jupyter imdb_top_250_movies_data.csv ✔ 2 minutes ago

Logout

File Edit View Language current mode

```
,movie-names,movie-ratings,movie-year
0,The Shawshank Redemption,9.2,(1994)
1,The Godfather,9.2,(1972)
2,The Dark Knight,9.0,(2008)
3,The Godfather: Part II,9.0,(1974)
4,12 Angry Men,9.0,(1957)
5,Schindler's List,8.9,(1993)
6,The Lord of the Rings: The Return of the King,8.9,(2003)
7,Pulp Fiction,8.9,(1994)
8,The Lord of the Rings: The Fellowship of the Ring,8.8,(2001)
9,"Il buono, il brutto, il cattivo",8.8,(1966)
10,Forrest Gump,8.8,(1994)
11,Fight Club,8.8,(1999)
12,Inception,8.7,(2010)
13,The Lord of the Rings: The Two Towers,8.7,(2002)
14,The Empire Strikes Back,8.7,(1980)
15,The Matrix,8.7,(1999)
16,Goodfellas,8.7,(1990)
17,One Flew Over the Cuckoo's Nest,8.6,(1975)
18,Se7en,8.6,(1995)
19,Shichinin no samurai,8.6,(1954)
20,It's a Wonderful Life,8.6,(1946)
21,The Silence of the Lambs,8.6,(1991)
22,Saving Private Ryan,8.6,(1998)
23,Cidade de Deus,8.6,(2002)
24,La vita è bella,8.6,(1997)
25,The Green Mile,8.6,(1999)
26,Star Wars,8.6,(1977)
27,Interstellar,8.6,(2014)
28,Terminator 2: Judgment Day,8.5,(1991)
29,Back to the Future,8.5,(1985)
```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [4]:
```
# web scrapping
'''
Web Scrapping extracts the data from websites in the unstructured format.
It helps to collect these unstructured data and convert it in a structured form.
'''

'''
pip:

pip is the standard package manager for Python.
It allows you to install and manage additional packages.'''
```

In [5]:
```
from bs4 import BeautifulSoup

#installations
!pip install pandas
!pip install requests
!pip install beautifulsoup4

# importing
from urllib import response
import requests
import pandas as pd
from bs4 import BeautifulSoup4
```

```
Requirement already satisfied: pandas in c:\users\hp\anaconda3\lib\site-packages (1.2.4)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\hp\anaconda3\lib\site-packages (from pandas) (2.8.1)
Requirement already satisfied: pytz>=2017.3 in c:\users\hp\anaconda3\lib\site-packages (from pandas) (2021.1)
Requirement already satisfied: numpy>=1.16.5 in c:\users\hp\anaconda3\lib\site-packages (from pandas) (1.20.1)
Requirement already satisfied: six>=1.5 in c:\users\hp\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas) (1.15.
0)
```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [6]:
```
response=requests.get('https://www.imdb.com/chart/top/')
print(response)
```

```
<Response [200]>
```

In [7]:
```
response=requests.get('https://www.imdb.com/chart/top/')
bs=BeautifulSoup(response.content,"html.parser")
bs
```

Out[7]:

```
<!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>Top 250 Movies - IMDb</title>
<script>(function(t){ (t.events = t.events || {})["csm_head_post_title"] = new Date().getTime(); })(IMDbTimer);</script>
<script>
    if (typeof uet == 'function') {
      uet("be", "LoadTitle", {wb: 1});
```

Run Code

In [8]:
```
response=requests.get('https://www.imdb.com/chart/top/')
bs=BeautifulSoup(response.content,"html.parser")
bs.prettify
```

Out[8]:
```
<bound method Tag.prettify of
<!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>Top 250 Movies - IMDb</title>
<script>(function(t){ (t.events = t.events || {})["csm_head_post_title"] = new Date().getTime(); })(IMDbTimer);</script>
<script>
    if (typeof uet == 'function') {
      uet("be", "LoadTitle", {wb: 1});
```

In [9]:

```
response=requests.get('https://www.imdb.com/chart/top/')
bs=BeautifulSoup(response.content,"html.parser")
imdb_250movies=bs.find_all('td', class_='titleColumn')
imdb_250movies
```

