

# XML : TP XSL

## Feuilles de style XSLT

Nicolas Singer

### Mise en place de l'environnement de développement

Dans ce TP, nous allons appliquer des feuilles de style au document XML définissant une filmographie, que nous avons conçu lors des TP précédents.

Nous utiliserons le navigateur Firefox pour visualiser l'application de la feuille de style. Pour qu'il puisse interpréter une feuille de style locale, il faut paramétrer son paramètre de sécurité comme ceci :

1. Ouvrez les paramètres en tapant `about:config` dans la barre d'adresse ;
2. Fixez le paramètre `security.fileuri.strict.origin.policy` à `false`

Ceci fait, créez à côté de `filmographie.xml` un nouveau fichier du nom de `filmographie.xsl` et ajoutez au début de `filmographie.xml` la ligne :

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

### 1 Exercice 1

1. Mettez en place l'élément racine de la feuille de style, ainsi que l'instruction définissant le type de document qu'elle produit en ajoutant dans `filmographie.xsl` les lignes suivantes :

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="iso-8859-1" indent="yes"/>
</xsl:stylesheet>
```

Chargez `filmographie.xml` dans votre navigateur et vérifiez que l'affichage est bien soumis à cette feuille de style vide (par conséquent vous devez obtenir soit une page vide soit un document sans mise en forme suivant le navigateur).

2. Ajoutez une règle qui dit que quand la racine du document est rencontrée, on doit ajouter au document produit un entête et un pied de page HTML. Cet entête met en place les balises définissant une page web de titre *cinématographie*. Cette règle prend la forme suivante :

```
<xsl:template match="/">
<html><head> <title> cinématographie </title> </head>
<body>

</body>
</html>
</xsl:template>
```

Insérez cette règle dans la feuille de style et vérifiez le résultat obtenu (normalement une page web vide avec un nouveau titre dans la fenêtre du navigateur).

3. Définissez une règle qui s'applique à tous les éléments de type `film` destinée à afficher uniquement le titre du film (qui normalement est un attribut de cet élément). Cette règle prend la forme suivante :

```
<xsl:template match="film">
<xsl:value-of select="titre" />
</xsl:template>
```

Rappelons que l'élément `xsl:value-of` a pour effet d'afficher le résultat de l'expression placée dans le `select`. Ici cette expression demande d'évaluer le contenu de l'attribut `titre` de l'élément courant. Insérez cette règle dans la feuille de style et vérifiez le résultat obtenu. Rien ne se passe car nous devons expliquer au processeur XSL quand il doit appeler cette règle.

4. Ajoutez entre les balises `BODY` de la règle portant sur la racine, la ligne suivante :

```
<xsl:apply-templates select="films/film" />
```

L'instruction `xsl:apply-templates` dit d'appeler la règle concernant tous les éléments `film` fils de l'élément `films` au moment où elle apparaît. Vérifiez le résultat et constatez que les titres de film apparaissent tous sur la même ligne.

5. Modifiez la règle portant sur les éléments `film` en faisant en sorte que le titre des films apparaisse entre des balises `H2`. De même modifiez la règle portant sur la racine pour que le titre de la page web (cinématographie) apparaisse dans la page HTML entre des balises `H1`.
6. En complétant la règle concernant l'élément `film` réalisez la mise en forme suivante :
  - (a) Sous le titre du film on doit trouver en italique une ligne indiquant : *film <genre du film> <nationalité> de <durée> sorti en <année de sortie>*. Par exemple on aura *film fantastique américain de 2h35 sorti en 2005*.
  - (b) Encore dessous on doit faire apparaître en gras une ligne disant : **réalisé par <réalisateur>**.
7. Ajoutez à la feuille de style une règle destinée à afficher les acteurs d'un film. Cet affichage doit se faire sous la forme d'une liste HTML du type :

```
avec
<ul>
<li> Acteur 1 </li>
<li> Acteur 2 </li>
...
</ul>
```

Appelez cette règle à l'intérieur de la règle concernant l'élément `film`. Notez bien que les balises `ul` doivent être placées par la règle `film` alors que les balises `li` doivent être placées par la règle concernant un acteur. Remarque : Quand on est dans une règle portant sur un élément et que l'on veut afficher le contenu de cet élément (et pas le contenu d'un de ses sous-éléments) on peut utiliser la notation `.` (point) à l'intérieur du `select`. On peut aussi utiliser la notation `text()`.

8. On peut évidemment produire une page HTML qui utilise les styles CSS pour sa mise en page. Par exemple nous souhaitons que l'histoire du film soit encadré en rouge sur fond gris. Ajoutez dans votre feuille de style l'instruction nécessaire pour ajouter après les auteurs un paragraphe contenant l'histoire du film. Précisez la classe de ce paragraphe en utilisant l'attribut `class="histoireStyle"`. Vérifiez que l'histoire du film apparaît bien.
9. Créez une feuille de style `film.css` contenant les styles applicables aux éléments de type *histoire*. Pour cela la feuille de style doit contenir

```
.histoireStyle {
text-align: justify;
border-color: red;
border-style: solid;
background-color: #BBBBBB;
}
```

Pour que la feuille de style soit utilisée, on doit ajouter dans l'entête du document HTML produit la ligne :

```
<link rel="stylesheet" type="text/css" href="film.css" />
```

Vérifiez que tout cela fonctionne.

A ce point du TP nous manipulons donc trois fichiers pour mettre en place notre mise en forme :

- Le document XML à mettre en forme
- La feuille de style XSL qui appelle transforme ce document en HTML en spécifiant le style qui doit s'appliquer à chaque élément.
- La feuille de style CSS qui donne l'implémentation de ces styles.

## 2 Exercice 2

Nous allons dans cette section utiliser les feuilles de style XSL de façon un peu plus *savante*.

1. Nous voulons afficher les films sortis en 2006 avant ceux sortis à d'autres dates. Pour vérifier le fonctionnement de ce que nous allons écrire, modifiez éventuellement votre document XML pour qu'il se termine par un ou plusieurs films sortis en 2006 (et qu'il commence par un ou plusieurs films sortis à d'autres dates).

2. Trouvez l'expression XPath qui permet de sélectionner tous les films sortis en 2006. Même chose pour les films sortis à une date différente de 2006. Dans la règle de l'élément racine remplacez l'instruction `xsl:apply-templates` par deux instructions du même type, une sélectionnant les film de 2006, suivie par une autre sélectionnant les autres films. Vérifiez le résultat produit.
3. Nous voulons présenter les films de 2006 en précédant le titre du mot *Nouveauté* écrit en rouge et en tout petit. Pour cela nous pouvons écrire deux règles différentes, une s'appliquant aux films sortis en 2006, l'autre s'appliquant aux autres. Modifiez donc votre feuille de style pour réaliser cela (vous pouvez dupliquer la règle s'appliquant aux éléments `film` et modifiez légèrement le `match` de ces deux règles).
4. Cette méthode est un peu lourde car elle nous oblige à écrire deux règles différentes alors qu'il n'y a qu'un seul élément de mise en page à changer. XSL propose une instruction conditionnelle qui se note

```
<xsl:if test="expression XPath"> forme </xsl:if>
```

Si l'expression est évaluée à vrai, alors la forme est instanciée et placée dans le document. A l'aide de cette instruction, modifiez la feuille de style XSL de façon à n'avoir plus qu'une seule règle pour l'élément `film` qui à l'intérieur teste l'année de sortie du film et qui produit la mise en forme souhaitée.

5. On peut également lors d'un appel à `xsl:apply-templates` trier les résultats obtenus. Cela se fait par une instruction du type

```
<xsl:apply-templates select="expression">
<xsl:sort select="expression" order="ascending" data-type="text" />
</xsl:apply-templates>
```

Notez bien que l'instruction `xsl:sort` figure en tant que contenu de l'instruction `xsl:apply-templates`. L'expression à l'intérieur du `select` définit la clé de tri. `order` peut prendre comme valeur *ascending* ou *descending* suivant l'ordre de tri souhaité et `data-type` peut valoir *text* ou *number* selon que la clé de tri est une chaîne de caractère ou un nombre.

Appliquez cette instruction pour trier les films suivant leur date de sortie, cette fois-ci avec une seule instruction `apply-templates`. Ceci fait, essayez de les trier suivant le nombre d'entrées qu'ils ont fait.

### 3 Exercice 3

Nous voulons dans cet exercice, afficher le document XML en le restructurant.

Par exemple nous voulons créer une table des matières des films. Pour cela, nous voulons commencer le document HTML par la liste des titres de film suivie par la liste détaillée telle que nous l'avons mise au point dans les exercices précédents. La difficulté ici est d'afficher un même film sous deux formats différents. Un format très simplifié pour son entrée dans la table des matières, et un autre pour son apparition dans le document. Ceci peut s'obtenir grâce à un attribut *mode* que l'on peut ajouter à une règle et à l'appel d'une règle.

1. Ajoutez à la règle `xsl:template` de l'élément `film` un attribut `mode="complet"`. Ajoutez une nouvelle règle s'appliquant elle aussi à un élément de type `film` mais avec un attribut `mode="tdm"`. Cette nouvelle règle doit simplement afficher le titre du film.
2. Dans la règle de l'élément racine, avant la liste détaillée des films, faites appel à la règle d'affichage simplifiée en insérant un

```
<h1>Table des matières des films</h1>
<xsl:apply-templates select="films/film" mode="tdm">
```

De même modifiez l'ancienne appel à la règle d'affichage d'un film en la transformant en :

