

## TEI Transformation Exercise

All the exercises will be based on formatting the TEI XML file which describes gravestones in a cemetery, the people commemorated on them, and the inscriptions. This is the file `data.xml` in the `samples` directory, which is a subset of the cemetery data. You can safely ignore most of it. A separate document shows you what is in it.

### 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 `data.xml`; open this in your editor and check it looks like a TEI XML document.
2. Open `data.xml` in Mozilla Firefox (start the browser, then use the File/Open File menu). Mozilla should show you the XML structure.
3. Now edit the file and add, before `<teiCorpus>` the line

```
<?xml-stylesheet type="text/xsl" href="ex1.xsl"?>
```

This is an instruction to the browser to process the file using `ex1.xsl`. Save the file, and then reload it in the browser. Did your changes take effect? What has happened?

4. Open the file `ex1.xsl` in your editor and consider how to break up the display for each gravestone. Let us make a new `<h2>` division titled 'Stone' plus the value of the 'id' attribute. Go to the end of `ex1.xsl` and put in this before `</xsl:stylesheet>`:

```
<xsl:template match="tei:TEI">
  <h1>Stone <xsl:value-of select="@xml:id"/></h1>
  <xsl:apply-templates/>
</xsl:template>
```

Now save the XSL file, and reload the data file in the browser. Did it work?

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 run the processor on the command line. Thus to make `sample.html` from `data.xml` using `ex1.xsl` as your stylesheet, you would type

```
xsltproc -o sample.html ex1.xsl data.xml
```

You can type in a command line using the Tools/Shell Command menu in Emacs, if you like.

### 2 Exercises part 2

Your initial set of tasks are as follows:

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

```
<xsl:template match="tei:persName">
  <b><xsl:apply-templates/></b>
</xsl:template>
```

2. Put spaces between surname and forename
3. Put each inscription in a block quote, and put a break at the end of every line
4. Omit everything from the <person> except the contents of the <persName>, and put each person in a separate paragraph
5. Put the <death> element nicely after each person's name, and show their sex in brackets
6. Improve the formatting or information display in whatever way interests you; can you, for instance, left-justify the lines which have a with a 'rend' attribute with a value of 'Alignl'?

### 3 Exercises part 3

Now that we have the gravestones looking quite respectable, 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 the catalogue as we left it in the final exercise of part 1; give each heading an identifier (the stone number will do), and produce a list of numbers at the start, linked to the right stone.
2. Produce the catalogue sorted in order of surname and forename of people commemorated
3. Produce the ordinary catalogue, but append an index of surnames (sorted in alphabetical order), where each name is an HTML link to the right stone number
4. Produce a summary table listing the stone number, the number of people commemorated, and the number of lines of inscription

### 4 Exercises part 4

Now it starts getting a little less obvious:

1. Do the same catalogue that we produced at the end of the second set of exercises, but show only the *first* person and the *first* inscription. If there are other people or inscriptions, put in a paragraph saying eg 'there are *n* other people commemorated'. When you come to print an inscription, *number* the lines.
2. Make an exact copy of the input file, but put the stones in order of the year of death of the first person commemorated

3. Using the file from the previous exercise, produce a catalogue of years in which people died. The year should be a <h2> heading, and should be followed by a numbered, list of the people (just surname and forenames) who died in that year, sorted by surname.
4. Can you do the last two exercises in a single step, with no intermediate file?