Out[9]:

```
[<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="titleColumn">
      2.
      <a href="/title/tt0068646/" title="Francis Ford Coppola (dir.), Marlon Brando, Al Pacino">The Godfather</a>
<span class="secondaryInfo">(1972)</span>
</td>,
<td class="titleColumn">
      3.
      <a href="/title/tt0468569/" title="Christopher Nolan (dir.), Christian Bale, Heath Ledger">The Dark Knight</a>
<span class="secondaryInfo">(2008)</span>
</td>,
<td class="titleColumn">
      4.
      <a href="/title/tt0071562/" title="Francis Ford Coppola (dir.), Al Pacino, Robert De Niro">The Godfather: Part II</a>
<span class="secondaryInfo">(1974)</span>
```

jupyter web-scrappping Last Checkpoint: 15 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [10]:

```
# for structured data of imdb-250 movies-names

response=requests.get('https://www.imdb.com/chart/top/')
bs=BeautifulSoup(response.content,"html.parser")
imdb_250movies=bs.find_all('td', class_='titleColumn')

imovies=[]
for i in range(0,len(imdb_250movies)):
    imovies.append(imdb_250movies[i].a.get_text().strip())  ## using strip() for remove white spaces

imovies
```

Out[10]:

```
['The Shawshank Redemption',
 'The Godfather',
 'The Dark Knight',
 'The Godfather: Part II',
 '12 Angry Men',
 "Schindler's List",
 'The Lord of the Rings: The Return of the King',
 'Pulp Fiction',
 'The Lord of the Rings: The Fellowship of the Ring',
 'Il buono, il brutto, il cattivo',
 'Forrest Gump',
 'Fight Club',
 'Inception',
 'The Lord of the Rings: The Two Towers',
 'The Empire Strikes Back',
 'The Matrix',
 'Goodfellas',
 "One Flew Over the Cuckoo's Nest",
 'Se7en'
```

Run Code

In [11]:

```
# imdb-top 250 movies ratings

response=requests.get('https://www.imdb.com/chart/top/')
bs=BeautifulSoup(response.content,"html.parser")
imdb_ratings=bs.find_all('td', class_='ratingColumn imdbRating')

ratings=[]
for i in range(0,len(imdb_ratings)):
    ratings.append(imdb_ratings[i].get_text().strip())
ratings

```

Out[11]:

```
['9.2',
 '9.2',
 '9.0',
 '9.0',
 '9.0',
 '8.9',
 '8.9',
 '8.9',
 '8.8',
 '8.8',
 '8.8',
 '8.8',
 '8.8',
 '8.7',
 '8.7',
 '8.7',
 '8.7',
 '8.7',
 '8.7',
 '8.6',
```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [12]:

```
## imdb top 250 movies-releasing years

response=requests.get('https://www.imdb.com/chart/top/')
bs=BeautifulSoup(response.content,"html.parser")
imdb_movie_years=bs.find_all('span', class_='secondaryInfo')

years=[]
for i in range(0,len(imdb_movie_years)):
    years.append(imdb_movie_years[i].get_text().strip())  ## using strip() for remove white spaces

years
```

Out[12]:

```
['(1994)',
 '(1972)',
 '(2008)',
 '(1974)',
 '(1957)',
 '(1993)',
 '(2003)',
 '(1994)',
 '(2001)',
 '(1966)',
 '(1994)',
 '(1999)',
 '(2010)',
 '(2002)',
 '(1980)',
 '(1999)',
 '(1990)',
```

jupyter web-scrappping Last Checkpoint: 15 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

(1995)',

In [21]:
```
## create dataframe using pandas

df=pd.DataFrame()
df['movie-names']=imovies
df['movie-ratings']=ratings
df['movie-year']=years
print(df)
```

```
                movie-names movie-ratings movie-year
0    The Shawshank Redemption           9.2     (1994)
1               The Godfather           9.2     (1972)
2             The Dark Knight           9.0     (2008)
3      The Godfather: Part II           9.0     (1974)
4                  12 Angry Men           9.0     (1957)
..                          ...           ...        ...
245      Beauty and the Beast           8.0     (1991)
246                    Gandhi           8.0     (1982)
247                  The Help           8.0     (2011)
248                 Ah-ga-ssi           8.0     (2016)
249        Dances with Wolves           8.0     (1990)

[250 rows x 3 columns]
```

In [22]:
```
df.to_csv('imdb_top_250_movies_data.csv')
```