

Séance 5 : les boucles

xsl:for-each, xsl:sort et xsl:for-each-group

xsl:for-each

L'instruction xsl:for-each itère sur les nœuds sélectionnés par son attribut *@select* et applique le modèle de son contenu à chacun des éléments du nœud.

```
<xsl:template match="mon_element">
  <xsl:for-each select="sous_elements_a_traiter">
    [motifs à appliquer]
  </xsl:for-each>
</xsl:template>
```

Exercice

Exercice

À partir du XML lerailleur.xml, créez un index des personnages avec la liste des répliques de chacun des personnages.

- Modifier la règle qui porte sur castItem
 - Créer une variable pour enregistrer l'xml:id de l'élément
 - Après le nom du personnage, ajoutez " : "
 - Commencer une boucle for-each qui itère sur tous les
 - Trouver le xpath pour atteindre ces éléments depuis castItem
 - Ajouter un prédicat où xml:id du doit être égal à l'attribut who (trouver l'erreur dans le xml)
 - Pour chaque élément sp, écrire la valeur du contenu de
 - Ajouter entre parenthèses le numéro de l'acte, ., le numéro de la scène. Ex : (1.1)

xsl:sort

L'instruction xsl:sort apparaît comme un enfant de xsl:apply-templates ou xsl:for-each. Il change l'ordre des nœuds contextuels du document source en un autre ordre, comme l'ordre alphabétique.

Liste des attributs :

- @select : clé du tri;
- @data-type : par défaut le type est alphabétique, mais avec la valeur "number", on peut passer sur un tri numérique
- @order : par défaut "ascending"
- @case-order : "upper-first" ou "lower-first".

```
<xsl:template match="mon_element">
  <xsl:for-each select="sous-elements">
    <xsl:sort select="cle_tri"
              order='ascending|descending' />
    <xsl:copy-of select="."/>
  </xsl:for-each>
</xsl:template>
```

xsl:for-each-group

L'instruction *xsl:for-each-group* itère sur les groupes de nœuds sélectionnés par son attribut *@select* et applique le modèle de son contenu à chacun d'entre eux, tandis que l'attribut *@group-by* permet de rassembler les nœuds sélectionnés en sous-groupes en fonction d'un critère défini dans *@group-by* (par exemple une valeur d'attribut). Dans le motif, la fonction *xpath current-grouping-key()* permet de retourner la valeur de la clé de regroupement de la boucle en cours.

```
<xsl:template match="mon_element">
  <xsl:for-each-group select="sous-elements"
                    group-by="cle_regroupement">
    [motifs à appliquer]
  </xsl:for-each-group>
</xsl:template>
```