

CSCI 366 (Database and Web Dev)

Dr. Schwartz

Lab 6: PHP

(Adapted from Web Programming Step by Step)

Due Wednesday, April 5th at 11:59pm, 100 pts

Introduction

For this assignment, you will write PHP code for movie review pages much like your TMNT page from the previous assignment. Your PHP code will allow you to generate reviews for a variety of movies using the same code. Turn in these files to files AutoLab (in a single .zip archive):

- movie.php, the PHP code to produce review pages for movies
- movie.css, the style sheet for movie.php (probably does not need to be modified from HW2 version)

You will recreate the page below. Your page must match the appearance specified in this document. Although it does not need to be green, your page should follow the styles specified in this document and match the look, layout and behavior shown here as closely as possible.

The overall page looks the same as it did in the previous assignment. The difference is that the page might show reviews and information for a film other than TMNT. For example, the following output shows The Princess Bride:

The screenshot shows a web page for 'The Princess Bride (1987)' on a site called 'Rancid Tomatoes'. The page has a green header with the site logo. The main content area has a yellow background with a green border. On the left, there's a 'FRESH 95%' badge. Below it, several review snippets are listed, each with a tomato icon and a quote. On the right, there's a 'GENERAL OVERVIEW' section with a movie poster for 'The Princess Bride'. Below the poster, there's a list of movie details: STARRING (Cary Elwes, Robin Wright, Andre the Giant, Mandy Patinkin), DIRECTOR (Rob Reiner), PRODUCER (Arnold Scheinman, Rob Reiner), SCREENWRITER (William Goldman), RATING (PG), RELEASE DATE (September 25, 1987 (USA)), RUNTIME (98 min), and SYNOPSIS (Director Rob Reiner breathes vividly colored cinematic life into William Goldman's THE PRINCESS BRIDE, effectively evoking the wondrous, wide-eyed spirit of the witty 1973 novel). At the bottom right, there are icons for W3C HTML and W3C CSS validation.

Rancid Tomatoes

The Princess Bride (1987)

FRESH 95%

"One of Reiner's most entertaining films, effective as a swashbuckling epic, romantic fable, and satire of these genres."
Emanuel Levy
emanuellevy.com

"An effective comedy, an interesting bedtime tale, and one of the greatest date rentals of all time."
Brad Laidman
Film Threat

"Based on William Goldman's novel, this is a post-modern fairy tale that challenges and affirms the conventions of a genre that may not be flexible enough to support such horseplay."
Variety Staff
Variety

"The lesson it most effectively demonstrates is that cinema has the power to turn you into a kid again. As we wish."
Phil Villarreal
Arizona Daily Star

"Rob Reiner's friendly 1987 fairy-tale adventure delicately mines the irony inherent in its make-believe without ever undermining the effectiveness of the fantasy."
Jonathan Rosenbaum
Chicago Reader

"My name is Marty Stepp. You killed my father. Prepare to die."
Marty Stepp
Step by Step Publishing

"One of the Top films of the 1980s, if not of all time. A treasure of a film that you'll want to watch again and again."
Clint Morris
Moviehole

GENERAL OVERVIEW

THE PRINCESS BRIDE

STARRING
Cary Elwes, Robin Wright, Andre the Giant, Mandy Patinkin

DIRECTOR
Rob Reiner

PRODUCER
Arnold Scheinman, Rob Reiner

SCREENWRITER
William Goldman

RATING
PG

RELEASE DATE
September 25, 1987 (USA)

RUNTIME
98 min

SYNOPSIS
Director Rob Reiner breathes vividly colored cinematic life into William Goldman's THE PRINCESS BRIDE, effectively evoking the wondrous, wide-eyed spirit of the witty 1973 novel.

RELEASE COMPANY
20th Century Fox

(1-7) of 7

W3C HTML
W3C CSS

Development

You should use your vm from this assignment. Under your www directory, create a directory called Lab6. Store your php and css files for this lab under this directory. I should be able to view your lab at this URL <yourVM>/Lab6/movie.php. Part of your grade will depend on this being true.

