UDACITY

PROJECT

Create a Movie Website

A part of the Intro to Programming Nanodegree Program

PROJECT REVIEW

CODE REVIEW 4

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Congratulations, the project meets specifications! Excellent work!

Here are some extra tips I'd like to provide:

- As it gets progressively more challenging, my best advice is to be patient. Sometimes it can take days of rereading before something clicks. Eventually everything will click, it just takes time. All of this stuff was invented in the human mind, but sometimes that's hard to believe.
- When something doesn't work, asking myself, "What can I do differently? What did I learn? What would work now?" opens up the door to new possibilities. These questions allow the brain to activate creativity as part of the learning process and use the growth mindset.

Awesome job, keep up the hard work. 👍 🍻

Functionality

The page contains all of the required elements (title, box with art imagery, and trailer).

Great work! ⭐

Here's a little hint for how to hack the `fresh_tomatoes.py` file and add some more information to the page, if you want to hack on it:

On line 147 of fresh_tomates.py, add the movie.storyline string

```
content += movie_tile_content.format(
    movie_title=movie.title,
    poster_image_url=movie.poster_image_url,
    trailer_youtube_id=trailer_youtube_id,
    storyline=movie.storyline # add a storyline
)
```

Then, in movie_title_content string above this:

```
   <h2>{movie_title}</h2>
   <p>{storyline}</p>
```

Page is generated dynamically from a Python data structure.

Code has no errors, bugs, or glitches.

Great job! 👍

No bugs or glitches.

Code Review

- **Code uses variables to avoid magic numbers.**
- **Each variable name reflects the purpose of the value stored in it.**
- **After initialized, the purpose of each variable is kept throughout the program.**
- **No variable overrides Python's built-in values (i.e., `def`).**

- **Functions are used as tools to automate tasks that are likely to be repeated**
- **Functions generate the appropriate output (usually with a return statement) from the appropriate input (function parameters).**
- **No function contains codes longer than 18 lines (not including blank spaces, comments, variable and function definitions).**

The appropriate data types are consistently used (strings for text, lists for ordered data, and nested lines, as appropriate).

The student demonstrates coding techniques like branching and loops appropriately (i.e., for loop in a list, `for element in list:`; for testing if something is in a list, `if name in list_names:`).

Code defines classes appropriately and uses instances of those classes in it.

All functions include a comment, which explains its expected behavior, inputs and outputs (if applicable).

Very thorough commenting! 👍

Commenting is an important aspect of programming many overlook. Good commenting and commenting a lot in the beginning will train your brain to think about code by making you put the code into words. So comment as much as you can and you'll build great habits that will improve not just your commenting skills, but your coding skills as well! That's my tip. It’s a little biased because my first employer kept complimenting my comments in my projects. I feel like he hired me for that reason as much as what my project did.

Here's a great resource for new Python programmers. The PEP8 style guide. It's the style guide from Python's website highly regarded for every Python pro. Reading it is one of the best things any new Pythonista could do. 😄 https://www.python.org/dev/peps/pep-0008/#comments

DOWNLOAD PROJECT

4 CODE REVIEW COMMENTS ›

RETURN TO PATH