CMPE 326 Concepts of Programming Languages

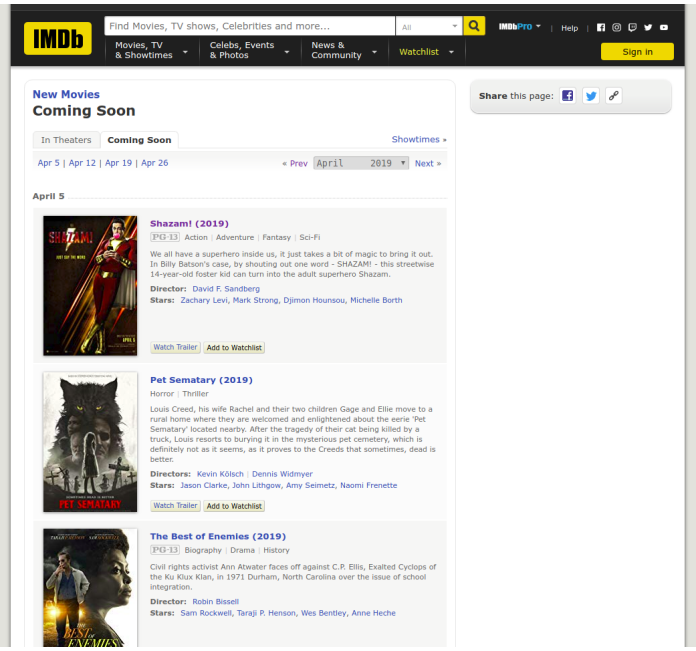
Spring 2019

Homework 1

Due date: 15/03/2019 23:59

In this homework you will develop a system in Python that processes semi-structured data from Web. You will use the parsed text from a web page that has information in a semi-structured way. Using Python you will fetch the necessary parts from this text and form a system organizes the data for later querying.

You will use the IMDB (Internet Movie Database) web site for getting information about upcoming films. For instance, below you can see a screenshot from the page `https://www.imdb.com/movies-coming-soon/2019-04/`. As depicted in the screenshot, Shazam! will be releasing on April 5. Its production year is 2019. You can see its genre information, which is represented by tags Action, Advanture, Fantasy, Sci-Fi. Some synopsis text along with information about its director and stars can also be found. Basically, given a text compiled out of a page like this, your system will fetch these information about films and answer queries of the user.



Here is a snippet of text parsed from the page depicted in the above screenshot. The complete text file can be found in Moodle as a sample input. You can check section on Implementation Language for a hint on generating more input data.

```
 Shazam! (2019)


Action
|
Adventure
|
Fantasy
|
Sci-Fi



    We all have a superhero inside us, it just takes a bit of magic to bring it
    out. In Billy Batson's case, by shouting out one word - SHAZAM! - this
    streetwise 14-year-old foster kid can turn into the adult superhero Shazam.

Director:

David F. Sandberg


Stars:
Zachary Levi,
Mark Strong,
Djimon Hounsou,
Michelle Borth
```

Your program must process an input parsed text to fetch necessary information about films and release dates. You need to form patterns to fetch specific information from the text. Python regular expressions (the `re` package from the standard library) is quite powerful and can be used for this task.

Your program should accept commands from the **standard input**. Of course, it is better for you to write test cases to files and direct them to the program as standard input. Here are the list of commands that you need to implement.

