

Feuilles de style XSLT : transformation de XML en XML ou XHTML

Andrei Popescu-Belis
TIM / ETI, Université de Genève

Cours n°8

CSS pour XML : possibilités

- Paramétrer l'affichage d'un document XML
 - avec « display: block » → passage à la ligne
 - avec « display: none » → effacer un élément
- Changer la police, la taille, la couleur, la position
- Créer des cadres, changer le fond, etc.
- Recréer une structure d'affichage de liste (« display: list-style ») ou de tableau (« display: table »), si le document XML s'y prête

2

CSS pour XML : limitations

- Impossible de...
 - déplacer ou réorganiser des éléments
 - ex: mettre les auteurs après le titre
 - changer le nom des balises
 - ex: convertir l'élément `<lien>URL</lien>` en un véritable hyperlien `URL`
 - afficher du texte supplémentaire
 - ex: introduire des virgules entre deux éléments
 - ex: écrire RÉSUMÉ quand on rencontre `<resume>`

3

Pour aller plus loin avec XML

- XSL : eXtensible Stylesheet Language
 - XSLT : XSL Transformations
 - XSL-FO : XSL Formatting Objects
- XSLT
 - pour transformer un fichier XML en un autre fichier XML, ou en un fichier XHTML qui pourra être affiché dans un navigateur
- XSL-FO
 - pour transformer un fichier XML en un fichier directement affichable ou imprimable (p.ex. PDF)

4

Les feuilles de style XSLT

- Indiquent comment un fichier XML doit être transformé en un autre fichier XML
 - ou même en un fichier texte, ou HTML, ou XHTML
 - solution élégante au problème de l'affichage en passant par un document XHTML (ou d'autres formats avec XSL-FO)
- Sont aussi des documents XML bien formés et valides
 - déclaration XML en première ligne
 - déclaration de l'encodage (UTF-8, ISO-8859-1, etc.)
 - règles de fermeture des balises, etc.

5

Appliquer une feuille de style XSLT à un document XML

- Utiliser un processeur XSLT séparé (p.ex. *saxon.exe*)
- Avec XMLSpy
 - ouvrir le document XML
 - choisir XSL → XSL Transformation
 - ou bien la touche F10
 - ou bien le bouton `XSL`
 - indiquer la feuille de style (un fichier ou une fenêtre)
 - ou associer de manière permanente une feuille de style et un doc. XML
- Très utile : XSL → Start Debugger (Alt+F11)
 - appliquer la feuille pas à pas (touche F11 pour avancer) ©

6

Associer une feuille de style XSLT à un document XML

- Indiquer l'emplacement de la feuille de style au début du fichier XML (chemin absolu ou relatif ou URL)

```
<?xml version="1.0" encoding="us-ascii"?>
<?xml-stylesheet type="text/xsl"
href="feuille-test.xsl"?>
<!DOCTYPE livre SYSTEM "livre.dtd"> ...
```

- XMLSpy
 - menu XSL → 'Assign XSL'
 - puis indiquer le fichier, sur disque ou bien ouvert dans XML Spy
- ☺ : indiquée une fois pour toutes → **affichage direct du document transformé dans Firefox ou Netscape ou IE**
- ☹ : pour changer de feuille de style, il faut éditer le document

7

Structure d'une feuille de style XSLT (1)

- Déclaration XML habituelle
- Élément racine `xsl:stylesheet` + attributs fixés
 - `<xsl:stylesheet`
 - déclaration de version*
 - déclaration du nom d'espace xsl*
- Instructions de formatage générales, paramètres
- Instructions de transformation des éléments
 - pour balises et attributs
 - `<xsl:template match="balise">... </xsl:template>`
- Fermeture : `</xsl:stylesheet>`

