

Instructions for Wisconsin Museums 4 Coding Assignment

Assignment

The goal of this assignment is to create a printable PDF document that contains formatted information regarding Wisconsin Museums. You will achieve this by creating an XSLT stylesheet and using that stylesheet to transform an XML instance document regarding Wisconsin Museums into an XSL-FO page. Using the Oxygen XML Editor and the XEP print formatting engine, you will generate the PDF output document

The files that you will need to begin this coding assignment have been made available as part of a downloadable Oxygen project .zip file in the Weekly Schedule. The project file is named “wisconsin_museums_4_starter_code.zip”. The files provided include the following:

```
imls_wisconsin_museums_1Q2015.xml
museums.xsd
generate_museum_report.xsl
wisconsin_museums.html
museum_style.css
```

You will be expected to create an XSLT stylesheet (.xsl) and to use that stylesheet to transform the XML instance document. The result of the transformation should be a well-formatted XSL-FO document that presents that information regarding Wisconsin Museums that originated on the XML instance document. Because you will be performing the entire transformation and PDF generation with a single Oxygen XML Editor transformation scenario, the XSL-FO document created from your XSLT stylesheet will be a temporary file and not easy for you to locate. Yet, the final output of the XSL-FO print formatting process should be a proper PDF document that is easy for you to locate.

In all, you should have coded or generated 2 files when you have completed the exercise:

```
generate_print_version_museum_report.xsl
wisconsin_museums.pdf
```

Method

Use the method that was demonstrated in my tutorial video on creating XSLT stylesheets for the creation of XSL-FO documents. In that video, I take the approach of converting the last version of the XSLT transformation (that created an HTML5 page) into a new version of the XSLT transformation (that creates an XSL-FO page). It is especially important to develop your stylesheet incrementally, testing it as you add each feature. XSLT processing engines are notoriously bad at identifying errors in your code. This problem is made even worse by the fact that we are now performing one additional

processing operation, the rendering of the XSL-FO document to create a PDF document. So, you do best to add just a little bit of code. Then, test the new code until you get the desired results.

Please take special care when coding for optional elements in the input document.

Since we will mainly be following the navigational stylesheet design pattern, we will need to test for the presence or absence of optional elements with the `<xsl:if>` statement. Remember that all code to be inserted into the output document for an optional element must be placed inside of the `<xsl:if>` statement. This includes any `<fo:block>` statements.

The XML instance document includes records for a large number of museums. While your final test should include all of the museum data, you may find it easier to do your early testing with a subset of museum data. You can create a subset simply by placing an XML comment around the museum data that you want to exclude:

```
<!-- Place comment text insider here -->
```

When deciding how to style your print output, use my demo work as an example of the level of design that is appropriate.

Please remember to use the `xsl:sort` element within your `xsl:for-each` code to put your report into sorted order based upon one of the names available for all museums.

Please remember to add the `keep-together` attribute to the `fo:block` element that holds the grouping of lines for each museum. This will prevent the print output for a single museum from being split between two pages of output.

Tools

Use the Oxygen XML Editor to create the XSLT document. Use an Oxygen transformation scenario to generate your PDF document.

Length

Your final test should use all of the museums in the XML instance document. Make sure that your XSLT code makes allowances for all optional elements that occur in the test data. Refer to the schema provided to see which elements are optional. Make sure that the `keep-together` attribute is being used to keep entries from a single museum from spanning more than one page.

Format and File Naming Conventions

When unzipped, the Oxygen project directory that is available in the Weekly Schedule will be named:

wisconsin_museums_4_starter_code

Copy and/or rename this directory so that it has a name that takes the following form:

trainor_kevin_wisconsin_museums_4

When your work is complete, zip this directory and submit it to the D2L dropbox. The name of the file submitted should take the following form:

trainor_kevin_wisconsin_museums_4.zip

Due By

Please submit this assignment by the date and time shown in the Weekly Schedule.