```
pip install opencv-python
```

```
Requirement already satisfied: opencv-python in c:\users\xander sam\anaconda3\lib\site-packages (4.9.0.80)Note: you may need to restart the kernel

Requirement already satisfied: numpy>=1.17.3 in c:\users\xander sam\anaconda3\lib\site-packages (from opencv-python) (1.23.5)
```

```
import cv2

key = cv2.waitKey(1)
webcam = cv2.VideoCapture(0)
while True:
    try:
        check, frame = webcam.read()
        print(check)  # prints true as long as the webcam is running
        print(frame)  # prints matrix values of each frame
        cv2.imshow("Capturing", frame)
        key = cv2.waitKey(1)
        if key == ord('s'):
            cv2.imwrite(filename='saved_img.jpg', img=frame)
            webcam.release()
            img_new = cv2.imread('saved_img.jpg', cv2.IMREAD_GRAYSCALE)
            cv2.imshow("Captured Image", img_new)
            cv2.waitKey(1650)
            cv2.destroyAllWindows()
            print("Processing Image.....")
            img_ = cv2.imread('saved_img.jpg', cv2.IMREAD_ANYCOLOR)
            print("Converting RGB IMAGE to grayscale.....")
            gray = cv2.cvtColor(img_, cv2.COLOR_BGR2GRAY)
            print("Converted RGB image to grayscale.....")
            print("Resizing image to 28x28 pixels.....")
            img_resized = cv2.resize(gray, (28, 28))
            print("Resized.......")
            cv2.imwrite(filename='saved_img-final.jpg', img=img_resized)
            print("Image saved!!!")
            break
        elif key == ord('q'):
            print("Turn off the camera.")
            webcam.release()
            print("Camera off.")
            print("Program Ended.")
            cv2.destroyAllWindows()
            break
    except KeyboardInterrupt:
        print("Turning off the camera.")
        webcam.release()
        print("Camera off.")
        print("Program Ended.")
        cv2.destroyAllWindows()
        break
```

```
       ...
       [12  5  8]
       [12  5  8]
       [12  5  8]]

      [[19 11 18]
       [19 11 18]
       [19 11 18]
       ...
       [12  5  8]
       [12  5  8]
       [12  5  8]]

      ...

      [[23 19 38]
       [22 18 37]
       [21 17 36]
       ...
       [17 14 23]
       [17 14 23]
```

```
!pip3 install sounddevice
```

```
     Requirement already satisfied: sounddevice in c:\users\xander sam\anaconda3\lib\site-packages (0.4.6)
     Requirement already satisfied: CFFI>=1.0 in c:\users\xander sam\anaconda3\lib\site-packages (from sounddevice) (1.15.1)
     Requirement already satisfied: pycparser in c:\users\xander sam\anaconda3\lib\site-packages (from CFFI>=1.0->sounddevice) (2.21)
```

```
!pip3 install wavio
```

```
     Requirement already satisfied: wavio in c:\users\xander sam\anaconda3\lib\site-packages (0.0.8)
     Requirement already satisfied: numpy>=1.19.0 in c:\users\xander sam\anaconda3\lib\site-packages (from wavio) (1.23.5)
```

```
!pip3 install scipy
```

```
     Requirement already satisfied: scipy in c:\users\xander sam\anaconda3\lib\site-packages (1.10.0)
     Requirement already satisfied: numpy<1.27.0,>=1.19.5 in c:\users\xander sam\anaconda3\lib\site-packages (from scipy) (1.23.5)
```

```
!apt-get install libportaudio2
```

```
     'apt-get' is not recognized as an internal or external command,
     operable program or batch file.
```

```
import sounddevice as sd
from scipy.io.wavfile import write
import wavio as wv

#Sampling Freq
freq = 44100

#Recording Duration
duration = 5

#Start record with the given Values
#of duration and sample frequency
recording = sd.rec(int(duration * freq),
                   samplerate = freq, channels = 2)

#Record Audio for the given number of seconds
sd.wait()

#This will covner the NumPY array to an audio
#file with the given sampling frequency
write("recording012.wav", freq, recording)

#COnvert the numPY array to audio file
wv.write("recording123.wav", recording, freq, sampwidth=2)
```

```
##WEBSCRAPING
```

```
!pip install bs4
```

```
     Requirement already satisfied: bs4 in c:\users\xander sam\anaconda3\lib\site-packages (0.0.2)
     Requirement already satisfied: beautifulsoup4 in c:\users\xander sam\anaconda3\lib\site-packages (from bs4) (4.11.1)
     Requirement already satisfied: soupsieve>1.2 in c:\users\xander sam\anaconda3\lib\site-packages (from beautifulsoup4->bs4) (2.3.2.post1)
```

```
pip install request
```

```
     Note: you may need to restart the kernel to use updated packages.
     ERROR: Could not find a version that satisfies the requirement request (from versions: none)
     ERROR: No matching distribution found for request
```

```
import requests
from bs4 import BeautifulSoup

def getdata(url):
    r = requests.get(url)
    return r.text

htmldata = getdata("https://www.google.com/")
soup = BeautifulSoup(htmldata, 'html.parser')
for item in soup.find_all('img'):
    print(item['src'])
```

```
 /images/branding/googlelogo/1x/googlelogo_white_background_color_272x92dp.png
```

```
pip install selenium
```

```
 Requirement already satisfied: selenium in c:\users\xander sam\anaconda3\lib\site-packages (4.18.1)
 Requirement already satisfied: trio-websocket~=0.9 in c:\users\xander sam\anaconda3\lib\site-packages (from selenium) (0.11.1)
 Requirement already satisfied: certifi>=2021.10.8 in c:\users\xander sam\anaconda3\lib\site-packages (from selenium) (2022.12.7)
 Requirement already satisfied: urllib3[socks]<3,>=1.26 in c:\users\xander sam\anaconda3\lib\site-packages (from selenium) (1.26.14)
 Requirement already satisfied: trio~=0.17 in c:\users\xander sam\anaconda3\lib\site-packages (from selenium) (0.25.0)
 Requirement already satisfied: typing_extensions>=4.9.0 in c:\users\xander sam\anaconda3\lib\site-packages (from selenium) (4.10.0)
 Requirement already satisfied: sniffio>=1.3.0 in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.3.1)
 Requirement already satisfied: cffi>=1.14 in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.15.1)
 Requirement already satisfied: attrs>=23.2.0 in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (23.2.0)
 Requirement already satisfied: sortedcontainers in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (2.4.0)
 Requirement already satisfied: idna in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (3.4)
 Requirement already satisfied: outcome in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.3.0.post0)
 Requirement already satisfied: exceptiongroup in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.2.0)
 Requirement already satisfied: wsproto>=0.14 in c:\users\xander sam\anaconda3\lib\site-packages (from trio-websocket~=0.9->selenium) (1.2.0)
 Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\users\xander sam\anaconda3\lib\site-packages (from urllib3[socks]<3,>=1.26->selen
 Requirement already satisfied: pycparser in c:\users\xander sam\anaconda3\lib\site-packages (from cffi>=1.14->trio~=0.17->selenium) (2.21)
 Requirement already satisfied: h11<1,>=0.9.0 in c:\users\xander sam\anaconda3\lib\site-packages (from wsproto>=0.14->trio-websocket~=0.9->selenium
 Note: you may need to restart the kernel to use updated packages.
```

