
Lab 05

Please note:

For the manipulation of a *source.xml* document by the *model.xsl* model you can use the command:

xsltproc model.xsl source.xml > sortie.html (.xml or .txt)

Exercise 01: We consider the following XML document:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<biblio>
<livre>
<!-- Élément enfant titre -->
<titre>Les Misérables</titre>
<auteur>Victor Hugo</auteur>
<nb_tomes>3</nb_tomes>
</livre>
<livre>
<titre>L'Assomoir</titre>
<auteur>Émile Zola</auteur>
</livre>
<livre lang="en">
<titre>David Copperfield</titre>
<auteur>Charles Dickens</auteur>
<nb_tomes>3</nb_tomes>
</livre>
</biblio>
```

1. Define a *biblio.xsl* stylesheet to produce the display as follows: title, book names in bold, book elements separated by line breaks, book elements separated by paragraph marks):

<p>Ma Bibliothèque</p> <p>Titre : Les Misérables Auteur : Victor Hugo Nombre de tomes : 3</p> <p>Titre : L'Assomoir Auteur : Émile Zola Nombre de tomes :</p> <p>Titre : David Copperfield Auteur : Charles Dickens Nombre de tomes : 3</p>

2. Define *biblio2.xsl* stylesheet in order to arrange the books in alphabetical order, without displaying the number of volumes when it is not filled in, and indicate when a book is in English. Produce the display as an HTML table as below:

Ma Bibliothèque

Titre: David Copperfield
Auteur: Charles Dickens
Nombre de tomes: 3
Ce livre est en anglais.

Titre: L'Assomoir
Auteur: Émile Zola

Titre: Les Misérables
Auteur: Victor Hugo
Nombre de tomes: 3

3. Define a third stylesheet *biblio3.xsl* to produce the display as an HTML table as below.

Ma bibliotheque

Titre	Auteur	Nombre de tomes	Langue
David Copperfield	Charles Dickens	Nombre de tomes :3	<i>Ce livre est en anglais.</i>
L'Assomoir	Émile Zola	-	-
Les Misérables	Victor Hugo	Nombre de tomes :3	-

Exercise 02:

We consider XML files which contain a list of documents on the internet. Each document is associated with its type (web page or image), its URL and a brief description of the document. These XML files follow the following docs.dtd DTD:

```
<!ELEMENT docs (doc*) >
<!ELEMENT doc (type, url, description) >
  <!ATTLIST doc number ID #REQUIRED>
  <!ELEMENT type (#PCDATA) >
  <!ELEMENT url (#PCDATA) >
<!ELEMENT description (#PCDATA) >
```

For example, the example.xml file below contains two documents (one of type “web page” and another of type “image”):

```
<docs> <doc number='1'>
  <type>webpage</type>
  <url>http://www.upmc.fr</url>
<description>UPMC website</description>
</doc>
<doc number='2'>
<type>image</type>
  <url>http://www.upmc.fr/logo.jpg</url>
<description>New UPMC logo</description>
</doc>
</docs>
```

Note:

The example above contains only two documents, but you will answer the questions so that your XSL sheets can process any XML files in *docs.dtd* format.

1. Write an XSL sheet that transforms an XML document (which follows the docs.dtd DTD) into another XML document containing only a list of URLs. This new document follows the DTD:

```
<!ELEMENT addresses (ad*) >
```

```
<!ELEMENT ad (#PCDATA)>
```

For example, the result obtained for the exe file mple.xml is:

```
<addresses>
```

```
  <ad>http://www.upmc.fr</ad>
```

```
  <ad>http://www.upmc.fr/logo.jpg</ad>
```

```
</addresses>
```

2. Write an XSL sheet that generates a new XML document (which respects the DTD docs.dtd) containing only “image” type documents.