→ Voir la feuille aide-mémoire indiquée sur *moodle*

8

Structure d'une feuille de style XSLT (2)

```
{
  <?xml version="1.0"?>
  <xsl:stylesheet
    version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    I1 {
      <xsl:template match="NOM-ELEMENT-OU-JOKER">
        ... INSTRUCTIONS-OU-BALISES-OU-TEXTE ...
      </xsl:template>
    }
    I2 {
      <xsl:template match="NOM-AUTRE-ELEMENT-OU-JOKER">
        ... AUTRES-INSTRUCTIONS-OU-TEXTE-OU-BALISES ...
      </xsl:template>
    }
    In ... AUTANT DE BLOCS XSL:TEMPLATE QUE NECESSAIRE ...
  }
  </xsl:stylesheet>
}
```

9

Commandes de base (1)

- Instruction de formatage principale
 - `<xsl:template match="NOM-ELEMENT-OU-JOKER">`
 - PAR QUOI REMPLACER CET ELEMENT, TEXTE-OU-BALISES**
 - `</xsl:template>`
- « Jokers »
 - * signifie « tout fils »
 - . dénote l'élément courant
 - plus généralement : le langage XPath

10

Commandes de base (2)

- A l'intérieur de `xsl:template` on peut écrire
 - Commande de recopie du contenu, sans balises
 - `<xsl:value-of select="BALISE-OU-JOKER"/>`
 - Commande de recopie de l'élément
 - `<xsl:copy-of select="BALISE-OU-JOKER"/>`
 - Commande de traitement des éléments suivants
 - `<xsl:apply-templates select="BALISE"/>`

11

Commandes de base (3)

- A l'intérieur de `xsl:template`
 - si on écrit du texte, il sera recopié dans le fichier résultat
 - si on écrit des balises, aussi (ne pas oublier des les fermer)
 - si on veut ne pas recopier certains éléments, il ne faut pas utiliser `xsl:apply-templates` pour eux

12

Comment recopier un document XML ?

Feuille de transformation XSLT « identique »

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:output method="xml"/>
  <xsl:template match="* | @*">
    <xsl:copy>
      <xsl:copy-of select="@*" />
      <xsl:apply-templates />
    </xsl:copy>
  </xsl:template>
</xsl:stylesheet>
```

13

Comment une feuille de style XSLT est appliquée (1)

- Le processeur lit la feuille de style et mémorise ses instructions de type `xsl:template`
 - si des erreurs sont détectées dans la feuille de style, STOP
- Le processeur lit le fichier XML à transformer et le vérifie
 - s'il est mal formé ou non valide, STOP
- Le processeur parcourt le document XML initial
 - du début à la fin
 - sous forme d'arbre
 - des éléments parents vers les éléments enfants
 - sauf si la feuille de style change l'ordre de parcours

14

Comment une feuille de style XSLT est appliquée (2)

- Pour chaque élément du document XML
 - le processeur cherche l'instruction de formatage XSLT (`<xsl:template match="...">`) qui s'applique
 - s'il la trouve, il fait ce qu'elle dit
 - sinon, il existe deux *instructions par défaut* (si rien n'est prévu dans une feuille de style pour un élément) qui sont:
 1. éléments avec du texte : recopier seulement le texte, *sans* les balises
 2. ensuite passer aux éléments enfants (contenus dans l'élément courant)

15

Instructions XSLT par défaut

- Elles sont connues « par défaut » de tout processeur XSLT
- Elles ont la priorité la plus faible : s'appliquent seulement si aucune autre instruction `xsl:template` ne s'applique

1. Recopier tout texte dans le fichier XML résultat

```
<xsl:template match="text()">
  <xsl:value-of select="." />
</xsl:template>
```
2. Propagation du traitement aux éléments enfants

```
<xsl:template match="*">
  <xsl:apply-templates />
</xsl:template>
```

16

Principe de « propagation »

- Si dans la feuille de style on prévoit une instruction `xsl:template` pour traiter un certain élément, ne pas oublier de préciser dans l'instruction ce qui doit se passer avec ses enfants (= balises contenues)
 - si on veut les ignorer, on ne dit rien
 - si on veut les traiter, écrire :

```
<xsl:apply-templates select="balises-à-traiter"/>
```
 - pour traiter tous les enfants, écrire :

