

# Transformation de documents XML avec XSLT

## Présentation de documents HTML et XML avec CSS

### Objectif

L'objectif de ce TP est d'utiliser le langage XSLT pour (1) transformer la structure d'un ensemble de documents XML vers un format pivot, (2) puis de transformer ce format "pivot" vers un ou plusieurs formats applicatifs (XHTML dans un premier temps).

Outils utilisés pour l'application de feuilles XSLT:

- Processeur XSLT : Xalan (<http://xml.apache.org/xalan-j/index.html>)

### 1 Transformation pas à pas de XML vers XML

La commande pour appliquer une feuille de style est : `/usr/local/bin/xalan -IN <source.xml> -XSL <stylesheet.xsl> [-OUT result.???` (Regarder le script et les autres paramètres !)

Un squelette de feuille XSLT est fourni dans le fichier `squel.xsl`, le compléter pour que l'application de cette feuille de style au document `books.xml` des TP précédents crée un document qui réponde aux exigences suivantes :

Le document résultat contient un élément racine `Result` qui contient trois éléments `Bloc`.

- Le premier contient la liste des catégories et de leurs descriptions
  - Faire une exception pour la catégorie `C` dont la description est secrète, on affichera alors " catégorie secrète ".
- Le second élément `bloc` contient les informations qui décrivent le " top author " de la semaine (indiqué dans la variable `top-author`) dans un élément `topauthor`
  - Ajouter un élément fils `Books` contenant un élément `Title` pour chacun de ses livres.
  - ajouter un attribut `nbLivres` à l'élément `Books` dont la valeur est le nombre de ses livres.
  - Créer un élément `LastBook` qui contient le texte du titre du dernier livre de cet auteur dans le document source. Ajouter à cet élément un attribut `price` dont la valeur est le prix de ce livre.
  - Créer un élément `Domain` qui la liste des catégories des livres de cet auteur.
  - Ajouter un attribut `averageBookPrice` dont la valeur est le prix moyen de ses livres.
- Le troisième élément `bloc` contient la liste complètes des livres triées par premier auteur et par catégorie dans des éléments de la forme `<Book category='...'><FirstAuthor>...</FirstAuthor><Title></Title></Book>`

Avez-vous pensé à utiliser les modes ?

### 2 De XML à HTML

Ecrire une feuille XSLT qui transforme le document XML obtenu dans l'exercice précédent en un document XHTML.

#### 2.1 Le document XHTML sans instructions de présentation

Un introduction très basique à HTML : <http://www.w3.org/TR/html/>.

Important : Votre document XHTML ne contiendra que des informations liées à la structure logique du document mais **aucune** instructions de présentation.

## 2.2 La présentation avec une feuille de style CSS

Créer une feuille de style CSS (<http://www.w3.org/TR/CSS2/>) qui décrit précisément la présentation à associer au document XHTML précédent.

Des éditeurs CSS existent topstyle : <http://www.bradsoft.com/topstyle> (Windows), un mode emacs (mode css, limité), et bluefish <http://bluefish.openoffice.nl/> (Linux)

Vous pouvez vous inspirer de la feuille de style de cette page :  
<http://www.w3.org/Style/CSS/> (regarder le source pour avoir l'adresse de la feuille CSS)

## 3 Pour aller plus loin

### 3.1 XML+CSS

Il est possible d'écrire une feuille de style qui s'applique directement au document XML précédent (CSS s'applique à HTML et à XML).

### 3.2 XML vers XSL-FO

La présentation du document pourra être réalisée en utilisant le langage XSLT-FO à la place de XHTML+CSS. Ecrire une feuille XSLT qui transforme le document sortie.xml en XSL-FO. Le document obtenu pour être transformé en pdf avec fop (cf. TP 1).

Une description de XSL-FO pourra être trouvée ici :  
<http://www.w3.org/TR/xsl>  
et un tutoriel ici :  
<http://www.renderx.com/Tests/doc/html/tutorial.html>