Cartouche du document

Année : ING 2
Activité : Travail dirigé

Objectifs

L'objectif de cette série d'exercices est :

- d'approfondir les langages XSL et XPath
- de découvrir des régles propres à XSLT 2.0

Quelques liens pour ce travail dirigé : XSLT 2.0, XPath 2.0 les fonctions XPath et SAXON (pour télécharger le moteur XSLT).

Sommaire des exercices

Corps des exercices

1 - Produits et Familles

Énoncé :

Dans cet exercice, on travaille avec le fichier XML qui suit :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<produits>
 <produit nom="p1" famille="f1" prixUnitaire="20" stock="15">
 descriptif de p1
 bla bla
 </produit>
 <produit nom="p2" famille="f2" prixUnitaire="50" stock="2">
 descriptif de p2
 bli bli
 </produit>
 <produit nom="p3" famille="f1" prixUnitaire="1" stock="23">
 descriptif de p3
 blo blo
 </produit>
 <produit nom="p4" famille="f2" prixUnitaire="3" stock="2">
 descriptif de p4
 bli bli
```

```
 </produit>
</produits>
```

Question 1)

Énoncé de la question

Ecrire la feuille XSL qui permet d'afficher dans un fichier html les caractéristiques regroupées par famille.

Question 2)

Énoncé de la question

On désire ajouter dans la feuille précédente l'affichage pour chaque famille de la valeur moyenne (en euros) du stock. Les familles doivent être ordonnées. L'ordre se fait par rapport aux stocks cumulés des produits du plus grand au plus petit.

On vous demande d'utiliser les objets suivants :

- la règle <xsl:for-each-group ....
- la fonction current-group
- les fonctions d'agrégation : sum et avg

Pour une bonne présentation du résultat, vous devez associer à chaque balise
 un retour à la ligne. Ce traitement devra se faire à l'aide d'un template nommé.

Question 3)

Énoncé de la question

On désire inclure la notion de langue. On passera en paramètre la langue de l'utilisateur et la sortie sera présentée dans la langue choisie.

On devra créer autant de fichiers XML qu'il y a de langues. Tous ces fichiers auront le même schéma. Ils contiendront chacun les expressions utilisées dans la sortie traduite dans la langue. Dans le fichier XSL on utilisera la fonction document qui permettra de charger dans une variable globale l'arbre des expressions et donc d'accéder aux différentes expressions à n'importe quel endroit de la feuille XSL.

Rappel : Pour passer un paramètre dans la ligne de commande, on écrit **nomParam=valParam**.

2 - Recherche de mots dans un texte

Énoncé :

Dans cet exercice, on travaille avec le fichier XML de la forme suivante :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<texte>
 <motCle>???</motCle>
 <motCle>???</motCle>
 <contenu>
 bla bla
 </contenu>
```

```
</texte>
```

Question 1)

Énoncé de la question

Ecrire la feuille XSL qui reçoit en paramètre un mot et qui affiche dans un fichier texte le nombre d'occurences de ce mot dans le texte contenu dans le noeud contenu.

On pourra s'inspirer du code xsl qui suit :

```
<!-- le patron nommé pour remplacer dans une chaîne les retours à
la ligne par la balise br -->
<xsl:template name="nlTobr">
<xsl:param name="string"/>
 <xsl:choose>
 <xsl:when test="contains($string,'&#10;')">
  <xsl:call-template name="tabToli">
   <xsl:with-param name="string" select="substring-
   before($string,'&#10;')"/>
  </xsl:call-template>
  <br/>
  <xsl:call-template name="nlTobr">
  <xsl:with-param name="string" select="substring-
  after($string,'&#10;')"/>
  </xsl:call-template>
 </xsl:when>
 <xsl:otherwise>
  <xsl:call-template name="tabToli">
   <xsl:with-param name="string" select="$string"/>
  </xsl:call-template>
 </xsl:otherwise>
 </xsl:choose>
</xsl:template>
<!-- le patron nommé pour remplacer dans une chaîne les
tabulations par une série d'espaces insécables -->
<xsl:template name="tabToli">
<xsl:param name="string"/>
 <xsl:choose>
 <xsl:when test="contains($string,'&#9;')">
  <xsl:value-of select="substring-before($string,'&#9;')" disable-
  output-escaping="yes"/>
```

```
 <!--   est le code ascii de l'espace insécable -->
    
 <xsl:call-template name="tabToli">
 <xsl:with-param name="string" select="substring-
 after($string,'&#9;')"/>
 </xsl:call-template>
</xsl:when>
<xsl:otherwise>
 <xsl:value-of select="$string" disable-output-escaping="yes"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
```

Question 2)

Énoncé de la question

Ecrire la feuille XSL qui affiche pour chaque mot-clé, le mot-clé et son nombre d'occurences dans le texte. L'affichage se fera par ordre décroissant d'occurences.