

# XSLT Publishing Exercises 2



Sebastian Rahtz

February 2004

## 1 Basic exercises

In your sample files, you will find an XSLT stylesheet called `punch-tei.xml`. Run this stylesheet with the command-line

```
xsltproc -o punch.html punch-tei.xml punch.xml
```

and look at the resulting `punch.html` with your browser. The stylesheet is calling a set of generalized XSLT stylesheets for rendering TEI documents. You can alter the effects by editing `punch-tei.xml`, which has a series of high-level variables. We suggest you start by trying

- Change

```
<xsl:param name="splitLevel">-1</xsl:param>
```

to

```
<xsl:param name="splitLevel">0</xsl:param>
```

and see the sections split into separate files (this will NOT work if you look at the XML file in IE5)

- Change

```
<xsl:param name="numberHeadings"></xsl:param>
```

to

```
<xsl:param name="numberHeadings">true</xsl:param>
```

to see how the section numbering appears.

- Change

```
<xsl:param name="leftLinks"/>
```

to

```
<xsl:param name="leftLinks">true</xsl:param>
```

and

```
<xsl:param name="autoToc">true</xsl:param>
```

to

```
<xsl:param name="autoToc"></xsl:param>
```

Observe how the table of contents is now in a table column on the left

- Write a template matching the `<speaker>` element, starting like this:

```
<xsl:template match="speaker">
```

Finish it to output a centred statement '[xx speaks here]'

- Write a template in `punch-tei.xml` to output the line numbers before each line of verse.
- In your new template for `<l>` elements, create an `xsl:if` condition before the `<xsl:apply-templates>` instruction. Use this to output several hard spaces (&#160;) before the text of even-numbered lines. This is not easy. You'll need to save the result of `<xsl:number>` in a variable and examine it.

There is a full explanation of these variables at <http://www.tei-c.org.uk/Stylesheets/teixsl.html>.

You can use the same stylesheet techniques on the `sample.xml` file, or one of your own.

## 2 Exercises for advanced students

We have so far concentrated on rendering the TEI XML in HTML for display on the web. That is only one of the things we can do with XSLT, however. The aim of this set of exercises is to stretch your scripting skills a bit more. We show the first one worked out, your task is to attempt the others. In each case, we will write an HTML file containing the results.

- Count the number of pages in each chapter of A Christmas Carol. This is a simple use of the count function.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                version="1.0">

  <xsl:output method="html"/>

  <xsl:template match="/">
    <html>
      <body>
        Pages: <xsl:value-of select="count(../pb)"/>
      </body>
    </html>
  </xsl:template>

</xsl:stylesheet>
```

- Make a sorted list of all the `<soCalled>` elements in A Christmas Carol
- Extend that list of `<soCalled>` to give the page number next to each one
- List the `<div>` elements in Punch which have more than one `<head>`
- Write out a table of the `<l>` elements in Punch sorted in order of size (remember to use the data-type attribute of `<xsl:sort>`)
- Can you make a table of the first word of each paragraph in Dickens?