

Mise en forme XML avec XSLT XML schéma (XSD)

BOUALEM ABID & LEBCIR FAHIMA

ENS KOUBA ALGER ÉÀ&(

Introduction

XSL : eXtensible Stylesheet Langage, ou langage extensible de feuille de style.

- **XPath** : un moyen d'accéder à un nœud quelconque de l'arborescence d'un document XML à partir d'un autre nœud quelconque.
- **XSLT** : eXtensible Stylesheet Langage Transformation.
- **XSL-FO** : eXtensible Stylesheet Langage - Formating Objects, et désigne un langage permettant le contrôle de la mise en page finale de la transformation. Ce langage est particulièrement destiné à la production de contenus au format PDF.

Introduction

- Le fonctionnement du XSL est fondé sur les manipulations de modèles (templates).
- Les éléments du document XML d'origine sont remplacés (ou plus ou moins légèrement modifiés) par ces modèles. Un modèle contient ainsi le texte (éléments, attributs, texte...) de remplacement d'un élément donné.
- Tout élément pouvant être remplacé dans le fichier de sortie par tout type de contenu texte, XSL est un outil privilégié de production de fichiers HTML à partir de sources XML.
- PHP fait ainsi appel à des bibliothèques de procédures de type XSL comme libxslt quand il doit gérer l'interfaçage avec des bases de données XML.
- Un fichier XSL étant un fichier XML, il doit respecter les normes de syntaxe de ce format.

Structure d'un document XSL

- La structure de base d'un document XSL commence par un prologue
- un élément `xsl:stylesheet` pouvant contenir quelques attributs, notamment une déclaration d'espace de noms ainsi que le numéro de version.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

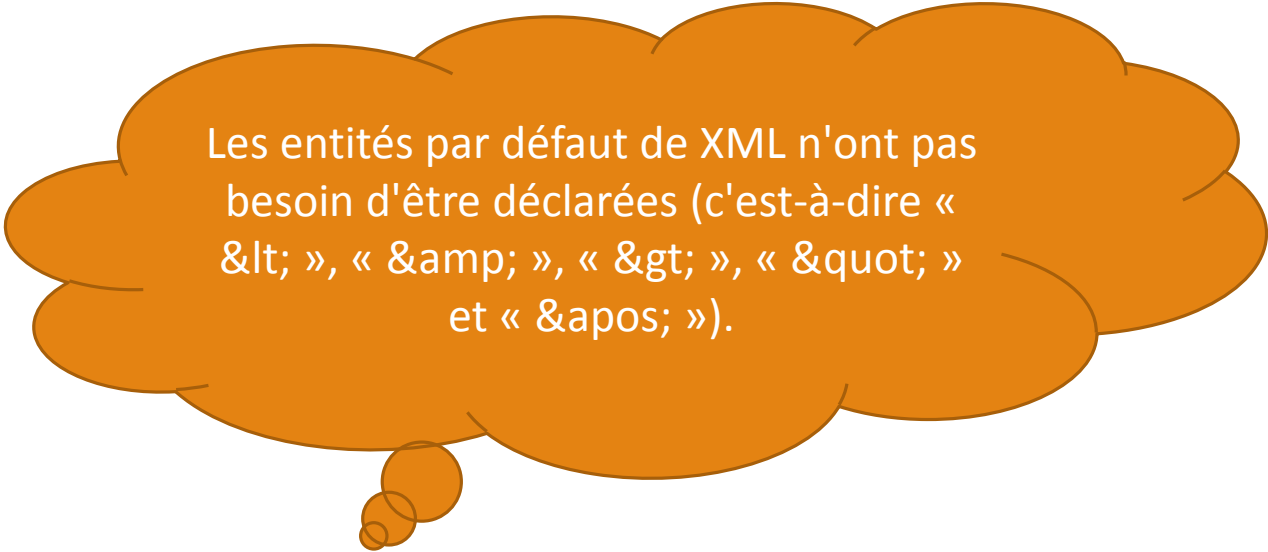
```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"> (...)  
</xsl:stylesheet>
```

Structure d'un document XSL

➤ Ensuite, si vous utilisez des entités dans la page XML (c'est-à-dire des choses telles que « é »), il vous faut les déclarer. Pour cela, écrivez à la suite :

➤ <!ENTITY le_nom "le_code">

```
<!DOCTYPE xsl:stylesheet [  
    <!ENTITY nbsp "&#160;">  
    <!ENTITY trade "&#8482;">  
    <!ENTITY mdash "&#8212;">  
    <!ENTITY ldquo "&#8220;">  
    <!ENTITY pound "&#163;">  
    <!ENTITY yen "&#165;">  
    <!ENTITY euro "&#8364;">  
>
```



Les entités par défaut de XML n'ont pas besoin d'être déclarées (c'est-à-dire « < », « & », « > », « " » et « ' »).

Structure d'un document XSL

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
<xsl:output method="html" encoding="utf-8" doctype-public="-//W3C//DTD XHTML 1.0  
Transitional//EN" doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"/>  
<xsl:template match="/">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>  
<title>Mise en forme avec XSLT</title>  
</head> <body> </body> </html>  
</xsl:template> </xsl:stylesheet>
```

Afficher du contenu XML

Lier la feuille XSLT au fichier XML

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

Régions répétées

```
<xsl:for-each select="test/jeu">
```

```
</xsl:for-each>
```

```
<xsl:for-each select="test/jeu">
```

```
  <tr>
```

```
    <td><xsl:value-of select="test/jeu/@id"/></td>
```

```
    <td><xsl:value-of select="test/jeu/nom"/></td>
```

```
    <td><xsl:value-of select="test/jeu/type"/></td>
```

```
    <td><xsl:value-of select="test/jeu/prix"/></td>
```

```
  </tr>
```

```
</xsl:for-each>
```

```
<xsl:for-each select="test/jeu">
```

```
  <tr>
```

```
    <td><xsl:value-of select="@id"/></td>
```

```
    <td><xsl:value-of select="nom"/></td>
```

```
    <td><xsl:value-of select="type"/></td>
```

```
    <td><xsl:value-of select="prix"/></td>
```

```
  </tr>
```

Les filtres

```
<xsl:for-each select="test/jeu[nom = 'Super Mario Galaxy']">
  <tr>
    <td><xsl:value-of select="@id"/></td>
    <td><xsl:value-of select="nom"/></td>
    <td><xsl:value-of select="type"/></td>
    <td><xsl:value-of select="prix"/></td>
  </tr>
</xsl:for-each>
```

Régions conditionnelles

```
<xsl:if test="">
```

```
</xsl:if>
```

```
<xsl:for-each select="test/jeu">
```

```
<xsl:if test="nom = 'Mario Sokoban'">
```

```
<p>J'ai Mario Sokoban !</p>
```

```
</xsl:if>
```

```
<xsl:if test="nom = 'Metroid Prime'">
```

```
<p>J'ai Metroid Prime !</p>
```

```
</xsl:if>
```

```
</xsl:for-each>
```

Régions conditionnelles

```
<xsl:choose>
```

```
  <xsl:when test="">
```

```
    </xsl:when>
```

```
  <xsl:otherwise>
```

```
    </xsl:otherwise>
```

```
</xsl:choose>
```

Inclure une page XML associée à une page XSLT dans PHP

```
<?php
    $xslDoc = new DOMDocument();
    $xslDoc->load("sommaire.xsl");

    $xmlDoc = new DOMDocument();
    $xmlDoc->load("chapitre.xml");

    $proc = new XSLTProcessor();
    $proc->importStylesheet($xslDoc);
    echo $proc->transformToXML($xmlDoc);
?>
```



