

# Introduction à XSLT

## Introduction

On considère le document XML `biblio.xml` suivant :

```
1  <?xml version="1.0"?>
2  <!DOCTYPE bibliotheque [
3  <!ELEMENT bibliotheque (livre+)>
4  <!ELEMENT livre (titre, auteur, ref)>
5  <!ELEMENT titre (#PCDATA)>
6  <!ELEMENT auteur (#PCDATA)>
7  <!ELEMENT ref (#PCDATA)>
8  ]
9  >
10 <bibliotheque>
11 <livre>
12 <titre>N ou M</titre>
13 <auteur>Agatha Christie</auteur>
14 <ref>Policier-C-15</ref>
15 </livre>
16 <livre>
17 <titre>Le chien des Baskerville</titre>
18 <auteur>Sir Arthur Conan Doyle</auteur>
19 <ref>Policier-D-3</ref>
20 </livre>
21 <livre>
22 <titre>Dune</titre>
23 <auteur>Franck Heckbert</auteur>
24 <ref>Fiction-H-1</ref>
25 </livre>
26 </bibliotheque>
```

Et on considère la feuille de style `biblio.css` suivante :

```
1  livre{
2  display:block; margin-left:10pt; margin-bottom:5pt; font-size:12pt
3  }
4  titre{
5  margin-right:10pt; color:blue;
6  }
7  auteur{
8  margin-right:10pt
9  }
10 ref{
11 color:red;
12 }
```

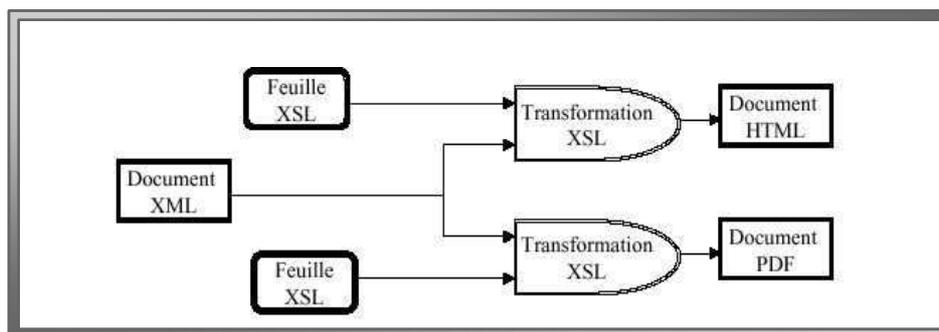
Alors, Comment associer cette feuille de style au document XML `biblio.xml` ? Que voit un utilisateur lorsqu'il le charge dans un navigateur, associé à cette feuille de style ?

## Description générale

L'utilisation d'une feuille de style est obligatoire en XML pour contrôler la mise en page du document, en effet un document XML ne contient que des informations sur la structure, aucune information relative à la mise en page n'apparaît dans le document. Il est possible d'utiliser pour présenter un document XML les feuilles de style CSS ou les feuilles de style XSL (*eXtensible StylesheetLanguage*).

XSL est beaucoup plus qu'un simple langage pour le formatage de documents XML (ce que pourrait très bien faire CSS - Cascading Style Sheets). Il permet de retraiter un document XML et ainsi de réarranger sa structure. En fait XSL permet de transformer un document XML en un autre document, souvent XML, mais pas forcément, par exemple en HTML, TeX, RTF, PostScript, etc.

Le schéma ci-dessous représente le principe de fonctionnement.