##Image Scraipng using Selenium

```
!pip install selenium
!apt-get update # to update ubuntu to correctly run apt install
!apt install chromium-chromedriver
!cp /usr/lib/chromium-browser/chromedriver /usr/bin

import sys
sys.path.insert(0,'/usr/lib/chromium-browser/chromedriver')

from selenium import webdriver
import time
import requests
import shutil
import os
import getpass
import urllib.request
import io
import time
from PIL import image

user = getpass.getuser()
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument('--headless')
chrome_options.add_argument('--no-sandbox')
chrome_options.add.argument('--disable-dev-shm-usage')
driver = webdriver.Chrome('chromedriver', chrome_options = chrome_options)

search_url = "https://www.google.com/search?q={q}&tbm=isch&tbs=sur%3Afc&hl=en&ved=0CAIQpwVqFwoTCKCa1c6s4-oCFQAAAAAdAAAAABAC&biw=1251&bih=568"
driver.get(search_url.format(q='Car'))

def scroll_to_end(driver):
    driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
    time.sleep(5)#Sleep_between_interactions

def getImageUrls(name,totalImgs,driver):

    search_url = "https://www.google.com/search?q={q}&tbm=isch&tbs=sur%3Afc&hl=en&ved=0CAIQpwVqFwoTCKCa1c6s4-oCFQAAAAAdAAAAABAC&biw=1251&bih=568"
    driver.get(search_url.format(q=name))
    img_urls = set()
    img_count = 0
    results_start = 0

    while(img_count<totalImgs): #EXTRACT_actual_image-now

        scroll_to_end(driver)

        thumbnail_results = driver.find_elements_by_xpath("//img[contains(@class,'Q4LuWd')]")
        totalResults=len(thumbnail_results)
        print(f"Found: {totalResults} search results. Extracting links from{results_start}:{totalResults}")

        for img in thumbnail_results[results_start:totalResults]:

            img.click()
            time.sleep(2)
            actual_images = driver.find_elements_by_css_selector('img.n3VNCb')
            for actual_image in actual_images:
                if actual_image.get_attribute('src') and 'https' in actual_image.get_attribute('src'):
                    img_urls.add(actual_image.get_attribute('src'))

            img_count = len(img_urls)

            if img_count >= totalImgs:
                print(f"Found: {img_count} image links")
                break
            else:
                print("Found:", img_count, "looking for more image links.....")
                load_more_button = driver.find_element_by_css_selector(".mye4qd")
                driver.execute_script("document.querySelector('.myeqd').click();")
                results_start = len(thumbnail-results)
return img_ruls

def downloadImages(folder_path,file_name,url):
    try:
        image_content = requests.get(url).content
    except Exception as e:
        print(f"ERROR - COULD NOT DOWNLOAD {url} - {e}")
    try:
        image_file = io.BytesIO(image_content)
        image = Image.open(image_file).convert('RGB')

        file_path = os.path.join(folder_path, file_name)

        with open(file_path, 'wb') as f:
            image.save(f, "JPEG", quality = 85)
        print(f"SAVED - {url} - AT: {file_path}")
    except Exception as e:
        print(f"ERROR - COULD NOT SAVE {url} - {e}")

def saveInDestFolder(searchNames,destDir,totalImgs,driver):
```

```
    for name in list(searchNames):
        path=os.path.join(destDir,name)
        if not os.path.isdir(path):
            os.mkdir(path)
        print('Current Path',path)
        totalLinks=getImageUrls(name,totalImgs,driver)
        print('totalLinks', totalLinks)

    if totalLinks is None:
        print('image not found for: ', name)

    else:
        for i, link in enumerate(totalLinks):
            file_name = f"{i:150}.jpg"
            downloadImages(path,file_name,link)

searchNames = ['cat']
destDir = f'/content/driver/My Drive/ Colab Notebooks/Dataset/'
totalImgs=5

saveInDestFolder(searchNames,destDir, totalImgs, driver)
```

```
Requirement already satisfied: selenium in c:\users\xander sam\anaconda3\lib\site-packages (4.18.1)
Requirement already satisfied: trio-websocket~=0.9 in c:\users\xander sam\anaconda3\lib\site-packages (from selenium) (0.11.1)
Requirement already satisfied: urllib3[socks]<3,>=1.26 in c:\users\xander sam\anaconda3\lib\site-packages (from selenium) (1.26.14)
Requirement already satisfied: typing_extensions>=4.9.0 in c:\users\xander sam\anaconda3\lib\site-packages (from selenium) (4.10.0)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\xander sam\anaconda3\lib\site-packages (from selenium) (2022.12.7)
Requirement already satisfied: trio~=0.17 in c:\users\xander sam\anaconda3\lib\site-packages (from selenium) (0.25.0)
Requirement already satisfied: exceptiongroup in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.2.0)
Requirement already satisfied: outcome in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.3.0.post0)
Requirement already satisfied: sortedcontainers in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: sniffio>=1.3.0 in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.3.1)
Requirement already satisfied: cffi>=1.14 in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.15.1)
Requirement already satisfied: idna in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (3.4)
Requirement already satisfied: attrs>=23.2.0 in c:\users\xander sam\anaconda3\lib\site-packages (from trio~=0.17->selenium) (23.2.0)
