



## Python coding challenge : **Level # 2**

### Title: "Movie Ratings Analyzer"

#### Assignment Description:

In this Python programming assignment, you are tasked with creating a movie ratings analyzer that processes movie ratings data. The program should utilize lists, functions, dictionaries, and if-else statements to achieve the desired functionality. The assignment's main objective is to assess the students' understanding of requirements and their ability to figure out the logic.

#### Instructions:

Create a function called "**add\_movie\_rating**" that takes two arguments:

the movie name (string) and its corresponding rating (float).

The function should add the movie and its rating to a dictionary called "movie\_ratings."

Implement a function named "get\_average\_rating"  
that calculates and returns the average rating of all the movies in the "movie\_ratings" dictionary.

Develop a function called "recommend\_movie" that takes the "movie\_ratings" dictionary as input and recommends a movie based on the following criteria:

- If a movie's rating is above or equal to 8.5, recommend it as "Must Watch!"
- If a movie's rating is between 7.0 and 8.4 (inclusive), recommend it as "Recommended."
- If a movie's rating is below 7.0, recommend it as "Consider Later."

Create a function named "find\_highest\_rated\_movie" that searches for the movie with the highest rating in the "movie\_ratings" dictionary and returns its name.

Implement the `"main" function`, which will serve as the entry point of the program.

In the `"main" function`, prompt the user to input five movie names and their corresponding ratings.

Use the `"add_movie_rating"` function to store this data in the `"movie_ratings"` dictionary.

Print the average rating of all the movies using the `"get_average_rating"` function.

Call the `"recommend_movie"` function and pass the `"movie_ratings"` dictionary as an argument to get the movie recommendation.

Print the name of the highest-rated movie using the `"find_highest_rated_movie"` function.

Note: Remember to use appropriate variable names, user-friendly prompts, and descriptive function names to enhance the code's readability.

## Example Output:

Welcome to the Movie Ratings Analyzer!

Please enter the name and rating of 5 movies.

Movie 1 Name: Avengers: Endgame

Rating: 9.2

Movie 2 Name: The Shawshank Redemption

Rating: 8.7

Movie 3 Name: Inception

Rating: 8.9

Movie 4 Name: Jurassic Park

Rating: 7.5

Movie 5 Name: Twilight

Rating: 5.3

## Output :

The average rating of all movies: 7.72

Recommended movie: Avengers: Endgame - Must Watch!

Highest-rated movie: Avengers: Endgame

## Extra help

1. The `add_movie_rating` function takes two arguments: `movie_name` (string) and `rating` (float) and adds the movie name and its corresponding rating to the `movie_ratings` dictionary.
2. The `get_average_rating` function calculates and returns the average rating of all the movies in the `movie_ratings` dictionary by summing up the ratings and dividing by the number of movies.
3. The `recommend_movie` function iterates through the `movie_ratings` dictionary and recommends a movie based on its rating using if-else statements.
4. The `find_highest_rated_movie` function finds the movie with the highest rating in the `movie_ratings` dictionary and returns its name using the `max` function with a custom key.
5. The `main` function serves as the entry point of the program. It prompts the user to input five movie names and their corresponding ratings, stores the data in the `movie_ratings` dictionary, and then calls the other functions to print the results.
6. When running the program, the user will be prompted to input the movie names and ratings. The program will then calculate and display the average rating, recommend movies, and show the name of the highest-rated movie.

Note: Remember to run this code in your Python environment to interactively input the movie names and ratings as shown in the example output in the previous answer.

```

Def add_movie_rating(movie_name, rating):
    movie_ratings[movie_name] = rating

def get_average_rating(movie_ratings):
    total_ratings = sum(movie_ratings.values())
    num_movies = len(movie_ratings)
    return total_ratings / num_movies

def recommend_movie(movie_ratings):
    for movie, rating in movie_ratings.items():
        if rating >= 8.5:
            print(f"{movie} - Must Watch!")
        elif 7.0 <= rating <= 8.4:
            print(f"{movie} - Recommended.")
        else:
            print(f"{movie} - Consider Later.")

def find_highest_rated_movie(movie_ratings):
    highest_rated_movie = max(movie_ratings, key=movie_ratings.get)
    return highest_rated_movie

def main():
    print("Welcome to the Movie Ratings Analyzer!")
    movie_ratings = {}

    for i in range(5):
        movie_name = input(f"Movie {i+1} Name: ")
        rating = float(input("Rating: "))
        add_movie_rating(movie_name, rating)

    print("\nResults:")
    average_rating = get_average_rating(movie_ratings)
    print(f"The average rating of all movies: {average_rating:.2f}")

    print("\nMovie Recommendations:")
    recommend_movie(movie_ratings)

    highest_rated_movie = find_highest_rated_movie(movie_ratings)
    print(f"\nHighest-rated movie: {highest_rated_movie}")

if __name__ == "__main__":
    main()

```