HW7 : BeautifulSoup

For this assignment, you will write five functions to get data from IMDB’s top 250 web page.

We have provided test cases that will pass if the functions are written correctly. You may ***not*** edit the test cases in anyway. ***It is okay for the extra credit test case (test_file_has_250_movies) to fail if you don’t attempt the extra credit.***

NOTE 1: The IMDB Top 250 list gets updated often. The movies or directors listed in your output may not match the output in these instructions.

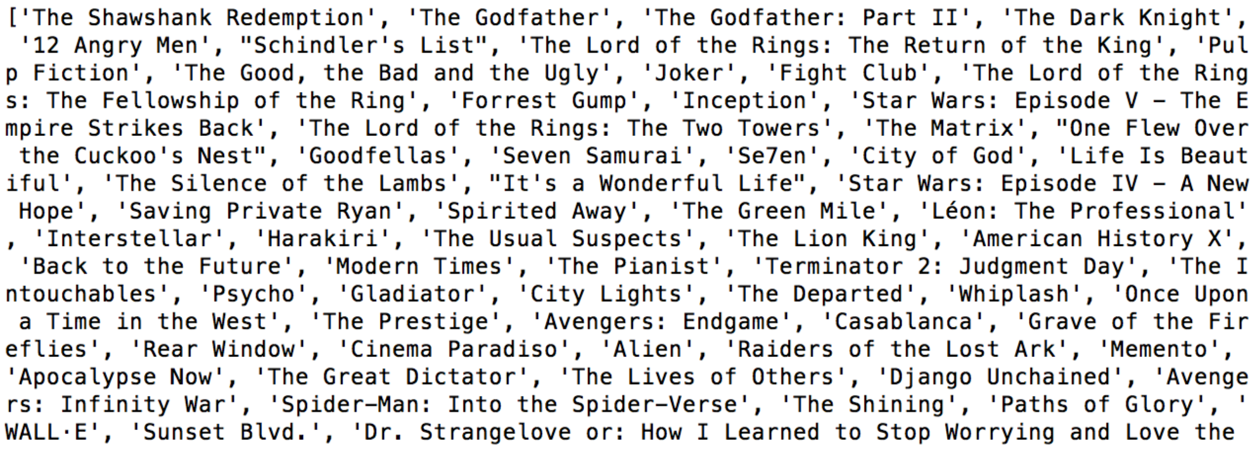
NOTE 2: You don’t have to write code to print the data shown below. We have provided code that will print the data for you.

Tasks

1. **getAllMovieRows() function**: The function takes the BeautifulSoup object as input. It finds and returns a list of all of the <tr> tags that contain movie information. Inspect the website and look at the html that displays the movie data to determine which tag and class you should search for in order to find the movie data.

2. **getAllMovies() function**: The function takes as input the list returned by the previous function. It goes through the list, finds all the movie names, and returns a list of the top 250 movie names.

   Expected printed output (the image does not show all 250 movies, but you should return all 250 movies in the list):

