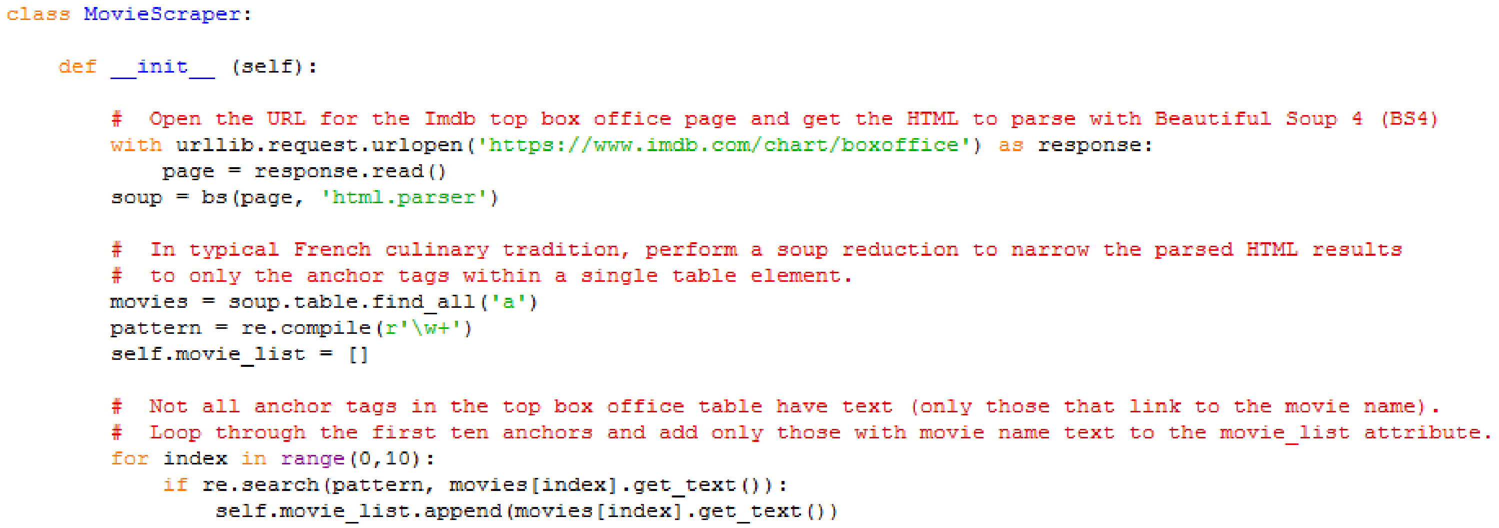
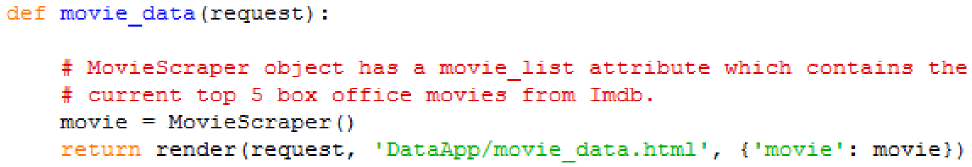
MovieScraper class written for DataScrape/DataApp/viewmodels.py to retrieve the current top 5 box office movies from the IMDb.com web site.



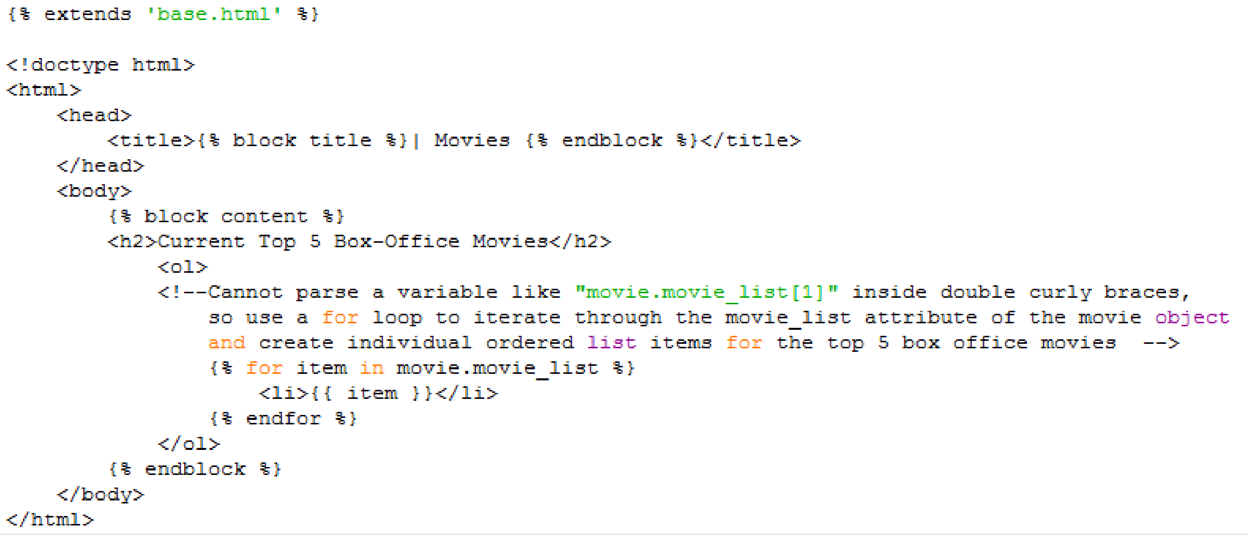
```
class MovieScraper:

    def __init__ (self):

        #  Open the URL for the Imdb top box office page and get the HTML to parse with Beautiful Soup 4 (BS4)
        with urllib.request.urlopen('https://www.imdb.com/chart/boxoffice') as response:
            page = response.read()
        soup = bs(page, 'html.parser')

        #  In typical French culinary tradition, perform a soup reduction to narrow the parsed HTML results
        #  to only the anchor tags within a single table element.
        movies = soup.table.find_all('a')
        pattern = re.compile(r'\w+')
        self.movie_list = []

        #  Not all anchor tags in the top box office table have text (only those that link to the movie name).
        #  Loop through the first ten anchors and add only those with movie name text to the movie_list attribute.
        for index in range(0,10):
            if re.search(pattern, movies[index].get_text()):
                self.movie_list.append(movies[index].get_text())
```

movie_data function written for DataScrape/DataApp/views.py to render user's request for current top 5 box office movies.



```
def movie_data(request):

    # MovieScraper object has a movie_list attribute which contains the list of the
    # current top 5 box office movies from Imdb.
    movie = MovieScraper()
    return render(request, 'DataApp/movie_data.html', {'movie': movie})
```

movie_data.html written for DataScrape/DataApp/templates/DataApp to display top 5 box office movies on user's browser.



```
{% extends 'base.html' %}

<!doctype html>
<html>
    <head>
        <title>{% block title %}| Movies {% endblock %}</title>
    </head>
    <body>
        {% block content %}
        <h2>Current Top 5 Box-Office Movies</h2>
            <ol>
            <!--Cannot parse a variable like "movie.movie_list[1]" inside double curly braces,
                so use a for loop to iterate through the movie_list attribute of the movie object
                and create individual ordered list items for the top 5 box office movies  -->
                {% for item in movie.movie_list %}
                    <li>{{ item }}</li>
                {% endfor %}
            </ol>
        {% endblock %}
    </body>
</html>
```