

Technique et chaîne de publication électronique avec XSLT (3/7)

Jean-Damien Généro

2025, 1^{er} dec. - 2026, 20 janv.

École nationale des chartes – M2 TNAH

XPath / Les fonctions

XPath/ Fonctions

- Les fonctions peuvent être utilisées dans des expressions XPath ou dans des prédictats :
 - Expression → $count(./p)$
 - Prédictat ($[]$) → $./[count(p)]$
- Cf. fiche : fichier xpath-nodes-axis-functions.

XSLT. Variables et paramètres

XSLT/ Différence entre variable et param

- Rappel → variable : nom (unique) + valeur (statique ou dynamique).
- Dans XSLT, portée globale (premier niveau) ou locale (intégrée à un *xsl:template*);
- *<xsl:variable/>* : valeur fixe (*NB : la valeur courante peut changer au sein d'une itération*);
- *<xsl:param/>* : valeur fixe ou dynamique, peut être pris en argument par le processeur (ex : saxon dans lxml).

XSLT/ <xsl:variable/> (1/3)

- Attributs : *@name* (*req.*) et *@select* (*opt.*);
- Deux manières de définir la valeur de <xsl:variable/>:
 - Par la valeur de l'*@select* (expression XPath ou texte);
 - Par le contenu de <xsl:variable/> (règle ou texte).

```
<xsl:variable name="w" select="Xpath"/>
<xsl:variable name="x">règle/texte</xsl:variable>
[<xsl:variable name="y" select="'texte'"/>]
[<xsl:variable name="z" select="1234"/>]
```

XSLT/ <xsl:variable/> (2/3)

- Pour appeler une variable : $\$ + \text{@name}$ dans le @select d'un <xsl:value-of/> (uniquement le texte) ou d'un <xsl:copy-of/> (texte + balises).

```
<xsl:template math="/">
    <xsl:value-of select="$x"/>
    <xsl:copy-of select="$y"/>
</xsl:template>
```

- Possibilité de les ajouter directement comme valeur d'attribut avec des { }.

```
<xsl:template math="/">
    <TEI xml:lang="{$language}">...</TEI>
</xsl:template>
```

XSLT/ <xsl:param/>

- Fonctionne de la même manière que *<xsl:variable/>*.
- En utilisation locale, doit être le premier enfant de *<xsl:template/>*.

XSLT. Conditions

XSLT/ Conditions

- Deux moyens d'exprimer une condition en XSLT : `<xsl:if/>` et `<xsl:choose/>`;
- Attention : les conditions XSLT ne fonctionnent pas de la même manière que celles d'autres langages de programmation, comme Python où l'on trouve le triplet `if`, `elif` et `else`.

XSLT/ <xsl:if/> (1/2)

- <xsl:if/> possède un *@test* obligatoire;
- Contenu appliqué si le *@test* est validé.
- Pas de *elif* : utiliser plusieurs <xsl:if/>;
- Pas de *else* : utiliser <xsl:choose>.

XSLT/ <xsl:if/> (2/2)

Exemple : dans le *Journal*, ajouter un *an* aux *<p/>* si ces derniers sont dans la *<div/>* n° 2 :

```
<xsl:template match="//text//div/p">
  <p>
    <xsl:if test="parent::div/@n = 2">
      <xsl:attribute name="n">
        <!-- à vous de compléter -->
      </xsl:attribute>
    </xsl:if>
    <xsl:value-of select=". />
  </p>
</xsl:template>
```

XSLT/ <xsl:choose>

- <xsl:choose/> a pour enfant un ou des <xsl:when/> (req.) et un seul <xsl:otherwise/> (opt.).
- <xsl:when/> a un *at test* obligatoire (même fonctionnement que <xsl:if/>);
- si aucun <xsl:when/> n'est réalisé, alors <xsl:otherwise/> est activé