FMIN201 : Galaxie XML
6 mai 2008 - 13h-15h : A.5.6 (59)
Partie Pierre Pompidor - Tout document autorisé
-
A composer sur une feuille séparée

Remarque préliminaire :
L'attribut `mode` permet de faire correspondre une sélection de noeuds dans l'arbre DOM (`<xsl:apply-templates ... mode="nomDuMode">`) à la définition d'un template particulier (`<xsl:template ... mode="nomDuMode">`).

Soit le fichier de données XML et le code XSL suivants :

Fichier de données XML

```
<?xml version="1.0" ?>
<?xml-stylesheet href="examen_GXML_2008.xsl" type="text/xsl" ?>

<A>
   <B a="b1"> B1
      <C a="c1"> C1
         <B a="b2"> B2 </B>
      </C>
      <C a="c2"> C2 </C>
   </B>
   <B a="b3"> B3 </B>
   <B a="b4"> B4
      <C a="c3"> C3 </C>
   </B>
</A>
```

Code XSL

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="ISO-8859-1" />
<xsl:strip-space elements="*"/>

<xsl:template match="/">
  <html>
    <body>
      <xsl:apply-templates select="//*"/>
    </body>
  </html>
</xsl:template>

<xsl:template match="*[not(preceding::node()[name()=name(current())])]">
  <br/><br/><xsl:value-of select="name(current())" /> : <br/>
  <xsl:apply-templates select="//*[name() = name(current())]" mode="m2">
     <!--xsl:sort select="@a" order="ascending" /-->
  </xsl:apply-templates>
</xsl:template>
```

```
<xsl:template match="*" mode="m2">
  <xsl:value-of select="@a" />
  <!--xsl:value-of select="." /-->
</xsl:template>

<xsl:template match="*">
</xsl:template>

</xsl:stylesheet>
```

Expliquez le fonctionnement de ce programme quand le document XML est chargé sous FireFox :

- en dessinant ce qui s'affiche dans le navigateur → 4 points
- en expliquant le rôle du dernier template → 2 points
- en expliquant ce qui se passe quand le premier commentaire est décommenté → 1 point
- en expliquant ce qui se passe quand le second commentaire est décommenté → 2 points

---

Si le code précédent est remplacé par celui-ci :

Nouveau code XSL

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:dyn="http://exslt.org/dynamic">

<xsl:output method="xml" encoding="ISO-8859-1" />
<xsl:strip-space elements="*"/>
<xsl:param name="requete" />

<xsl:template match="/">
  <xsl:variable name="xpath" select="concat('//',$requete)"></xsl:variable>
  <xsl:apply-templates select="dyn:evaluate($xpath)"/>
</xsl:template>

<xsl:template match="*">
  <xsl:value-of select="child::*" />
</xsl:template>

</xsl:stylesheet>
```

Expliquez le fonctionnement de ce programme quand la ligne de commande suivante est exécutée dans un terminal Unix/Linux :

```
xsltproc --param 'requete' "'B'" examen_GXML_2008.xsl donnees.xml > resultat.xml
```

- en décrivant le contenu du fichier `resultat.xml` → 3 points
- et en expliquant brièvement le rôle des deux templates → 2 points