## Visualisation d'un document sans feuille de style

Lorsqu'aucune précision n'est donnée quant à l'affichage (pas de feuille de style), le navigateur affichera le contenu du document XML. Prenons le document *biblio.xml* qui décrit une bibliothèque, nous aurons l'affichage ci-contre :

```
<?xml version="1.0" ?>
- <bibliotheque>
- <livre>
  <titre>N ou M</titre>
  <auteur>Agatha Christie</auteur>
  <ref>Policier-G-15</ref>
</livre>
- <livre>
  <titre>Le chien des Baskerville</titre>
  <auteur>Sir Arthur Conan Doyle</auteur>
  <ref>Policier-D-3</ref>
</livre>
- <livre>
  <titre>Dune</titre>
  <auteur>Franck Heckbert</auteur>
  <ref>Fiction-H-1</ref>
</livre>
</bibliotheque>
```

Il est possible de modifier l'affichage en appuyant sur les signes **+** et **-**

```
<?xml version="1.0" ?>
- <bibliotheque>
+ <livre>
+ <livre>
- <livre>
  <titre>Dune</titre>
  <auteur>Franck Heckbert</auteur>
  <ref>Fiction-H-1</ref>
</livre>
</bibliotheque>
```

## Utilisation de feuille de style CSS

Nous voulons afficher ce document XML avec le titre en bleu et la référence en rouge, nous allons pour cela définir une feuille de style CSS dans le fichier *biblio.css* et nous allons l'appeler dans le document XML *biblio.xml*.

**Appel de la feuille de style dans le document :**

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/css" href="biblio.css"?>
3 <bibliotheque>
4 <livre>
5   <titre>N ou M</titre>
6   <auteur>Agatha Christie</auteur>
7   <ref>Policier-C-15</ref>
8 </livre>
9 <livre>
10  <titre>Le chien des Baskerville</titre>
11  <auteur>Sir Arthur Conan Doyle</auteur>
12  <ref>Policier-D-3</ref>
13 </livre>
14 <livre>
15  <titre>Dune</titre>
16  <auteur>Franck Heckbert</auteur>
17  <ref>Fiction-H-1</ref>
18 </livre>
19 </bibliotheque>
```

Si on considère la feuille de style *biblio.css* précédente, on aura l'affichage dans un navigateur Web



:

## Utilisation de feuille de style XSL

La feuille de style XSL est enregistrée dans un fichier externe et son nom comporte l'extension ".xsl". Dans le document XML, l'instruction de traitement suivante indique le type de la feuille de style et son emplacement :

```
<?xml-stylesheet type="text/xml" href="URL"?>
```

Prenons l'exemple de la bibliothèque, chaque livre a un titre, un auteur et une référence, le document XML ci-dessous appelle une feuille de style XSL qui est dans le fichier *biblio.xsl* dans le répertoire courant. Appel de la feuille de style dans le document XML :

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xml" href="biblio.xsl"?>
3 <bibliotheque>
4 <livre>
5   <titre>N ou M</titre>
6   <auteur>Agatha Christie</auteur>
7   <ref>Policier-C-15</ref>
8 </livre>
9 <livre>
10  <titre>Le chien des Baskerville</titre>
11  <auteur>Sir Arthur Conan Doyle</auteur>
12  <ref>Policier-D-3</ref>
13 </livre>
14 <livre>
15  <titre>Dune</titre>
16  <auteur>Franck Heckbert</auteur>
17  <ref>Fiction-H-1</ref>
18 </livre>
19 </bibliotheque>
```

