Install Required Packages

In [1]:
```
!pip install bs4
```

```
Requirement already satisfied: bs4 in c:\users\nithish\anaconda3\lib\site-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in c:\users\nithish\anaconda3\lib\site-packages (from bs4) (4.11.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\nithish\anaconda3\lib\site-packages (from beautifulsoup4->bs4) (2.3.2.
post1)
```

In [2]:
```
!pip install requests
```

```
Requirement already satisfied: requests in c:\users\nithish\anaconda3\lib\site-packages (2.28.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\nithish\anaconda3\lib\site-packages (from requests) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\nithish\anaconda3\lib\site-packages (from requests) (1.26.14)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\nithish\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\nithish\anaconda3\lib\site-packages (from requests) (3.4)
```

In [3]:
```
#import pandas in order to convert the text datas into csv formate

!pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\nithish\anaconda3\lib\site-packages (1.5.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\nithish\anaconda3\lib\site-packages (from pandas) (2022.7)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\nithish\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: numpy>=1.21.0 in c:\users\nithish\anaconda3\lib\site-packages (from pandas) (1.23.5)
Requirement already satisfied: six>=1.5 in c:\users\nithish\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas)
(1.16.0)
```

In [4]:
```
!pip install openpyxl
```

```
Requirement already satisfied: openpyxl in c:\users\nithish\anaconda3\lib\site-packages (3.0.10)
Requirement already satisfied: et_xmlfile in c:\users\nithish\anaconda3\lib\site-packages (from openpyxl) (1.1.0)
```

Import required packages

In [5]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [6]:
```
url = 'https://www.imdb.com/chart/top/'
```

In [7]:
```
#Now we have to send a request to http of that link, to do that we have to include user agent and accept language as headers beca
#Every link has a user agent and accept language as headers.


#User agents -- it is basically tells us that you are trying to access this website and you are a genuine user by identifying our
#and some other required info.
```

In [8]:
```
#Headers for request


headers = ({'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safa
```

In [9]:
```
#HTTP request

webpage = requests.get(url, headers=headers)
```

In [10]:
```
webpage
```

Out[10]:
```
<Response [200]>
```

In [11]:
```
#<Response [200]> means our http request was successfull
```

In [12]:
```
webpage.content

#This is in bytes formate
```


```
dow.csa !== \'undefined\' && typeof window.csa === \'function\') {\n            var csaLatencyPlugin = window.csa(\'Content
\', {\n               element: {\n                  slotId: \'LoadTitle\',\n                  type: \'service-call\'\n
}\n            });\n            csaLatencyPlugin(\'mark\', \'clickToLoaded\', 1690043253611);\n        }\n    })</script><me
ta name="next-head-count" content="34"/><script>\nvar ue_t0=ue_t0||+new Date();\n\nwindow.ue_ihb = (window.ue_ihb || window.
ueinit || 0) + 1;\nif (window.ue_ihb === 1) {\n\nvar ue_csm = window,\n    ue_hob = +new Date();\n(function(d){var e=d.ue=d.
ue||{},f=Date.now||function(){return+new Date};e.d=function(b){return f()-(b?0:d.ue_t0)};e.stub=function(b,a){if(!b[a]){var
c=[];b[a]=function(){c.push([c.slice.call(arguments),e.d(),d.ue_id])};b[a].replay=function(b){for(var a;a=c.shift();)b(a[0],
a[1],a[2])};b[a].isStub=1}};e.exec=function(b,a){return function(){try{return b.apply(this,arguments)}catch(c){ueLogError(c,
{attribution:a||"undefined",logLevel:"WARN"})}}}})(ue_csm);\n\n\n    var ue_err_chan = \'jserr\';\n(function(d,e){function h
(f,b){if(!(a.ec>a.mxe)&&f){a.ter.push(f);b=b||{};var c=f.logLevel||b.logLevel;c&&c!==k&&c!==m&&c!==n&&c!==p||a.ec++;c&&c!=k|
|a.ecf++;b.pageURL=""+(e.location?e.location.href:"");b.logLevel=c;b.attribution=f.attribution||b.attribution;a.erl.push({e
x:f,info:b})}}function l(a,b,c,e,g){d.ueLogError({m:a,f:b,l:c,c:""+e,err:g,fromOnError:1,args:arguments},g?{attribution:g.at
tribution,logLevel:g.logLevel}:void 0);return!1}var k="FATAL",m="ERROR",n="WARN",p="DOWNGRADED",a={ec:0,ecf:0,\npec:0,ts:0,e
rl:[],ter:[],buffer:[],mxe:50,startTimer:function(){a.ts++;setInterval(function(){d.ue&&a.pec<a.ec&&d.uex("at");a.pec=a.ec},
1E4)}};l.skipTrace=1;h.skipTrace=1;h.isStub=1;d.ueLogError=h;d.ue_err=a;e.onerror=l})(ue_csm,window);\n\n\nvar ue_id = \'CHQ
```

