

# **séance 6 : Transformation XML vers HTML**

# Exercice

À partir du fichier TEI *sainteEulalie.xml*

Régler un scénario XSL avec en entrée *sainteEulalie.xml* /  
*sainteEulalie.xsl* / en sortie *sainteEulalie.html*

# Étape 1

- Paramétrez votre en-tête pour une sortie XML vers HTML;
- Créer une structure HTML d'accueil pour le texte de *sainte Eulalie* avec :

- Un titre (h1)
  - Un lien vers la reproduction du manuscrit sur Gallica
  - Une `<div>` pour le texte accueillant le texte original ;
  - Une `<div>` pour le texte accueillant le texte normalisé ;
  - Bonus : numéroter automatiquement les vers.

## Étape 2

Mettre en place un index des noms des noms de personnages avec :

- Entrée : occurrence1, occurrence2, etc.
- Trouver un moyen de séparer les occurrences par une virgule et de terminer la liste par un point.
- Ajouter entre parenthèses derrière chaque occurrence, le numéro du vers où apparaît le terme :
  - Entrée : occurrence1 (v.nb)

## fonction xpath count

La fonction *count* retourne le nombre de noeuds d'un ensemble de noeuds XML.

### *exemple*

```
<exemple>
  <x level="1">
    <x level="2">
      <x>
        <y/>
        <y/>
      </x>
    </x>
  </x>
  <x level="1">
    <x level="2">
      <y/>
      <y/>
    </x>
  </x>
</exemple>
```

```
<xsl:template match="/">  
  <xsl:value-of select="count(//x)"/>  
  <xsl:value-of select="count(//x[1])"/>  
  <xsl:value-of select="count(//x/y)"/>  
</xsl:template>
```

## Étape 2bis

Insérer cette règle dans l'architecture XML à l'aide de `<xsl :call-template>`

### xsl:call-template

`<xsl:call-template>` permet d'appeler, à l'intérieur d'une règle xsl une autre règle, grâce à l'attribut `@name`, à un endroit précis d'une architecture d'accueil.

La règle appelée doit être déclarée et nommée grace à `@name` qui permettra de l'identifier.

### Exemple

```
<xsl:call-template name="nom_règle"/>
[...]  
<xsl:template name="nom_règle">  
    [motifs de la règle]  
</xsl:template>
```

## Étape 3

À l'aide d'une seule XSL, en automatisant la production des documents de sortie avec `<xsl:result-document>` :

- Faire une page HTML avec la version normalisée ;
- Faire une page HTML avec la version allographétique ;
- Faire une page pour l'index des noms de personnages ;
- Créer une page de navigation avec les adresses des trois pages précédentes.

Documentation : <https://www.saxonica.com/html/documentation/xsl-elements/result-document.html>

Bonus : proposer un habillage CSS du poème