

University of Washington, CSE 190 M, Spring 2009 Homework Assignment 3: Movie Review Part Deux due Wednesday, April 22, 2009, 11:30pm electronically

For this assignment you will write PHP code for movie review pages much like your *TMNT* page from HW2. Your PHP code will allow you to generate reviews for a variety of movies using the same code. For example:

Rancid Tomatoes™

The Princess Bride (1987)

FRESH **95%** (17 reviews total)

GENERAL OVERVIEW

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 17

W3C XHTML 1.1 ✓
W3C CSS 2.1 ✓

Turn in the following files:

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

Your pages must match the appearance specified in this document. It would be difficult to produce a pixel-perfect representation of the page that matches the image shown here, and we do not expect you to do so. But your page should follow all of the styles specified in this document and should match the overall look, layout, and behavior of the page shown here as closely as possible.

You should base your `movie.php` on the `tmnt.html` you wrote in HW2. You can also reuse your style sheet, `movie.css`. The modifications you need to make to your HTML code are to cause it to read the movie information using PHP so that the same file can produce different movie review pages.

Appearance Details:

The overall page has the same appearance as it did in Homework 2. The difference is that the page might show reviews and information for a film other than *TMNT*. The page's title should reflect the movie; for example, when showing *The Princess Bride*, the page title should be **The Princess Bride - Rancid Tomatoes**.

The page uses the same images as HW2 with the same file/path names, with two minor changes. The `tmnt.html` page showed a large `rottenbig.png` icon next to its 32% overall rating. But in this version, some films have high overall ratings. Any film whose **overall rating is 60% or above** should instead show the new image `freshbig.png`, shown at right.



Also, the `generaloverview.png` image has moved, for reasons we will describe in the next section. As before, you should link to all images using their full absolute URLs, not relative ones. The general overview images are the one exception; since these now reside in the folder with the movie's information, you should link to these using a relative path such as `"tmnt2/generaloverview.png"`.

Dynamic Page Content:

Your `movie.php` page will show reviews of different movies based upon a *query parameter* that is passed from the browser to the page in its URL. The browser will request your page with a URL such as the following:

```
https://webster.cs.washington.edu/your_uw_netid/hw3/movie.php?film=princessbride
```

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

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

You may assume that the browser has properly supplied this parameter and has given it a valid value. 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.

Based upon the film requested by the browser, you are to display a review of that film. Each film is stored in a directory whose name is the same as the `film` parameter. For example, the film `princessbride` stores its files in a folder named `princessbride/`. The files associated with each movie are the following:

- **info.txt**, a file with exactly four lines of general information about the film: its title, year, overall rating, and number of reviews. Example:

```
The Princess Bride  
1987  
95  
17
```

- **generaloverview.txt**, a file with information to be placed in the General Overview section of your page. Each line contains one item of information, with its title and value separated by a colon. 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
RATING:PG
RELEASE DATE:September 25, 1987 (USA)
RUNTIME:98 min
SYNOPSIS:Director Rob Reiner breathes vividly colored cinematic life into ...
RELEASE COMPANY:20th Century Fox

- **generaloverview.png**, an image to display at the top of the General Overview section. This is the movie poster for the film, such as the one shown at right.
- **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. Example `review1.txt`:

One of Reiner's most entertaining films, effective as a swashbuckling...
FRESH
Emanuel Levy
emanuellevey.com



You may assume that all of the above files exist and are valid for the films your page is displaying. Part of your task is to successfully open these files, read the information from them, and display that information on the page. For example, if your page is requested as `movie.php?film=princessbride`, you should open `princessbride/info.txt` to read the film's title/year/etc. and display that in the headings and other text on the page. You would open the file `princessbride/generaloverview.txt` and display its contents in the General Overview section on the right side of the page. You would look for all review text files in the `princessbride/` folder and would display each of them in the reviews area of the page.

Different movies have different numbers of reviews. You should show the first half of the reviews in the left column on the page, and the rest on the right. If a given movie has an odd number of reviews, the right column should receive the extra review. For example, Princess Bride has 7 reviews, so the first three go into the left column and the last four into the right column. You do not have to worry about the possibility that the general overview section and the reviews sections will be wildly different in height; this would potentially upset the appearance of the page, but your page does not need to handle this.

The image for each review are affected by the contents of the corresponding review text file. If the critic gave the film a FRESH rating, you should display an icon of `fresh.gif` in the review. If the critic gave the film a ROTTEN rating, you should display an icon of `rotten.gif` in the review.

Several other various pieces of text on the page is now related to the contents of these input files. For example, the display of the number of reviews shown, the number of total reviews, and the large red rating text all adjust based on the film's input files. The number of total reviews comes from the `info.txt` file, and the number of reviews shown comes from the count of the number of review text files.

Development Strategy and Hints:

PHP code is difficult to debug if you have lots of errors on your page. It is wise to write this program incrementally, adding small pieces of code to a known working page, and not advancing until you have tested and debugged the new code. Rather than trying to generalize all of your page at once, focus on incorporating a particular aspect or input file. Once this is successful, move on to the next.

We suggest that 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`.

Since this program uses PHP code, you will need to upload your files to Webster to test your work. Changes you make to your files will not be shown in the browser until you re-upload the file and refresh.

The following PHP functions may be helpful to you in implementing this program:

- [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)

Submission and Grading:

Submit your assignment online from the course web site. You still need to turn in your `movie.css` even if it did not change from HW2. For reference, our `movie.php` solution is roughly 115 lines long (80 without blank lines and comments), though you do not need to match this exactly.

Implement the HTML output of your web page using XHTML 1.1 as taught in class. For full credit, your page's output for all films must successfully pass the W3C XHTML validator. (Not the PHP source code itself, but the HTML output it generates, must validate.) Do not use HTML tables on this assignment.

Your PHP code should generate no error or warning messages when run using reasonable sets of parameters. Your PHP code should have good style similar to a CSE 14x assignment. Decompose the problem into functions as appropriate, minimize global variables, utilize parameters/returns properly, correctly use indentation/spacing, and avoid long PHP lines over 100 characters. Use material that discussed during the first three weeks of class or the first six chapters of the textbook.

A grading focus in this assignment is avoiding redundancy. One possible source redundancy is between different movies. You should not have code that depends on particular movies or that uses lots of `if/else` statements to figure out which movie to display; the same code should be able to display any movie. Use loops, functions, variables, `if/else` factoring, etc. to ensure that your code is not redundant.

Another grading focus on this assignment is commenting. HTML and CSS are simple declarative languages; in PHP you are implementing complex algorithms. Therefore you should have much more commenting in your PHP than we have expected so far, similar to Java commenting in CSE 142/143. Put a descriptive header (name, course, section/TA, description of assignment) at the top of your code, and a well-written comment on each major section of code and function explaining in detail what that code is doing.

For full credit, you should reduce the number of large complex chunks of PHP code that are injected into the middle of your HTML code. When possible, try to replace such chunks with functions that are called in the HTML and declared at the bottom of your file. Also, you should minimize the use of `print` and `echo` statements. As much as possible you should insert dynamic content into the page using PHP expression blocks as taught in class. You will lose points if you `print` any HTML tags into the page such as the following:

```
print "<p>I am adding content to the page!</p>"; # bad
```

Format your HTML and PHP code similarly to the examples from class. You must properly use whitespace and indent your XHTML and PHP code following examples shown in class. To keep line lengths manageable, do not place more than one block element on the same line or begin any block element past the 100th character on a line. The source code that is output by your PHP pages will be graded by the same criteria as normal HTML code, meaning that it should be valid XHTML, should be indented, and so on.

CSS is not a major part of this assignment, but you should not introduce new poorly written CSS code. Do not express stylistic information in HTML, such as inline styles or presentational HTML tags.

Part of your grade will also come from uploading your files to the Webster server at the following URL:

```
https://webster.cs.washington.edu/your_uw_netid/hw3/movie.php
```

Please do not place a solution to this assignment online on a publicly accessible web site.