

TV Show Explorer Version 4: Episodes

Finally, your program will get and display information about the actors of a TV show.

What to do

- First, implement the `find_actor` function. This function will take a query as string argument, and a list of dictionaries with the cast crew of a show. It should be similar to the existing `find_show` function.

The function will:

- a) Access the dictionaries in the cast list to look for the matching actor based on the query.
 - b) Return a dictionary containing information about the actor of the matching result.
- Next, implement the function `format_actor_info`. This function will take a dictionary with the data about an actor as an argument, and return a string containing the properly formatted information of the actor with all necessary details.

For each actor, include the following information:

- Actor name
- Actor gender
- Actor birthday
- Actor country
- Role
- Character name

Here is an example of the formatted string that will be returned by this function:

Example 1

Jennifer Aniston (Female, born 1969-02-11 in United States) plays Rachel Green

Example 2

Leonard Nimoy (Male, born 1931-03-26 in United States) plays Mr. Spock

All functions defined in the template **must be present and implemented** in your code (you may **not** omit functions). You **may** add extra functions if needed.

```
import json

"""
New features in this version:
* Searching for an actor/character in the cast
* Showing detailed information about a selected actor/character
"""

def load_json(filename) -> dict:
    """
    Load JSON from a file
    """
    # Implement in ver. 1

def find_show(query: str, shows: dict) -> str:
    """
    Search for TV shows in the shows dictionary
    Return the name of the first (only one) result based on the query
    If the show is not found, return None
    """
    # Implement in ver. 1

def find_actor(query: str, cast: list[dict]) -> dict:
    """
    Search for an actor in the cast list
    Return the actor's data if found
    If the actor is not found, return None
    """
    # TODO: Implement the function

def get_show_data_by_name(show_name: str, shows: dict) -> dict:
    """
    Return the data for a show based on its name
    """
    # Implement in ver. 2

def format_show_details(show: dict) -> str:
    """
    Format the show name
    """
    # Implement in ver. 2

def get_cast_by_id(show_id: int) -> list[dict]:
    """
```

```

    Get the cast for a show
    """
    # Implement in ver.3

def format_cast(cast: list[dict]) -> str:
    """
    Format the cast
    """
    # Implement in ver.3

def format_actor_info(actor_dict: dict) -> str:
    """
    Format the actor's information
    """
    # TODO: Implement the function

def main():
    # TODO: Update main():
    # 1. Ask the user to input an actor of the selected show
    # 2. Display the selected actor's information of the selected season

if __name__ == '__main__':
    main()

```

Hints

- Your `main()` function can get quite long at this point. It's okay.

Program name

Save your program as `shows4.py`.

Demo

<https://asciinema.org/a/621400>

Testing

To make sure your program works correctly, you should test it.

- Run your program with `python shows4.py`. Type `friends`, then press Enter. The output of your program must resemble the example output below:

```
Found: Friends (1994 - 2004, comedy, romance)
Cast:
Jennifer Aniston as Rachel Green
Courteney Cox as Monica Geller
Lisa Kudrow as Phoebe Buffay
Matt LeBlanc as Joey Tribbiani
Matthew Perry as Chandler Bing
David Schwimmer as Ross Geller
```

Type `Jennifer Aniston`, then press Enter. The output of your program must resemble the output below:

```
Found: Jennifer Aniston (Female, born 1969-02-11 in United States) plays Rachel
Green
```

- Type `qwertyui`, then press Enter. Your program should print:

```
Can't find this actor or character in the cast!
```

Submitting

Submit `shows4.py` via eClass.

Copyright

I. Akhmetov, J. Schaeffer, M. Morris and S. Ahmed, Department of Computing Science, Faculty of Science, University of Alberta (2023).