 Python Programming

Unit 2 Lab: Control Flow

Overview

Welcome to the second unit lab! Remember, our goal is that at the end of the fifth lab, you're going to have an app that searches for movies to display all movies with titles containing the search term, or prints out the Rotten Tomatoes rating for any movie a user enters.

Right now, you have a variable to hold a movie title, a variable to hold a movie's rating, and a print statement to show the user.

Next, let's set up the functions and control flow to print out the values of our variables.

Deliverables

You're going to continue building this locally from the last lab. You'll write all of your code in the same `movie_app.py` file.

Note: It might be wise to save old copies of the code before starting each lab, so you can go back and see how things progressed.

Run the file from the command line to check your work.

Reminder: On your laptop, you can run the file from your command line with the following command:

```
python movie_app.py
```

> **Hint**: Make sure you are printing something out with the print statement! Otherwise, you won't see any output from running your program!

Requirements:

By the end of this, you will have edited your existing `movie_app.py`. At the top, you will have a variable called `mode`.

- If `mode` is equal to `search`, your app will then print `Back To The Future Blade Spirited Away`
- If `mode` is equal to `ratings`, your app will then print`The rating for Back To The Future is 8.`

```
The rating for Blade is 8.
The rating for Spirited Away is 8.
```

- **Bonus**: Accept user's input to determine the `mode` value at run time.

Directions

You'll augment the code you wrote for the Unit 1 lab, so leave your two variable declarations at the top of your program and don't delete the `print` statement!

1. Our program's going to get pretty complex. Let's have a definite starting point. At the bottom of your program, create one `main` function. From here, we'll call everything else.
2. In programming, if you have a `main` function, you can set it to automatically run when you start the program. In Python, there's a section of code that does this for us. At the very bottom of your file, put this code:

```
if __name__ == "__main__":
    main()
```

1. We want to make sure our code can support multiple movie titles. The best way to test code is to provide default values, run the code, and verify the output. Let's set up our test data by converting the `movie_title` variable to be called `movie_titles`, and replace the value with a list of your favorite movies. Our example uses `Back To The Future`, `Blade`, and `Spirited Away`, but please use whatever you'd like.
2. Create a function called `print_rating`, which will consume a movie title. This function should display the given movie title to the user along with the value stored in our `movie_rating`, in the following format: "The rating for is ". How can you test that this function works? Don't move on until you've run this code and verified it via the command line!!!!
3. Create a function called `print_all_ratings`, which will consume a list of movie titles, and will use the `print_rating` function to display each movie's rating.
4. Create a `print_search_results` function, which will consume a list of movie titles, and will print each title one after another.
5. Create a new global variable called `mode`, and set it equal to either "search" or "ratings". (**Hint:** Where do global variables go?)
6. Use our new variable to determine which functions should be called in our program. Remember that the `main` function is where we control the program execution from a high level.
7. **BONUS**: Do this if you have time! Use the `input()` function to allow the user to set the `mode` variable's value at run time.

You're done! Test it out to be sure you match the requirements above. Great job.