Video Explaination: https://drive.google.com/file/d/1vrrfbjUmM1LZ-0PpVssC3Ex76p5tiaU/view?usp=sharing

Goodreads Web Scraping

```
import requests
from requests import get
from bs4 import BeautifulSoup
import pandas as pd
```

```
url = "https://www.goodreads.com/list/show/1.Best_Books_Ever?page=1"
headers={'User-Agent':'Mozilla/5.0 (Windows NT 6.3; Win 64 ; x64) Apple WebKit /537.36(KHTML , like Gecko) Chrome/80.0.3987.162 Safari/537.36'}

response = requests.get(url,headers=headers)

soup = BeautifulSoup(response.text, "lxml")
```

```
containers = soup.find_all("tr",{"itemtype":"http://schema.org/Book"})
len(containers)
```

```
100
```

```
first_book = containers[0]
```

```
## Book Name
first_book.find("span",{"itemprop":"name"}).text
```

```
"The Hunger Games (The Hunger Games, #1)"
```

```
## Author
first_book.find("a",{"class":"authorName"}).text
```

```
'Suzanne Collins'
```

```
## Rating
first_book.find("span",{"class":"minirating"}).text.split(" — ")[0]
```

```
' 4.35 avg rating'
```

```
## Ratings Count
first_book.find("span",{"class":"minirating"}).text.split(" — ")[1]
```

```
'6,825,866 ratings'
```

```
## Book Link
link = first_book.find("a")["href"]
book_url = "https://www.goodreads.com" + link
book_url
```

```
"https://www.goodreads.com/book/show/2767052-the-hunger-games"
```

```
response = requests.get(book_url, headers=headers)
inner_soup = BeautifulSoup(response.text, "lxml")
```

```
## Genres
genres_list = inner_soup.find_all("span",{"class":"BookPageMetadataSection__genreButton"})
genres_list[0].text
```

```
'Young Adult'
```

```
genres = []
for i in genres_list:
  genres.append(i.text)
genres
```

```
['Young Adult',
 'Dystopia',
 'Fiction',
 'Fantasy',
 'Science Fiction',
 'Romance',
 'Adventure']
```

```
## Number of Pages and Book Type
inner_soup.find("p",{"data-testid":"pagesFormat"}).text
```

```
"374 pages, Hardcover"
```

```
# 1st published year
inner_soup.find("p",{"data-testid":"publicationInfo"}).text.split(", ")[1]
```

```
'2008'
```

```
## Score
first_book.find("a",{"onclick":"Lightbox.showBoxByID('score_explanation', 300); return false;"}).text
```

```
"score: 6,318,854"
```

```
## Vote
text = first_book.find("div",{"style":"margin-top: 5px"}).text
clean = " ".join(text.split())
parts = clean.split(" and ")
parts[1]
```

```
'43,905 people voted'
```

Single Page Web Scraping

```
url = "https://www.goodreads.com/list/show/1.Best_Books_Ever?page=1"
headers={'User-Agent':'Mozilla/5.0 (Windows NT 6.3; Win 64 ; x64) Apple WebKit /537.36(KHTML , like Gecko) Chrome/80.0.3987.162 Safari/537.36'}

response = requests.get(url,headers=headers)

soup = BeautifulSoup(response.text, "lxml")

containers = soup.find_all("tr",{"itemtype":"http://schema.org/Book"})


# Create an empty list to accumulate data
raw_data = []

# Create an empty DataFrame to store the raw data with predefined column names
books_raw_data = pd.DataFrame(columns=['book_name', 'author', 'genres_list', 'pages_format', 'first_published_year'
                                       'rating', 'ratings_count', 'score', 'votes','book_url'])



# Iterate over each container and extract relevant information
for i in containers:
  ## Book names
  book_name = i.find("span",{"itemprop":"name"}).text

  ## Author
  author = i.find("a",{"class":"authorName"}).text

  ## Rating
  rating = i.find("span",{"class":"minirating"}).text.split(" — ")[0]

  ## Ratings Count
  ratings_count = i.find("span",{"class":"minirating"}).text.split(" — ")[1]

  ## Score
  score = i.find("a",{"onclick":"Lightbox.showBoxByID('score_explanation', 300); return false;"}).text

  ## Votes
  text = i.find("div",{"style":"margin-top: 5px"}).text
  clean = " ".join(text.split())
  parts = clean.split(" and ")
  votes = parts[1]

  ## Book link
  book_url = "https://www.goodreads.com" + i.find("a")["href"]

  inner = requests.get(book_url, headers=headers)
  inner_soup = BeautifulSoup(inner.text, "lxml")

  ## Genres
  genres = inner_soup.find_all("span",{"class":"BookPageMetadataSection__genreButton"})
  genres_list = []
  for i in genres:
    genres_list.append(i.text)

  ## Number of Pages and Book Type
  pages_format = inner_soup.find("p",{"data-testid":"pagesFormat"}).text

  ## First published year
  first_published_year = inner_soup.find("p",{"data-testid":"publicationInfo"}).text.split(", ")[1]

  # Create a dictionary for the current data
  dictionary = {
              "book_name": book_name,
              "author": author,
              "genres_list": genres_list,
              "pages_format": pages_format,
              "first_published_year": first_published_year,
              "rating": rating,
              "ratings_count": ratings_count,
              "score": score,
              "votes": votes,
              "book_url": book_url
          }

  # Append the current movie's dictionary to the list of movie data
  raw_data.append(dictionary)

# Create a DataFrame from the list of movie data
books_raw_data = pd.DataFrame(raw_data)
```

```
books_raw_data.shape
```

```
(100, 10)
```

```
books_raw_data.head()
```

	book_name	author	genres_list	pages_format	first_published_year	rating	ratings_count	score	votes	book_url
0	The Hunger Games (The Hunger Games, #1)	Suzanne Collins	[Young Adult, Dystopia, Fiction, Fantasy, Scie...	374 pages, Hardcover	2008	4.35 avg rating	9,825,466 ratings	score: 4,318,854	43,905 people voted	https://www.goodreads.com/book/show/2767052-th...
1	Pride and Prejudice	Jane Austen	[Classics, Romance, Fiction, Historical Fictio...	279 pages, Paperback	1813	4.29 avg rating	4,762,379 ratings	score: 2,966,459	30,399 people voted	https://www.goodreads.com/book/show/1885.Pride...
2	To Kill a Mockingbird	Harper Lee	[Classics, Fiction, Historical Fiction, School...	323 pages, Paperback	1960	4.26 avg rating	6,833,157 ratings	score: 2,602,221	26,573 people voted	https://www.goodreads.com/book/show/2657.To_Ki...
3	Harry Potter and the Order of the Phoenix (Har...	J.K. Rowling	[Fantasy, Fiction, Young Adult, Harry Potter, ...	896 pages, Hardcover	2003	4.50 avg rating	3,771,882 ratings	score: 2,080,496	21,168 people voted	https://www.goodreads.com/book/show/5807451-h...
4	The Book Thief	Markus Zusak	[Historical Fiction, Fiction, Young Adult, Cla...	592 pages, Kindle Edition	2005	4.39 avg rating	2,859,724 ratings	score: 1,970,739	20,242 people voted	https://www.goodreads.com/book/show/19063.The_...

Multiple Pages Web Scraping

```
headers = {'User-Agent':'Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.162 Safari/537.36'}

# Create an empty list to accumulate data
all_books = []

# Create an empty DataFrame to store the raw data with predefined column names
books_raw_data = pd.DataFrame(columns=['book_name', 'author', 'genres_list', 'pages_format', 'first_published_year'
                                       'rating', 'ratings_count', 'score', 'votes','book_url'])


for page in range(1, 11):
    print(f"Scraping page {page}...")

    url = f"https://www.goodreads.com/list/show/1.Best_Books_Ever?page={page}"
    response = requests.get(url, headers=headers)
    soup = BeautifulSoup(response.text, "lxml")

    containers = soup.find_all("tr", {"itemtype":"http://schema.org/Book"})


    # Iterate over each container and extract relevant information
    for i in containers:
      ## Book names
      book_name = i.find("span",{"itemprop":"name"}).text

      ## Author
      author = i.find("a",{"class":"authorName"}).text

      ## Rating
      rating = i.find("span",{"class":"minirating"}).text.split(" — ")[0]

      ## Ratings Count
      ratings_count = i.find("span",{"class":"minirating"}).text.split(" — ")[1]

      ## Score
      score = i.find("a",{"onclick":"Lightbox.showBoxByID('score_explanation', 300); return false;"}).text

      ## Votes
      text = i.find("div",{"style":"margin-top: 5px"}).text
      clean = " ".join(text.split())
      parts = clean.split(" and ")
      votes = parts[1]

      ## Book link
      book_url = "https://www.goodreads.com" + i.find("a")["href"]

      inner = requests.get(book_url, headers=headers)
      inner_soup = BeautifulSoup(inner.text, "lxml")

      ## Genres
      genres = inner_soup.find_all("span",{"class":"BookPageMetadataSection__genreButton"})
      genres_list = []
      for i in genres:
          genres_list.append(i.text)


      ## Number of Pages and Book Type
      pages_format_tag = inner_soup.find("p",{"data-testid":"pagesFormat"})
      pages_format = pages_format_tag.text if pages_format_tag else None


      ## First published year
      first_published_year_tag = inner_soup.find("p",{"data-testid":"publicationInfo"})
      first_published_year = first_published_year_tag.text.split(", ")[1] if first_published_year_tag else None

      # Create a dictionary for the current data
      dictionary = {
                  "book_name": book_name,
                  "author": author,
                  "genres_list": genres_list,
                  "pages_format": pages_format,
                  "first_published_year": first_published_year,
                  "rating": rating,
                  "ratings_count": ratings_count,
                  "score": score,
                  "votes": votes,
                  "book_url": book_url
              }

      # Append the current movie's dictionary to the list of movie data
      all_books.append(dictionary)

# Create a DataFrame from the list of movie data
books_raw_data = pd.DataFrame(all_books)

# Chick the shape of df
books_raw_data.shape
```

```
Scraping page 1...
Scraping page 2...
Scraping page 3...
Scraping page 4...
Scraping page 5...
Scraping page 6...
Scraping page 7...
Scraping page 8...
Scraping page 9...
Scraping page 10...
(1000, 10)
```

```
books_raw_data.head()
```

	book_name	author	genres_list	pages_format	first_published_year	rating	ratings_count	score	votes	book_url
0	The Hunger Games (The Hunger Games, #1)	Suzanne Collins	[Young Adult, Dystopia, Fiction, Fantasy, Scie...	374 pages, Hardcover	2008	4.35 avg rating	9,825,466 ratings	score: 4,318,854	43,905 people voted	https://www.goodreads.com/book/show/2767052-th...
1	Pride and Prejudice	Jane Austen	[Classics, Romance, Fiction, Historical Fictio...	279 pages, Paperback	1813	4.29 avg rating	4,762,379 ratings	score: 2,966,459	30,399 people voted	https://www.goodreads.com/book/show/1885.Pride...
2	To Kill a Mockingbird	Harper Lee	[Classics, Fiction, Historical Fiction, School...	323 pages, Paperback	1960	4.26 avg rating	6,833,157 ratings	score: 2,602,221	26,573 people voted	https://www.goodreads.com/book/show/2657.To_Ki...
3	Harry Potter and the Order of the Phoenix (Har...	J.K. Rowling	[Fantasy, Fiction, Young Adult, Harry Potter, ...	896 pages, Hardcover	2003	4.50 avg rating	3,771,882 ratings	score: 2,080,496	21,168 people voted	https://www.goodreads.com/book/show/5807451-h...
4	The Book Thief	Markus Zusak	[Historical Fiction, Fiction, Young Adult, Cla...	592 pages, Kindle Edition	2005	4.39 avg rating	2,859,724 ratings	score: 1,970,739	20,242 people voted	https://www.goodreads.com/book/show/19063.The_...

```
books_raw_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   book_name            1000 non-null   object
 1   author               1000 non-null   object
 2   genres_list          1000 non-null   object
 3   pages_format         999 non-null    object
 4   first_published_year 999 non-null    object
 5   rating               1000 non-null   object
 6   ratings_count        1000 non-null   object
 7   score                1000 non-null   object
 8   votes                1000 non-null   object
 9   book_url             1000 non-null   object
dtypes: object(10)
memory usage: 78.3+ KB
```

```
books_raw_data.isnull().sum()
```

	0
book_name	0
author	0
genres_list	0
pages_format	1
first_published_year	1
rating	0
ratings_count	0
score	0
votes	0
book_url	0

dtype: int64

```
books_raw_data.to_csv("books_raw_data.csv", index=False)
```