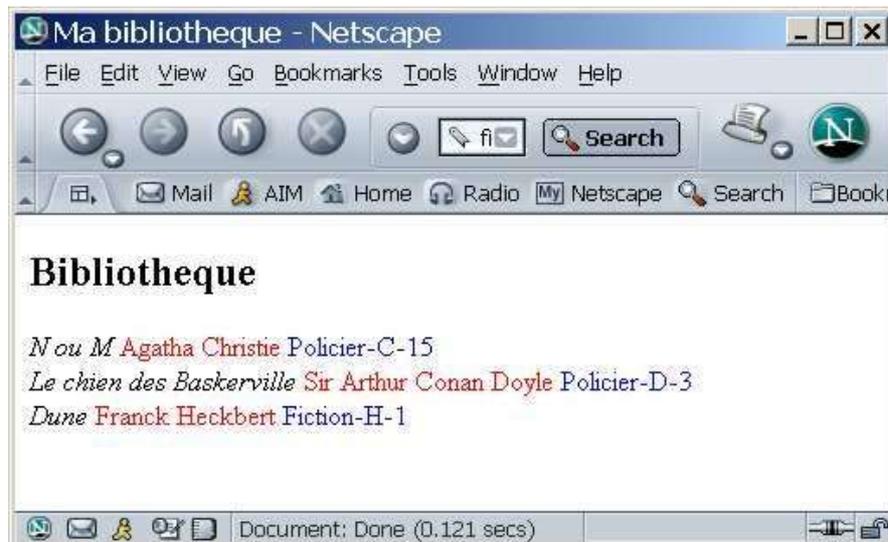
## Structure d'une feuille de style XSL

XSL est une application XML, une feuille de style XSL est donc un document XML. La feuille de style contient donc une déclaration XML et tous ses éléments sont placés dans l'élément racine. D'autre part, les éléments XSL sont préfixés par xsl: (XSL utilise les domaines de noms).

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

L'élément racine contient principalement des modèles (templates) pour l'affichage du document XML.

## La feuille XSL qui transforme le document `biblio.xml` en la page HTML



Une feuille de styles XSL contient un ou plusieurs modèles (templates), chaque modèle contient des informations sur l'affichage d'une branche des éléments du document. S'il n'y a qu'un seul modèle alors il s'applique sur la racine du document XML.

Prenons la feuille de style *biblio.xsl* qui applique un modèle unique au document XML bibliothèque *biblio.xml* :

```

1  <?xml version="1.0"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3  <xsl:output method="html"/>
4  <xsl:template match="/">
5  <html>
6  <head>
7  <title>Ma bibliothèque</title>
8  </head>
9  <body>
10 <H2>Bibliothèque</H2>
11 <xsl:for-each select="bibliotheque/livre">
12 <SPAN style="font-style:italic; padding-right:3pt">
13 <xsl:value-of select="titre"/>
14 </SPAN>
15 <SPAN style="color:red; padding-right:3pt">
16 <xsl:value-of select="auteur"/>
17 </SPAN>
18 <SPAN style="color:blue">
19 <xsl:value-of select="ref"/>
20 </SPAN>
21 <br/>
22 </xsl:for-each>
23 </body>
24 </html>
25 </xsl:template>
26 </xsl:stylesheet>

```

Cette feuille de style comporte une déclaration XML, un élément racine `xsl:stylesheet` qui englobe tous les autres éléments et précise que les éléments préfixés par `xsl:` appartiennent au domaine de nom `xsl`.

Le modèle est appliqué à la branche spécifiée par l'attribut `match` de l'élément `template`. En CSS la valeur de l'attribut `match` correspondrait au sélecteur de la règle. Dans notre exemple c'est à la racine du document XML (ne pas confondre avec l'élément racine `bibliotheque` qui est un enfant de la racine du document) que le modèle s'applique. La transformation d'un document XML par une feuille de style XSL s'effectue donc par un modèle traitant un nœud donné. Chaque modèle est divisé en deux parties : un nœud cible indiqué par l'attribut `match` et une action sur le nœud :

```
<xsl:template match="noeud_cible">
  action
</xsl:template>
```

Un modèle contient deux types d'éléments :

- ✓ des éléments XML bien formés pour représenter les éléments html, c'est le cas de H2, SPAN et BR dans notre exemple.
- ✓ des éléments XSL, dans notre exemple le `xsl:value-of`, qui permettent d'accéder au contenu des éléments du document XML, l'attribut `select` indique le nom de l'élément XML auquel on veut accéder. Ce nom est donné à partir de l'élément courant (ici c'est `bibliotheque/livre` donc on accède au titre du livre en donnant la valeur `titre` à l'attribut `select`)
- ✓ L'élément `xsl:`

