# Read file function

The films.csv file that you found here is a movie database that contains records of movies, their genres, year of release, rating, and more. Take a good look at the file and familiarize yourself with all its fields. You want to create a code-based service that allows the user to specify the genre, the year, and the number of top rated movies they want to receive and returns to them a list of matching movies ordered by rating in descending order.

To solve this problem, develop the following functions:

read_file(pathname: str, year: int=0) - reads data from the file whose path is passed through the pathname argument, starting with the year year, and returns a list of lists, where each internal list corresponds to one movie entry. An example of execution is found below, noting that we are showing only the first two lists in the resulting list.

```
>>> read_file('films.csv', 2014)[:2]
[['1',
  'Guardians of the Galaxy',
  'Action,Adventure,Sci-Fi',
  'A group of intergalactic criminals are forced to work together to stop a fanatical warrior from taking control of the universe.',
  'James Gunn',
  'Chris Pratt, Vin Diesel, Bradley Cooper, Zoe Saldana',
  '2014',
  '121',
  '8.1',
  '757074',
  '333.13',
  '76.0'],
 ['3',
  'Split',
  'Horror,Thriller',
  'Three girls are kidnapped by a man with a diagnosed 23 distinct personalities. They must try to escape before the apparent emergence of a frightful new 24th.',
  'M. Night Shyamalan',
```

'James McAvoy, Anya Taylor-Joy, Haley Lu Richardson, Jessica Sula',
'2016',
'117',
'7.3',
'157606',
'138.12',
'62.0']]

Write read_file function.
Here are some examples from films.csv
Rank;Title;Genre;Description;Director;Actors;Year;Runtime (Minutes);Rating;Votes;Revenue (Millions);Metascore
1;Guardians of the Galaxy;Action,Adventure,Sci-Fi;A group of intergalactic criminals are forced to work together to stop a fanatical warrior from taking control of the universe.;James Gunn;Chris Pratt, Vin Diesel, Bradley Cooper, Zoe Saldana;2014;121;8.1;757074;333.13;76.0
2;Prometheus;Adventure,Mystery,Sci-Fi;Following clues to the origin of mankind, a team finds a structure on a distant moon, but they soon realize they are not alone.;Ridley Scott;Noomi Rapace, Logan Marshall-Green, Michael Fassbender, Charlize Theron;2012;124;7.0;485820;126.46;65.0
3;Split;Horror,Thriller;Three girls are kidnapped by a man with a diagnosed 23 distinct personalities. They must try to escape before the apparent emergence of a frightful new 24th.;M. Night Shyamalan;James McAvoy, Anya Taylor-Joy, Haley Lu Richardson, Jessica Sula;2016;117;7.3;157606;138.12;62.0
4;Sing;Animation,Comedy,Family;In a city of humanoid animals, a hustling theater impresario's attempt to save his theater with a singing competition becomes grander than he anticipates even as its finalists' find that their lives will never be the same.;Christophe Lourdelet;Matthew McConaughey,Reese Witherspoon, Seth MacFarlane, Scarlett Johansson;2016;108;7.2;60545;270.32;59.0
5;Suicide Squad;Action,Adventure,Fantasy;A secret government agency recruits some of the most dangerous incarcerated super-villains to form a defensive task force. Their first mission: save the world from the apocalypse.;David Ayer;Will Smith, Jared Leto, Margot

Robbie, Viola Davis;2016;123;6.2;393727;325.02;40.0
6;The Great Wall;Action,Adventure,Fantasy;European
mercenaries searching for black powder become embroiled
in the defense of the Great Wall of China against a horde of
monstrous creatures.;Yimou Zhang;Matt Damon, Tian Jing,
Willem Dafoe, Andy Lau;2016;103;6.1;56036;45.13;42.0
7;La La Land;Comedy,Drama,Music;A jazz pianist falls for an
aspiring actress in Los Angeles.;Damien Chazelle;Ryan
Gosling, Emma Stone, Rosemarie DeWitt, J.K.
Simmons;2016;128;8.3;258682;151.06;93.0
8;Mindhorn;Comedy;"A has-been actor best known for
playing the title character in the 1980s detective series
""Mindhorn"" must work with the police when a serial killer
says that he will only speak with Detective Mindhorn, whom
he believes to be a real person.";Sean Foley;Essie Davis,
Andrea Riseborough, Julian Barratt,Kenneth
Branagh;2016;89;6.4;2490; ;71.0

To implement the `read_file` function, we'll follow these steps:

1. Open and read the file located at `pathname`.
2. Parse the data using the semicolon ( `;` ) as the delimiter.
3. Filter records by year if provided (only records from that year onward).
4. Convert each row into a list of values and store them in a list.
5. Return the list of lists with each list representing a movie record.