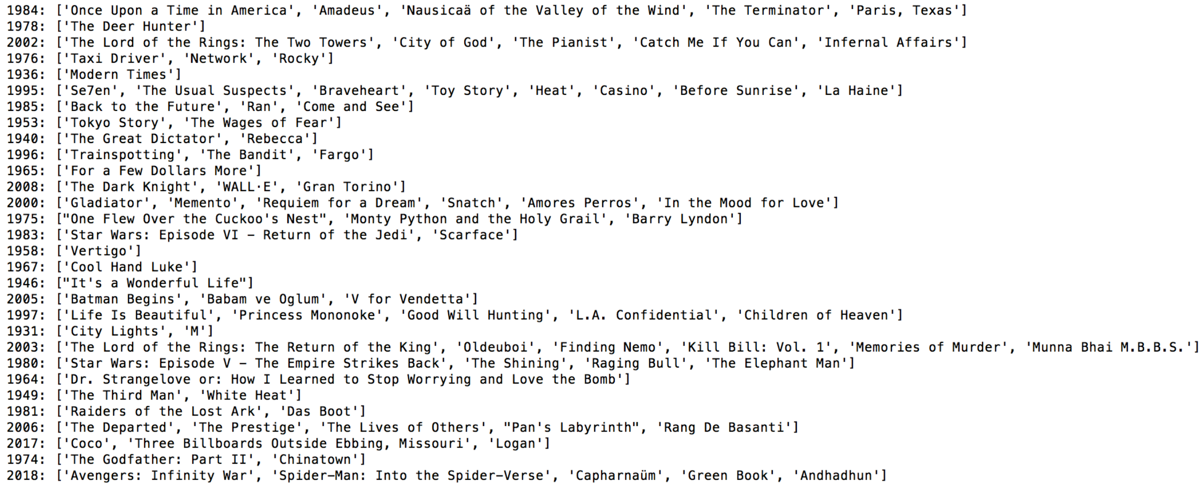


```
['The Shawshank Redemption', 'The Godfather', 'The Godfather: Part II', 'The Dark Knight',
 '12 Angry Men', "Schindler's List", 'The Lord of the Rings: The Return of the King', 'Pul
p Fiction', 'The Good, the Bad and the Ugly', 'Joker', 'Fight Club', 'The Lord of the Ring
s: The Fellowship of the Ring', 'Forrest Gump', 'Inception', 'Star Wars: Episode V - The E
mpire Strikes Back', 'The Lord of the Rings: The Two Towers', 'The Matrix', "One Flew Over
 the Cuckoo's Nest", 'Goodfellas', 'Seven Samurai', 'Se7en', 'City of God', 'Life Is Beaut
iful', 'The Silence of the Lambs', "It's a Wonderful Life", 'Star Wars: Episode IV - A New
 Hope', 'Saving Private Ryan', 'Spirited Away', 'The Green Mile', 'Léon: The Professional'
, 'Interstellar', 'Harakiri', 'The Usual Suspects', 'The Lion King', 'American History X',
 'Back to the Future', 'Modern Times', 'The Pianist', 'Terminator 2: Judgment Day', 'The I
ntouchables', 'Psycho', 'Gladiator', 'City Lights', 'The Departed', 'Whiplash', 'Once Upon
 a Time in the West', 'The Prestige', 'Avengers: Endgame', 'Casablanca', 'Grave of the Fir
eflies', 'Rear Window', 'Cinema Paradiso', 'Alien', 'Raiders of the Lost Ark', 'Memento',
'Apocalypse Now', 'The Great Dictator', 'The Lives of Others', 'Django Unchained', 'Avenge
rs: Infinity War', 'Spider-Man: Into the Spider-Verse', 'The Shining', 'Paths of Glory', '
WALL·E', 'Sunset Blvd.', 'Dr. Strangelove or: How I Learned to Stop Worrying and Love the
```

3. **getMovieByYears() function**: The function takes as input the list returned by the getAllMovieRows() function. It goes through the list, and creates a dictionary where the keys are years and values are list of movies released that year. (Do NOT convert the

year to integers)

Expected printed output (the image does not show all the years and corresponding movies)