Requirement already satisfied: wsproto>=0.14 in c:\users\xander sam\anaconda3\lib\site-packages (from trio-websocket~=0.9->selenium) (1.2.0)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\users\xander sam\anaconda3\lib\site-packages (from urllib3[socks]<3,>=1.26->selen
Requirement already satisfied: pycparser in c:\users\xander sam\anaconda3\lib\site-packages (from cffi>=1.14->trio~=0.17->selenium) (2.21)
Requirement already satisfied: h11<1,>=0.9.0 in c:\users\xander sam\anaconda3\lib\site-packages (from wsproto>=0.14->trio-websocket~=0.9->selenium
'apt-get' is not recognized as an internal or external command,
operable program or batch file.
'apt' is not recognized as an internal or external command,
operable program or batch file.
'cp' is not recognized as an internal or external command,
operable program or batch file.
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
Cell In[6], line 18
     16 import io
     17 import time
---> 18 from PIL import image
     20 user = getpass.getuser()
     21 chrome_options = webdriver.ChromeOptions()

ImportError: cannot import name 'image' from 'PIL' (C:\Users\Xander Sam\anaconda3\lib\site-packages\PIL\__init__.py)
```

```
from requests import get
url = 'https://www.imdb.com/search/title?release_date=2017&sort=num_votes,desc&page=1'
agent = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.186 Safari/537.36'
}
response = get(url, headers = agent)
print(response.text)
```

```
<!DOCTYPE html><html lang="en-US" xmlns:og="http://opengraphprotocol.org/schema/" xmlns:fb="http://www.facebook.com/2008/fbml"><head><meta charS
        if (typeof window.csa !== 'undefined' && typeof window.csa === 'function') {
            var csaLatencyPlugin = window.csa('Content', {
                element: {
                    slotId: 'LoadTitle',
                    type: 'service-call'
                }
            });
            csaLatencyPlugin('mark', 'clickToBodyBegin', 1711294540799);
        }
    })</script><title>Advanced search</title><meta name="description" content="" data-id="main"/><meta property="og:url" content="https://www.im
        if (typeof window.csa !== 'undefined' && typeof window.csa === 'function') {
            var csaLatencyPlugin = window.csa('Content', {
                element: {
                    slotId: 'LoadTitle',
                    type: 'service-call'
                }
            });
            csaLatencyPlugin('mark', 'clickToBodyEnd', 1711294540799);
        }
    })</script><script>if(typeof uex === 'function'){ uex('ld', 'LoadTitle', {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', 'clickToLoaded', 1711294540799);
        }
    })</script><meta name="next-head-count" content="36"/><script>
var ue_t0=ue_t0||+new Date();

window.ue_ihb = (window.ue_ihb || window.ueinit || 0) + 1;
if (window.ue_ihb === 1) {

var ue_csm = window,
    ue_hob = +new Date();
(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 ue_err_chan = 'jserr';
(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&&
pec:0,ts:0,erl:[],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


var ue_id = 'MRCT3QSFZ9H6H179QVMV',
    ue_url,
    ue_navtiming = 1,
    ue_mid = 'A1EVAM02EL8SFB',
    ue_sid = '146-8432665-9574439',
    ue_sn = 'www.imdb.com',
    ue_furl = 'fls-na.amazon.com',
    ue_surl = 'https://unagi-na.amazon.com/1/events/com.amazon.csm.nexusclient.prod',
    ue_int = 0,
    ue_fcsn = 1,
    ue_urt = 3,


from bs4 import BeautifulSoup
html_soup = BeautifulSoup(response.text, 'html.parser')
headers = {'Accept-Language': 'en-US,en;q=0.8'}
type(html_soup)


    bs4.BeautifulSoup


movie_containers = html_soup.find_all('div', class_ = 'ipc-metadata-list-summary-item__c')
print(type(movie_containers))
print(len(movie_containers))


    <class 'bs4.element.ResultSet'>
    50


first_movie = movie_containers[0]
first_movie


    <div class="ipc-metadata-list-summary-item__c"><div class="ipc-metadata-list-summary-item__tc"><span aria-disabled="false" class="ipc-metadata-
    list-summary-item__t"></span><div class="sc-ab6fa25a-3 bVYfLY dli-parent"><div class="sc-ab6fa25a-2 gOsifL"><div class="sc-e5a25b0f-0 jQjDIb dli-
    poster-container"><div class="ipc-poster ipc-poster--base ipc-poster--dynamic-width ipc-sub-grid-item ipc-sub-grid-item--span-2" role="group">
    <div aria-label="add to watchlist" class="ipc-watchlist-ribbon ipc-focusable ipc-watchlist-ribbon--s ipc-watchlist-ribbon--base ipc-watchlist-
    ribbon--loading ipc-watchlist-ribbon--onImage ipc-poster__watchlist-ribbon" role="button" tabindex="0"><svg class="ipc-watchlist-ribbon__bg"
    height="34px" role="presentation" viewbox="0 0 24 34" width="24px" xmlns="http://www.w3.org/2000/svg"><polygon class="ipc-watchlist-ribbon__bg-
    ribbon" fill="#000000" points="24 0 0 0 0 32 12.2436611 26.2926049 24 31.7728343"></polygon><polygon class="ipc-watchlist-ribbon__bg-hover"
    points="24 0 0 0 0 32 12.2436611 26.2926049 24 31.7728343"></polygon><polygon class="ipc-watchlist-ribbon__bg-shadow" points="24 31.7728343 24
    33.7728343 12.2436611 28.2926049 0 34 0 32 12.2436611 26.2926049"></polygon></svg><div class="ipc-watchlist-ribbon__icon" role="presentation">
    <svg class="ipc-loader ipc-loader--circle ipc-watchlist-ribbon__loader" data-testid="watchlist-ribbon-loader" height="48px" role="presentation"
    version="1.1" viewbox="0 0 48 48" width="48px" xmlns="http://www.w3.org/2000/svg"><g class="ipc-loader__container" fill="currentColor"><circle
    class="ipc-loader__circle ipc-loader__circle--one" cx="24" cy="9" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--two"
    cx="35" cy="14" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--three" cx="39" cy="24" r="4"></circle><circle class="ipc-
    loader__circle ipc-loader__circle--four" cx="35" cy="34" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--five" cx="24"
    cy="39" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--six" cx="13" cy="34" r="4"></circle><circle class="ipc-
    loader__circle ipc-loader__circle--seven" cx="9" cy="24" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--eight" cx="13"
    cy="14" r="4"></circle></g></svg></div></div><div class="ipc-media ipc-media--poster-27x40 ipc-image-media-ratio--poster-27x40 ipc-media--base
    ipc-media--poster-m ipc-poster__poster-image ipc-media__img" style="width:100%"><img alt="Hugh Jackman in Logan (2017)" class="ipc-image"
    loading="lazy" sizes="50vw, (min-width: 480px) 34vw, (min-width: 600px) 26vw, (min-width: 1024px) 16vw, (min-width: 1280px) 16vw"
    src="https://m.media-
    amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX140_CR0,1,140,207_.jpg"
    srcset="https://m.media-
    amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX140_CR0,1,140,207_.jpg 140w,
    https://m.media-
    amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX210_CR0,2,210,311_.jpg 210w,
    https://m.media-
    amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX280_CR0,3,280,414_.jpg 280w"
    width="140"/></div><a aria-label="View title page for Logan" class="ipc-lockup-overlay ipc-focusable" href="/title/tt3315342/?ref_=sr_i_1"><div
    class="ipc-lockup-overlay__screen"></div></a></div></div><div class="sc-b0691f29-0 jbYPfh"><div class="ipc-title ipc-title--base ipc-title--title
    ipc-title-link-no-icon ipc-title--on-textPrimary sc-b0691f29-9 klOwFB dli-title"><a class="ipc-title-link-wrapper" href="/title/tt3315342/?
    ref_=sr_t_1" tabindex="0"><h3 class="ipc-title__text">1. Logan</h3></a></div><div class="sc-b0691f29-7 hrgukm dli-title-metadata"><span
    class="sc-b0691f29-8 ilsLEX dli-title-metadata-item">2017</span><span class="sc-b0691f29-8 ilsLEX dli-title-metadata-item">2h 17m</span><span
    class="sc-b0691f29-8 ilsLEX dli-title-metadata-item">R-16</span></div><span class="sc-b0691f29-1 grHDBY"><div class="sc-e2dbc1a3-0 ajrIH sc-
    b0691f29-2 bhhtyj dli-ratings-container" data-testid="ratingGroup--container"><span aria-label="IMDb rating: 8.1" class="ipc-rating-star ipc-
    rating-star--base ipc-rating-star--imdb ratingGroup--imdb-rating" data-testid="ratingGroup--imdb-rating"><svg class="ipc-icon ipc-icon--star-
    inline" fill="currentColor" height="24" role="presentation" viewbox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M12
    20.1l5.82 3.682c1.066.675 2.37-.322 2.09-1.584l-1.543-6.926 5.146-4.667c.94-.85.435-2.465-.799-2.567l-6.773-.602L13.29.89a1.38 1.38 0 0 0-2.581
    0l-2.65 6.53-6.774.602C.052 8.126-.453 9.74.486 10.59l5.147 4.666-1.542 6.926c-.28 1.262 1.023 2.26 2.09 1.585L12 20.099z"></path></svg>8.1<span
    class="ipc-rating-star--voteCount"> (<!-- -->827K<!-- -->)</span></span><button aria-label="Rate Logan" class="ipc-rate-button sc-e2dbc1a3-1
```

```
jboOQc ratingGroup--user-rating ipc-rate-button--unrated ipc-rate-button--base" data-testid="rate-button"><span class="ipc-rating-star ipc-rating-star--base ipc-rating-star--rate"><svg class="ipc-icon ipc-icon--star-border-inline" fill="currentColor" height="24" role="presentation" viewbox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M22.724 8.217l-6.786-.587-2.65-6.22c-.477-1.133-2.103-1.133-2.58 0l-2.65 6.234-6.772.573c-1.234.098-1.739 1.636-.8 2.446l5.146 4.446-1.542 6.598c-.28 1.202 1.023 2.153 2.09 1.51l5.818-3.495 5.819 3.509c1.065.643 2.37-.308 2.089-1.51l-1.542-6.612 5.145-4.446c.94-.81.45-2.348-.785-2.446zm-10.726 8.891-5.272 3.174 1.402-5.983-4.655-4.026 6.141-.531 2.384-5.634 2.398 5.648 6.14.531-4.654 4.026 1.402 5.983-5.286-3.187z"></path></svg><span class="ipc-rating-star--rate">Rate</span></span></button></div><span class="sc-b0691f29-11 TmkKM"><span class="sc-b0901df4-0 bcQdDJ metacritic-score-box" style="background-color:#54A72A">77</span><span class="metacritic-score-label">Metascore</span></span></div><div class="sc-ab6fa25a-4 ggHbBR dli-post-element"><button aria-disabled="false" aria-label="See more information about Logan" class="ipc-icon-button dli-info-icon ipc-icon-button--base ipc-icon-button--onAccent2" role="button" tabindex="0" title="See more information about Logan"><svg class="ipc-icon ipc-icon--info" fill="currentColor" height="24" role="presentation" viewbox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M0 0h24v24H0V0z" fill="none"></path><path d="M11 7h2v2h-2zm0 4h2v6h-2zm1-9C6.48 2 2 6.48 2 12s4.48 10 10 10 10-4.48 10-10S17.52 2 12 2zm0 18c-4.41 0-8-3.59-8-8s3.59-8 8-8 8 3.59 8 8-3.59 8-8 8z"></path></svg></button></div></div><div class="sc-ab6fa25a-1 bBwFsP"><div class="ipc-html-content ipc-html-content--base sc-ab6fa25a-0 bhexuD dli-plot-container" role="presentation"><div class="ipc-html-content-inner-div">In a future where mutants are nearly extinct, an elderly and weary Logan leads a quiet life. But when Laura, a mutant child pursued by scientists, comes to him for help, he must get her to safety.</div></div></div></div></div></div>
```