```
<xsl:apply-templates select="films/film" mode="complet">
```

Vérifiez que cela fonctionne.

3. Embellissez la table des matières en mettant chaque titre de film à l'intérieur d'une balise `UL` comme vous l'avez fait pour les acteurs.
4. Dans la table des matières, ajoutez entre parenthèse après le titre, le nombre d'acteurs du film suivi de "acteurs".
5. Essayez de faire en sorte qu'un click sur le titre d'un film déplace la fenêtre du navigateur sur le film détaillé. Pour cela le titre du film détaillé doit être encadré par une balise `<a name="titre"> titre </a>` et le lien de l'entrée de la table des matières doit être encadré par `<a href="#titre"> titre </a>`. Réfléchissez à un moyen de faire cela et notez la difficulté rencontrée.

6. La difficulté vient du fait que les valeurs des attributs **name** et **href** doivent être calculées par le processeur XSL. Pour cela XSL propose l'instruction

```
<xsl:attribute name="expression donnant le nom de l'attribut">
expression donnant la valeur de l'attribut
</xsl:attribute>
```

Cette instruction calcule à la fois le nom de l'attribut et sa valeur. Elle s'utilise à la suite d'un nom d'élément. Par exemple :

```
<a>
<xsl:attribute name="href">
#
<xsl:value-of select="nom" />
</xsl:attribute>
<xsl:value-of select="@titre" />
</a>
```

renvoi `<a href="#King Kong"> King Kong </a>` si elle est appliquée au film portant ce titre. Corrigez la feuille de style pour arriver à ce que la table des matières soit cliquable.

## 4 Exercice 4

1. En utilisant les éléments **acteursDescription**, produisez une table des matières des acteurs de film devant afficher les noms et prénoms des acteurs, sa date de naissance et entre parenthèses le nombre de films de votre document dans lequel cet acteur joue. Le prénom des hommes devra figurer en bleu et celui des femmes en rose.
2. Faites suivre le nombre entre parenthèses d'une liste de numéros cliquables et séparés par des espaces, chacun devant mener à un des films dans lesquels l'acteur a tourné.
3. Ajoutez devant le titre du descriptif complet de chaque film, l'image que vous avez choisi pour le représenter. Précisez dans la feuille de style CSS un style propre à cette image pour régler sa taille ou sa bordure par exemple.
4. Modifiez l'affichage du scénario pour que les mots clés apparaissent en gras, les éléments **ev** en italique, et les éléments **personnage** soulignés. Pour cela on écrira des règles spécifiques pour afficher ces éléments et on utilisera une balise `<font class="ev">` par exemple pour encadrer l'élément en mettre en forme. On utilisera ensuite la feuille de style CSS pour spécifier précisément cette mise en forme.
5. Ajoutez dans la table des matières des films, juste après le nombre des acteurs, la liste des mots clés correspondant à chaque film (entre crochets).

## 5 Exercice 5

La page que nous allons désormais générer va intégrer des scripts JavaScript et faire appel à un plugin jquery pour réaliser un effet de scrolling entre les films.

1. Rendez-vous sur la page du plugin *One Page Scroll* d'adresse :

[http://www.thepetdesign.com/demos/onepage\\_scroll\\_demo.html](http://www.thepetdesign.com/demos/onepage_scroll_demo.html)

et téléchargez les parties javascript et css du plugin que vous placerez dans des fichiers appelés **scroll.js** et **scroll.css**.

2. Modifiez la feuille de style pour que la page construite soit désormais la suivante :
  - Une balise `<div id="main">` doit désormais encapsuler l'ensemble de la page (elle doit donc être fille directe de l'élément **body**).
  - Chaque section correspondant à un élément du scroll doit être incluse dans un élément `<section>`. Insérez par exemple chaque table des matières (acteurs et films) et chaque film dans son propre élément **section**.
  - Avant la fin du `<body>`, insérez un élément `<script>` contenant le code suivant :

```
$("#main").onepage_scroll({
  sectionContainer: "section",
  easing: "ease",
  animationTime: 1000,
```

```

    pagination: true,
    updateURL: false,
    beforeMove: function(index) {},
    afterMove: function(index) {},
    loop: false,
    keyboard: true,
    responsiveFallback: false
  });

```

— Dans la partie `<head>` insérez les lignes suivantes :

```

<script src ="https://code.jquery.com/jquery-2.2.4.min.js" > </script>
<script src="scroll.js"> </script>
<link rel="stylesheet" type="text/css" href="scroll.css" />

```

3. Testez l'effet obtenu. Vous pouvez corriger vos lien (ceux qui mènent des tables des matières au films) pour utiliser les cibles automatiquement créées par le plugin jquery. Pour plus d'informations concernant les paramètres du plugin, référez-vous à sa page GitHub.

## 6 Exercice 6

1. réalisez une nouvelle feuille de style XSL qui présente les films sous la forme d'un tableau où les lignes sont les différents films et les colonnes respectivement l'image du film, le titre du film, la liste des acteurs, le scénario, le nombre d'entrées, et la date de sortie. Faites en sorte que votre document XML utilise à présent cette nouvelle feuille de style. Pour ce qui est de la liste des acteurs et du scénario, vous prendrez soin de conserver le même formatage que pour la première feuille de style.