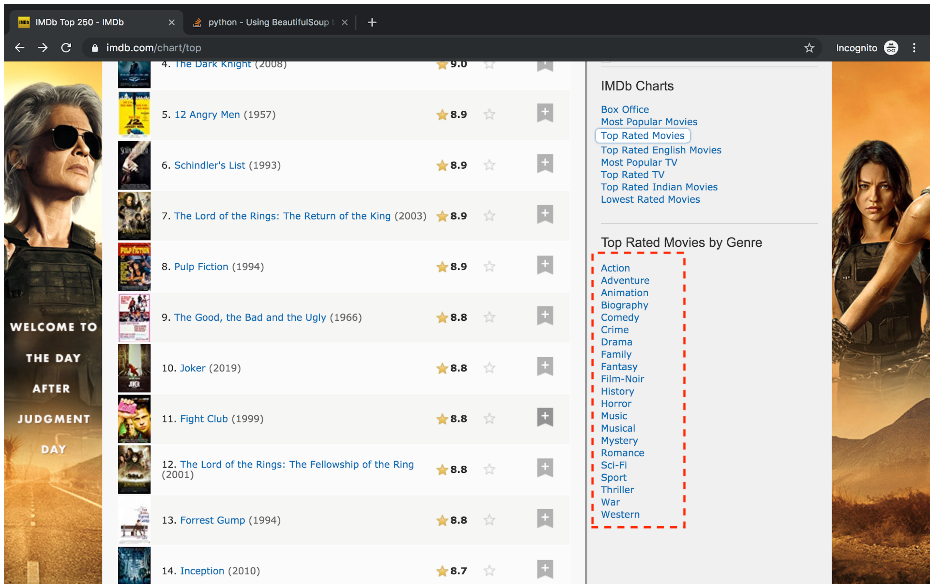
```
1984: ['Once Upon a Time in America', 'Amadeus', 'Nausicaä of the Valley of the Wind', 'The Terminator', 'Paris, Texas']
1978: ['The Deer Hunter']
2002: ['The Lord of the Rings: The Two Towers', 'City of God', 'The Pianist', 'Catch Me If You Can', 'Infernal Affairs']
1976: ['Taxi Driver', 'Network', 'Rocky']
1936: ['Modern Times']
1995: ['Se7en', 'The Usual Suspects', 'Braveheart', 'Toy Story', 'Heat', 'Casino', 'Before Sunrise', 'La Haine']
1985: ['Back to the Future', 'Ran', 'Come and See']
1953: ['Tokyo Story', 'The Wages of Fear']
1940: ['The Great Dictator', 'Rebecca']
1996: ['Trainspotting', 'The Bandit', 'Fargo']
1965: ['For a Few Dollars More']
2008: ['The Dark Knight', 'WALL·E', 'Gran Torino']
2000: ['Gladiator', 'Memento', 'Requiem for a Dream', 'Snatch', 'Amores Perros', 'In the Mood for Love']
1975: ["One Flew Over the Cuckoo's Nest", 'Monty Python and the Holy Grail', 'Barry Lyndon']
1983: ['Star Wars: Episode VI - Return of the Jedi', 'Scarface']
1958: ['Vertigo']
1967: ['Cool Hand Luke']
1946: ["It's a Wonderful Life"]
2005: ['Batman Begins', 'Babam ve Oglum', 'V for Vendetta']
1997: ['Life Is Beautiful', 'Princess Mononoke', 'Good Will Hunting', 'L.A. Confidential', 'Children of Heaven']
1931: ['City Lights', 'M']
2003: ['The Lord of the Rings: The Return of the King', 'Oldeuboi', 'Finding Nemo', 'Kill Bill: Vol. 1', 'Memories of Murder', 'Munna Bhai M.B.B.S.']
1980: ['Star Wars: Episode V - The Empire Strikes Back', 'The Shining', 'Raging Bull', 'The Elephant Man']
1964: ['Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb']
1949: ['The Third Man', 'White Heat']
1981: ['Raiders of the Lost Ark', 'Das Boot']
2006: ['The Departed', 'The Prestige', 'The Lives of Others', "Pan's Labyrinth", 'Rang De Basanti']
2017: ['Coco', 'Three Billboards Outside Ebbing, Missouri', 'Logan']
1974: ['The Godfather: Part II', 'Chinatown']
2018: ['Avengers: Infinity War', 'Spider-Man: Into the Spider-Verse', 'Capharnaüm', 'Green Book', 'Andhadhun']
```

4. **getAllDirectors() function**: The function takes as input the list returned by the getAllMovieRows() function. It goes through the list, finds all the directors, and returns a list of director names. The list does not have any repeat names, all names are unique.

