TEI Transformation Exercise

All the exercises will be based on formatting a BNC document containing a collection of advertising leaflets. This is the file `cfr.xml` in the `Samples` directory.

1 First look at using XSLT

For these exercises, we will be using the Mozilla Firefox web browser to transform our XML files as we view them (this will also work with Internet Explorer 5.5 or later).

1. In the Samples directory is a file called `cfr.xml`; open this in your editor and check it looks like a TEI XML document.

2. Now open `cfr.xml` in Mozilla Firefox or Internet Explorer (start the browser, then use the File/Open File menu). This should show you the XML structure (it's a large file with a lot of tags, so it may take a while to load).

3. Now edit the file and add, before `<bncDoc>` the line

   ```
   <?xml-stylesheet type="text/xsl" href="transform.xsl"?>
   ```

   This is an instruction to the browser to process the file using a stylesheet called transform.xsl. Save the file, and then reload it in the browser. Did your changes take effect? What has happened?

4. Open the file transform.xsl in your editor and consider how to format it better. Each leaflet in the original starts in a new `<div1>` element, so one simple thing to do would be just to put out an HTML `<h1>` tag with a label such as 'Leaflet' and a sequence number at the start of each of them . Go to the end of transform.xsl and add this before </xsl:stylesheet:

   ```
   <xsl:template match="div1">
     <h1>Leaflet number <xsl:number/></h1>
    <xsl:apply-templates/>
   </xsl:template>
   ```

   (You'll notice that Oxygen is just as helpful when editing XSLT files as it is for XML files by the way). Now save the XSLT file, and reload the data file in the browser. Did it work?

5. You may have noticed that the file `transform.xsl` also contains a template for the `<teiHeader>` element. What is it doing?

You should now be confident that you can control the display of an XML file using the XSLT stylesheet. Now we can move on to refine the stylesheet.

If you want to run the transformation statically and make an HTML file, you can configure oXygen to do it for you:

- Go to the `Document` menu, then `XML Document`, then `Configure Transformation Scenario`, and choose `New`
- Tick the box next to `Use "xml-stylesheet" declaration"`
- Open the `Output` tab, and select `Prompt for file`
- click OK, and then OK again

You can now run the transformation by clicking the  button. It will prompt you for a file name to save the HTML in.

2 Exercises part 2

Your initial set of tasks is to make the leaflets easier to read:

1. Print the main headings in bold. How? You'll need a new template like this:

```
<xsl:template match="head[@type='MAIN']">
 <h2><xsl:apply-templates/></h2>
</xsl:template>
```

2. Start each sentence (`<s>`)on a new line.
3. Put each caption (`<caption>`) in a block quote and italicize it
4. Replace each `<gap>` element by the content of its desc attribute enclosed in square brackets
5. Improve the formatting or information display in any other way that interests you; how about numbering the sentences for example?

3 Exercises part 3

Now that the leaflets are readable, let us consider how we can present them in different ways. Most of these exercises are much simpler than they may seem at first sight!

1. Produce a table of contents for this collection of leaflets. Use the number of each leaflet as a way of linking from the table of contents into the collection.
2. Produce the same list, sorted alphabetically by the main heading of each leaflet.
3. Produce the same list, but include in it a count of the number of sentences (`<s>` elements) in each leaflet.
4. Reformat the `<revisionDesc>` in the header as an HTML table
5. The header contains a taxonomy of "text-types". Make an index organized by text type, containing links to leaflets of each type and a count of the total number of sentences they contain.

4 Exercises part 4

As you know, the file `leaflets.xml` is richly encoded. Yesterday you saw how Xaira made use of the annotation for POS codes etc. We can process this annotation in just the same way using XSLT. We can also try to improve on the XML encoding itself.

1. Produce a print out of the text in which each word is replaced by its lemma or by its POS code

2. Produce a list of the headings of leaflet sections containg sentences with no verbs.

3. Because of a mixup during the production of the BNC, some editorial comments in these leaflets have been included in the text. They are marked up as `<note>` elements with a type of `ED`. First produce a list of them, together with the heading of the division they belong to; then produce a corrected XML file in which each of these notes is replaced by an XML comment containing the same markup.