New Perspectives on XML Comprehensive, 3rd Edition

Additional Project

Tutorial 6

Jenna has been working on creating a DVD library where she can list all of the DVDs that she has. Some she bought herself, others were gifts from her mom, dad, sister, grandpa, and grandma. The root element for her library is dvdlibrary with one or more dvd elements. Each dvd has a title, description, year, company, favorite characters, favorite bonus features, and purchase information. Her characters’ elements each have one or more character elements. Her bonus elements each have one or more features. Her purchased elements each have price, location, and date. The dvd element has two attributes, bonus element has one attribute, and the purchased element has no attributes.

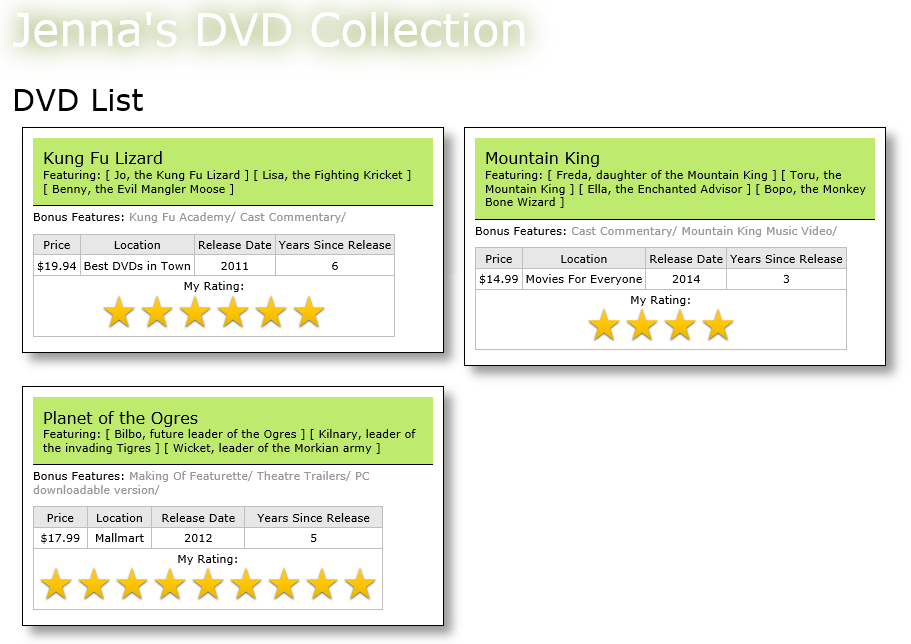
Jenna wishes to create an XSLT style sheet to use to display the DVD collection information attractively.She would like the price value to be clearly formatted as currency with a dollar sign. She would like the result document to show how many years it has been since each movie was released. Lastly, she would like to display a small chart illustrating her own review rating for each movie, based on a number from 1 to 10.

She has already created a CSS style sheet for the page. Complete the following:

1. Using your text editor, open the file ***dvdlibrary.txt***.Review the contents to familiarize yourself with the structure of the XML file.
2. Add a comment in the prolog with a description of the file, your name, date, and the filename.
3. Save the file as ***dvdlibrary.xml***.
4. Add a processing instruction after the comment section that attaches the ***dvdlibrary.xsl*** style sheet to this XML document. Close the file, saving your changes.
5. Using your text editor, open the file ***dvdlibrarytxt.xsl***.
6. Enter your name and the date in the comment section of the file, and save it as ***dvdlibrary.xsl***.
7. In the dvd template, locate the table cell that displays the value of the price element. Use the format-number() function to format the price value so it displays with a leading dollar sign.
8. Go to the top of the document and, directly after the opening <xsl:stylesheet> tag, insert code to create a global variable named *currentYear* to hold the value of the current year.
9. In the empty <td></td> tag at the bottom of the table modified in step 7, insert code to display the correct number of years since the movie was released. (Hint: Subtract the value of the year element from the value of the *$currentYear* variable.)
10. Below the dvd template, create a new named template named drawStars. Within the drawStars template:

* create two parameters named *starFile* and *starCount*
* use an xsl:if structure so that, as long as the *starCount* is greater than 0, the template continues to call itself, drawing the *starFile* image and decreasing the *starCount* value by 1 each time. (Hint: The structure should be similar to that used for the drawImages template in the tutorial.)

1. Scroll to the table modified in step 7. Below the <p> tag containing the text "My Rating:" (but before the </td> tag), insert code to use the call-template element to call the drawStars template using 'star.png' for the *starFile* parameter and rating for the value of the *starCount* parameter.
2. Save your changes to the file.
3. Generate a result document using either an XML editor or your web browser. Verify that the layout and content of each item matches that shown in the figure below:



1. Submit the completed files to your instructor.