   Expected Output that will get printed:

```
['Frank Darabont', 'Francis Ford Coppola', 'Christopher Nolan', 'Sidney Lumet', 'Steven Sp
ielberg', 'Peter Jackson', 'Quentin Tarantino', 'Sergio Leone', 'Todd Phillips', 'David Fi
ncher', 'Robert Zemeckis', 'Irvin Kershner', 'Lana Wachowski', 'Milos Forman', 'Martin Sco
rsese', 'Akira Kurosawa', 'Fernando Meirelles', 'Roberto Benigni', 'Jonathan Demme', 'Fran
k Capra', 'George Lucas', 'Hayao Miyazaki', 'Luc Besson', 'Masaki Kobayashi', 'Bryan Singe
r', 'Roger Allers', 'Tony Kaye', 'Charles Chaplin', 'Roman Polanski', 'James Cameron', 'Ol
ivier Nakache', 'Alfred Hitchcock', 'Ridley Scott', 'Damien Chazelle', 'Anthony Russo', 'M
ichael Curtiz', 'Isao Takahata', 'Giuseppe Tornatore', 'Florian Henckel von Donnersmarck',
 'Bob Persichetti', 'Stanley Kubrick', 'Andrew Stanton', 'Billy Wilder', 'Chan-wook Park',
 'Joon-ho Bong', 'Sam Mendes', 'Lee Unkrich', 'Makoto Shinkai', 'Mel Gibson', 'Wolfgang Pe
tersen', 'Rajkumar Hirani', 'Aamir Khan', 'Richard Marquand', 'John Lasseter', 'Nitesh Tiw
ari', 'Gus Van Sant', 'Fritz Lang', 'Darren Aronofsky', 'Michel Gondry', 'Orson Welles', '
Thomas Vinterberg', 'Guy Ritchie', 'Jean-Pierre Jeunet', 'Vittorio De Sica', 'Brian De Pal
ma', 'Stanley Donen', 'David Lean', 'George Roy Hill', 'Asghar Farhadi', 'Robert Mulligan'
, 'Nadine Labaki', 'Pete Docter', 'Curtis Hanson', 'Terry Gilliam', 'Denis Villeneuve', 'M
ichael Mann', 'John McTiernan', 'Clint Eastwood', 'Peter Farrelly', 'Oliver Hirschbiegel',
 'Majid Majidi', 'Jeethu Joseph', 'John Sturges', 'Joseph L. Mankiewicz', 'Ron Howard', 'G
uillermo del Toro', 'Elem Klimov', 'Juan José Campanella', 'Çagan Irmak', 'John Huston', '
Stanley Kramer', 'Martin McDonagh', 'James McTeigue', 'Paul Thomas Anderson', "Gavin O'Con
nor", 'Ingmar Bergman', 'Lenny Abrahamson', 'Danny Boyle', 'Ethan Coen', 'David Lynch', 'S
riram Raghavan', 'M. Night Shyamalan', 'John Carpenter', 'Victor Fleming', 'Carol Reed', '
Elia Kazan', 'Yavuz Turgul', 'Joel Coen', 'Michael Cimino', 'Yasujirô Ozu', 'Andrei Tarkov
sky', 'Peter Weir', 'Damián Szifron', 'Adam Elliot', 'Jim Sheridan', 'Buster Keaton', 'Cly
de Bruckman', 'Dean DeBlois', 'Wes Anderson', 'Richard Linklater', 'Michael Powell', 'Sean
 Penn', 'Henri-Georges Clouzot', 'Stuart Rosenberg', 'Steve McQueen', 'Terry Jones', 'Carl
 Theodor Dreyer', 'Rob Reiner', 'Rakeysh Omprakash Mehra', 'George Miller', 'Oliver Stone'
, 'Lasse Hallström', 'James Mangold', 'William Wyler', 'Raoul Walsh', 'Terry George', 'Ale
jandro G. Iñárritu', 'David Yates', 'Tom McCarthy', 'François Truffaut', 'John G. Avildsen
', 'Anurag Kashyap', 'Mathieu Kassovitz', 'Ashutosh Gowariker', 'Tate Taylor', 'Katsuhiro
Ôtomo', 'Luis Buñuel', 'Kar-Wai Wong', 'Wim Wenders', 'James Gunn', 'Andrew Lau', 'Nishika
nt Kamat', 'Ron Clements', 'Gillo Pontecorvo', 'Shimit Amin']
```

5. **getAllGenres() function**: The IMDB Top 250 webpage displays a list of genres on the right (see the red rectangle in image below). The function takes as input the BeautifulSoup object and returns a list of those genres.



4. The Dark Knight (2008) 9.0
5. 12 Angry Men (1957) 8.9
6. Schindler's List (1993) 8.9
7. The Lord of the Rings: The Return of the King (2003) 8.9
8. Pulp Fiction (1994) 8.9
9. The Good, the Bad and the Ugly (1966) 8.8
10. Joker (2019) 8.8
11. Fight Club (1999) 8.8
12. The Lord of the Rings: The Fellowship of the Ring (2001) 8.8
13. Forrest Gump (1994) 8.8
14. Inception (2010) 8.7

Expected printed output:

```
['Action', 'Adventure', 'Animation', 'Biography', 'Comedy', 'Crime', 'Drama', 'Family', 'F
antasy', 'Film-Noir', 'History', 'Horror', 'Music', 'Musical', 'Mystery', 'Romance', 'Sci-
Fi', 'Sport', 'Thriller', 'War', 'Western']
```

Grading Rubric : Total 60 points

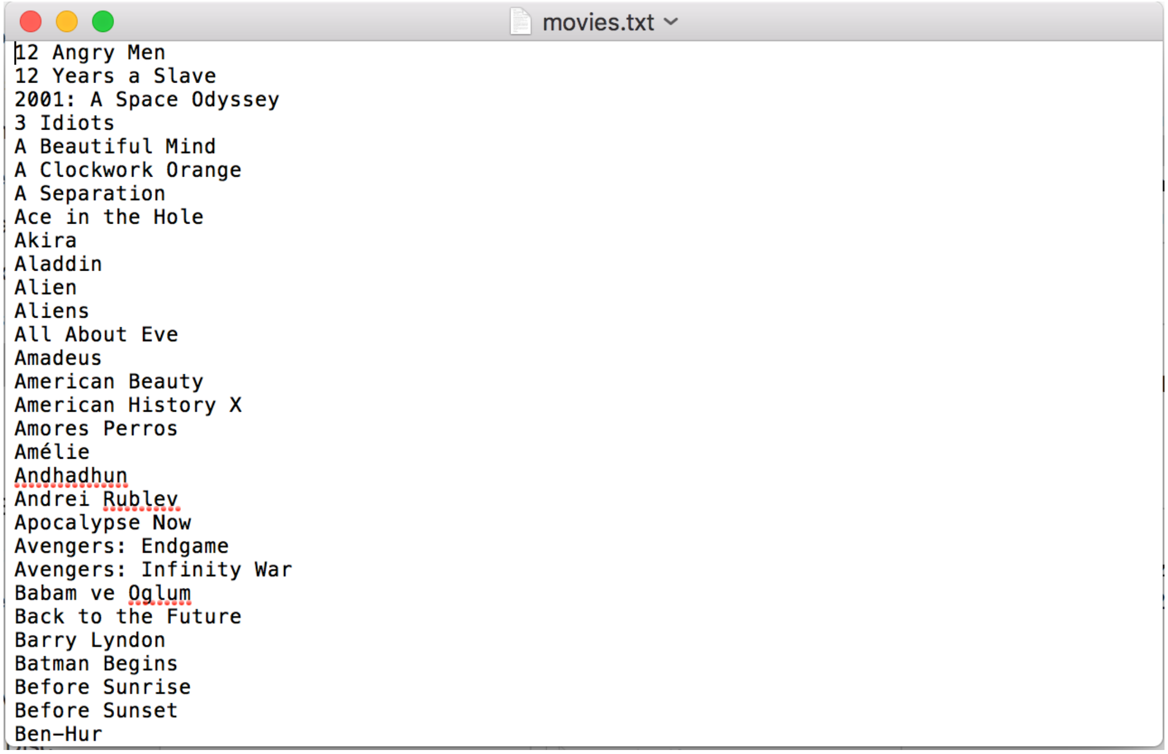
15 points: getAllMovieRows() function
10 points: getAllMovies() function
10 points: getMovieByYears() function
10 points: getAllDirectors() function
15 points: getAllGenres() function

Git Commits

Make at least 3 git commits before the deadline. Each commit is worth 5 points. Please upload a link to your github repository URL to canvas.

Extra Credit - 6 points

Complete the saveMovieNames() function. It writes the 250 movies to a file called “movies.txt”. The movies are sorted alphabetically before they are written. Each movie is written on a separate line as shown below. The file is closed after writing the movies.



```
12 Angry Men
12 Years a Slave
2001: A Space Odyssey
3 Idiots
A Beautiful Mind
A Clockwork Orange
A Separation
Ace in the Hole
Akira
Aladdin
Alien
Aliens
All About Eve
Amadeus
American Beauty
American History X
Amores Perros
Amélie
Andhadhun
Andrei Rublev
Apocalypse Now
Avengers: Endgame
Avengers: Infinity War
Babam ve Oglum
Back to the Future
Barry Lyndon
Batman Begins
Before Sunrise
Before Sunset
Ben-Hur
```

NOTE: Please push the text file to the git repository too!