`for-each select=`

"`bibliotheque/livre`" permet de parcourir tous les éléments qui portent le nom `livre` et qui sont des enfants de `bibliotheque` et donc d'afficher tous les éléments `livre` du document XML. Ce qui permet d'obtenir l'affichage :



## 1. Classement

Si nous souhaitons classer la bibliothèque par titre en respectant l'ordre alphabétique nous utiliserons : `<xsl:sort select="." order="ascending" />`, la feuille de style devient alors :

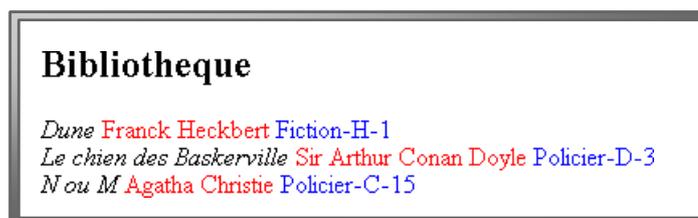
```

1  <?xml version="1.0"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3    <xsl:output method="html"/>
4    <xsl:template match="/">
5      <html>
6        <head>
7          <title>Ma bibliotheque</title>
8        </head>
9        <body>
10         <H2>Bibliotheque</H2>
11
12         <xsl:for-each select="bibliotheque/livre">
13           <xsl:sort select="." order="ascending"/>
14           <SPAN style="font-style:italic; padding-right:3pt">
15             <xsl:value-of select="titre"/>
16           </SPAN>
17           <SPAN style="color:red; padding-right:3pt">
18             <xsl:value-of select="auteur"/>
19           </SPAN>
20           <SPAN style="color:blue">
21             <xsl:value-of select="ref"/>
22           </SPAN>
23           <br />
24         </xsl:for-each>
25       </body>
26     </html>
27   </xsl:template>
28 </xsl:stylesheet>

```

L'élément `<xsl:sort select="."/>` permet de définir l'ordre dans lequel les éléments seront affichés. C'est un sous-élément de `xsl:apply-templates` ou de `xsl:for-each`. L'attribut `order = "ascending" | "descending"` permet d'obtenir un ordre croissant ou décroissant.

Ce qui fournit l'affichage ci-dessous dans le navigateur :



## 2 Filtrage

Ajoutons dans le document XML un attribut `type` qui correspond au type du livre :

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xml" href="biblio.xsl"?>
3 <bibliotheque>
4 <livre type="policier">
5 <titre>N-ou-M</titre>
6 <auteur>Agatha-Christie</auteur>
7 <ref>Policier-C-15</ref>
8 </livre>
9 <livre type="policier">
10 <titre>Le chien des Baskerville</titre>
11 <auteur>Sir-Arthur-Conan-Doyle</auteur>
12 <ref>Policier-D-3</ref>
13 </livre>
14 <livre type="science-fiction">
15 <titre>Dune</titre>
16 <auteur>Franck Heckbert</auteur>
17 <ref>Fiction-H-1 </ref>
18 </livre>
19 </bibliotheque>
```

Nous pouvons par exemple décider de n'afficher que les livres dont le type est "policier", il suffit d'ajouter dans la feuille de style dans le tri le nom de l'attribut. Ce nom est précédé de `@` pour indiquer que `type` n'est pas un élément mais un attribut (de la même manière il est possible d'accéder à la valeur d'un attribut dans un `value-of` en précédant l'attribut de `@`).

