

Séance 4 (correction)

Exercice 4.1 : <xsl:for-each/>

- Faire une boucle sur les deux <title> du <titleStmt> pour
 - les reproduire à l'identique (avec leur attribut) ;
 - ajouter leur texte dans une sous balise <title> ;
 - ajouter les prénoms et noms de l'auteur dans une balise <persName> juste après ;
 - ajouter la date de publication entre parenthèse après <persName> ;
 - mettre par entre le nouveau <title> et le <persName>.

```
<xsl:template match="//titleStmt">
  <xsl:copy> <!-- copie de la balise <titleStmt> -->
    <!-- Boucle sur les <title/> enfants de <titleStmt/> -->
    <xsl:for-each select="./title">
      <xsl:copy> <!-- copie de la balise <title> -->
        <xsl:attribute name="level"> <!-- ajout de l'@level -->
          <xsl:value-of select="./@level"/>
        </xsl:attribute>
        <!-- copie du texte de <title> dans un nouveau <title> -->
        <title><xsl:value-of select="."/></title>
        <!-- ajout de "par" -->
        par
        <!-- ajout d'un <persName avec pour valeur le texte de <author> -->
        <persName><xsl:value-of select="parent::titleStmt/author"/></persName>
        <!-- ajout entre () de la date de publication -->
        (<xsl:value-of select="ancestor::fileDesc//sourceDesc/biblStruct//date"/>)
      </xsl:copy> <!-- copie de la balise </title> -->
    </xsl:for-each>
  </xsl:copy> <!-- copie de la balise </titleStmt> -->
</xsl:template>
```

Exercice 4.2 : <xsl:sort/>

- Faire une boucle sur les deux références dans <listBibl> :
 - Reproduire les <biblStruct> sans changement ;
 - Mais les trier :
 - * Une première fois par date de parution ;
 - * Une deuxième fois par ordre alphabétique (en vous basant sur le nom de famille).

```
<xsl:template match="//sourceDesc">
  <xsl:copy> <!-- copie de la balise <sourceDesc> -->
    <xsl:copy-of select="./biblStruct"/> <!-- copie-conforme de <biblStruct/> -->
  <listBibl>
    <!-- boucle sur les <biblStruct/> dans <listBibl/> -->
```

```

        <xsl:for-each select="./listBibl/biblStruct">
            <!-- tri des <biblStruct/> en fonction de leur date -->
            <xsl:sort select="//date" data-type="number"/>
            <!-- copie-conforme des <biblStruct/> -->
            <xsl:copy-of select="."/>
        </xsl:for-each>
    </listBibl>
</xsl:copy> <!-- copie de la balise </sourceDesc> -->
</xsl:template>

```

Exercice 4.3 : <xsl:for-each-group/>

- Sélectionner le <text/> et reproduire les balises <text>.
- Reproduire les balises <body> ;
- Reproduire les deux <div> des chapitres sans changement ;
- Ajouter un index des noms de personnes et un index des noms de lieux dans une nouvelle <div> après celles des chapitres.
 - Ajouter un <head>Index</head> ;
 - Utiliser plusieurs <xsl:for-each-group/> :
 1. Un premier <xsl:for-each-group/> pour sélectionner les <hi> et les grouper en fonction de leur attribut. À l'intérieur, créer une <div> avec un @type. Utiliser une condition pour spécifier la valeur de l'@type : name si l'itération est sur les <hi rend="b"/> et place si elle est sur les <hi rend="i"/> ;
 2. Un deuxième <xsl:for-each-group/> pour créer **un seul** <p/> par entrée d'index (attention, il y a un doublon : Julien). Ajouter le nom de l'individu ou du lieu dans le <p> et deux points (:) devant. Exemple : <p>Julien :</p> ;
 3. Un troisième <xsl:for-each-group/> pour ajouter devant les deux points le numéro des chapitres où apparaissent les entrées d'index. Exemple : <p>Julien : 1, 2.</p>.

```

<xsl:template match="//text">
    <xsl:copy>
        <body>
            <xsl:copy-of select="./body/div"/> <!-- copie-conforme des deux chapitres -->
            <div> <!-- ajout d'une nouvelle <div> -->
                <head>Index</head> <!-- ajout du <head> -->

                <!-- BOUCLE N°1 -->
                <xsl:for-each-group select="//hi" group-by="//@rend">
                    <div> <!-- ouverture de la <div> -->
                        <!-- attribut @type de la <div> -->
                        <xsl:attribute name="type">
                            <xsl:choose>
                                <!-- si on boucle sur les individus (<hi rend="b"/>) -->

```

```

        <xsl:when test="./@rend='b'">
            <xsl:text>name</xsl:text>
        </xsl:when>
        <!-- si on boucle sur les lieux (<hi rend="i"/>) -->
        <xsl:when test="./@rend='i'">
            <xsl:text>place</xsl:text>
        </xsl:when>
    </xsl:choose>
</xsl:attribute>

<!-- BOUCLE N°2 -->
<xsl:for-each-group select="current-group()" group-by=".">
    <p> <!-- création des <p> -->
        <!-- ajout de l'entrée d'index + les deux points -->
        <xsl:value-of select="."/> :

        <!-- BOUCLE N°3 -->
        <xsl:for-each-group select="current-group()"
            group-by="ancestor::div">
            <!-- ajout de la valeur du @n de la <div>-->
            <xsl:value-of select="ancestor::div/@n"/>
            <!-- ajout d'une virgule s'il y a un numéro après-->
            <!-- ou d'un point dans le cas contraire -->
            <xsl:if test="position() != last()">,
            </xsl:if><xsl:if test="position() = last()">.</xsl:if>
        </xsl:for-each-group>
    </p>
</xsl:for-each-group>
</div> <!-- fermeture de la <div> -->
</xsl:for-each-group>
</div>
</body>
</xsl:copy>
</xsl:template>

```

- Explications :
 - Boucle n°1 :
 - * Objectif : créer une <div> pour les individus et une autre pour les lieux.
 - * Solution : sélectionner les <hi> et les grouper en fonction de leur @rend
 - Boucle n°2 :
 - * Objectif : créer un seul <p> par entrée d'index.
 - * Solution : itérer sur le même élément que la boucle précédente [les <hi>, auxquels la fonction `current-group()` permet d'accéder]. Comme on souhaite éviter les doublons, il faut

grouper ces `<hi>` en fonction d'eux-mêmes. Ce qui signifie que les deux `<hi rend="b">Julien</hi>` seront transformés en un seul `<p>Julien</p>`.

- Boucle n°3 :
 - * Objectif : ajouter le numéro de la ou des `<div>` où apparaissent les entrées d'index.
 - * Solution : itérer une nouvelle fois sur les `<hi>`, à la différence que cette fois les groupes sont constitués en fonction de la `<div>` à laquelle ils appartiennent.