].append(get_show_Director(docs_2_list, i))
        scraped_dict['Cast'].append(get_show_cast(docs_2_list, i))

    return pd.DataFrame(scraped_dict)
```

In [34]:
```
tag_title_, tag_user_score_, doc_2_list_ = get_show_info(doc)
```

In [35]:
```
import csv
```

In [37]:
```
x = get_show_details(tag_title_, tag_user_score_, doc_2_list_)
x.to_csv('check.csv')
pd.read_csv('check.csv',index_col=[0])
```

Out[37]:

	Title	User_rating	Release_date	Genre	Director	Cast
0	Black Adam	73.0	Oct 19, 2022	['Action', 'Fantasy', 'Science Fiction']	[]	['Dwayne Johnson', 'Aldis Hodge', 'Noah Centin...
1	R.I.P.D. 2: Rise of the Damned	68.0	Nov 15, 2022	['Fantasy', 'Action', 'Comedy', 'Crime']	[]	['Jeffrey Donovan', 'Penelope Mitchell', 'Rich...
2	Paradise City	63.0	Nov 11, 2022	['Crime', 'Action', 'Thriller']	[]	['John Travolta', 'Bruce Willis', 'Blake Jenne...
3	Corrective Measures	50.0	Apr 29, 2022	['Science Fiction', 'Action']	[]	['Bruce Willis', 'Hayley Sales', 'Michael Rook...
4	Hex	43.0	Nov 01, 2022	['Action', 'Horror', 'Thriller']	[]	['Kayla Adams', 'Matthew Holcomb', 'Bryan Davi...
5	The Woman King	79.0	Sep 15, 2022	['Action', 'Drama', 'History']	[]	['Viola Davis', 'Thuso Mbedu', 'Lashana Lynch'...
6	Emily the Criminal	69.0	Aug 12, 2022	['Crime', 'Drama', 'Mystery', 'Thriller']	[]	['Aubrey Plaza', 'Theo Rossi', 'Megalyn Echiku...
7	Lost Bullet 2	68.0	Nov 10, 2022	['Action', 'Drama', 'Thriller']	[]	['Alban Lenoir', 'Stéfi Celma', 'Pascale Arbil...
8	The Minute You Wake Up Dead	49.0	Nov 04, 2022	['Thriller', 'Crime']	[]	['Cole Hauser', 'Jaimie Alexander', 'Morgan Fr...
9	Disenchanted	73.0	Nov 16, 2022	['Comedy', 'Family', 'Fantasy']	[]	['Amy Adams', 'Patrick Dempsey', 'Maya Rudolph...
10	Frank and Penelope	75.0	Jun 03, 2022	['Thriller', 'Horror', 'Crime']	[]	['Kevin Dillon', 'Sean Patrick Flanery', 'John...
11	Margaux	68.0	Sep 09, 2022	['Horror', 'Science Fiction']	[]	['Madison Pettis', 'Vanessa Morgan', 'Richard ...
12	Black Panther: Wakanda Forever	75.0	Nov 09, 2022	['Action', 'Adventure', 'Science Fiction']	[]	['Letitia Wright', "Lupita Nyong'o", 'Danai Gu...
13	Lyle, Lyle, Crocodile	77.0	Oct 07, 2022	['Comedy', 'Family', 'Music']	[]	['Winslow Fegley', 'Javier Bardem', 'Constance...

	Title	User_rating	Release_date	Genre	Director	Cast
14	Sniper: The White Raven	75.0	May 03, 2022	['Drama', 'Action', 'War']	[]	['Pavlo Aldoshyn', 'Maryna Koshkina', 'Andrei ...
15	Medieval	72.0	Sep 08, 2022	['History', 'Action', 'Drama']	[]	['Ben Foster', 'Sophie Lowe', 'Michael Caine',...
16	Smile	68.0	Sep 23, 2022	['Horror', 'Mystery', 'Thriller']	[]	['Sosie Bacon', 'Kyle Gallner', 'Caitlin Stase...
17	On the Line	65.0	Oct 31, 2022	['Thriller']	[]	['Mel Gibson', 'Kevin Dillon', 'William Mosele...
18	Blue's Big City Adventure	75.0	Nov 18, 2022	['Family', 'Adventure', 'Music', 'Animation']	[]	['Joshua Dela Cruz', 'Steve Burns', 'Donovan P...
19	Slumberland	79.0	Nov 09, 2022	['Family', 'Fantasy', 'Adventure', 'Drama']	[]	['Jason Momoa', 'Marlow Barkley', "Chris O'Dow...

In [39]:
```
import os
base_link = "https://www.themoviedb.org/movie"

