

## Cartouche du document

Année : ING 2 - Activité : Travail dirigé

## Objectifs

L'objectif de cette série d'exercices de travailler avec des possibilités spécifiques à XSLT 2.0 :

- xsl:for-each-group
- les séquences
- les fonctions proposés dans XSLT 2.0
- la construction de fonctions

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

## Sommaire des exercices

### 1 - Lycéens

## Corps des exercices

### 1 - Lycéens

#### Énoncé :

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

```
<?xml version="1.0" encoding="iso-8859-1"?>
<lyceens>
  <lyceen>
    <nom>Nguyen</nom>
    <prenom>Sarah</prenom>
    <moyenne>18</moyenne>
    <sexe>F</sexe>
    <classe>Terminale</classe>
  </lyceen>
  <lyceen>
    <nom>Dupont</nom>
    <prenom>Christophe</prenom>
    <moyenne>5</moyenne>
    <sexe>M</sexe>
    <classe>première</classe>
  </lyceen>
  <lyceen>
    <nom>Saint Cyr</nom>
    <prenom>Christophe</prenom>
    <moyenne>20</moyenne>
    <sexe>M</sexe>
    <classe>Terminale</classe>
```

```
</lyceen>
<lyceen>
  <nom>Bulot</nom>
  <prenom>Pierre</prenom>
  <moyenne>12</moyenne>
  <sexe>M</sexe>
  <classe>Seconde</classe>
</lyceen>
<lyceen>
  <nom>de Vinci</nom>
  <prenom>Françoise</prenom>
  <moyenne>13</moyenne>
  <sexe>F</sexe>
  <classe>Terminale</classe>
</lyceen>
<lyceen>
  <nom>Amboise</nom>
  <prenom>Augustine</prenom>
  <moyenne>7</moyenne>
  <sexe>F</sexe>
  <classe>première</classe>
</lyceen>
<lyceen>
  <nom>Durand</nom>
  <prenom>Pierre</prenom>
  <moyenne>9</moyenne>
  <classe>Seconde</classe>
</lyceen>
</lyceens>
```

### Question 1)

#### Énoncé de la question

Ecrire les fonctions XSLT 2.0 appelées identification qui permet de retourner sortie

- le nom et le prénom d'un lycéen si on ne connaît pas son sexe
- la civilité (Mlle ou Mr), le nom et le prénom si on connaît son sexe

#### Solution de la question

### Question 2)

#### Énoncé de la question

En utilisant les fonctions précédentes, écrire la feuille xsl 2.0 qui permet d'afficher l'identification de tous les lycéens.

### Solution de la question

### Question 3)

#### Énoncé de la question

Ecrire la fonction XSLT 2.0 appelée *mediane* qui reçoit une liste de valeurs réelles et qui renvoie la médiane.

### Solution de la question

### Question 4)

#### Énoncé de la question

En utilisant la fonction précédente, écrire la feuille xsl 2.0 qui permet d'afficher la moyenne et la médiane des notes de tous les lycéens.

### Solution de la question

### Question 5)

#### Énoncé de la question

En utilisant l'instruction `xsl:for-each-group`, écrire la feuille xsl 2.0 qui permet d'afficher la moyenne et la médiane des notes de tous les lycéens par sexe.

### Solution de la question

## Réponse globale à tout l'exercice

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:fonc="http://www.eisti.fr"
  xmlns:fn="http://www.w3.org/2006/xpath-functions"
  >
<xsl:output method="text" encoding="iso-8859-1"/>
<xsl:template match="lyceens">
  <!-- exercice 1 -->
  <xsl:value-of select="fonc:titre(concat('Exo 1 :
  ','Indentification des lycéens'))"/>
  <xsl:for-each select="lyceen">
    <xsl:choose>
```

```

<xsl:when test="sexe"><xsl:value-of
select="fonc:identification(nom,prenom,sexe)"/></xsl:when>
<xsl:otherwise><xsl:value-of
select="fonc:identification(nom,prenom)"/></xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:for-each select="(1 to 5)">
<xsl:call-template name="nl"/>
</xsl:for-each>
<!-- exercice 2 -->
<!-- on crée la séquence des moyennes -->
<xsl:variable name="moyennes" as="xsd:double *">
<xsl:sequence select="lyceen/moyenne"/>
</xsl:variable>
<!-- On crée la liste des moyennes -->
<xsl:value-of select="fonc:titre(concat('Exo 2 : ','Statistiques
sur les moyennes des lycéens'))"/>
<!-- On affiche la série des moyennes ordonnées -->
<xsl:for-each select="$moyennes">
<xsl:sort select="."/>
<xsl:text> </xsl:text><xsl:value-of select="."/>
</xsl:for-each>
<xsl:call-template name="nl"/>
<xsl:text>La moyenne des moyennes vaut </xsl:text>
<xsl:value-of select="format-number(avg($moyennes),'##.00')"/>
<xsl:call-template name="nl"/>
<xsl:text>La médiane des moyennes vaut </xsl:text>
<xsl:value-of select="format-
number(fonc:mediane($moyennes),'##.00')"/>
<xsl:for-each select="(1 to 5)">
<xsl:call-template name="nl"/>
</xsl:for-each>
<!-- Exercice 3 -->
<xsl:value-of select="fonc:titre(concat('Exo 3 : ','Statistiques
sur les moyennes des lycéens par sexe'))"/>
<xsl:for-each-group select="lyceen" group-by="sexe">
<xsl:value-of select="fonc:affSexe(current-grouping-key())"/>
<xsl:variable name="moyennesParGroupe" as="xsd:double *">
<xsl:sequence select="current-group()/moyenne"/>
</xsl:variable>

```

