In [ ]:

```
import requests
import csv
from bs4 import BeautifulSoup as bs
import urllib
import os

def scrape_and_run(genre):
    # scrape on goodreads.com using desire genre type or key word
    # and save the titles and autors in a csv file
    page = requests.get("https://www.goodreads.com/shelf/show/" + genre)
    soup = bs(page.content, 'html.parser')
    titles = soup.find_all('a', class_='bookTitle')
    authors = soup.find_all('a', class_='authorName')


    image_dir = os.getcwd() + "/images/" + genre

    ## check if the desire genre path exists
    ## create a new one if it doesnt
    if not os.path.exists(image_dir):
        os.makedirs(image_dir)

    with open(genre + '.csv', 'w') as csvfile:
        fieldnames = ['title', 'author']
        csv_write = csv.DictWriter(csvfile, fieldnames=fieldnames)
        books_save = 0

        for title, author in zip(titles, authors):

            try:
                ## single book page
                book_page = requests.get("https://www.goodreads.com" + title['href'])
                soup = bs(book_page.content, 'html.parser')
                # get image id
                image = soup.find('img', id='coverImage')

                title_name = title.get_text()

                save_dir = image_dir + "/" + title_name
                urllib.request.urlretrieve(image['src'], save_dir)
```

```
                csv_write.writerow({'title': title_name, 'author': author.get_text()})
                books_save += 1
                ## error handelling for long file names
            except OSError as exc:
                if exc.errno == 36:
                    print(exc)

        print("%d %s books saved." % (books_save, genre)) # books count feedback



if __name__ == '__main__':

    ## run ifinite till user tells you to stop
    ## to avoid having to compile again and again
    while True:
        genre = input("Enter the genre (or quit to stop): ").lower() # input case lowered
        if(genre == "quit"):
            break
        else:
            scrape_and_run(genre)
```

In [ ]:

In [ ]: