Lab#3, NLP@CGU Spring 2023

This is due on 2023/03/20 16:00, commit to your github as a PDF (lab3.pdf) (File>Print>Save as PDF).

IMPORTANT: After copying this notebook to your Google Drive, please paste a link to it below. To get a publicly-accessible link, hit the *Share* button at the top right, then click "Get shareable link" and copy over the result. If you fail to do this, you will receive no credit for this lab!

LINK: paste your link here

https://colab.research.google.com/drive/17MvoLJSvGrTYAuiTCsLuQWeKSbLlyVu3?usp=sharing

---

Student ID:

Name:

Question 1 (100 points)

Implementing Yahoo Movies Crawler.

1. Design a Yahoo! Movie Crawler.
2. Crawl all the movie information listed in movie_intheaters page
3. The more movie data crawled, the higher the score

按兩下 (或按 Enter 鍵) 即可編輯

```
import requests
import re
from bs4 import BeautifulSoup

Y_MOVIE_URL = "https://movies.yahoo.com.tw/movie_intheaters.html"

# YOUR CODE HERE!
# IMPLEMENTIG YAHOO MOVIES CRAWLER
```

```
class MovieCrawler(object):
    def __init__(self):
      self.movies = []

    def get_movies(self, page_url):
      resp = requests.get(page_url)
      soup = BeautifulSoup(resp.text, 'html.parser')
      movie_list = soup.find_all("div", class_ = "release_info_text")
      page = soup.find("div", class_ = "page_numbox").find("ul")

      pages = []
      for p in page:
        if (p.find("a") != -1 and p.find("a") != None):
          if (p.find("a").text.isnumeric()):
            pages.append(p.find("a")["href"])

      for movie in movie_list:
        temp = {}

        ch_name = movie.find("div", class_ = "release_movie_name").find("a")
        temp["ch_name"] = ch_name.text[19: ]

        en_name = movie.find("div", class_ = "release_movie_name").find("div", c
        temp["en_name"] = en_name.text[21: ]

        movie_url = ch_name["href"]
        temp["movie_url"] = movie_url

        release_date = movie.find("div", class_ = "release_movie_time")
        t = ""
        for n in release_date.text:
          if (n not in ("\n", " ")):
            t += n
        temp["release_date"] = t

        intro = movie.find("div", class_ = "release_text").find("span")
        temp["intro"] = intro.text

        self.movies.append(temp)

      for p in pages:
        resp = requests.get(p)
        soup = BeautifulSoup(resp.text, 'html.parser')
        movie_list = soup.find_all("div", class_ = "release_info_text")

        for movie in movie_list:
          temp = {}
          ch_name = movie.find("div", class_ = "release_movie_name").find("a")
          temp["ch_name"] = ch_name.text[19: ]

          en_name = movie.find("div", class_ = "release_movie_name").find("div",
          temp["en_name"] = en_name.text[21: ]

          movie_url = ch_name["href"]
          temp["movie_url"] = movie_url

          release_date = movie.find("div", class_ = "release_movie_time")
          t = ""
          for n in release_date.text:
            if (n not in ("\n", " ")):
              t += n
          temp["release_date"] = t

          intro = movie.find("div", class_ = "release_text").find("span")
          temp["intro"] = intro.text

          self.movies.append(temp)

      return self.movies


# # DO NOT MODIFY THE VARIABLES
crawler = MovieCrawler()
movies = crawler.get_movies(Y_MOVIE_URL)

# # THE RESULTS : AS THE FOLLOWING SECTION
# # {'ch_name', 'en_name', 'movie_url', 'release_date', 'intro'}
print(len(movies))
print(*movies, sep="\n")
```

```
77
{'ch_name': '配樂大師顏尼歐', 'en_name': 'Ennio: The Maestro', 'movie_url': '
{'ch_name': '熊蓋毒', 'en_name': 'Cocaine Bear', 'movie_url': 'https://movie
{'ch_name': '若愛重來', 'en_name': 'Marriages', 'movie_url': 'https://movies
{'ch_name': '無人相信的真相', 'en_name': 'La syndicaliste', 'movie_url': 'htt
{'ch_name': '闇黑對決', 'en_name': "The Devil's Deal", 'movie_url': 'https:/
{'ch_name': '噩夢輓歌 4K數位修復版', 'en_name': 'Requiem For A Dream', 'movie_
{'ch_name': '人體動物圖鑑：烏龜的殼其實是肋骨', 'en_name': 'Turtle’s Shell is a 
{'ch_name': '流水落花', 'en_name': 'Lost Love', 'movie_url': 'https://movies
{'ch_name': '聖蛛', 'en_name': 'Holy Spider', 'movie_url': 'https://movies.
{'ch_name': '沙贊！眾神之怒', 'en_name': 'Shazam! Fury of the Gods', 'movie_u
{'ch_name': '夢遊樂園', 'en_name': 'Melody-Go-Round', 'movie_url': 'https://
{'ch_name': '黑的教育', 'en_name': 'Bad Education', 'movie_url': 'https://mo
{'ch_name': 'TÁR塔爾', 'en_name': 'Tár', 'movie_url': 'https://movies.yahoo.
{'ch_name': '驚聲尖叫6', 'en_name': 'Scream VI', 'movie_url': 'https://movie
{'ch_name': '怪談比留子 數位修復版', 'en_name': 'Hiruko The Goblin', 'movie_ur
{'ch_name': '天生一對2大電影：再續前緣', 'en_name': 'Love Destiny: The Movie',
{'ch_name': '尋找第5味', 'en_name': 'Umami', 'movie_url': 'https://movies.yal
{'ch_name': '超完美狗保姆', 'en_name': 'My Puppy', 'movie_url': 'https://movi
{'ch_name': '蓋世棋蹟', 'en_name': 'The Royal Game', 'movie_url': 'https://mo
{'ch_name': '斷網', 'en_name': 'Cyberheist', 'movie_url': 'https://movies.ya
{'ch_name': '所有的美麗與血淚', 'en_name': 'All the Beauty and the Bloodshed',
{'ch_name': '過時·過節', 'en_name': 'Hong Kong Family', 'movie_url': 'https:/
{'ch_name': '8釐米：詛咒影帶', 'en_name': '8MM: The Sinister Record', 'movie_u
{'ch_name': '屍蹤天使', 'en_name': 'Mindcage', 'movie_url': 'https://movies.y
{'ch_name': '貓王艾維斯', 'en_name': 'Elvis', 'movie_url': 'https://movies.ya
{'ch_name': '媽的多重宇宙', 'en_name': 'Everything Everywhere All at Once', '
{'ch_name': '光影帝國', 'en_name': 'Empire Of Light', 'movie_url': 'https://m
{'ch_name': '金牌拳手3', 'en_name': 'Creed III', 'movie_url': 'https://movies
{'ch_name': '本日公休', 'en_name': 'Day Off', 'movie_url': 'https://movies.ya
{'ch_name': '玩具當家', 'en_name': 'The New Toy', 'movie_url': 'https://movie
{'ch_name': '驚爆點', 'en_name': 'Point Break', 'movie_url': 'https://movies
{'ch_name': '火線埋伏', 'en_name': 'Ambush', 'movie_url': 'https://movies.yal
{'ch_name': '小熊維尼：血與蜜', 'en_name': 'Winnie the Pooh：Blood and Honey',
{'ch_name': '鈴芽之旅', 'en_name': 'Suzume', 'movie_url': 'https://movies.yal
{'ch_name': '法貝爾曼', 'en_name': 'The Fabelmans', 'movie_url': 'https://mov
{'ch_name': '人肉搜索2：失蹤搜救', 'en_name': 'Missing', 'movie_url': 'https:/
{'ch_name': '悲情城市', 'en_name': 'A City of Sadness', 'movie_url': 'https:/
{'ch_name': '風再起時', 'en_name': 'Where The Wind blows', 'movie_url': 'http
{'ch_name': '胡桃鉗與魔笛公主的奇幻冒險', 'en_name': 'The Nutcracker And The Mag
{'ch_name': '我們的黎明', 'en_name': 'Break of Dawn', 'movie_url': 'https://m
{'ch_name': '不離職冒險王', 'en_name': 'Irreductible', 'movie_url': 'https://
{'ch_name': '「鬼滅之刃」上弦集結，前進刀匠村', 'en_name': 'Demon Slayer Kimetsu
{'ch_name': '追海豚的長崎夏日', 'en_name': 'Sabakan', 'movie_url': 'https://mo
{'ch_name': '蟻人與黃蜂女：量子狂熱', 'en_name': 'Ant-Man and the Wasp: Quantum
{'ch_name': '超難搞先生', 'en_name': 'A Man Called Otto', 'movie_url': 'https
{'ch_name': '關於我和鬼變成家人的那件事', 'en_name': 'Marry My Dead Body', 'movi
{'ch_name': '山椒魚來了', 'en_name': '', 'movie_url': 'https://movies.yahoo.c
{'ch_name': '僕愛君愛：致深愛妳的那個我', 'en_name': 'To me, The One Who Loved Y
{'ch_name': '僕愛君愛：致我深愛的每個妳', 'en_name': 'To Every You I’ve Loved Be
{'ch_name': '日麗', 'en_name': 'Aftersun', 'movie_url': 'https://movies.yahc
{'ch_name': '新世紀福音戰士新劇場版：終', 'en_name': 'Evangelion:3.0+1.0 Thrice
{'ch_name': '瑪琳艾索普：首席女指揮', 'en_name': 'Conductor', 'movie_url': 'htt
{'ch_name': '我的鯨魚老爸', 'en_name': 'The Whale', 'movie_url': 'https://mov
{'ch_name': '幻影', 'en_name': 'Phantom', 'movie_url': 'https://movies.yahoo
{'ch_name': '伊尼舍林的女妖', 'en_name': 'The Banshees of Inisherin', 'movie_u
{'ch_name': '鱷魚歌王', 'en_name': 'Lyle, Lyle, Crocodile', 'movie_url': 'htt
{'ch_name': '巴比倫', 'en_name': 'Babylon', 'movie_url': 'https://movies.yah
{'ch_name': '詐團圓', 'en_name': 'Scamsgiving', 'movie_url': 'https://movies
{'ch_name': '天龍八部之喬峰傳', 'en_name': 'Sakra', 'movie_url': 'https://movi
```