```

<!-- On affiche la série des moyennes ordonnées -->
<xsl:for-each select="$moyennesParGroupe">
  <xsl:sort select="."/>
  <xsl:text> </xsl:text><xsl:value-of select="."/>
</xsl:for-each>
<xsl:call-template name="nl"/>
<xsl:text>La moyenne des moyennes vaut </xsl:text>
<xsl:value-of select="format-
number(avg($moyennesParGroupe),'###.00')"/>
  <xsl:call-template name="nl"/>
<xsl:text>La médiane des moyennes vaut </xsl:text>
<xsl:value-of select="format-
number(fonc:mediane($moyennesParGroupe),'###.00')"/>
<xsl:for-each select="(1 to 2)">
  <xsl:call-template name="nl"/>
</xsl:for-each>
</xsl:for-each-group>
</xsl:template>

<!--
Identification d'un élève avec son nom et son prénom
-->
<xsl:function name="fonc:identification" as="xsd:string">
  <xsl:param name="nom" as="xsd:string"/>
  <xsl:param name="prenom" as="xsd:string"/>
  <xsl:variable name="res">
    <xsl:value-of select="$nom"/><xsl:text> </xsl:text><xsl:value-of
select="$prenom"/>
    <xsl:call-template name="nl"/>
  </xsl:variable>
  <xsl:value-of select="$res"/>
</xsl:function>

<!--
Identification d'un élève avec son nom, son prénom et son sexe
-->
<xsl:function name="fonc:identification" as="xsd:string">
  <xsl:param name="nom" as="xsd:string"/>
  <xsl:param name="prenom" as="xsd:string"/>
  <xsl:param name="sexe" as="xsd:string"/>
  <xsl:variable name="res">
    <xsl:choose>

```

```

<xsl:when test="$sexe = 'F' or $sexe = 'f'">
  <xsl:text>Mlle </xsl:text><xsl:value-of select="$nom"/>
  <xsl:text> </xsl:text><xsl:value-of select="$prenom"/>
  <xsl:call-template name="nl"/>
</xsl:when>
<xsl:when test="$sexe = 'M' or $sexe = 'm'">
  <xsl:text>Mr </xsl:text><xsl:value-of select="$nom"/><xsl:text>
  </xsl:text><xsl:value-of select="$prenom"/>
  <xsl:call-template name="nl"/>
</xsl:when>
<xsl:otherwise>
  <xsl:value-of select="fonc:identification($nom,$prenom)"/>
</xsl:otherwise>
</xsl:choose>
</xsl:variable>
<xsl:value-of select="$res"/>
</xsl:function>
<!--
  template nommé : Pour passer à la ligne dans une sortie en mode
  texte
-->
<xsl:template name="nl">
  <xsl:text>&#10;</xsl:text>
</xsl:template>
<!--
  fonction : Pour passer à la ligne dans une sortie en mode texte
-->
<xsl:function name="fonc:nl" as="xsd:string">
  <xsl:text>&#10;</xsl:text>
</xsl:function>
<!--
  fonction : Pour faire une tabulation dans une sortie en mode texte
-->
<xsl:function name="fonc:tabulation" as="xsd:string">
  <xsl:text>&#09;</xsl:text>
</xsl:function>
<!--
  fonction : Pour afficher dans la sortie le sexe
-->
<xsl:function name="fonc:affSexe">

```

```

<xsl:param name="sexe" as="xsd:string"/>
<xsl:variable name="res">
  <xsl:choose>
    <xsl:when test="$sexe = 'F' or $sexe = 'f'"><xsl:text>Groupe des
    filles</xsl:text></xsl:when>
    <xsl:when test="$sexe = 'M' or $sexe = 'm'"><xsl:text>Groupe des
    garçons</xsl:text></xsl:when>
  </xsl:choose>
  <xsl:text>&#10;</xsl:text>
</xsl:variable>
<xsl:value-of select="$res"/>
</xsl:function>

<!--
fonction : ecriture d'un titre avec passage à la ligne
-->
<xsl:function name="fonc:titre" as="xsd:string">
  <xsl:param name="titre" as="xsd:string"/>
  <xsl:variable name="res">
    <xsl:value-of select="$titre"/>
    <xsl:value-of select="fonc:nl()"/>
  </xsl:variable>
  <xsl:value-of select="$res"/>
</xsl:function>

<!--
Calcul de la médiane d'une série quelconques de nombres réels
-->
<xsl:function name="fonc:mediane" as="xsd:double">
  <xsl:param name="valeurs" as="xsd:double *"/>
  <!-- on crée la même série en ordonnant les valeurs -->
  <xsl:variable name="valeursOrdonnees" as="xsd:double *">
    <xsl:for-each select="$valeurs" >
      <xsl:sort select="."/>
      <xsl:value-of select="."/>
    </xsl:for-each>
  </xsl:variable>
  <!-- on calcule le rang du milieu de la série -->
  <xsl:variable name="milieu" as="xsd:decimal"
  select="floor(1+count($valeursOrdonnees) div 2)"/>
  <!-- on extrait la valeur du milieu de la série ordonnée -->

```

```
<xslvalue-of select="$valeursOrdonnees[position() = $milieu]"/>  
</xsl:function>  
</xsl:stylesheet>
```