first_movie.div


```
<div class="ipc-metadata-list-summary-item__tc"><span aria-disabled="false" class="ipc-metadata-list-summary-item__t"></span><div class="sc-ab6fa25a-3 bVYfLY dli-parent"><div class="sc-ab6fa25a-2 gOsifL"><div class="sc-e5a25b0f-0 jQjDIb dli-poster-container"><div class="ipc-poster ipc-poster--base ipc-poster--dynamic-width ipc-sub-grid-item ipc-sub-grid-item--span-2" role="group"><div aria-label="add to watchlist" class="ipc-watchlist-ribbon ipc-focusable ipc-watchlist-ribbon--s ipc-watchlist-ribbon--base ipc-watchlist-ribbon--loading ipc-watchlist-ribbon--onImage ipc-poster__watchlist-ribbon" role="button" tabindex="0"><svg class="ipc-watchlist-ribbon__bg" height="34px" role="presentation" viewbox="0 0 24 34" width="24px" xmlns="http://www.w3.org/2000/svg"><polygon class="ipc-watchlist-ribbon__bg-ribbon" fill="#000000" points="24 0 0 0 0 32 12.2436611 26.2926049 24 31.7728343"></polygon><polygon class="ipc-watchlist-ribbon__bg-hover" points="24 0 0 0 0 32 12.2436611 26.2926049 24 31.7728343"></polygon><polygon class="ipc-watchlist-ribbon__bg-shadow" points="24 31.7728343 24 33.7728343 12.2436611 28.2926049 0 34 0 32 12.2436611 26.2926049"></polygon></svg><div class="ipc-watchlist-ribbon__icon" role="presentation"><svg class="ipc-loader ipc-loader--circle ipc-watchlist-ribbon__loader" data-testid="watchlist-ribbon-loader" height="48px" role="presentation" version="1.1" viewbox="0 0 48 48" width="48px" xmlns="http://www.w3.org/2000/svg"><g class="ipc-loader__container" fill="currentColor"><circle class="ipc-loader__circle ipc-loader__circle--one" cx="24" cy="9" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--two" cx="35" cy="14" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--three" cx="39" cy="24" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--four" cx="35" cy="34" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--five" cx="24" cy="39" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--six" cx="13" cy="34" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--seven" cx="9" cy="24" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--eight" cx="13" cy="14" r="4"></circle></g></svg></div></div><div class="ipc-media ipc-media--poster-27x40 ipc-image-media-ratio--poster-27x40 ipc-media--base ipc-media--poster-m ipc-poster__poster-image ipc-media__img" style="width:100%"><img alt="Hugh Jackman in Logan (2017)" class="ipc-image" loading="lazy" sizes="50vw, (min-width: 480px) 34vw, (min-width: 600px) 26vw, (min-width: 1024px) 16vw, (min-width: 1280px) 16vw" src="https://m.media-amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX140_CR0,1,140,207_.jpg" srcset="https://m.media-amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX140_CR0,1,140,207_.jpg 140w, https://m.media-amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX210_CR0,2,210,311_.jpg 210w, https://m.media-amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX280_CR0,3,280,414_.jpg 280w" width="140"/></div><a aria-label="View title page for Logan" class="ipc-lockup-overlay ipc-focusable" href="/title/tt3315342/?ref_=sr_i_1"><div class="ipc-lockup-overlay__screen"></div></a></div></div><div class="sc-b0691f29-0 jbYPfh"><div class="ipc-title ipc-title--base ipc-title--title ipc-title-link-no-icon ipc-title--on-textPrimary sc-b0691f29-9 klOwFB dli-title"><a class="ipc-title-link-wrapper" href="/title/tt3315342/?ref_=sr_t_1" tabindex="0"><h3 