In [13]:
```
#Change from bytes formate into html

soup = BeautifulSoup(webpage.content, 'html.parser')      #This means send your web content and parse it into html formate.
```

In [14]:
```
print(soup)
```

```
<!DOCTYPE html>
<html lang="en-US" xmlns:fb="http://www.facebook.com/2008/fbml" xmlns:og="http://opengraphprotocol.org/schema/"><head><meta
charset="utf-8"/><meta content="width=device-width" name="viewport"/><script>if(typeof uet === 'function'){ uet('bb', 'LoadT
itle', {wb: 1}); }</script><script>window.addEventListener('load', (event) => {
        if (typeof window.csa !== 'undefined' && typeof window.csa === 'function') {
            var csaLatencyPlugin = window.csa('Content', {
                element: {
                    slotId: 'LoadTitle',
                    type: 'service-call'
                }
            });
            csaLatencyPlugin('mark', 'clickToBodyBegin', 1690043253611);
        }
    })</script><title>IMDb Top 250 Movies</title><meta content="IMDb Top 250 as rated by regular IMDb voters" data-id="main"
name="description"/><meta content="IMDb" property="og:site_name"/><meta content="IMDb Top 250 Movies" property="og:title"/><
meta content="IMDb Top 250 as rated by regular IMDb voters" property="og:description"/><meta content="website" property="og:
type"/><meta content="https://m.media-amazon.com/images/G/01/imdb/images/social/imdb_logo.png" property="og:image"/><meta co
ntent="1000" property="og:image:height"/><meta content="1000" property="og:image:width"/><meta content="en_US" property="og:
locale"/><meta content="es_ES" property="og:locale:alternate"/><meta content="es_MX" property="og:locale:alternate"/><meta c
```

Fetch links as list of tag objects

In [15]:
```
#Since class is a special keyword in python,i used class_ instead class

movies = soup.find('ul',  attrs={'class': 'ipc-metadata-list ipc-metadata-list--dividers-between sc-3a353071-0 wTPeg compact-list
```

In [16]:
```
for movie in movies:

    #Rank and name of movie
    rank_name = movie.find('h3', attrs={'class': 'ipc-title__text'}).text

    #Rank and name of movie after split
    rank_name_split = movie.find('h3', attrs={'class': 'ipc-title__text'}).text.split('.')

    #Rank of the movie
    rank = movie.find('h3', attrs={'class': 'ipc-title__text'}).text.split('.')[0]

    #Year of the movie
    year = movie.find('span', attrs={'class': 'sc-14dd939d-6 kHVqMR cli-title-metadata-item'}).text

    #Rating of the movie
    rating = movie.find('span', attrs={'class': 'ipc-rating-star ipc-rating-star--base ipc-rating-star--imdb ratingGroup--imdb-ra



    print(rank_name, year, rating)
```

```
1. The Shawshank Redemption 1994 9.3
2. The Godfather 1972 9.2
3. The Dark Knight 2008 9.0
4. The Godfather Part II 1974 9.0
5. 12 Angry Men 1957 9.0
6. Schindler's List 1993 9.0
7. The Lord of the Rings: The Return of the King 2003 9.0
8. Pulp Fiction 1994 8.9
9. The Lord of the Rings: The Fellowship of the Ring 2001 8.8
10. The Good, the Bad and the Ugly 1966 8.8
11. Forrest Gump 1994 8.8
12. Fight Club 1999 8.8
13. The Lord of the Rings: The Two Towers 2002 8.8
14. Spider-Man: Across the Spider-Verse 2023 8.9
15. Inception 2010 8.8
16. Star Wars: Episode V - The Empire Strikes Back 1980 8.7
17. The Matrix 1999 8.7
18. Goodfellas 1990 8.7
19. One Flew Over the Cuckoo's Nest 1975 8.7
```

In [ ]: