In [1]:
```
import requests
import time
import random
import selenium
from selenium import webdriver
from selenium.webdriver.chrome.service import Service as ChromeService
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By
from time import sleep
from random import randint
options = webdriver.ChromeOptions()
#options.headless = True --> deprecated
driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()))

import pandas as pd
```

In [2]:
```
urlList = ['avengers_infinity_war','spider_man_across_the_spider_verse','top_gun_ma
#You can manually put movies into the URL list or you can have a webscraper pull mo
#Please make sure to follow roten tomatoes formatting, all spaces replaced with _ a
#the movies are separated by year. Ex: smile_2022

#Also, shows have a different URL, so be sure to customize per your needs
for url in urlList:
    finalUrl = 'https://www.rottentomatoes.com/m/'+url+'/reviews?type=user&intcmp=r
    data = []
    stars = []
    rmList = []
    df = pd.DataFrame()
    driver.get(finalUrl)
    #Will run for 50 pages
    for i in range(0,50) :
        rmList = []
        #Find star section, count how many full stars and half star in each section
        starSection = driver.find_elements(By.CLASS_NAME, 'star-display')
        for section in starSection:
            count = 0.0
            fullStars = section.find_elements(By.CLASS_NAME, 'star-display__filled'
            for fullStar in fullStars:
                count +=1.0
            halfStars = section.find_elements(By.CLASS_NAME, 'star-display__half')
            for halfStar in halfStars:
                count +=0.5
            stars.append(count)
        sleep(randint(2,10))
        #Find review element, then append each review to the data frame
        elements = driver.find_elements(By.CLASS_NAME, 'audience-reviews__review')
        index = 0
        for title in elements:
            data.append(title.text)

        sleep(randint(2,10))
        #Keep a time difference between each parse to avoid overloading rotten toma
        #The scraper SHOULD NOT interfere with the website at all
```

```
        #This part will click the "next" button
        driver.find_element("xpath", '/html/body/div[3]/main/div/div/section/div/di
        sleep(10)
    finalData = {'Review':data,'Stars':stars}
    df = pd.DataFrame(finalData)
    #df = df.rename(columns={0:'Review', 1:'Stars'})
    #Send the data frame to a CSV file that we can edit
    df.to_csv('reviewsRated.csv', mode='a', index=False, header=False)
    sleep(randint(2,10))
```

In [3]:
```
driver.close()
```

In [4]:
```
driver.quit()
```