jupyter Beautiful_soup_WebScarpping By Kundan Last Checkpoint: 11 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [7]:
```
!pip install bs4
```

```
Requirement already satisfied: bs4 in c:\anaconda\lib\site-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in c:\anaconda\lib\site-packages (from bs4) (4.9.3)
Requirement already satisfied: soupsieve>1.2; python_version >= "3.0" in c:\anaconda\lib\site-packages (from beautifulsoup4->bs4) (2.0.1)
```

In [8]:
```
!pip install requests
```

```
Requirement already satisfied: requests in c:\anaconda\lib\site-packages (2.24.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\anaconda\lib\site-packages (from requests) (1.25.11)
Requirement already satisfied: certifi>=2017.4.17 in c:\anaconda\lib\site-packages (from requests) (2020.6.20)
Requirement already satisfied: idna<3,>=2.5 in c:\anaconda\lib\site-packages (from requests) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\anaconda\lib\site-packages (from requests) (3.0.4)
```

In [9]:
```
from bs4 import BeautifulSoup
import requests
```

In [10]:
```
source=requests.get('https://www.imdb.com/chart/top/')
```

In [11]:
```
source.raise_for_status() #in case some issue
```

In [33]:
```
try:
    source =requests.get('https://www.imdb.com/chart/top/')
    source.raise_for_status()
    soup= BeautifulSoup(source.text,'html.parser')
    movies =soup.find('tbody',class_="lister-list").find_all('tr')


    print(movies)

except Exception as e:
    print(e)
```

```
[<tr>
<td class="posterColumn">
<span data-value="1" name="rk"></span>
<span data-value="9.234788363719158" name="ir"></span>
<span data-value="7.791552E11" name="us"></span>
<span data-value="2640805" name="nv"></span>
<span data-value="-1.765211636280842" name="ur"></span>
<a href="/title/tt0111161/"> <img alt="The Shawshank Redemption" height="67" src="https://m.media-amazon.com/images/M/MV5BMDFkYTc0MGEtZmNhMC00ZDIzLWFmNTEtODM1ZmRlYWMwMWFmXkEyXkFqcGdeQXVyMTMxODk2OTU@._V1_UY67_CR0,0,45,67_AL_.jpg" width="45"/>
</a> </td>
<td class="titleColumn">
      1.
      <a href="/title/tt0111161/" title="Frank Darabont (dir.), Tim Robbins, Morgan Freeman">The Shawshank Redemption</a>
<span class="secondaryInfo">(1994)</span>
</td>
<td class="ratingColumn imdbRating">
<strong title="9.2 based on 2,640,805 user ratings">9.2</strong>
</td>
<td class="ratingColumn">
```

In [16]:
```
len(movies)
```

Out[16]: 250

In [57]:
```
data =[]
S_no =[]
Movie_name=[]

for movie in movies:

    name =movie.find('td',class_="titleColumn").get_text(strip=True).split('.')[0]
    #     movies =soup.find('tbody',class_="lister-list").text
#         movies =soup.find('tbody',class_="lister-list")

#     data.append(data)
    print(name)
```

In [ ]:
```

```

In [112]:
```
import openpyxl
excel =openpyxl.Workbook()
sheet=excel.active
sheet.title ="Top Rated Movies"
print(excel.sheetnames)
Movie_Name = []
Year=[]
s_no=[]
i=1
ratings=[]
sheet.append(["S_no","Movies Name",'Year of Release',"IMBD Rating"])
for movie in movies:
#     s_no.append(i)
    serial_no=name =movie.find('td',class_="titleColumn").get_text(strip=True).split('.')[0]
```

```
#     print(serial_no)
#     print(movie)
#     name =movie.find('td',class_="titleColumn").find('a').text
    name =movie.find('td',class_="titleColumn").a.get_text(strip=True)


    Movie_Name.append(name)
    year =movie.find('td',class_="titleColumn").span.text.strip('()')
    Year.append(year)
    i=i+1
    rate=movie.find('td', class_="ratingColumn imdbRating").text.strip()


    s_no.append(serial_no)
    ratings.append(rate)
#     Data_without_list =print([i,name,year,rate])
#     print(s_no,Movie_Name,Year,ratings)
    sheet.append([serial_no,name,year,rate])
# print(ratings)
# print(s_no)
# print(Movie_Name)
# print(Year)
# print(s_no,Movie_Name,Year,ratings)
excel.save('IMBD Movie Ratings.xlsx')

```

```
['Top Rated Movies']
```

In [113]:
```
import pandas as pd
```

In [116]:
```
Datas =pd.DataFrame({"S.No":s_no,"Name":Movie_Name,"Year":Year,"Rating":ratings})
```

In [117]:
```
Datas
```

Out[117]:

	S.No	Name	Year	Rating
0	1	The Shawshank Redemption	1994	9.2
1	2	The Godfather	1972	9.2
2	3	The Dark Knight	2008	9.0
3	4	The Godfather Part II	1974	9.0
4	5	12 Angry Men	1957	8.9
...	...	...	...	...
245	246	Dersu Uzala	1975	8.0
246	247	Aladdin	1992	8.0
247	248	Gandhi	1982	8.0
248	249	The Help	2011	8.0
249	250	The Iron Giant	1999	8.0

250 rows × 4 columns

In [119]:
```
Datas.to_excel("RatingsfromPandas.xlsx",sheet_name="sheetname12",index=False)
```

In [ ]:
```

```