

Création de variables, fonctions Xpath et Conditions

La création de variables

« L'élément *xsl:variable* attache une valeur de n'importe quel type (chaîne de caractères, nombre, ensemble de nœuds, etc.) à un nom. Cette variable peut être déréférencée par la suite en utilisant la forme \$nom dans une expression Xpath. »

Elliott Rusty Harold, W. Scott Means, Philippe Ensarguet[et al.], *XML en concentré*, Paris, O'Reilly, 2005, p. 544.

```
<xsl:variable name="nom_variable" select="Xpath"/>  
ou  
<xsl:variable name="variable"> Règles internes à la variable  
</xsl:variable>
```

La variable est appelée par la suite par \$nom_variable

```
<xsl:value-of select="$variable"/>
```

Fontions Xpath et traitement des chaînes de caractères

- Concat : permet de souder ensemble des chaînes de caractères

ex :

```
<xsl:value-of select="concat('element1', 'element2')"
```

- Replace : permet de modifier une chaîne de caractères)

ex :

[illegible]

- Upper-case : permet de passer en majuscule une chaîne de caractères
 - upper-case(string)

```
<xsl:value-of select="upper-case(chaineCaractères)"/>
```

- Lower-case : permet de passer en minuscule une chaîne de caractères.
 - lower-case(string)
- count : retourne le nombre de nœuds pour un élément donné

```
<xsl:value-of select="count(element-a-compter)"/>
```

Pour voir plus de fonctions Xpath :

https://www.w3schools.com/xml/xsl_functions.asp

Exercice

Transformer le fichier XML mon rêve familial en fichier HTML à afficher.

1. Créer un nouveau fichier XSLT
2. Définir le namespace `tei` par défaut dans votre Xpath.
3. Paramétrer HTML comme format de sortie avec un encodage `utf-8` à l'aide de `xsl:output @method @encoding`.
4. Créer une structure HTML d'accueil des données XML :

```
<html>
  <head><meta charset="UTF-8"/>
    <title>Exercice de structuration HTML</title>
  </head>
  <body><h1> Ajouter ici le titre demandé (voir A) </h1>
    <ul>
      Remplacer les lg du poeme en XML par des ul (Voir B)
      <ul>
        <li></li>
        Remplacer les l du poeme en XML par des li (voir B)
        [...]
      </ul>
      [...]
    </ul>
    <div> Ajouter ici le texte demandé en 7</div>
  </body>
</html>
```

- A- Ajouter dans h1 le titre du poeme, comprenant le contenu de la balise `<title>` , « écrit par », le contenu de la balise `author` .
- B- Structurer votre poème à l'aide de `<u1>` pour les tercets et les quatrains et `` pour les vers

5. Remplacer dans le titre « Mon » par « Un »
6. Ajouter à la fin du poème, la ligne "Ce poème contient [x=nb de vers] vers", puis remplacer « poème » par la valeur de @type du lg qui contient l'intégralité du poème.

Les Conditions

xsl:if

xsl:if contient un modèle, instancié si et seulement si l'expression Xpath contenue dans son attribut obligatoire *@test* est vraie.

```
<xsl:template match="mon_element">
  <xsl:if test="chemin_Xpath_a_verifier">
    <xsl:copy-of select="."/>
  </xsl:if>
</xsl:template>
```

xsl:choose, xsl:when, xsl:otherwise

L'élément `xsl:choose` sélectionne une possibilité dans une liste de choix. Cet élément contient au moins un `xsl:when` avec un attribut `@test`, quand la condition de `@test` est vérifiée, les motifs contenus par `xsl:when` sont exécutés. L'élément `xsl:choose` peut avoir un sous-élément optionnel `xsl:otherwise` qui traite tous les cas qui ne correspondent pas à ceux sélectionnés par `xsl:when`.

```
<xsl:template match="mon_element">
  <xsl:choose>
    <xsl:when test="chemin_xpath_a_verifier">
      [motifs à appliquer]
    </xsl:when>
    <xsl:otherwise/>
  </xsl:choose>
</xsl:template>
```