class="ipc-title__text">1. Logan</h3></a></div><div class="sc-b0691f29-7 hrgukm dli-title-metadata"><span class="sc-b0691f29-8 ilsLEX dli-title-metadata-item">2017</span><span class="sc-b0691f29-8 ilsLEX dli-title-metadata-item">2h 17m</span><span class="sc-b0691f29-8 ilsLEX dli-title-metadata-item">R-16</span></div><span class="sc-b0691f29-1 grHDBY"><div class="sc-e2dbc1a3-0 ajrIH sc-b0691f29-2 bhhtyj dli-ratings-container" data-testid="ratingGroup--container"><span aria-label="IMDb rating: 8.1" class="ipc-rating-star ipc-rating-star--base ipc-rating-star--imdb ratingGroup--imdb-rating" data-testid="ratingGroup--imdb-rating"><svg class="ipc-icon ipc-icon--star-inline" fill="currentColor" height="24" role="presentation" viewbox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M12 20.1l5.82 3.682c1.066.675 2.37-.322 2.09-1.584l-1.543-6.926 5.146-4.667c.94-.85.435-2.465-.799-2.567l-6.773-.602L13.29 1.38 1.38 0 0 0-2.581 01-2.65 6.53-6.774.602C.052 8.126-.453 9.74.486 10.59l5.147 4.666-1.542 6.926c-.28 1.262 1.023 2.26 2.09 1.585L12 20.099z"></path></svg>8.1<span class="ipc-rating-star--voteCount"> (<!-- -->827K<!-- -->)</span></span><button aria-label="Rate Logan" class="ipc-rate-button sc-e2dbc1a3-1 jboOQc ratingGroup--user-rating ipc-rate-button--unrated ipc-rate-button--base" data-testid="rate-button"><span class="ipc-rating-star ipc-rating-star--base ipc-rating-star--rate"><svg class="ipc-icon ipc-icon--star-border-inline" fill="currentColor" height="24" role="presentation" viewbox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M22.724 8.217l-6.786-.587-2.65-6.22c-.477-1.133-2.103-1.133-2.58 0l-2.65 6.234-6.772.573c-1.234.098-1.739 1.636-.8 2.446l5.146 4.446-1.542 6.598c-.28 1.202 1.023 2.153 2.09 1.51l5.818-3.495 5.819 3.509c1.065.643 2.37-.308 2.089-1.51l-1.542-6.612 5.145-4.446c.94-.81.45-2.348-.785-2.446zm-10.726 8.891-5.272 3.174 1.402-5.983-4.655-4.026 6.141-.531 2.384-5.634 2.398 5.648 6.14.531-4.654 4.026 1.402 5.983-5.286-3.187z"></path></svg><span class="ipc-rating-star--rate">Rate</span></span></button></div><span class="sc-b0691f29-11 TmkKM"><span class="sc-b0901df4-0 bcQdDJ metacritic-score-box" style="background-color:#54A72A">77</span><span class="metacritic-score-label">Metascore</span></span></div><div class="sc-ab6fa25a-4 ggHbBR dli-post-element"><button aria-disabled="false" aria-label="See more information about Logan" class="ipc-icon-button dli-info-icon ipc-icon-button--base ipc-icon-button--onAccent2" role="button" tabindex="0" title="See more information about Logan"><svg class="ipc-icon ipc-icon--info" fill="currentColor" height="24" role="presentation" viewbox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M0 0h24v24H0V0z" fill="none"></path><path d="M11 7h2v2h-2zm0 4h2v6h-2zm1-9C6.48 2 2 6.48 2 12s4.48 10 10 10 10-4.48 10-10S17.52 2 12 2zm0 18c-4.41 0-8-3.59-8-8s3.59-8 8-8 8 3.59 8 8-3.59 8-8 8z"></path></svg></button></div></div><div class="sc-ab6fa25a-1 bBwFsP"><div class="ipc-html-content ipc-html-content--base sc-ab6fa25a-0 bhexuD dli-plot-container" role="presentation"><div class="ipc-html-content-inner-div">In a future where mutants are nearly extinct, an elderly and weary Logan leads a quiet life. But when Laura, a mutant child pursued by scientists, comes to him for help, he must get her to safety.</div></div></div></div></div></div>
```


first_movie.a


```
<a aria-label="View title page for Logan" class="ipc-lockup-overlay ipc-focusable" href="/title/tt3315342/?ref_=sr_i_1"><div class="ipc-lockup-overlay__screen"></div></a>
```


first_movie.h3


```
<h3 class="ipc-title__text">1. Logan</h3>
```


first_movie.h3.a

Movie Name of the first

```
first_name = first_movie.find('h3', class_="ipc-title__text").text
first_name
```

```
'1. Logan'
```

Year of the First Movie

```
first_year = first_movie.find('span',class_="sc-b0691f29-8 ilsLEX dli-title-metadata-item").text
first_year
```

```
'2017'
```

First Movie Ratings

```
first_movie_rate = first_movie.find('span',class_="ipc-rating-star ipc-rating-star--base ipc-rating-star--imdb ratingGroup--imdb-rating").text[:3]
first_movie_rate
```

```
'8.1'
```

First Movie meta score

```
first_metscore = first_movie.find('span', class_="sc-b0901df4-0 bcQdDJ metacritic-score-box").text
first_metscore
```

```
'77'
```

First Movie Votes

```
fvote = first_movie.find('span', class_="ipc-rating-star--voteCount").text[2:6]
fvote
```

```
'827K'
```

The Script

```
#Lists to store the scraped data in
names = []
years = []
imdb_ratings = []
metascores = []
votes = []

for container in movie_containers:
# If the movie has Metascore, then extract:
    if container.find('span', class_ = 'sc-b0901df4-0 bcQdDJ metacritic-score-box') is not None:
# The name
        name = container.find('h3', class_ = 'ipc-title__text' ).text
        names.append(name)
# The year
        year = container.find('span', class_ = 'sc-b0691f29-8 ilsLEX dli-title-metadata-item').text
        years.append(year)
# The IMDB rating
        imdb = float(container.find('span', class_='ipc-rating-star ipc-rating-star--base ipc-rating-star--imdb ratingGroup--imdb-rating').text[:3])
        imdb_ratings.append(imdb)
# The Metascore
        m_score = container.find('span', class_ = 'sc-b0901df4-0 bcQdDJ metacritic-score-box').text
        metascores.append(int(m_score))
# The number of votes
        vote = container.find('span', class_ = 'ipc-rating-star--voteCount').text[2:6]
        votes.append(vote)

print(names)
print(years)
print(imdb_ratings)
print(metascores)
print(votes)
```

```
['1. Logan', '2. Thor: Ragnarok', '3. Guardians of the Galaxy Vol. 2', '4. Dunkirk', '5. Spider-Man: Homecoming', '6. Wonder Woman', '7. Get Out',
['2017', '2017', '2017', '2017', '2017', '2017', '2017', '2017', '2017', '2017', '2017', '2017', '2017', '2017', '2017', '2017', '2017', '2017', '
[8.1, 7.9, 7.6, 7.8, 7.4, 7.3, 7.8, 6.9, 8.0, 7.5, 7.3, 8.4, 8.1, 7.4, 6.1, 7.3, 6.9, 6.7, 6.7, 6.5, 7.1, 7.4, 7.8, 7.5, 6.4, 6.5, 7.4, 7.7, 6.6,
[77, 74, 67, 94, 73, 76, 85, 84, 81, 86, 69, 81, 88, 75, 45, 87, 58, 44, 62, 39, 65, 93, 94, 48, 65, 52, 82, 73, 56, 54, 76, 47, 77, 41, 52, 75, 6
['827K', '813K', '756K', '736K', '716K', '698K', '691K', '670K', '658K', '605K', '603K', '586K', '553K', '509K', '477K', '446K', '436K', '361K', '
```

```
import pandas as pd
test_df = pd.DataFrame({'movie': names,
'year': years,
'imdb': imdb_ratings,
'metascore': metascores,
'votes': votes
})
print(test_df.info())
test_df
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41 entries, 0 to 40
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   movie      41 non-null     object
 1   year       41 non-null     object
 2   imdb       41 non-null     float64
 3   metascore  41 non-null     int64
 4   votes      41 non-null     object
dtypes: float64(1), int64(1), object(3)
memory usage: 1.7+ KB
None
```

	movie	year	imdb	metascore	votes
0	1. Logan	2017	8.1	77	827K
1	2. Thor: Ragnarok	2017	7.9	74	813K
2	3. Guardians of the Galaxy Vol. 2	2017	7.6	67	756K
3	4. Dunkirk	2017	7.8	94	736K
4	5. Spider-Man: Homecoming	2017	7.4	73	716K
5	6. Wonder Woman	2017	7.3	76	698K
6	7. Get Out	2017	7.8	85	691K
7	8. Star Wars: Episode VIII - The Last Jedi	2017	6.9	84	670K
8	9. Blade Runner 2049	2017	8.0	81	658K
9	10. Baby Driver	2017	7.5	86	605K
10	11. It	2017	7.3	69	603K
11	12. Coco	2017	8.4	81	586K
12	13. Three Billboards Outside Ebbing, Missouri	2017	8.1	88	553K
13	15. John Wick: Chapter 2	2017	7.4	75	509K
14	16. Justice League	2017	6.1	45	477K
15	17. The Shape of Water	2017	7.3	87	446K
16	19. Jumanji: Welcome to the Jungle	2017	6.9	58	436K
17	20. Kingsman: The Golden Circle	2017	6.7	44	361K
18	21. Kong: Skull Island	2017	6.7	62	345K
19	23. Pirates of the Caribbean: Salazar's Revenge	2017	6.5	39	344K
20	24. Beauty and the Beast	2017	7.1	65	333K
21	26. Lady Bird	2017	7.4	93	326K
22	28. Call Me by Your Name	2017	7.8	94	313K
23	29. The Greatest Showman	2017	7.5	48	310K
24	30. Alien: Covenant	2017	6.4	65	302K
25	31. Murder on the Orient Express	2017	6.5	52	295K
26	32. War for the Planet of the Apes	2017	7.4	82	280K
27	33. Wind River	2017	7.7	73	279K
28	36. Fast & Furious 8	2017	6.6	56	253K
29	37. Life	2017	6.6	54	252K
30	38. Mother!	2017	6.6	76	249K
31	39. The Hitman's Bodyguard	2017	6.9	47	246K
32	40. I, Tonya	2017	7.5	77	242K
33	41. King Arthur: Legend of the Sword	2017	6.7	41	232K
34	42. Ghost in the Shell	2017	6.3	52	227K
35	44. Darkest Hour	2017	7.4	75	220K
36	46. American Made	2017	7.1	65	207K
37	47. Atomic Blonde	2017	6.7	63	206K
38	48. The Mummy	2017	5.4	34	206K
39	49. Baywatch	2017	5.5	37	201K
40	50. Bright	2017	6.3	29	201K

```
from time import time
from time import sleep
from requests import get
from random import randint

from IPython.core.display import clear_output
pages = [ '1','2','3','4','5']
years_url = [ '2015','2016','2017', '2018', '2019', '2020']


# Redeclaring the lists to store data in
names = []
years = []
imdb_ratings = []
metascores = []
votes = []

# Preparing the monitoring of the loop
start_time = time()
requests = 0

# For every year in the interval 2015-2020
for year_url in years_url:


        # Make a get request
        url = f'https://www.imdb.com/search/title?release_date={year_url}-01-01,{year_url}-12-31&sort=num_votes,desc&page=1'
        agent = {"User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.186 Safari/537.36"}
        response = get(url,headers = agent)

        # Pause the loop
        sleep(randint(1,5))

        # Monitor the requests
        requests += 1
        elapsed_time = time() - start_time
        print('Request:{}; Frequency: {} requests/s'.format(requests, requests/elapsed_time))
        clear_output(wait = True)

        # Throw a warning for non-200 status codes
        if response.status_code != 200:
            print('Request: {}; Status code: {}'.format(requests, response.status_code))

        # Break the loop if the number of requests is greater than expected
        if requests > 72:
            print('Number of requests was greater than expected.')
            break
        # Parse the content of the request with BeautifulSoup
        page_html = BeautifulSoup(response.text, 'html.parser')

        # Select all the 50 movie containers from a single page
        mv_containers = page_html.find_all('div', class_ = 'sc-ab6fa25a-3 bVYfLY dli-parent')

        # For every movie of these 50
        for container in movie_containers:
            # If the movie has Metascore, then extract:
            if container.find('span', class_ = 'sc-b0901df4-0 bcQdDJ metacritic-score-box') is not None:
                # The name
                name = container.find('h3', class_ = 'ipc-title__text' ).text
                names.append(name)
                # The year
                year = container.find('span', class_ = 'sc-b0691f29-8 ilsLEX dli-title-metadata-item').text
                years.append(year)
                # The IMDB rating
                imdb = float(container.find('span', class_='ipc-rating-star ipc-rating-star--base ipc-rating-star--imdb ratingGroup--imdb-rating').tex
                imdb_ratings.append(imdb)
                # The Metascore
                m_score = container.find('span', class_ = 'sc-b0901df4-0 bcQdDJ metacritic-score-box').text
                metascores.append(int(m_score))
                # The number of votes
                vote = container.find('span', class_ = 'ipc-rating-star--voteCount').text[2:6]
                votes.append(vote)



    Request:6; Frequency: 0.1947938483663179 requests/s


movie_ratings = pd.DataFrame({'movie': names,
                              'year': years,
                              'imdb': imdb_ratings,
                              'metascore': metascores,
                              'votes': votes})
print(movie_ratings.info())
movie_ratings.head(10)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246 entries, 0 to 245
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   movie      246 non-null    object
 1   year       246 non-null    object
 2   imdb       246 non-null    float64
 3   metascore  246 non-null    int64
 4   votes      246 non-null    object
dtypes: float64(1), int64(1), object(3)
memory usage: 9.7+ KB
None
```

	movie	year	imdb	metascore	votes
0	1. Logan	2017	8.1	77	827K
1	2. Thor: Ragnarok	2017	7.9	74	813K
2	3. Guardians of the Galaxy Vol. 2	2017	7.6	67	756K
3	4. Dunkirk	2017	7.8	94	736K
4	5. Spider-Man: Homecoming	2017	7.4	73	716K
5	6. Wonder Woman	2017	7.3	76	698K
6	7. Get Out	2017	7.8	85	691K
7	8. Star Wars: Episode VIII - The Last Jedi	2017	6.9	84	670K
8	9. Blade Runner 2049	2017	8.0	81	658K

```
movie_ratings.tail(10)
```

	movie	year	imdb	metascore	votes
236	39. The Hitman's Bodyguard	2017	6.9	47	246K
237	40. I, Tonya	2017	7.5	77	242K
238	41. King Arthur: Legend of the Sword	2017	6.7	41	232K
239	42. Ghost in the Shell	2017	6.3	52	227K
240	44. Darkest Hour	2017	7.4	75	220K
241	46. American Made	2017	7.1	65	207K
242	47. Atomic Blonde	2017	6.7	63	206K
243	48. The Mummy	2017	5.4	34	206K
244	49. Baywatch	2017	5.5	37	201K
245	50. Bright	2017	6.3	29	201K

```
movie_ratings.to_csv('movie_rating.csv',index=False, encoding='utf-8', sep=':')
```

Data Prep

```
movie_ratings['year'].unique()
```

```
array(['2017'], dtype=object)
```

```
movie_ratings.dtypes
```

```
movie         object
year          object
imdb         float64
metascore      int64
votes         object
dtype: object
```

```
movie_ratings['year'] = (movie_ratings.year.apply(lambda x:x.replace('(I)','')))
```

```
movie_ratings['year'].unique()
```

```
array(['2017'], dtype=object)
```