Here's the Python code:

python

```python
import csv

def read_file(pathname: str, year: int = 0):
    movies = []
    with open(pathname, mode='r', encoding='utf-8') as file:
        reader = csv.reader(file, delimiter=';')
        headers = next(reader)  # Skip the header row

        for row in reader:
            movie_year = int(row[6])  # Year is the 7th field in the file (index 6)
            if movie_year >= year:
                movies.append(row)
```

```
    return movies
```

## Explanation:

1. **File Opening**: Opens the file at `pathname` in read mode with UTF-8 encoding.
2. **CSV Reader**: Uses `csv.reader` to handle the semicolon-delimited format.
3. **Header Skipping**: Skips the header row to only process movie data.
4. **Year Filtering**: Checks if the movie's year (7th field) is greater than or equal to the specified `year`.
5. **Appending Data**: Adds the movie entry to `movies` if it meets the criteria.

## Example Usage

Given `films.csv` in the format provided, calling:

python

```
read_file('films.csv', 2014)[:2]
```

should return:

python

```
[
    ['1', 'Guardians of the Galaxy', 'Action,Adventure,Sci-Fi', 'A group of intergalactic criminals are forced to work together to stop a fanatical warrior from taking control of the universe.', 'James Gunn', 'Chris Pratt, Vin Diesel, Bradley Cooper, Zoe Saldana', '2014', '121', '8.1', '757074', '333.13', '76.0'],
    ['3', 'Split', 'Horror,Thriller', 'Three girls are kidnapped by a man with a diagnosed 23 distinct personalities. They must try to escape before the apparent emergence of a frightful new 24th.', 'M. Night Shyamalan', 'James McAvoy, Anya Taylor-Joy, Haley Lu Richardson, Jessica Sula', '2016', '117', '7.3', '157606', '138.12', '62.0']
]
```

This function will provide the data we need to implement additional filtering and sorting based on genre and rating in the next steps.

don't use any additional libraries

If we need to avoid using any additional libraries, we can manually read and process the file. Here's an updated `read_file` function that reads the file line by line, splits each line by the semicolon delimiter, and filters by year without using the `csv` library:

```python
def read_file(pathname: str, year: int = 0):
    movies = []
    with open(pathname, mode='r', encoding='utf-8') as file:
        next(file)  # Skip the header row

        for line in file:
            row = line.strip().split(';')
            movie_year = int(row[6])  # Year is the 7th field in the file (index 6)
            if movie_year >= year:
                movies.append(row)

    return movies
```

## Explanation

1. **File Opening**: Opens the file at `pathname` in read mode with UTF-8 encoding.
2. **Skipping Header**: Uses `next(file)` to skip the header line.
3. **Processing Each Line**: Reads each line, strips any extra whitespace, and splits it by `;`.
4. **Year Filtering**: Checks if the movie's year is greater than or equal to `year` before appending to `movies`.

This code provides the same functionality without relying on the `csv` library.

make it follow pep8 documentation

To make the function follow PEP 8 documentation conventions, we'll add a docstring describing the function's purpose, parameters, and return values. Here's the revised version of `read_file`:

```python
def read_file(pathname: str, year: int = 0):
    """
```

```
    Reads movie data from a semicolon-delimited file and returns a list of movies.

    Each movie is represented as a list of values from the file.
    Optionally, only movies from the specified `year` onward are included.

    Parameters:
        pathname (str): The path to the file containing the movie data.
        year (int): The starting year for filtering movies (default is 0, meaning no
filter).

    Returns:
        list: A list of lists, where each inner list contains details of one movie.
    """
    movies = []
    with open(pathname, mode='r', encoding='utf-8') as file:
        next(file)  # Skip the header row

        for line in file:
            row = line.strip().split(';')
            movie_year = int(row[6])  # Year is the 7th field in the file (index 6)
            if movie_year >= year:
                movies.append(row)

    return movies
```

## Explanation of Changes

- **Docstring**: Added a multi-line docstring according to PEP 257 (PEP 8 documentation style).
- **Description**: The docstring explains what the function does, including its parameters (`pathname` and `year`) and return value.

This format adheres to PEP 8 documentation guidelines for readability and maintainability.