

Basic Transformations with XSLT

Exercise 1: Create a stylesheet in Oxygen

1. Open the TEI document(s) you've been working on in Oxygen
2. Select File > New
3. Select XSLT Stylesheet, tick Save as, and choose a location to save it. Use your name as the filename (e.g. andrew.xml)
4. Add the following to the `xsl:stylesheet` element:
`xpath-default-namespace=http://www.tei-c.org/ns/1.0`
5. Add the following as a child of the `xsl:stylesheet`:
`<xsl:import href="https://raw.githubusercontent.com/andrew-morrison/xsl_training/master/base.xml"/>`
6. Select Window > Open Perspective > XSLT Debugger
7. Choose a TEI file as the XML, choose your .xml file as the XSL, and choose a location to save the output (as a .html file)
8. Select Saxon-HE 9.x.x.x
9. Select Debugger > Run (or click the blue left-arrow in the toolbar)
10. Open the output file in a web browser.
You should see your TEI converted to HTML, with very limited formatting, just enough to display the hierarchy. This is provided by the base.xml stylesheet imported into your own.

Exercise 2: Customize by creating simple templates

Add templates to your XSL file to override the default display:

1. Start with something simple: filter out something you do not want in the HTML at all.
2. Then skip over something but display its contents.
3. Create simple one-to-one mappings of TEI elements to HTML elements.
4. Create a template which chooses what to output based on one or more conditions.
5. Create a template which re-sequences child elements using a for-each loop.

Test each new template by selecting Debugger > Run. If it fails, look at the error messages. Oxygen will also highlight any problems with red underlining.

Exercise 3: Adding a table of contents

1. Review your TEI document to decide what you want to include in a table of contents (e.g. *msItem* elements in a *teiHeader*? *div* elements in *body*?)
2. Each should have an *xml:id* attribute in the source document. If not, add these now.
3. Start a new template in your XSL file, matching the root *TEI* element.
4. This should contain two *xsl:apply-templates* instructions, the first with a mode of "toc", the second without a mode.

5. Enclose the first one with `` tags (this is the HTML code for bullet-point lists).
6. Create another template, with a mode of "toc", which matches whichever TEI element you've decided to include in the table of contents.
7. Within that, add `` tags, and inside those `<a>` tags followed by sibling `` tags.
8. Inside the `<a>`, add an `xsl:value-of`, selecting whatever you want to display for each in the table of contents (e.g. the *title* of an *msItem*, or the *head* of a *div*.)
9. Give the `<a>` tag a `href` attribute containing `#{ @xml:id }`
10. Inside the ``, add an `xsl:apply-templates` instruction with a mode of "toc".
11. Create another template with a mode of "toc", matching "*", containing only an `xsl:apply-templates` instruction, itself with a mode of "toc".
12. Create a final template with a mode of "toc", matching "text()", containing nothing.
13. Select Debugger > Run to test whether the table of contents has been generated correctly.

Exercise 4: Combining everyone's XSL

1. Select Document > Validate > Validate to check you XSLT
2. Zip up your XSL and TEI (if you don't mind sharing, this is optional)
3. Email it to me.
4. I will combine everyone's stylesheets and we can see what the result is.