Images:

From AutoLab, you will download a .ZIP archive of input files for many movies, described on the next page. Unzip this file into the same directory as your Lab6 files, so that the files will be located in relative paths such as `tmnt2/info.txt` and `princessbride/overview.png`. Your code should assume these are the relative paths to use. (Don't put them in a `moviefiles/` folder. Put each film's folder, such as `tmnt2/` or `mortalkombat/`, directly in your Lab6 folder.)

The page uses the same images as Lab5, with two minor changes.

1. The `tmnt.html` page showed `rottenbig.png` next to its 33% rating. But now some films have high overall ratings. Any film whose rating is 60% or above should show `freshbig.png`, shown at right. This image can be found at http://cs.millersville.edu/~sschwartz/366/HTML_CSS_Lab/Images/freshbig.png
2. The `overview.png` has moved, since each movie has its own overview image. Although you should link to all of the "fixed" (or shared) images using their full absolute URLs (same as in the last lab), the overview images now come from the support ZIP file and reside in the folder with the movie's information. Link to these using a relative path such as `"tmnt2/overview.png"`.



All style elements not described here should be as specified in Lab5 or subject to the preference of the web browser. The screenshots in this document were taken on Windows in the latest version of Firefox, which may differ from your system.

Appearance and Behavior Details:

- **Title:** To make things simpler, the page's title should now always be just "Rancid Tomatoes".
- **URL and parameters:** Your `movie.php` page will show reviews of different movies based upon a query parameter named *film* that is passed from the browser to the page in its URL. The value of this variable should be a string representing the movie to display. The browser will request your page with a URL such as the following:

`movie.php?film=princessbride`

Your PHP code can store this parameter's into a variable using code such as the following:

```
$movie = $_GET["film"];
```

All of your PHP code should use this parameter's value, and you should never hard-code particular movie names. Your code may assume that the browser has properly supplied this parameter and has given it a valid value. You may assume that the movie exists and has a corresponding folder of valid input data and images. You do not have to handle the case of a missing or empty film parameter, a value that contains bad characters, a value of a movie that is not found, etc.

- **Files:** Each film is stored in a directory named the same as your query parameter. For example, the film `princessbride` stores its files in a folder named `princessbride/`. The files associated with each movie are the following. You may assume that the files exist and are valid at all times.
 1. `info.txt`, a file with three lines of information about the film: its title, year, and overall rating percentage. The information is used as the page heading and overall red movie rating number. Here's an example file:
The Princess Bride 1987
95
 2. `overview.txt`, a file with information to be placed in the General Overview section of your page. Each pair of neighboring lines contains the title and value for one item of information, to be displayed as a definition list term (dt) and its description (dd). The number of lines in the file varies from movie to movie. Example:
STARRING:Cary Elwes, Robin Wright, Andre the Giant, Mandy Patinkin
DIRECTOR:Rob Reiner
PRODUCER:Arnold Scheinman, Rob Reiner
SCREENWRITER:William Goldman
...

3. overview.png, an image to display at the top of the General Overview section. This is the movie poster for the film.
4. review1.txt, review2.txt, ..., files containing information for each review of the film. Each review file contains exactly four lines: The review, a fresh/rotten rating, the reviewer's name, and the publication. For example, review1.txt might store the following:
One of Reiner's most entertaining films, effective as a ...
FRESH
Emanuel Levy
emanuellevy.com

If a review has 10 or more reviews (such as tmnt2), the names will be e.g. review01.txt, Don't hard-code file names like "review1.txt"; look for all files that begin with "review" and end with ".txt".

For example, if your variable stores "princessbride", open princessbride/info.txt to read the film's title/year/etc. and display that on the page. Open princessbride/overview.txt and display its contents in the General Overview section on the right side. Then look for all review files in princessbride/ and display each in the reviews area.

- **Displaying Reviews:** Based upon the movie name variable's value, display reviews of that film. Different movies have different numbers of reviews. Show the first half of the reviews in the left column, and the rest on the right. If a movie has an odd number of reviews, the left column receives the extra review. For example, Princess Bride has 7 reviews, with the first four go on the left and the last three on the right. If a movie has only a single review, it goes in the left column. Do not worry about the possibility that the overview and reviews sections will be wildly different in height. (You may assume that the film has at least 1 review, but it might have 10 or more; you should display them all.) The green bar along the bottom of the page also should now display "(1 - N) of N" where N is the number of reviews for the current movie.

The image to show for each review is affected by the second line of the review's text file. If the critic gave the film a FRESH rating, display fresh.gif. If it's ROTTEN, display rotten.gif.

Development Strategy and Hints:

PHP code is difficult to debug if you have errors. Write the program incrementally, adding small pieces of code to a working page, and not advancing until you have tested the new code. A good strategy is to get your page to display a single film (such as princessbride or tmnt2) first, then generalize it for other films. None of your code should refer to specific names of films such as princessbride. The following functions may be helpful:

- count: returns the number of elements in an array
- explode: breaks apart a string into an array of smaller strings based on a delimiter
- file: reads the lines of a file and returns them as an array
- glob: given a file path or wildcard such as "foo/bar/*.jpg", returns an array of matching file names
- list: unpacks an array into a set of variables; useful for dealing with fixed-size arrays of data
- trim: removes whitespace at the start/end of a string (gets rid of stray spaces in output)

Submitting Your Assignment

Create a .zip file containing your movie.php and movie.css files and submit this to AutoLab. Be sure that you've used the proper relative paths for the relative file paths for the movie-specific files and the absolute paths for all shared image files.

Grading

- **HTML:** For full credit, your page must use **valid HTML5** and successfully pass the W3C HTML5 validator. (Not the PHP source code itself, but the HTML output it generates, must validate. To validate your page, you can view the page in the browser, then choose View Source in your browser and copy/paste it into the validator.) **Do not express style information in the HTML, such as inline styles or presentational HTML tags such as b, i, or font. Do not use tables for layout.**

- Your PHP code should not cause errors or warnings. The style of your PHP code will be graded (much like in a CSCI 362/162/161 assignment). Do not use the global keyword. Use indentation/spacing, and avoid long lines over 100 characters. Use material from the slides used in class and the first five book chapters. (Textbook Chapter 5 is a good resource.)
- Be mindful of "mode switches" between PHP code and HTML tags using `<?php` and `?>` . Produce as much of your output in HTML mode as possible; only switch to "PHP mode" to compute or print dynamic content. For full credit, you should reduce the number of large complex chunks of PHP code that are placed in the middle of your HTML code. When possible, replace such chunks with functions that are called in the HTML and declared at the bottom of your file.
- Your page should not contain any print and echo statements. Instead, insert dynamic content into the page using PHP expression blocks as taught in class.
- Another grading focus is redundancy. You should not have code that depends on particular movies or uses if/else statements to see which movie to display. Use loops, variables, if/else factoring, etc. to avoid redundancy.
- PHP commenting will also count in grading. More comments are expected here. Put a descriptive header (name, course, description) at the top of your code and comment each non-trivial section of PHP code explaining the purpose of that code.
- CSS is not a major part of this assignment, but your CSS file should validate and you should not introduce new bad/invalid CSS code.
- Publishing: Part of your grade will also come from successfully uploading your files to the web. You should use your vm for this assignment. Under your `/var/www` directory, create a directory called Lab6. Store your html and css files for this lab under this directory. I should be able to view your lab at this URL `<yourVM>/Lab6/movie.php` (with the appropriate file parameter).

Copyright © Marty Stepp/Jessica Miller, licensed under Creative Commons Attribution 2.5 License. All rights reserved.