INPUT april_coming.txt

The `INPUT` command is used to give a parsed text file as input to the system; in the above case it is the `april_coming.txt` file. Note that the user can give more than one input file and a relative path can be given as an argument.

```
INPUT data/april_coming.txt
Loading data/april_coming.txt ...
INPUT data/may_coming.txt
Loading data/may_coming.txt ...
```

LIST

The `LIST` command is used for listing names of all films stored in the system. Below can be seen

a sample run.

```
LIST
Listing ...
Shazam!
Pet Sematary
The Best of Enemies
...
```

LIST from:2019-04-08

The `LIST` command with `from` argument is used for listing names of all films that have the release date given in the `from` argument and on; in the above case starting (and including) from 2019-04-08 and on. Below can be seen a sample run.

```
LIST from:2019-04-10
Listing from:2019-04-10 ...
Hellboy
Missing Link
...
```

LIST from:2019-04-05 to:2019-04-10

The above `LIST` command is similar to the previous command. It lists names of all films that have the release date in between (inclusive) the `from` and `to` arguments. Below can be seen a sample run.

```
LIST from:2019-04-05 to:2019-04-10
Listing from:2019-04-10 to:2019-04-10 ...
Shazam!
Pet Sematary
The Best of Enemies
Peterloo
The Biggest Little Farm
Teen Spirit
```

LIST genre:Action

The above `LIST` command lists names of all films of genre Action. In the `genre` argument there can be more than one genres that are separated by commas. In that case the films whose genres including all the ones in the argument should be listed, i.e., the semantics of comma is *and*. Below can be seen a sample run.

```
LIST genre:Action,Sci-Fi
Listing genre:Action,Sci-Fi ...
Shazam!
Hellboy
...
```

INFO Shazam!

The `INFO` command gets a film name as argument. The system outputs all the information about the film with that name. Here is a sample run.

```
INFO Shazam!
Info ...
Shazam!
Production year: 2019
Release date: 2019-04-05
Genre: Action, Advanture, Fantasy, Sci-Fi
Synopsis: We all have a superhero inside us, it just takes a bit of magic to
bring it out. In Billy Batson's case, by shouting out one word - SHAZAM! - this
streetwise 14-year-old foster kid can turn into the adult superhero Shazam.
Director: David F. Sandberg
Stars: Zachary Levi, Mark Strong, Djimon Hounsou, Michelle Borth
```

1 Implementation Language

You must use Python as the implementation language. You can appreciate some of the properties of Python while performing this task.

First of all since Python is dynamically typed language, you can prototype and develop programs fast.

Due to its scripting nature, it has powerful string processing features. This will be quite handy when you are implementing the part of the homework that you process the parsed web page text. You can rely on the well-documented and powerful standard library package *re* for writing regular expressions.

Additionally, since high-level data structures like *lists* and *dictionaries* are built-in for Python. Hence, you can benefit from them to form advanced data structures easily for storing and querying the data used in the homework.

Hints on generating input data: You can easily generate input data from the IMDb web site. The following snippet is used to generate the input text file (the file `april_coming.txt`). You need to install the BeautifulSoup 4 package for Python. Note that all the tests will use data created in this way.

```
>>> u = urllib.request.urlopen('https://www.imdb.com/movies-coming-soon/2019-04/')
>>> x = u.read().decode('UTF-8')
>>> soup = BeautifulSoup(x,'html.parser')
>>> f = open('april_coming.txt','w')
>>> f.write(soup.text)
>>> f.close()
```

Hints on working with dates: You can work with dates easily by using the *datetime* package from the standard library. Check *datetime.date.fromisoformat()* method for creating a date object from a text. You can do comparison between to date objects.

You will be provided with some test cases by your teaching assistant.

Submission

Each person must submit **his or her own work**.

You need to submit your Python file `hw1.py` using the course Moodle page.

You are **not allowed** to use special packages. You can only use modules from the standard Python libraries.

Your program will be evaluated using the Python 3.X (min version 3.6) interpreter.

You may be asked for a demo session.

The final submission must be one file named `hw1.py`. Do not submit any compressed file or a file having a different name.

There will be 1 day **questions fence** for this homework. You are not allowed to ask questions to the instructor or the teaching assistant in 1 day period before the deadline (i.e., during 15/3/2019).

Your submission will be graded w.r.t. the maximum points calculated according to the following formula: $100 - (2^{NumOfLateDays} \times 5)$.