New Perspectives on XML Comprehensive, 3rd Edition

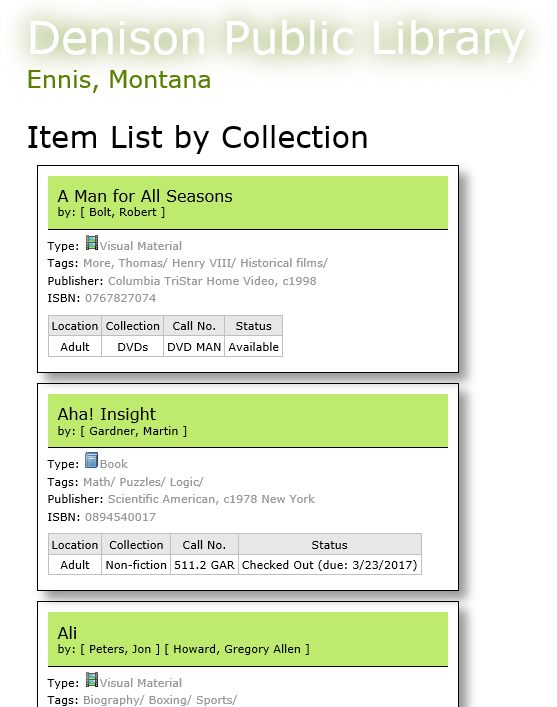
Additional Project

Tutorial 7

Maria Soares works at the Denison Public Library. She recently developed a basic XSLT style sheet that displays library items in an attractiveweb page. She would like to use more advanced techniques to improve the look and functionality of the page.

Maria would like to create a custom display of all library items organized by collection type. The top of the page should contain a list of all collection types, with a tally of how many items are available of each type. The body of the page should group the items by collection type, with clear headings separating each group.

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

1. Using your text editor, open the file ***itemstxt.xml***.Review the contents to familiarize yourself with the structure of the XML file.
2. Enter your name and the date in the comment section at the top of the file, and save the file as ***items.xml***.
3. Add a processing instruction after the comment section that attaches the ***library.xsl*** style sheet to this XML document. Close the file, saving your changes.
4. Using your text editor, open the file ***librarytxt.xsl***.
5. Enter your name and the date in the comment section of the file, and save it as ***library.xsl***.
6. Use your web browser or an XLST processor to generate the result document for the ***library.xml*** file based on the ***library.xsl*** style sheet. The figure below shows the initial appearance of Maria's library item web page.  
   
7. Go to the root template and, between the Item List by Collection h1 heading and the <xsl:apply-templates> element, insert the following code to generate a node set containing the unique list of collections. (Note the addition of the mode attribute to insure the collection item list is formatted uniquely from the general item list).

<section id="collection\_list">

|

<xsl:apply-templates

select="itemlist/item[not(collection=preceding::item/collection)]"

mode="collectionList">

<xsl:sort select="collection" />

</xsl:apply-templates>

</section>

1. Go to the bottom of the file and insert the following template directly before the closing </xsl:stylesheet> tag to display the list of each collection name:

<xsl:template match="item" mode="collectionList">

<xsl:value-of select="collection" /> |

</xsl:template>

1. Refresh the result document in your web browser. A list of the three collection types (DVDs, Fiction, and Non-fiction) should appear below the "Item List by Collection" heading.
2. Next, Maria would like to add a tally of how many items are available in each collection type. To do this, first you will create a key indexing all of the collection elements.
   1. Return to ***library.xsl*** and, directly below the opening <xsl:stylesheet> tag, insert the code to create a key named collectionNames that matches the item element and uses the collection element as the values to be used in the key's index table.
   2. Save your changes to the file.
3. Now, you can use the collectionNames key to add a tally of items by collection to the list at the top of the page. Return to the item template belonging to the collectionList mode at the bottom of the style sheet. Directly before the | character, insert the following code, leaving a spacebefore the | character:

(<xsl:value-of select="count(key('collectionNames', collection))" />)

1. Refresh the result document in your web browser. Confirm that a tally displays for how many of each collection type is available (5 DVDs, 5 Fiction, and 10 Non-fiction).
2. Next, Maria would like the items to be grouped on the page by collection type. You will use Muenchian grouping to achieve this result.  
     
   First, you will generate a sorted list of collection names formatted as headings.Go to the root template, and directly above the closing </section> tag and above the closing </body> tag, delete the three lines that apply the item template:

<xsl:apply-templates select="itemlist/item">

<xsl:sort select="title" />

</xsl:apply-templates>

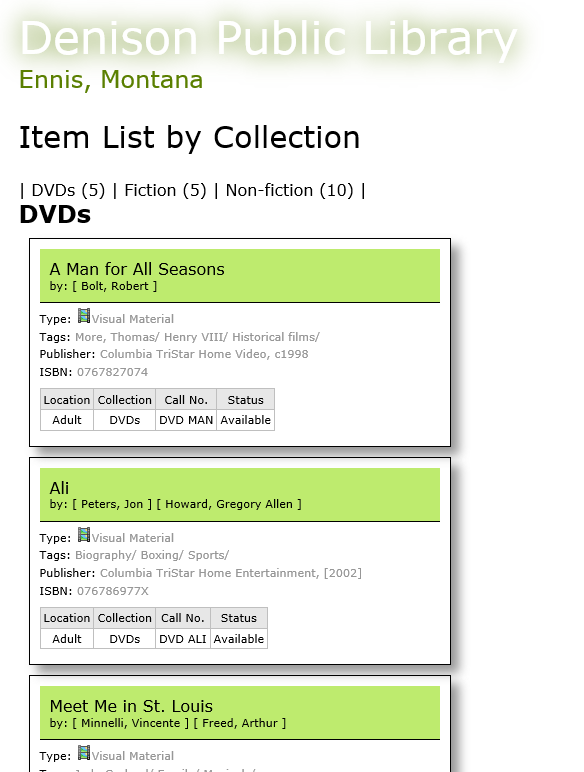
1. Replace the deleted code with a for-each loop that generates output for each unique collection name. Within the for-each loop, include code to sort by collection and to display the name of each colelction as an h2 heading. (Hint: Refer to the Employing Muenchian Grouping Reference box on page XML 433 in the text for any assistance in code syntax for the select attribute of the for-each loop.)
2. Lastly, you will display the descriptions for each item within each collection group, sorted by title. Within the for-each element in the root template, insert the following code below the h2 heading:

<xsl:apply-templates select="key('collectionNames', collection)">

<xsl:sort select="title" />

</xsl:apply-templates>

1. Save your changes to the file.
2. Refresh the result document in your web browser. Verify that the layout of the web page matches that shown in the figure below:



1. Submit the completed files to your instructor.