```
<xsl:apply-templates select="*" /> = tous ceux avec des balises
```

```
<xsl:apply-templates /> = balises ou texte
```

17

Autres possibilités de XSLT

- Mécanisme pour traiter les attributs
- Mécanisme pour passer des paramètres
- Instructions de formatage générales, à mettre au début de la feuille XSL (p.ex. pour générer du HTML ou TXT)
- Instructions conditionnelles (`xsl:choose`, `xsl:otherwise`, `xsl:if`, etc.)
- Les expressions dans `match="..."` et `select="..."` sont des expressions **XPath**
 - langage à part entière, très riche

18

Exemples

- Instruction de formatage principale

```
<xsl:template match="nom | prenom"> = <nom> OU <prenom>
<xsl:template match="*">          = tout élément fils de .
<xsl:template match="auteur/nom"> = <nom> fils de
                                   <auteur> fils de .
```

- Commande de recopie du contenu +/- balises

```
<xsl:value-of select="."/> = 1'élément courant
<xsl:copy-of select="nom"/> = premier <nom>, enfant de
                             l'élément courant
```

- Commande de traitement des éléments suivants

```
<xsl:apply-templates/> = traiter tous les fils de
                        l'élément courant (select="*")
```

19

Conversion d'un document XML en XHTML grâce à XSLT

- Transformer chaque balise XML initiale en une balise XHTML pour affichage (il faut connaître XHTML)

- exemple : une balise <titre> peut devenir <h1>

- Instruction XSLT nécessaire

```
<xsl:template match="titre">
  <xsl:element name="h1">
    <xsl:apply-templates/>
  </xsl:element>
</xsl:template>
```

Appliquer les autres templates aux autres balises

- Variante simplifiée autorisée

```
<xsl:template match="titre">
  <h1> <xsl:apply-templates/> </h1>
</xsl:template>
```

20

Pour transformer un document XML en XHTML

- Transformer les entêtes, générer les balises <html>

- Décider comment chaque élément XML (balise initiale) sera affiché en XHTML (balise finale)

- Quelques options

- ne pas transformer tous les éléments (= effacer)
- changer l'ordre de certains éléments
- dupliquer certains éléments (exemple : les mettre dans le titre et dans le contenu)
- mettre comme attributs certains contenus et vice-versa

21

XML → XHTML : deux stratégies

- Dans les deux cas

- traiter la racine avec un <xsl:template> pour insérer les déclarations HTML (<html>, <head>, <body>)

- Option A

- traiter dans le <xsl:template> initial tous les autres éléments, en indiquant dans quel ordre et avec quelles balises ils seront affichés

- Option B

- pour chaque élément XML à traiter, écrire un <xsl:template>
 - insérer la balise HTML souhaitée (écrire directement la balise [ouverture et fermeture] ou utiliser <xsl:element>), ajouter éventuellement des attributs HTML avec <xsl:attribute>
 - recopier le contenu de l'élément XML dans le fichier résultat
 - dire quels enfants seront traités et comment

- Option C : mélanger A et B

22

Exemples

- Option A

- fichier xml2html-A.html

- Option B

- fichier xml2html-B.html

- Discussion de ces fichiers

- Essais pendant la séance d'exercices

23

Exercices

- Associez les feuilles de style xml2html-A.xsl et xml2html-B.xsl respectivement aux documents livreC.xml et livreD.xml (les mêmes qu'au cours 7)

- testez les résultats XHTML obtenus : comment s'affichent-ils ?
- inverser les documents et les feuilles de style : résultats ?

- Objectif : développer une feuille de style XSLT pour améliorer l'affichage des fiches bibliographiques

- changer l'ordre des éléments, afficher du texte supplémentaire, etc.
- par exemple, format « entrée bibliographique » pour un mémoire

- Méthode

- enrichissez progressivement chaque feuille de style, *en testant après chaque modification* le résultat obtenu
- utilisez les exemples du cours et l'aide-mémoire indiqué sur Moodle

24