```

1  <?xml version="1.0"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3  <xsl:output method="html"/>
4  <xsl:template match="/">
5  <html>
6  <head>
7  <title>Ma bibliotheque</title>
8  </head>
9  <body>
10 <H2>Bibliotheque</H2>
11 <xsl:for-each select="bibliotheque/livre[@type='policier']">
12 <SPAN style="font-style:italic; padding-right:3pt">
13 <xsl:value-of select="titre"/>
14 </SPAN>
15 <SPAN style="color:red; padding-right:3pt">
16 <xsl:value-of select="auteur"/>
17 </SPAN>
18 <SPAN style="color:blue">
19 <xsl:value-of select="ref"/>
20 </SPAN>
21 <br/>
22 </xsl:for-each>
23 </body>
24 </html>
25 </xsl:template>
26 </xsl:stylesheet>

```

Ce qui fournit dans le navigateur l'affichage ci-contre :

**Bibliotheque**

*N ou M Agatha Christie Policier-C-15*  
*Le chien des Baskerville Sir Arthur Conan Doyle Policier-D-3*

### 3 Modèle multiples

Pour afficher un élément XML comme livre dans notre exemple, il est possible d'utiliser `xsl:apply-templates` dans un modèle multiple (à la place du `xsl:for-each`).

Lorsqu'il y a plusieurs modèles il faut toujours qu'il y en ait un pour l'affichage de la racine du document (le /). Dans le premier modèle le `xsl:apply-template` indique que pour chaque élément livre enfant de bibliothèque il faut appliquer le deuxième modèle (celui pour lequel l'attribut `match` a pour valeur `livre`).

```

1  <?xml version="1.0"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3  <xsl:output method="html"/>
4  <xsl:template match="/">
5  <html>
6  <head>
7  <title>Ma bibliotheque</title>
8  </head>
9  <body>
10 <H2>Bibliotheque</H2>
11 <xsl:apply-templates select="bibliotheque/livre" />
12 </body>
13 </html>
14 </xsl:template>
15
16 <xsl:template match="livre">
17 <SPAN style="font-style:italic; padding-right:3pt">
18 <xsl:value-of select="titre" />
19 </SPAN>
20 <SPAN style="color:red; padding-right:3pt">
21 <xsl:value-of select="auteur" />
22 </SPAN>
23 <SPAN style="color:blue">
24 <xsl:value-of select="ref" />
25 </SPAN>
26 <br />
27 </xsl:template>
28 </xsl:stylesheet>

```

Le navigateur affichera comme avec le for-each la page ci-contre :

## Bibliotheque

*N ou M* Agatha Christie Policier-C-15

*Le chien des Baskerville* Sir Arthur Conan Doyle Policier-D-3

*Dune* Franck Heckbert Fiction-H-1

## En conclusion

Comment visualiser un document XML à l'aide d'une feuille de style XSL :

1- Appel de la feuille de style dans le document XML :

```

1  <?xml version="1.0"?>
2  <?xml-stylesheet type="text/xml" href="biblio.xsl"?>
3  <bibliotheque>
4  <livre>
5  <titre>N ou M</titre>
6  <auteur>Agatha Christie</auteur>
7  <ref>Policier-C-15</ref>
8  </livre>
9  <livre>
10 <titre>Le chien des Baskerville</titre>
11 <auteur>Sir Arthur Conan Doyle</auteur>
12 <ref>Policier-D-3</ref>
13 </livre>
14 <livre>
15 <titre>Dune</titre>
16 <auteur>Franck Heckbert</auteur>
17 <ref>Fiction-H-1</ref>
18 </livre>
19 </bibliotheque>

```

2- Contenu de la feuille de style *biblio.xsl* :

```

1  <?xml version="1.0"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3  <xsl:output method="html"/>
4  <xsl:template match="/">
5  <html>
6  <head>
7  <title>Ma bibliotheque</title>
8  </head>
9  <body>
10 <H2>Bibliotheque</H2>
11 <xsl:for-each select="bibliotheque/livre">
12 <SPAN style="font-style:italic; padding-right:3pt">
13 <xsl:value-of select="titre"/>
14 </SPAN>
15 <SPAN style="color:red; padding-right:3pt">
16 <xsl:value-of select="auteur"/>
17 </SPAN>
18 <SPAN style="color:blue">
19 <xsl:value-of select="ref"/>
20 </SPAN>
21 <br />
22 </xsl:for-each>
23 </body>
24 </html>
25 </xsl:template>
26 </xsl:stylesheet>

```

### 3- Affichage dans un navigateur Web :