# 'i' here means the number of page we want to extract
def create_page_df( i, dataframe_list):
    os.makedirs('shows-data', exist_ok = True)
    next_url = base_link + '?page={}'.format(i)
    doc_top = get_page_content(next_url)
    name_tag, viewer_score_tag, doc_2_lis = get_show_info(doc_top)
    print('scraping page {} :- {}'.format(i, next_url))
    dataframe_data = get_show_details(name_tag, viewer_score_tag, doc_2_lis)
    dataframe_data.to_csv("shows-data/shows-page-{}.csv".format(i) , index = None
    print(" ---> a CSV file with name shows-page-{}.csv has been created".format(
    dataframe_list.append(dataframe_data)
```

In [40]:
```
test_list = []
create_page_df(50 , test_list)
```

```
scraping page 50 :- https://www.themoviedb.org/movie?page=50 (https://www.themo
viedb.org/movie?page=50)
 ---> a CSV file with name shows-page-50.csv has been created
```

In [41]:
```
import pandas as pd
base_link = "https://www.themoviedb.org/movie"

def scrape_top_1000_shows(base_link):
    dataframe_list = []
    # we are going to keep range up to 1001 because we just need up to 1000 movie
    for i in range(1,101):
        create_page_df(i, dataframe_list)
    # here we are using concat function so that we can merge the each dataframe t
    total_dataframe = pd.concat(dataframe_list, ignore_index = True)

    # with the simple command of to_csv() we can create a csv file of all the pag
    csv_complete =  total_dataframe.to_csv('shows-data/Total-dataframe.csv', inde
    print(" \n a CSV file named Total-dataframe.csv with all the scraped shows ha
```

In [42]:
```
scrape_top_1000_shows(base_link)
```

```
scraping page 68 :- https://www.themoviedb.org/movie?page=68 (https://www.the
moviedb.org/movie?page=68)
 ---> a CSV file with name shows-page-68.csv has been created
scraping page 69 :- https://www.themoviedb.org/movie?page=69 (https://www.the
moviedb.org/movie?page=69)
 ---> a CSV file with name shows-page-69.csv has been created
scraping page 70 :- https://www.themoviedb.org/movie?page=70 (https://www.the
moviedb.org/movie?page=70)
 ---> a CSV file with name shows-page-70.csv has been created
scraping page 71 :- https://www.themoviedb.org/movie?page=71 (https://www.the
moviedb.org/movie?page=71)
 ---> a CSV file with name shows-page-71.csv has been created
scraping page 72 :- https://www.themoviedb.org/movie?page=72 (https://www.the
moviedb.org/movie?page=72)
 ---> a CSV file with name shows-page-72.csv has been created
```

```
In [43]: pd.read_csv('shows-data/Total-dataframe.csv')[0:100]
```

Out[43]:

	Title	User_rating	Release_date	Genre	Director	Cast
0	Black Adam	73.0	Oct 19, 2022	['Action', 'Fantasy', 'Science Fiction']	[]	['Dwayne Johnson', 'Aldis Hodge', 'Noah Centin...
1	R.I.P.D. 2: Rise of the Damned	68.0	Nov 15, 2022	['Fantasy', 'Action', 'Comedy', 'Crime']	[]	['Jeffrey Donovan', 'Penelope Mitchell', 'Rich...
2	Paradise City	63.0	Nov 11, 2022	['Crime', 'Action', 'Thriller']	[]	['John Travolta', 'Bruce Willis', 'Blake Jenne...
3	Corrective Measures	50	Apr 29, 2022	['Science Fiction', 'Action']	[]	['Bruce Willis', 'Hayley Sales', 'Michael Rook...
4	Hex	43.0	Nov 01, 2022	['Action', 'Horror', 'Thriller']	[]	['Kayla Adams', 'Matthew Holcomb', 'Bryan Davi...
...	...	...	...	...	...	...
95	Batman and Superman: Battle of the Super Sons	80	Oct 17, 2022	['Animation', 'Action', 'Science Fiction']	[]	['Jack Dylan Grazer', 'Jack Griffo', 'Laura Ba...
96	Samaritan	69.0	Aug 25, 2022	['Action', 'Drama', 'Science Fiction']	[]	['Javon Walton', 'Sylvester Stallone', 'Dascha...
97	After Ever Happy	70	Aug 24, 2022	['Romance', 'Drama']	[]	['Josephine Langford', 'Hero Fiennes Tiffin', ...
98	The Addams Family	70	Nov 22, 1991	['Comedy', 'Fantasy']	[]	['Raúl Juliá', 'Anjelica Huston', 'Christopher...
99	Doctor Strange in the Multiverse of Madness	74.0	May 04, 2022	['Fantasy', 'Action', 'Adventure']	[]	['Benedict Cumberbatch', 'Elizabeth Olsen', 'C...

100 rows × 6 columns

In [42]:
```
# Reading the csv file
df_new = pd.read_csv('shows-data/Total-dataframe.csv')

# saving xlsx file
GFG = pd.ExcelWriter('Names.xlsx')
df_new.to_excel(GFG, index=False)

GFG.save()
```

In [43]:
```
final = pd.ExcelWriter('GFG.xlsx')
```