

## 1. Introduction

The project is about crawling IMDB's TOP 250 movies list on [http://www.imdb.com/chart/top?ref=nm\\_wl\\_img\\_3](http://www.imdb.com/chart/top?ref=nm_wl_img_3) and then exporting it as a txt file. The information includes the ranks, the titles, the years and the rates of the movies.

## 2. Requirements

- (1) mechanize : it needs to be run on Python 2.7.
- (2) lxml

## 3. Description of the Python program

- (1) class Imdb\_Crawler(object): crawling the top 250 movie list.
- (2) def \_\_init\_\_(self) : setting initial values
- (3) def find\_title(self) : using the module mechanize to get the html of the website. Xpath helps find the informations I want for the project, which are lists. Put those data into one list called list.
- (4) def writefile(self): writing the information I want into a txt file called movie.txt
- (5) def main(): the main funcation

## 4. Screenshots of the program output

```
import mechanize
import lxml.html

class Imdb_Crawler(object):

    def __init__(self):
        self.cur_url = "http://www.imdb.com/chart/top?ref=nm_wl_img_3"
        self.txtfile = "movie.txt"
        print "The program is ready to crawl the data of movies..."

    def find_title(self):
        b = mechanize.Browser()
        fd = b.open(self.cur_url)
        doc = lxml.html.fromstring(fd.read())

        self.Title = doc.xpath('//*[contains(concat(" ", @class, " "), "titleColumn", " ")]//a/text()')
        self.Year = doc.xpath('//*[contains(concat(" ", @class, " "), "secondaryInfo", " ")]//text()')
        self.IMDBRating = doc.xpath('//*[contains(concat(" ", @class, " "), "ratingColumn imdbRating", " ")]//strong/text()')
        self.list = []
        for x in xrange(0, len(self.Title)):
            self.list.append(self.Title[x])
            self.list.append(self.Year[x].strip("0"))
            self.list.append(self.IMDBRating[x])
```

Run imdbmovie

C:\Python27\python.exe D:/program/python/crawlingimdbmovie/imdbmovie.py

The program is ready to crawl the data of movies...

Done...

Process finished with exit code 0

```
Rank - Title - Year - Rating
1 - The Shawshank Redemption - 1994 - 9.2
2 - The Godfather - 1972 - 9.2
3 - The Godfather: Part II - 1974 - 9.0
4 - The Dark Knight - 2008 - 8.9
5 - 12 Angry Men - 1957 - 8.9
6 - Schindler's List - 1993 - 8.9
7 - Pulp Fiction - 1994 - 8.9
8 - The Good, the Bad and the Ugly - 1966 - 8.9
9 - The Lord of the Rings: The Return of the King - 2003 - 8.9
10 - Fight Club - 1999 - 8.8
11 - The Lord of the Rings: The Fellowship of the Ring - 2001 - 8.8
12 - Star Wars: Episode V - The Empire Strikes Back - 1980 - 8.7
13 - Forrest Gump - 1994 - 8.7
14 - Inception - 2010 - 8.7
15 - One Flew Over the Cuckoo's Nest - 1975 - 8.7
16 - The Lord of the Rings: The Two Towers - 2002 - 8.7
17 - Goodfellas - 1990 - 8.7
18 - The Matrix - 1999 - 8.7
19 - Star Wars - 1977 - 8.7
20 - Seven Samurai - 1954 - 8.7
21 - City of God - 2002 - 8.6
22 - Se7en - 1995 - 8.6
```

## 5. Conclusion

I used 6 Python main elements to write the program such as data structure list, classes, importing external modules , functions, list comprehension, and file input and output.  
The output is to check whether the program runs properly.  
The information I want to crawl is exported to the file movie.txt

## 6. Python program

imdbmovie.py